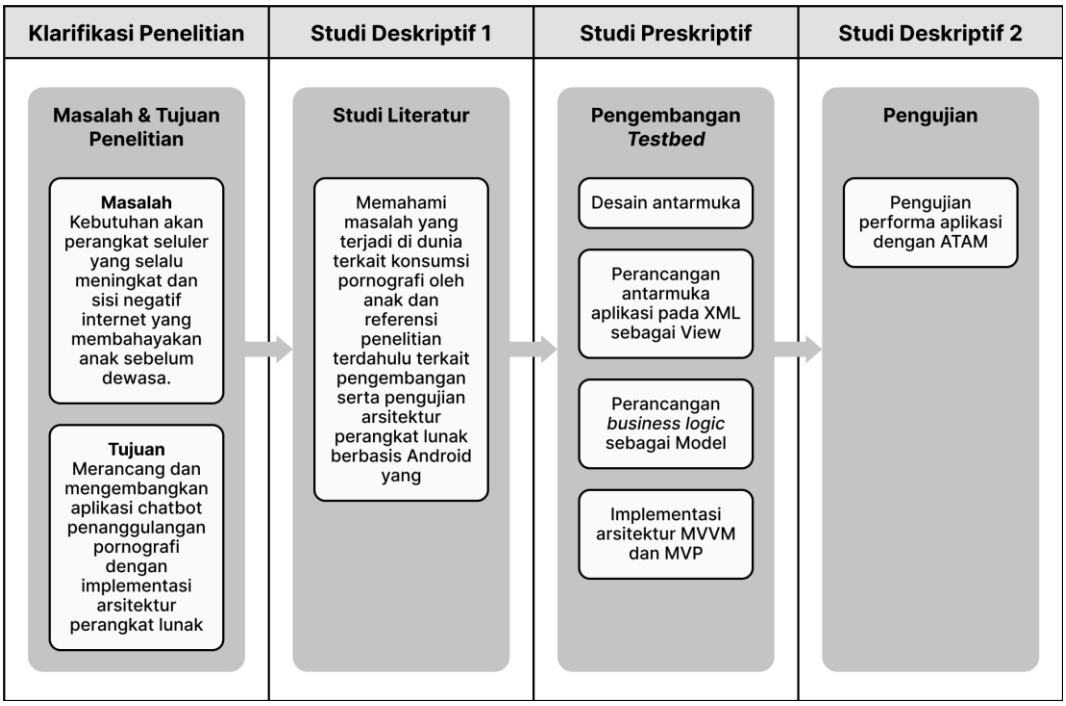
BAB III
METODOLOGI PENELITIAN

3.1 Desain Penelitian

Penelitian yang dilakukan diawali dengan perancangan dan pengembangan aplikasi chatbot berbasis Android untuk deteksi kecanduan pornografi serta menguji aplikasi yang menggunakan arsitektur MVVM dengan arsitektur MVP berdasarkan performa aplikasi. Atas tujuan dari penelitian ini, desain metode penelitian yang digunakan adalah *Design Research Methodology* (DRM). Hal ini ditenkankan sebagaimana metode ini digunakan sebagai kerangka penelitian yang mendukung pendekatan yang lebih akurat dengan perencanaan dan pengimplementasian dari penelitian yang dilakukan (Chakrabarti dan Blessing, 2009).

Untuk mencapai tujuan dari penelitian ini, diperlukan adanya tahapan yang sesuai. Prosedur penelitian ini pada dasarnya bertujuan untuk merancang, mengembangkan perangkat lunak, dan menguji perangkat lunak yang dibuat berdasarkan performanya. Menggunakan pola kerja pada DRM, penelitian ini akan memiliki alur sebagai berikut:



Gambar 3.1. Alur penelitian dengan *Design Research Methodology* (DRM)

3.1.1 Klarifikasi Penelitian

Di tahap pertama, dilakukan identifikasi masalah berdasarkan pada latar belakang yang telah didapatkan untuk mendapatkan masalah utama yang akan menjadi perhatian utama pada penelitian. Hasil dari identifikasi masalah tersebut menghasilkan tujuan dari penelitian yang akan dicapai sebagai sasaran dari hasil penelitian.

3.1.2 Studi Deskriptif 1

Pada tahap ini, dilakukan studi literatur dengan pemahaman lebih dalam terhadap topik yang diangkat untuk penelitian. Pemahaman dilakukan dengan melihat pada referensi dari penelitian terdahulu terkait dengan topik pengembangan aplikasi dalam maksud untuk mengidentifikasi dan mengklarifikasi secara lebih rinci. Pemahaman ini ada pada *state of the art* di poin 2.5 yang membahas sebanyak enam penelitian yang terkait dengan penelitian yang dilakukan.

