

MVVM Perangkat Lunak Android dan Analisis Arsitektur MVP dengan Studi Kasus Aplikasi iTourism

Akhmad Hilmy Zakaria, I Kadek Dwi Nuryana

^{1,2} Jurusan Teknik Informatika, Universitas Negeri Surabaya)

¹akmad.17051204028@mhs.unesa.ac.id

²dwinuryana@unesa.ac.id

Abstrak – Pariwisata merupakan salah satu sektor pendapatan penting bagi suatu negara. Indonesia merupakan salah satu negara yang memiliki pendapatan dari industri pariwisata. Indonesia memiliki banyak industri pariwisata yang tersebar di berbagai daerah, bahkan hampir di setiap kota. Kota Batu merupakan salah satu contoh kota yang dijuluki Kota Pariwisata karena memiliki keragaman dan jumlah daya tarik wisata yang tinggi. Berbagai tempat wisata dapat ditemukan di kota Batu, mulai dari alam yang indah, pemandangan yang menyejukkan, masakan otentik, dan banyak kafe yang nyaman untuk dikunjungi wisatawan. Di era teknologi, menggabungkan sektor pariwisata dengan teknologi merupakan strategi pemasaran yang baik untuk meningkatkan pendapatan pariwisata. Hal ini ditunjukkan dengan menggunakan pengembangan aplikasi android sebagai pemandu wisata, adanya fitur yang akan memandu wisatawan ke tempat-tempat populer terdekat merupakan bentuk kerjasama dari sektor pariwisata dengan teknologi. Aplikasi yang baik tidak hanya memiliki aplikasi dengan UI yang mewah, namun aspek UX juga dianggap penting karena aplikasi yang berat akan membuat penggunaannya tidak nyaman.

Kata Kunci — MVVM, MVP, GIS, Arsitektur, Android Studio, Pengembangan Android.

I. LATAR BELAKANG

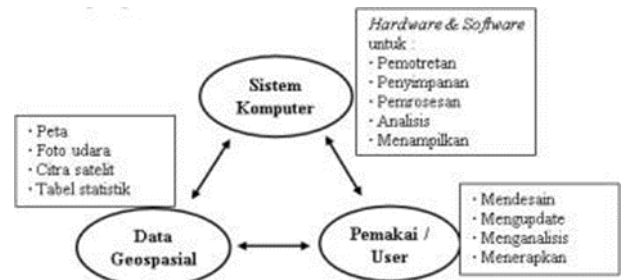
Proses pengembangan aplikasi android tidak hanya menulis kode yang berfungsi dengan UI yang terlihat menarik, aspek kinerja harus dipertimbangkan dengan cermat saat proses pengembangan dilakukan. Aplikasi yang berkinerja buruk membuat pengguna tidak nyaman saat menggunakan aplikasi. Dalam pemilihan arsitektur mana yang harus diimplementasikan dalam tahap pengembangan aplikasi android untuk memberikan kinerja terbaik, pengembang dihadapkan pada banyak pilihan. Arsitektur yang paling banyak digunakan selama proses pengembangan android adalah MVC, MVVM dan MVP. Masing-masing memiliki pendekatan dan hasil kinerja yang berbeda. Dalam studi ini, diskusi difokuskan pada mengapa MVVM digunakan pada proyek ini dibandingkan MVP. Performa yang lebih baik antara MVVM dan MVP diukur dengan penggunaan CPU yang lebih rendah, konsumsi memori yang lebih rendah dan waktu eksekusi yang lebih sedikit. Penelitian ini menggunakan Android Studio Profile sebagai alat *benchmark* untuk mengatur konsistensi memori dan CPU.

Dalam studi ini, *avd (android Virtual Device)* dan *Smartphone Device* digunakan sebagai platform pengujian. Pengujian dilakukan dalam aplikasi panduan wisata sederhana yang dikembangkan bernama iTourism yang dikembangkan menggunakan kedua arsitektur dan diuji menggunakan alat yang telah disebut pada paragraf sebelumnya.

II. TINJAUAN LITERATUR

A. GIS (Sistem Informasi Geografis)

Menurut Lembaga Penelitian Sistem Lingkungan atau ESRI (1990), Sistem Informasi Geografis (SIG) didefinisikan sebagai sistem komputer yang mampu menyimpan dan menggunakan data yang menggambarkan lokasi di permukaan bumi.

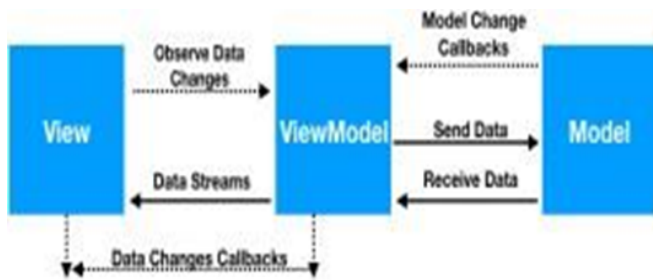


Gambar 1 diatas telah menunjukkan tentang beberapa komponen Sistem Informasi Geografis (SIG) menurut Sulastio et al (2021), diantaranya :

- 1) Sistem Komputer: perangkat keras dan perangkat lunak GIS dapat digunakan untuk entri data, pemrosesan dan pemrosesan data, penyimpanan data, analisis, tampilan, pemodelan, pemotretan, dan tampilan data geospasial.
- 2) Data Geospasial: Sumber data geospasial seperti peta digital, foto udara, citra satelit, tabel statistik, dan dokumen penting lainnya.
- 3) User : Pekerja manusia yang tugasnya mengumpulkan, menganalisis, dan memutakhirkan data pada Sistem Informasi Geografis (SIG).

B. MVVM (Model-View-ViewModel)

VVM adalah arsitektur pengembangan aplikasi seluler yang memiliki struktur Model–View–ViewModel dimana Model adalah truktur data yang digunakan dalam aplikasi sedangkan View.



Pada Gambar 2 di atas ditampilkan pemetaan arsitektur MVVM. Menurut Akhtar et al (2021), disebutkan bahwa Model-View-ViewModel atau MVVM mendukung bidirectional data binding antara komponen View dan ViewModel, dimana View dapat berupa Activity atau Fragment, dengan bantuan Android MVVM Library itu juga dapat mempermudah pekerjaan dengan arsitektur MVVM.

C. Retrofit

Retrofit merupakan library yang banyak digunakan dalam pengembangan aplikasi android untuk mengolah sebuah web service API dan mengolah data yang berformat JSON dari sebuah API. Berdasarkan penelitian dari An Observational Study on the State of REST API Uses in Android Mobile Applications, Retrofit dan OKHttp yang dirilis pada tahun 2014, library pihak ketiga untuk memproses penggunaan REST API dan permintaan HTTP terus digunakan sejak rilis awal mereka pada tahun 2014 hingga Sekarang. OkHttp dan Retrofit terus digunakan karena memudahkan pengembang untuk menyederhanakan permintaan HTTP.

III. METODE

Metode penelitian dalam penelitian ini dilakukan dalam dua skenario, dimana setiap skenario dilakukan pada platform yang berbeda. Setiap skenario memiliki 5 iterasi pengujian yang dapat menghasilkan utilisasi CPU dan Konsumsi Memori rata-rata. alat dan persyaratan yang digunakan selama penelitian ini akan dijelaskan di bawah ini :

a. Alat dan Layanan yang dibutuhkan selama penelitian ini

1. API <https://api.imgflip.com/> Ttopi terdiri dari string dan gambar dengan total 100 data
2. Profiler Android Studio
3. Perangkat ponsel pintar (Redmi 8)
4. Perangkat Virtual Android dengan API > 28

b. Library yang digunakan selama pengembangan :

1. Squareup.Retrofit2

Retrofit2 adalah sebuah *library API Call* yang dikembangkan oleh Squareup. Retrofit2 berfungsi

untuk melakukan proses *API Call* atau bisa disebut dengan *HTTP Request* pada suatu *API* untuk melakukan operasi CRUD (*Create, Read, Update, Delete*).