3.1.3 Studi Preskriptif

Pada tahap ini, dilakukan pengembangan aplikasi perangkat lunak yang direncanakan untuk dibuat sebagai *testbed* penelitian. Tahapan pengembangan perangkat lunak tersebut didapatkan berdasarkan hasil pemahaman pada studi deskriptif 1. Hasil pemahaman yang didapatkan lalu disesuaikan dengan tujuan penelitian hingga menjadi rancangan pengembangan yang teratur. Dalam upaya terjalankannya tahap pengembangan ini, digunakan metode pengembangan yaitu Personal Extreme Programming (PXP) untuk dapat melakukan pengembangan yang bertahap dan *agile* oleh satu orang pengembang yaitu peneliti.

Pada penelitian ini, pengembangan aplikasi dirancang dalam basis Android dengan arsitektur MVVM. Dalam pemahaman pada studi literatur beserta tujuan penelitian yang akan membandingkan performa antara arsitektur MVVM dan MVP, pengembangan dirancang untuk tidak dibuat satu-per-satu arsitektur karena terdapat kesamaan antara MVVM dengan MVP, yaitu sisi View dan Model. Hal yang paling membedakan antara kedua arsitektur ini adalah penggunaan ViewModel pada arsitektur MVVM dan Presenter pada arsitektur MVP. Perbedaan tersebut dapat memberikan hasil yang berbeda pada sisi kinerja karena cara kerja

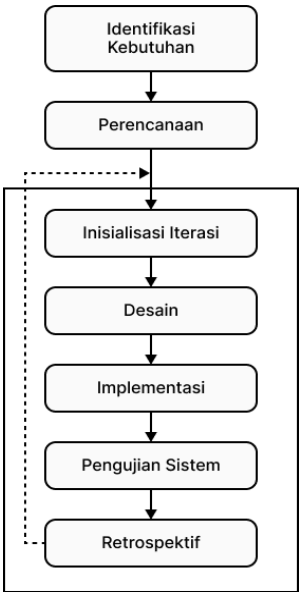
antara interaksi ViewModel ke komponen lain dalam arsitekturnya dengan Presenter yang terhubung dengan komponen lain yang berbeda.

3.1.4 Studi Deskripitif 2

Pada tahap terakhir, dilakukan identifikasi terhadap pengembangan perangkat lunak yang dibuat dengan tujuan utama dari penelitian. Identifikasi ini dilakukan dengan pengujian terhadap perangkat lunak. Untuk mencapai pengujian yang baik, kode dianalisa dan dibersihkan menggunakan Lint supaya kode tidak lagi menjadi halangan atau *bottleneck* pada pengujian lainnya. Pengujian utama dilakukan untuk menguji kinerja atau performa kedua aplikasi yang telah dirancang dengan arsitektur MVVM dan MVP. Pengujian ini akan menggunakan metode *Architecture Tradeoff Analysis Method* (ATAM) dengan menggunakan skenario pengujian beserta metrik pengujiannya.

3.2 Desain Pengembangan

Untuk dapat mengembangkan perangkat lunak dengan efisien berdasarkan perencanaan pengembangan dan dapat menghasilkan produk yang siap untuk menjadi alat uji penelitian atau *testbed*, diperlukan metode pengembangan yang mampu digunakan dan sesuai dengan kebutuhan. Pengembangan aplikasi pada penelitian ini akan menggunakan metode *Personal Extreme Programming* (PXP). Metode PXP dipilih karena merupakan salah satu metode pengembangan berbasis *agile* yang dapat dilakukan khusus untuk satu orang dengan pengembangan yang dapat dilakukan dengan cepat, fleksibel, serta adaptif dalam adanya umpan balik setelah setiap tahapannya dilalui sehingga dapat dilakukan perubahan yang bertahap (Putra dkk., 2022).



Gambar 3.2. Alur Pengembangan dengan *Personal Extreme Programming*

Tahapan dari pengembangan menggunakan metode PXP ini adalah seperti pada Gambar 3.2. Dalam alurnya, pengembangan diawali dengan identifikasi kebutuhan dan perencanaan pengembangan perangkat lunak, lalu masuk ke tahap pengembangan dimana terdapat beberapa tahapan utama yaitu inisialisasi iterasi, desain, implementasi, pengujian sistem, dan retrospektif. Alur pengembangan tersebut dapat kembali ke tahap inisialisasi iterasi setelah satu alur pengembangan dilakukan hingga mendapatkan hasil retrospektif.

3.2.1 Tahap Pengembangan

3.2.1.1 Identifikasi Kebutuhan

Tahap pertama dalam pengembangan dengan metode PXP adalah identifikasi kebutuhan. Tahap ini bertujuan untuk mendapatkan kebutuhan dasar dari sistem aplikasi yang akan dikembangkan, baik secara kebutuhan fungsional maupun nonfungsional.

Pada pengembangan dalam penelitian ini, identifikasi kebutuhan secara fitur akan dilakukan berdasarkan kebutuhan aplikasi yang dapat mendeteksi kecanduan pornografi. Kebutuhan fitur lainnya juga akan ditambahkan berdasarkan pengamatan peneliti terhadap beberapa aplikasi *mobile* yang dapat meningkatkan kemudahan pemakaian, seperti otentikasi pengguna supaya data dapat tersimpan dengan aman dan dapat digunakan pada perangkat lain.

3.2.1.2 Perencanaan

Tahap kedua pada pengembangan dengan metode PXP adalah perencanaan. Perencanaan pengembangan aplikasi ini akan disesuaikan dengan hasil identifikasi kebutuhan yang telah didapatkan. Perencanaan ini akan memberikan keluaran berupa tahapan pengembangan aplikasi berdasarkan kebutuhan fungsional dan nonfungsional beserta pengembangan yang diawali dengan penggunaan arsitektur MVVM dan MVP menyesuaikan dengan tujuan dari penelitian ini. Selain itu, hasil dari perencanaan juga akan menggambarkan bagaimana bentuk sistem yang akan dibuat baik dalam segi antarmuka aplikasi, interaksi data dalam aplikasi, rancangan logika sebagai model, serta sumber data yang akan digunakan sebagai sumber utama dari perjalannya fitur aplikasi yang dikembangkan.

3.2.1.3 Inisialisasi Iterasi

Tahap ketiga pada metode PXP adalah inisialisasi iterasi. Iterasi pada metode pengembangan ini memiliki arti yaitu memahami setiap alur akan dikerjakan, apa saja yang menjadi prioritas utama dalam pengembangan, bagaimana tingkat kesulitannya, dan seberapa lama setiap pekerjaan akan dilakukan. Dari definisi tersebut, maka tahap inisialisasi iterasi ini memerlukan pembentukan tabel dengan tugas yang terurut beserta rincian pekerjaan dan estimasi waktu pengerjaan (Wiyana dkk., 2021).

3.2.1.4 Desain

Tahap keempat pada metode PXP adalah desain. Pada penelitian ini desain yang dimaksud merupakan desain dari aplikasi baik dari antarmukanya serta dengan basis datanya. Desain tersebut diperlukan supaya dapat menjadi acuan dari aplikasi yang umumnya digunakan dalam pendukung kerja jarak jauh, sehingga pengembangan aplikasi hingga menjadi versi prototipe siap menjadi *testbed* dan dilakukan berbagai pengujian atau simulasi yang akan dilakukan.

Untuk dapat membuat desain aplikasi yang akan dikembangkan, dibutuhkan alat bantu atau *tools* yang sesuai. Untuk pembuatan desain antarmuka aplikasi akan digunakan Figma sebagai media untuk pembuatan desain dan prototipe desain yang gratis. Untuk pembuatan basis data akan memanfaatkan MockAPI sebagai basis

data yang dapat digunakan secara prototipe namun dibuat menyerupai basis data yang digunakan pada aplikasi jadi.

3.2.1.5 Implementasi

Tahap kelima pada metode PXP adalah implementasi. Pada tahap ini, dilakukan pengembangan aplikasi sesuai dengan perencanaan, iterasi yang dilakukan, serta desain yang telah dibuat pada tahap sebelumnya.

Pada penelitian ini, pengembangan aplikasi akan menggunakan bahasa pemrograman Kotlin dengan lingkup pengembangan atau *Integrated Development Environment* (IDE) Android Studio. Penggunaan bahasa pemrograman Kotlin sendiri dipilih didasarkan pada hasil penelitian studi komparatif yang dilakukan tahun 2021 mengenai efisiensi penggunaan energi antara berbagai bahasa pemrograman dengan Kotlin yang memiliki *runtime* yang sama dengan Java, termasuk bahasa pemrograman populer yang digunakan dalam pengembangan aplikasi Android (Pereira dkk., 2021).