2. Bumptech.glide

Library Glide adalah sebuah *Library* yang berfungsi sebagai *Image Loader* untuk memuat sebuah gambar dari sebuah *URL* atau *API* dengan mudah. Kelebihan lain dari *Library Glide* adalah mempunyai fitur caching agar sebuah aplikasi tidak perlu memuat gambar berulang kali, Namun kelemahan dari *Library Glide* adalah ukuran nya yang lebih besar dibandingkan *Library Image Loader* lainnya seperti *Picasso* dikarenakan banyak nya fitur *Glide* yang dapat digunakan.

3. Lifecycle.viewmodel

Lifecycle.viewmodel adalah library bagian dari android jetpack yang berfungsi sebagai komponen dasar untuk membuat class *ViewModel* dalam arsitektur MVVM yang memudahkan pengembang mengirim hasil pemrosesan atau pengolahan data menggunakan *LiveData* dan menggunakan fitur databinding.

Prosedur Uji Tolok Ukur

Tabel 1 Prosedur Uji Tolok Ukur

Activity	No.	User Interaction	Benchmark Time
Cafe Activity	1.	User down scroll through "RecyclerView"	30 second
	2.	User click "RecyclerView" item	
Home Activity	1.	User swipe scroll through "ViewPager"	60 second
	2.	User down scroll through "Activity"	
	3.	User change to other Fragment	
Kuliner Activity	1.	User down scroll through "RecyclerView"	30 second
	2.	User click "RecyclerView" item	
Wisata Activity	1.	User down scroll through "RecyclerView"	30 second
	2.	User click "RecyclerView" item	
Penginapan Activity	1.	User down scroll through "RecyclerView"	30 second
	2.	User click "RecyclerView" item	

Skenario Uji 1 (AVD)

a. Persyaratan dan Alat: Android Studio Profiler

- b. Kondisi : Tidak ada aplikasi lain yang berjalan selama pengujian
- c. Prosedur:
 1. Pengguna akan diberikan daftar tertentu tentang apa yang harus dilakukan di setiap aktivitas secara berurutan.
 2. Tolok ukur dimulai saat AVD siap dimulai.
 3. Waktu benchmark diberikan paling kanan dari tabel.
 4. Saat acara selesai, pengguna dilarang melakukan interaksi lain hingga waktu benchmark habis.
 5. Keluaran dari pengujian akan berupa perekaman jejak sistem dengan ekstensi .trace yang menunjukkan penggunaan sistem secara keseluruhan.

Skenario Uji 2 (Perangkat Smartphone)

- a. Persyaratan dan Alat:
 1. Profiler Android Studio
 2. Perangkat Smartphone (Redmi 8)
- b. Kondisi
 1. Memori dibersihkan sebelum menjalankan tes
 2. Tidak ada aplikasi lain yang berjalan di latar belakang selama pengujian
- c. Prosedur:
 1. Pengguna akan diberikan daftar tertentu tentang apa yang harus dilakukan di setiap aktivitas secara berurutan.
 2. Benchmark dimulai pada saat Perangkat Smartphone siap start dalam kondisi bangun.
 3. Waktu benchmark diberikan paling kanan dari tabel.
 4. Saat acara selesai, pengguna dilarang melakukan interaksi lain hingga waktu benchmark habis.
 5. Keluaran dari pengujian akan berupa perekaman jejak sistem dengan ekstensi .trace yang menunjukkan penggunaan sistem secara keseluruhan.

D. Rumus Rata-Rata

Untuk menghitung selisih persentase rata-rata antara MVVM dan MVP, rumus selisih rata-rata yang digunakan adalah sebagai berikut :

$$\mu_{M1-M2} = \mu_1 - \mu_2$$

Gambar 3.2 Rumus Beda Rata-Rata (Sumber: Munawar, 2020)

Gambar 3.2 di atas menunjukkan rumus selisih rata-rata antara MVVM dan MVP, yang dimana "M1" adalah MVVM dan "M2" adalah MVP. Rumus perhitungan pada Gambar 4 di bawah ini akan digunakan dalam menghitung Build Execution Time dan CPU Utilization.