Pengembangan aplikasi pada penelitian ini juga menerapkan dua arsitektur yaitu MVVP dan MVP. Kedua arsitektur tesebut dipilih untuk digunakan atas tujuan penelitian ini yang akan membuat analisis performa antara implementasi dari kedua arsitektur tersebut.

3.2.1.6 Pengujian Sistem

Pada tahap keenam dari metode PXP dilakukan pengujian sistem. Pengujian sistem yang dilakukan pada pengembangan ini adalah analisa kode dengan Lint untuk membersihkan kode sehingga mencapai *clean code* dan pengujian fitur yang telah dibuat dengan *black box testing*. Pengujian *black box* akan membantu peneliti dalam mengetahui apakah fitur yang dibuat telah sesuai dengan perencanaan dan desain. Pengujian tersebut akan dilakukan dalam skala kecil yang memilki sedikit skenario sesuai dengan fitur yang dimiliki dan telah dibuat dalam aplikasi.

3.2.1.7 Retrospektif

Tahap terakhir dari metode PXP adalah retrospektif. Retrospektif merupakan tahapan untuk melihat atau meninjau kembali hasil pengembangan yang telah dibuat selama tahap iterasi hingga implementasi. Tahap ini bertujuan untuk

mengetahui bagaimana tingkat kesesuaian aplikasi yang telah dikembangkan dengan recana awal pengembangan aplikasinya. Dengan tahap retrospektif ini, pengembangan akan mendapatkan keputusan untuk menyatakan pengembangan telah selesai ketika setiap kebutuhan, rencana, dan desain telah sesuai dan terpenuhi, atau melanjutkan pengerjaan iteratif terhadap fitur atau bagian tertentu yang masih belum sesuai dengan hasil perencanaan.

3.2.2 Lingkungan Pengembangan

Pengembangan aplikasi pada penelitian ini akan dilakukan menggunakan perangkat komputer milik peneliti. Perangkat yang digunakan sendiri memiliki spesifikasi sebagai berikut:

Tabel 3.1. Spesifikasi Lingkungan Pengembangan

CPU	AMD Ryzen 5 2600
GPU	Nvidia GeForce GTX 1050 Ti
RAM	16 GB
OS	Windows 11 Pro Build 22000.978
IDE	Android Studio Arctic Fox
Emulator	Google Pixel 2

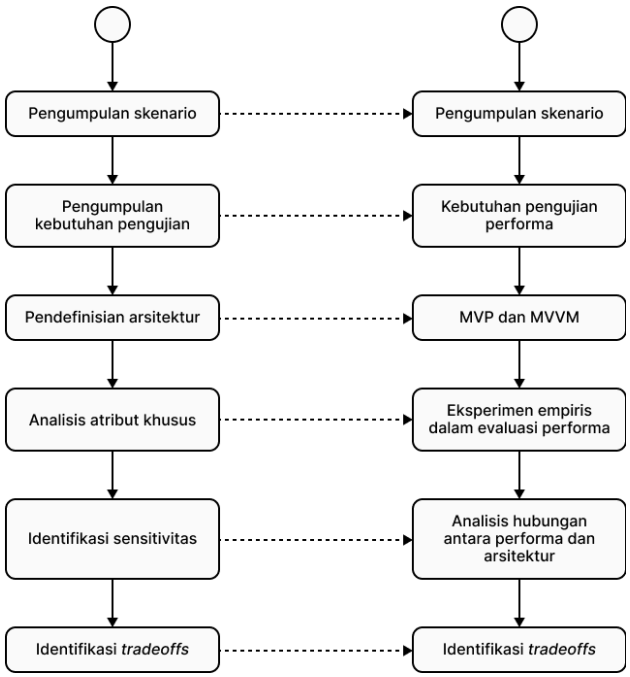
3.3 Desain Pengujian

Untuk dapat mencapai salah satu dari tujuan dari penelitian ini yaitu mengetahui bagaimana performa aplikasi *chatbot* berbasis Android dengan penggunaan antara arsitektur MVVM dan MVP, dibuatkan rancangan desain pengujiannya. Dalam pengujian komparasi performa antar arsitektur perangkat lunak, digunakan metode pengujian yang sesuai supaya bisa mendapatkan hasil yang sesuai. Pada penelitian ini, digunakan metode *Architecture Tradeoff Analysis Method* (ATAM) untuk pengujian arsitektur MVVM dan MVP.

ATAM merupakan metode yang digunakan dalam evaluasi desain arsitektur yang menilai kualitas dari berbagai atribut seperti *modifiability* atau kemudahan untuk dilakukan modifikasi, *performance* atau performa dari penggunaan arsitektur tersebut, dan *availability* atau ketersediaan atau kesiapan produk saat dibutuhkan dalam mendapatkan wawasan tentang apakah arsitektur telah sepenuhnya sempurna dalam memenuhi persyaratan tersebut (Kazman dkk., 2000). Metode ATAM

memiliki 9 tahapan sebagai langkah-langkah yang lengkap, yaitu mempresentasikan konsep ATAM kepada pemangku kepentingan, presentasi penggerak bisnis pada sistem, presentasi arsitektur, identifikasi pendekatan arsitektur yang berbeda, menentukan skenario dan persyaratan, analisis skenario dengan peringkat berdasarkan prioritas, realisasi skenario sesuai prioritas, analisis skenario pada arsitektur, dan mempresentasikan hasil (Kazman dkk., 2000; Mellon, 2018).

Meskipun terdapat tahapan yang lengkap, pengujian dapat dilakukan dengan menggunakan empat tahap utama pada metode ini, yaitu pengumpulan skenario dan persyaratan, realisasi skenario, analisis hasil skenario, dan keluaran hasil pengujian (Kazman dkk., 2000). Atas batasan dalam penelitian ini yang dilakukan secara individu, setiap posisi seperti pengembang dan penguji dilakukan oleh satu orang sehingga implementasi seluruh tahapan dalam metode ATAM kurang memungkinkan untuk dilakukan, karena pasa dasarnya terdapat tahapan yang membutuhkan grup dari kumpulan *stakeholder* untuk mencapai keputusan bersama (Kazman dkk., 2000). Karena keterbatasan tersebut, dilakukan modifikasi untuk mencapai kesesuaian terhadap metode ini sebagaimana studi komparasi arsitektur yang menggunakan metode ATAM yang dilakukan pada tahun 2016 (Lou, 2016).



Gambar 3.3. Alur evaluasi dengan ATAM

3.3.1 Tahap Pengujian

3.3.1.1 Pengumpulan Skenario

Pada tahap ini, skenario penggunaan aplikasi secara umum dan skenario penting diidentifikasi. Skenario dapat terbentuk dari pertanyaan-pertanyaan seperti bagaimana proses dialokasikan ke perangkat keras, bagaimana protokol *synchronous* atau *asynchronous* dapat memengaruhi kinerja aplikasi, dan lainnya sebagaimana dalam pertanyaan terkait performa pada panduan ATAM (Kazman dkk., 2000). Pengujian pada penelitian ini pengacu pada perangkat lunak yang spesifik, sehingga skenario yang dirancang akan lebih spesifik juga dengan penyesuaian sebagaimana perangkat lunak yang dikembangkan disesuaikan dengan kebutuhan umum dalam pekerjaan.

Tabel 3.2. Tabel Skenario Pengujian Performa

No.	ID Uji	Skenario	Tujuan
1	TC-01	Membuka aplikasi (tampilan splash sampai halaman login)	Mengetahui *startup time* dan penggunaan sumber daya (CPU, memori, dan energi)
2	TC-02	Membuka aplikasi (tampilan splash sampai halaman chatbot)	Mengetahui *startup time* dan penggunaan sumber daya (CPU, memori, dan energi)
3	TC-03	*Idle* di halaman chatbot setelah screening	Mengetaui penggunaan sumber daya aplikasi setelah pengguna melakukan screening dengan bot
4	TC-04	Navigasi dari halaman detail skor kembali ke halaman chat	Mengetahui perilaku penggunaan sumber daya saat aplikasi menampilkan activity/halaman yang berbeda
5	TC-05	Login	Mengetahui penggunaan sumber daya saat aplikasi menjalankan logika berdasarkan data input pengguna
6	TC-06	Melihat panduan cepat dan kalimat sambutan pertama kali dari bot	Mengetahui penggunaan sumber daya saat aplikasi menampilkan dialog fragment berisi gambar dan

			menampilkan kalimat pada *chat* oleh bot pertama kali
7	TC-07	Memulai *chat* dengan mengatakan halo kepada bot	Mengetahui penggunaan sumber daya saat pengguna mencoba interaksi dengan bot pertama kali
8	TC-08	Melakukan *screening* (pertanyaan dari bot dan jawaban dari pengguna)	Mengetahui penggunaan sumber daya saat dilakukan proses pengambilan data *screening* dari basis data, penjalanan logika untuk menampilkan pertanyaan berurutan, dan menerima jawaban dari pengguna.
9	TC-09	Membuka detail skor *screening*	Mengetahui penggunaan sumber daya saat pengguna akan membuka dialog fragment berisikan data dan navigasi ke *activity*/halaman lain
10	TC-10	*Treatment* melalui curhat	Mengetahui penggunaan sumber daya aplikasi saat dilakukan logika *decision tree* berdasarkan input pengguna

Tabel 3.2 di atas merupakan daftar skenario untuk pengujian performa. Kinerja aplikasi untuk kedua arsitektur MVP dan MVVM akan dilihat saat dijalankannya skenario di atas.