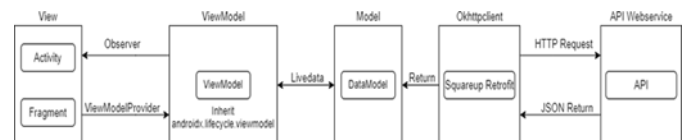
$$\text{Jumlah data} \times \frac{|\mu_{MVVM} - \mu_{MVP}|}{100}$$

Gambar 4 Rumus Menghitung Persentase Rata-Rata Perbedaan

Rumus pada Gambar 4 yang dapat menghitung rata-rata persentase selisih tiap data dikalikan dengan jumlah populasi data. Untuk menghitung rata-rata selisih 2 data dalam satu grafik, seperti penggunaan memori yang memiliki nilai minimal dan maksimal, digunakan rumus pada Gambar 5, dimana nilai min dan maks diperoleh dengan menghitung rumus pada Gambar 4.

III. IMPLEMENTASI

A. MVVP



Gambar 6. Arsitektur MVVM

MVVM adalah akronim dari Model-View-ViewModel yaitu arsitektur perangkat lunak yang menggunakan library lifecycle.viewmodel dan terdiri dari 3 komponen yaitu Model, View dan ViewModel. Setiap komponen dalam arsitektur MVVM memiliki fungsi berbeda dalam perangkat lunak.

Komponen MVVM Utama:

- 1) Melihat.View bertanggung jawab sebagai komponen yang berinteraksi dengan pengguna secara langsung baik itu input atau output pengguna, view bertanggung jawab atas bagaimana arsitektur menampilkan data dari sumber data ke pengguna. Antarmuka Pengguna dan Pengalaman Pengguna juga menjadi pertimbangan besar dalam mengimplementasikan komponen-komponen ini.
- 2) ViewModel.ViewModel dalam arsitektur MVVM berfungsi sebagai logika bisnis dimana semua pemrosesan dan pengolahan data dilakukan sebelum ditampilkan menggunakan komponen View menggunakan LiveData. ViewModel adalah class yang menggunakan inheritance dari android library yaitu lifecycle.ViewModel dimana

implementasi nya untuk mengirim data ke View menggunakan LiveData

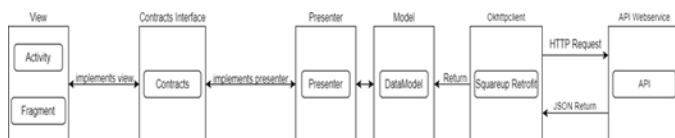
- 3) *Model*. Komponen model memiliki peran kunci dalam bagaimana data dibentuk atau dimodelkan dari JSON Response. Peran kunci ini membuat pemrosesan data jauh lebih mudah daripada memproses respons JSON mentah langsung dari Webservice API.

Komponen Tambahan:

- 1) *Retrofit Squareup*. Komponen tambahan ini berfungsi sebagai parser respons JSON dan panggilan Http. Retrofit menggunakan OkHttp3 sebagai lapisan panggilan HTTP untuk berinteraksi dengan layanan web, kemudian setelah mendapatkan respons dari API yang diberikan, Retrofit memodelkan ulang respons menjadi model data tertentu dan kemudian diproses di ViewModel.
- 2) *API*. API berfungsi sebagai gateway dari sumber data. API oh bagaimana tanggapan yang diberikan untuk permintaan yang masuk seperti: GET, POST, UPDATE, DELETE.

IV. HASIL

A. MVP



Gambar 5. Arsitektur MVP

MVP adalah akronim dari Model-View-Presenter yaitu arsitektur perangkat lunak yang terdiri dari 3 komponen yaitu Model, View dan Presenter. Setiap komponen dalam arsitektur MVP memiliki fungsi berbeda dalam perangkat lunak.