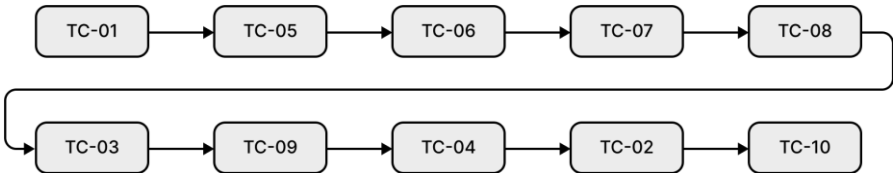
3.3.1.2 Pengumpulan Kebutuhan Pengujian

Kebutuhan pengujian di penelitian ini memerhatikan fokus uji yang dilakukan yaitu performa. Atas hal tersebut, kriteria yang dirancang akan dibuat sebagaimana membuat aplikasi dapat memperlihatkan performanya tanpa adanya gangguan yang berarti atau diperlukan perhatian khusus terhadap lingkungan pengujian yang bermaksud untuk menghilangkan variabel lain yang dapat mengubah hasil pengujian yang sebenarnya.

Tabel 3.3. Daftar Kriteria Khusus Skenario Pengujian

ID Uji	Kriteria Khusus
TC-01	Aplikasi baru dipasang di perangkat dan tidak ada data apapun terkait aplikasi yang tersimpan di perangkat
TC-02	Sudah dilakukan login dan dilakukan aktivitas chat biasa dan screening pada aplikasi
TC-03	Dilakukan setelah skenario TC-08
TC-04	Dilakukan setelah skenario TC-09
TC-05	Akun sudah dibuat sehingga pengujian langsung dilakukan login
TC-06	Dilakukan setelah TC-05
TC-07	Dilakukan setelah TC-06
TC-08	Dilakukan setelah TC-07
TC-09	Dilakukan setelah TC-03
TC-10	Aplikasi ditutup dan dilakukan *Warm Start* (TC-02). Dilakukan screening ulang dengan skor 50. Selanjutnya dilakukan treatment curhat

Tabel 3.3 memerlihatkan daftar kriteria khusus untuk setiap skenario pengujian. Kriteria tersebut dibuat supaya setiap pengujian pada arsitektur maupun perangkat yang berbeda tetap memiliki kondisi yang sama dan mendapatkan hasil pengujian yang sesuai serta tidak adanya faktor atau variabel lain yang mengubah hasil pengujian sebenarnya.



Gambar 3.4 Urutan Skenario Pengujian Berdasarkan Kriteria

Hasil dari pengumpulan kriteria khusus untuk setiap skenario menghasilkan tahapan pengujian yang harus diikuti pada Gambar 3.4.

3.3.1.3 Pendefinisian Arsitektur

Membandingkan beberapa arsitektur memerlukan pendefinisian untuk pemahaman arsitektur yang lebih baik. Setiap arsitektur dideskripsikan baik dari komponen yang membangun arsitektur tersebut dan cara kerjanya. Pada penelitian ini, pendefinisian arsitektur telah dibahas pada Bab 2.

3.3.1.4 Analisis Atribut Khusus

Tahap ini akan melihat fokus penelitian yang dilakukan sehingga menghasilkan keluaran yang sesuai. Untuk bisa mendapatkan hasil yang jelas dan persuasif, pengujian atau eksperimen empiris dilakukan. Pada metode ATAM tidak dijelaskan analisis yang spesifik untuk metode eksperimennya, sehingga proses analisis menjadi fleksibel dan dapat disesuaikan dengan penelitian maupun proyeknya. Pada penelitian ini, pengujian empiris dilakukan dengan menjalankan berbagai skenario dengan acuan parameter seperti pada poin 3.3.3.

3.3.1.5 Identifikasi Sensitivitas

Tahap ini dilakukan berdasarkan hasil yang didapatkan pada tahap ke-4. Tahap ini bertujuan untuk memahami bagaimana hasil pengujian menampilkan nilai tertentu dengan rancangan arsitektur yang dibangun. Identifikasi ini akan menghasilkan pemahaman mengenai kualitas dari arsitektur yang digunakan.