Komponen MVP Utama:

1. *Melihat*. View adalah komponen yang menangani interaksi pengguna. Interaksi pengguna memiliki berbagai bentuk seperti bagaimana User Experience dan User Interface dibentuk untuk menampilkan data yang telah diproses dari Presenter, yang kedua adalah menangani input pengguna yang nantinya diproses di komponen Presenter.
2. *Presenter*. Presenter adalah komponen dalam MVP sebagai logika bisnis dimana semua proses pengolahan dan pemrosesan data dilakukan dalam komponen Presenter. Dalam arsitektur MVP, Presenter mengimplementasikan interface Presenter yang terdapat pada komponen Contracts agar dapat mengirimkan hasil yang telah diproses ke dalam komponen View

3. *Model*. Model sering disebut bentuk atau model dari data yang diterima dari permintaan API. Respons API seringkali berbentuk respons JSON, untuk mempermudah pemrosesan data, model data membentuk kembali atau merombak respons menjadi objek.

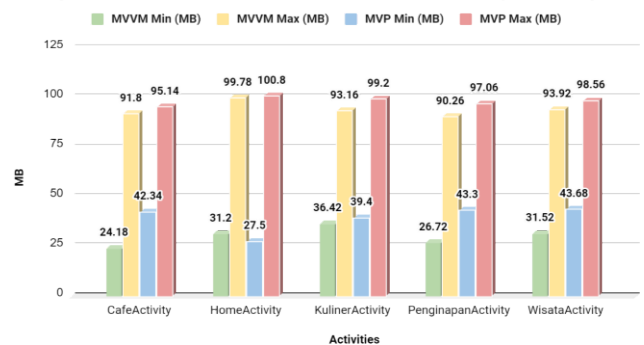
Komponen Tambahan:

1. *Contracts*. Contracts adalah komponen pembantu dalam arsitektur MVP dengan menggunakan class interface. Contracts terdiri dari 2 class interface lainnya, yaitu View dan Presenter dimana kedua class interface di dalam komponen Contracts memiliki implementasi dan fungsi yang berbeda.
2. *Interface View* di implementasikan pada komponen View yang dimana bentuk implementasi dari Interface View adalah berfungsi untuk mengolah interaksi pengguna untuk diproses menuju komponen Presenter. Sedangkan Interface Presenter di implementasikan pada komponen Presenter dan berfungsi untuk mengirimkan hasil pemrosesan dari komponen Presenter.
3. *Retrofit Squareup*. Retrofit merupakan library yang dikhususkan untuk melakukan API Call. Retrofit menggunakan OkHttp3 sebagai network layer untuk melakukan HTTP Request seperti GET, POST, PUT, PATCH dan DELETE. Fungsi lain dari Library Retrofit yaitu adalah JSON Parsing.
4. *API*. API adalah akronim dari Application Programming Interface yang dimana API adalah antarmuka untuk menghubungkan antar aplikasi dalam sistem, sebagai contoh Android App membutuhkan data dari Database menggunakan API yang berbentuk endpoint.

B. Skenario Uji 1 (AVD)

a. Penggunaan Memori

Comparison Between MVVM & MVP Allocated Memory Consumption

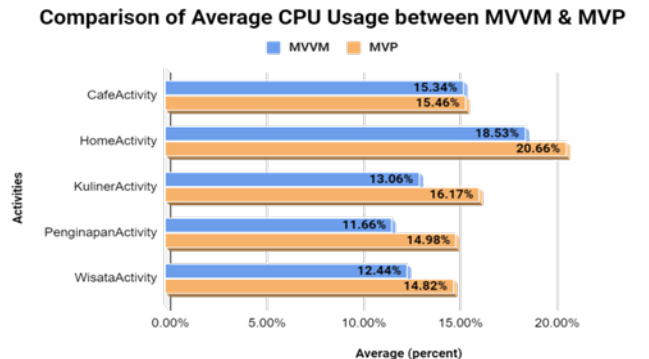


Gambar 8. Perbandingan Grafik Konsumsi Memori yang Dialokasikan

Setelah Pengujian pada AVD dilakukan, konsumsi memori yang tercatat pada Android Studio Profiler

menunjukkan penggunaan memori minimum dan maksimum dengan hasil arsitektur MVVM unggul dengan konsumsi memori rata rata 6,8% lebih rendah dibandingkan dengan arsitektur MVP.

b. penggunaan CPU

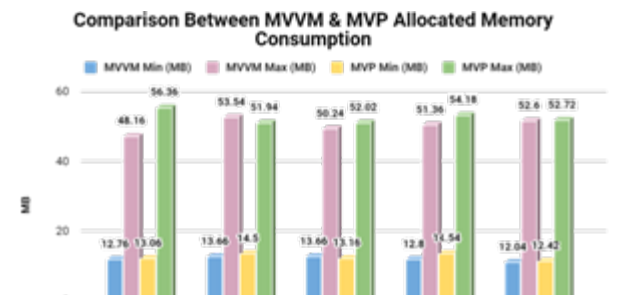


Gambar 9. Grafik Penggunaan CPU Rata-Rata

Grafik pada gambar 9 adalah grafik visualisasi data hasil pengujian penggunaan CPU dari kedua arsitektur yang dilakukan dengan AVD. Hasilnya, MVVM menggunakan sumber daya CPU lebih rendah dibandingkan dengan MVP dengan skor selisih penggunaan sumber daya CPU 2,21% lebih rendah.

C. Skenario Uji 2 (Perangkat Smartphone)

a. Penggunaan Memori



Gambar 10. Grafik Penggunaan memori

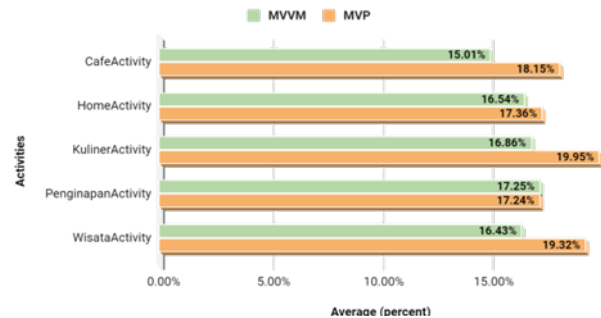
Data yang divisualisasikan dengan grafik pada gambar 7 menunjukkan bahwa pengujian yang dilakukan pada smartphone menggunakan 2 arsitektur perangkat lunak yaitu MVVM dan MVP. Pengujian pada perangkat smartphone dilakukan untuk mengetahui perbandingan performa dalam penggunaan smartphone di dunia nyata. Hasilnya, MVVM unggul dengan konsumsi memori 1.41% lebih rendah dibandingkan dengan arsitektur MVP.

Hasil pengujian memori yang dilakukan pada perangkat smartphone lebih rendah dibandingkan dengan pengujian yang

dilakukan pada AVD, hal ini disebabkan oleh fitur perangkat smartphone yang diuji yaitu Optimisasi MIUI.

b. penggunaan CPU

Comparison of Average CPU Usage between MVVM & MVP



Gambar 10. Grafik Penggunaan CPU Rata-Rata

Data yang divisualisasikan dengan grafik diatas menunjukkan bahwa MVVM unggul dalam 5 activity dengan tingkat penggunaan CPU yang lebih rendah dibandingkan dengan MVP dengan selisih rata rata 2% penggunaan CPU lebih rendah sehingga lebih efisien daya.

Data pengujian yang dilakukan pada AVD dan smartphone memiliki perbedaan dimana data yang dilakukan pada AVD memiliki konsumsi CPU lebih rendah dibandingkan pengujian yang dilakukan pada smartphone. Hal ini disebabkan oleh perbedaan CPU yang digunakan, AVD menggunakan prosesor desktop x86 dengan performa lebih baik dibandingkan dengan prosesor yang digunakan pada smartphone.

V. ANALISIS

A. Perbedaan Implementasi MVVM dan MVP

Faktor lainnya adalah karena kopling yang lebih rendah dan komponen yang lebih sedikit digunakan, yang berarti lebih sedikit kode yang ditulis dan pengujian yang lebih mudah. Ini dibuktikan dengan menyelami lebih dalam ke file pelacakan Android Studio menemukan bahwa di dalam MVVM memiliki jumlah utas yang lebih rendah untuk diproses daripada MVP.