3.3.1.6 Identifikasi *Tradeoffs*

Tahap *tradeoffs* atau pengorbanan merupakan identifikasi mengenai bagaimana keuntungan dan kerugian yang dimiliki dari satu arsitektur dengan yang lainnya. Keuntungan dan kerugian dari pengguaan arsitektur sendiri akan didapatkan setelah setiap tahapan dari ATAM telah dilakukan. Keuntungan dan kerugian dari penggunaan suatu arsitektur sendiri akan memberikan wawasan baru supaya dapat menjadi penguat atas keputusan penggunaan arsitektur dalam proyek tertentu.

3.3.2 Lingkungan Pengujian

Lingkungan pengujian terhadap aplikasi yang telah dikembangkan akan menggunakan perangkat yang dapat memberikan gambaran sesuai dari kinerja penggunaan aplikasi. Pada penelitian ini, digunakan dua perangkat sebagai lingkungan pengujian dari performa aplikasi seperti pada Tabel 3.3. Pengujian dengan perangkat Redmi Note 5 digunakan atas harapan untuk bisa melihat

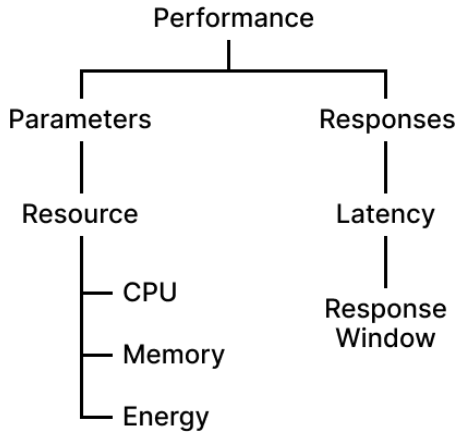
bagaimana performa nyata pada perangkat yang terbiasa digunakan untuk kegiatan keseharian. Penggunaan emulator Google Pixel 2 digunakan atas harapan untuk bisa mengetahui bagaimana performa terbaik dari setiap aplikasi yang menggunakan arsitektur MVP dan MVVM.

Tabel 3.3. Perangkat untuk Pengujian Performa Aplikasi

	Perangkat Pengujian	
	Redmi Note 5	Google Pixel 2 (Emulator)
Chipset	Qualcomm Snapdragon 636	-
CPU	Octa-core	-
GPU	Adreno 509	-
OS	Android 9.0	Android 9.0
Memori	3 GB	2 GB
Kondisi	Berbagai aplikasi berjalan di *background*	Bersih tanpa aplikasi di *background*

3.3.3 Metrik Pengujian

Mengikuti fokus pengujian yaitu performa penggunaan arsitektur pada aplikasi Android, metrik pengujian berawal dari karakteristik performa yang telah menjadi standar *architectural parameter* dalam metode ATAM (Kazman dkk., 2000). Karakteristik performa yang diambil disesuaikan dengan tujuan penelitian ini dan karakteristik sebagai kebutuhan terpenting pada aplikasi *mobile* seperti pada Gambar 3.5.



Gambar 3.5. Parameter karakteristik performa yang digunakan

Setiap metrik pada Gambar 3.5 menjadi acuan observasi pada pengujian. Observasi sendiri akan memanfaatkan alat yang telah disediakan pada lingkungan pengembangan IDE Android Studio, yaitu Android Profiler. Setiap komponen metrik performa akan diperhatikan sebagai berikut:

3.3.3.1 CPU

Penggunaan CPU dari kedua arsitektur MVVM dan MVP akan diperhatikan saat dijalankannya skenario yang telah dibuat. Penggunaan CPU sendiri akan dilihat berdasarkan besar persentase (%) dari besar CPU yang dimiliki pada perangkat pengujian yang telah dipaparkan pada poin 3.3.2. Penggunaan CPU juga akan dikatakan sebagai yang lebih baik ketika dijalankannya proses sesuai skenario dan penggunaan CPU dari satu arsitektur lebih sedikit dibandingkan arsitektur yang lainnya.