B. Kesenjangan Kinerja Antara MVVM dan MVP

Berdasarkan data yang telah divisualisasikan dengan grafik diatas, menghasilkan informasi bahwa MVVM memiliki performa yang lebih baik dibanding MVP pada kedua skenario pengujian dengan konsumsi memori yang lebih rendah dan tingkat penggunaan CPU yang lebih rendah. Perbedaan performa atas dua arsitektur yang telah ditunjukkan dengan grafik diatas disebabkan oleh fitur utama MVVM yaitu lifecycle-aware yang dimana komponen Ui akan menyesuaikan LiveData atau dikenal sebagai databinding.

C. Kasus Uji Fungsional

Testcase ID	Deskripsi Testcase	Hasil	
		Berjalan	Tidak Berjalan
	Aplikasi dapat memunculkan <i>location permission</i>	✓	
	Aplikasi dapat tetap berjalan meskipun <i>location permission</i> ditolak	✓	
	Aplikasi dapat memunculkan lokasi pengguna terkini	✓	
	Aplikasi dapat mengetahui apakah pengguna sedang berada di kota Batu	✓	
	Aplikasi dapat menampilkan hasil <i>request</i> pada setiap kategori	✓	
	Pengguna dapat melakukan <i>downscroll</i> pada hasil <i>request</i>	✓	
	Pengguna dapat melihat secara detail setiap item hasil <i>request</i>	✓	
	Aplikasi dapat menampilkan tempat terdekat pada pengguna	✓	
	Pengguna dapat melihat alamat tempat	✓	
	Pengguna dapat melihat jumlah ulasan	✓	

Tabel 3. Testcase Fungsional

Aplikasi i-Tourism memiliki fungsi utama sebagai penunjang kebutuhan dalam kebutuhan pariwisata di Kota Batu. Dalam menunjang kebutuhan pariwisata di Kota Batu, Aplikasi i-Tourism terdiri atas beberapa fitur fungsional yang telah diuji apakah berfungsi dengan baik sesuai kebutuhan pengguna.

Hasil pengujian kebutuhan fungsional menunjukkan hasil positif, dimana hal ini berarti semua fungsi berjalan dengan baik. Aplikasi i-Tourism juga memberikan perlindungan privasi dimana pengembang tidak memaksa pengguna untuk memberikan informasi lokasi saat dialog akses lokasi terkini muncul, namun hal tersebut menyebabkan aplikasi berjalan tidak semestinya dikarenakan Aplikasi i-Tourism adalah aplikasi yang menunjang kebutuhan pariwisata berbasis geospasial.

D. Validasi Pengalaman Pengguna

No	Aspek	Indikator	Skor				
			1	2	3	4	5
1.	Performance (Performa)	Aplikasi dapat di install di perangkat milik validator					
		Aplikasi berjalan di perangkat milik validator					
		Aplikasi ini dapat berjalan tanpa memberatkan perangkat milik validator					
2.	User Experience (Pengalaman Pengguna)	Aplikasi ini berfungsi dengan semestinya dengan internet					
		Aplikasi ini dapat menangkap lokasi dengan akurat					
		menu aplikasi dapat dipilih dan tampil dengan cepat					
		Tidak perlu melakukan request berulang saat pergantian orientasi layar					
		-Kesesuaian tema warna dengan fungsi aplikasi					
		-menu aplikasi dapat menampilkan informasi yang sesuai dengan Antarmuka					
		konten yang dimuat dapat tampil di daftar menu yang dipilih					
		Kemudahan mencari informasi yang dibutuhkan					
		Informasi yang dimuat sesuai dengan kategori yang dipilih					
		Perangkat lunak tidak melanggar privasi pengguna					

Tabel 4. Validitas Pengalaman Pengguna

Pada tabel 4 di atas dijelaskan bahwa penilaian User Experience menggunakan dua aspek penilaian yaitu performance dan user experience yang masing-masing indikator dinilai. Lembar validasi pada tabel 4 diserahkan kepada 8 responden yang terdiri dari 5 Software Engineer dan 3 user biasa menghasilkan nilai yang dipetakan pada tabel 5.

RESPONDEN	No. Item															
	Performa							Pengalaman								Total
	P1	P2	P3	P4	P5	P6	P7	Total P1 - P7	P8	P9	P10	P11	P12	P13	Total P8 - P13	
RESPONDEN KE-1	4	4	4	5	4	4	5	30	3	4	3	3	4	4	21	51
RESPONDEN KE-2	5	5	5	5	5	5	5	35	4	5	4	5	5	5	28	63
RESPONDEN KE-3	4	4	4	4	4	4	5	29	3	4	3	3	3	4	20	49
RESPONDEN KE-4	4	4	3	4	5	3	4	27	2	3	3	4	3	5	20	47
RESPONDEN KE-5	4	4	4	5	5	4	4	30	3	3	3	3	4	5	21	51
RESPONDEN KE-6	4	5	4	5	4	4	5	31	2	3	5	4	3	5	22	53
RESPONDEN KE-7	4	4	4	5	5	4	4	30	3	4	4	3	3	5	22	52
RESPONDEN KE-8	4	5	4	4	5	5	5	32	3	5	3	4	3	5	23	55
TOTAL	33	35	32	37	33	37	37	244	23	31	28	29	28	38	177	421

Tabel 5. Hasil Validasi

Indikator-indikator dimana masing-masing indikator mewakili suatu aspek aspek kinerja juga terlibat dalam pengujian. Kedua aspek tersebut akan dinilai dengan menggunakan skala 1 sampai dengan 5. Setelah melalui pengujian diperoleh hasil 85,8% atau termasuk dalam kategori “sangat layak”.

VI. KESIMPULAN

Arsitektur yang digunakan selama pengembangan perangkat lunak akan berpengaruh pada besaran memori yang digunakan dan tingkat penggunaan *CPU* dalam menyelesaikan proses intruksi dimana kedua faktor ini adalah faktor penting dalam penentu aspek performa, semakin besar memori yang dipakai dan tinggi nya utilisasi *CPU* maka aspek performa akan menurun.

Pengujian arsitektur perangkat lunak dilakukan pada 2 arsitektur perangkat lunak android yaitu *MVVM* dan *MVP*

dimana kedua arsitektur ini banyak dipilih oleh pengembang perangkat lunak. Pengujian dilakukan dengan mengembangkan dua perangkat lunak yang identik namun dengan penggunaan arsitektur perangkat yang berbeda. Pengujian menghasilkan data yang menunjukkan bahwa *MVVM* unggul dengan konsumsi memori yang lebih rendah sehingga perangkat lunak lebih ringan saat digunakan dibandingkan dengan perangkat lunak yang dikembangkan menggunakan *MVP*, dalam faktor penggunaan *CPU*, *MVVM* memiliki tingkat penggunaan *CPU* yang lebih rendah dibandingkan perangkat lunak yang dikembangkan dengan *MVP* sehingga perangkat keras tidak perlu bekerja lebih dalam memproses instruksi.

Selain memiliki performa perangkat lunak yang lebih baik dibandingkan dengan arsitektur *MVP*, arsitektur *MVVM* memiliki fitur yaitu *lifecycle aware* dimana memiliki fungsi *data binding* yang berfungsi mengikat data dan *UI* dengan *android lifecycle* sehingga perubahan data dan *UI* dapat menyesuaikan *android lifecycle*.

VII. DAFTAR PUSTAKA

- [1] Akhtar, Nayab dkk. (2021). Analisis Pola Arsitektur Untuk Pengembangan Android. Departemen Rekayasa Perangkat Lunak
- [2] Munawar, Ghifari dkk. (2020). Perbandingan Kinerja dari yang memengaruhi Pengalaman Pengguna. Selain Pengalaman Pengguna, Aplikasi Android Asli di MVP dan MVVM.
- [3] Seminar Internasional Sains dan Teknologi Terapan. Kemajuan dalam Riset Rekayasa. Vol. 198.
- [4] Sulastio, Bezaliel dkk. (2021). Sistem Informasi Geografis Untuk Menentukan Lokasi Rawan Macet di Jam Kerja Pada Kota Bandar Lampung Pada Berbasis Android. Jurnal Teknologi dan Sistem Informasi (JTSI). 2(1), hlm. 104-111.