3.3.3.2 Memory

Penggunaan memori perangkat menjadi komponen parameter pengujian. Berdasarkan pemaparan cara kerja arsitektur MVP dan MVVM pada bab dua, pola penggunaan memori akan turut berbeda. Penggunaan memori akan dikatakan lebih baik ketika satu arsitektur dapat menggunakan lebih sedikit memori dibandingkan yang lainnya saat dijalankan skenario pengujian yang sama. Penggunaan memori sendiri akan dihitung dalam satuan Mega Byte (MB)

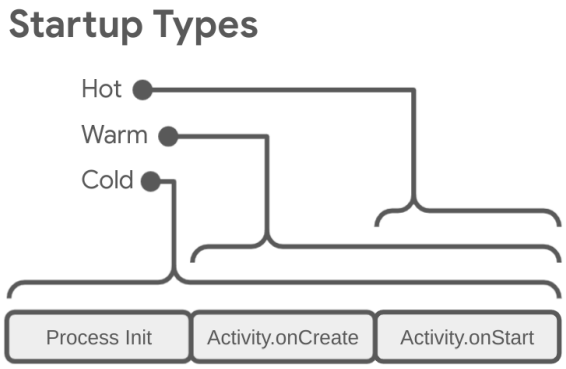
3.3.3.3 Energy

Besaran energi yang digunakan berbanding lurus dengan penggunaan sumber daya lain seperti CPU dan memori. Semakin rendah penggunaan energi pada pengujian sesuai skenario maka akan semakin baik. Pengukuran energi dalam Android Profiler akan dibagi menjadi tiga kategori, yaitu *light*, *medium*, dan *heavy* dengan besaran energi secara rinci pada grafiknya.

3.3.3.4 Startup

Startup atau membuka aplikasi akan dihitung dari lamanya waktu eksekusi. Skenario untuk pengujian ini dapat berupa pengujian waktu saat aplikasi dibuka (*app startup time*) maupun saat satu pengguna melakukan input yang membuat *event* untuk membuka suatu halaman. Terdapat submetrik pada parameter ini, yaitu *Time To Initial Display* (TTID) yang mengukur seberapa lama waktu yang

digunakan saat aplikasi memperlihatkan *frame* pertama, dan *Time to Full Display* (TTFD) yang mengukur seberapa lama waktu yang digunakan saat aplikasi menampilkan *frame* pertama dengan isi kontennya. Obrservasi pada parameter ini akan dilakukan melalui Logcats pada IDE Android Studio.



Gambar 3.6. Tipe *startup* pada aplikasi Android

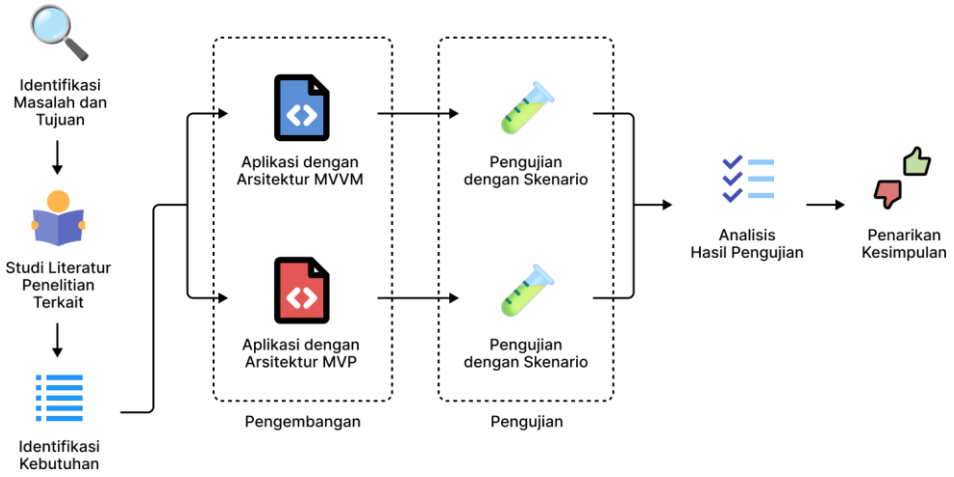
Pada dokumentasi pengembangan Android, terdapat tiga tipe untuk *startup*, yaitu *cold start*, *warm start*, dan *hot start*. Berdasarkan dokumentasi, penggunaan waktu dikatakan berlebihan saat:

- *Cold startup* memakan waktu 5 detik atau lebih
- *Warm startup* memakan waktu 2 detik atau lebih
- *Hot startup* memakan waktu 1,5 detik atau lebih

Pada penelitian ini, data yang diambil dari metrik waktu kecepatan membuka aplikasi atau *startup* akan melihat pada *startup* tipe *cold startup* dan *warm startup*. Poin metrik tersebut diambil karena merupakan tipe *startup* yang paling menantang untuk sebuah aplikasi sebagaimana yang dinyatakan pada dokumentasi Android.

3.4 Prosedur Penelitian

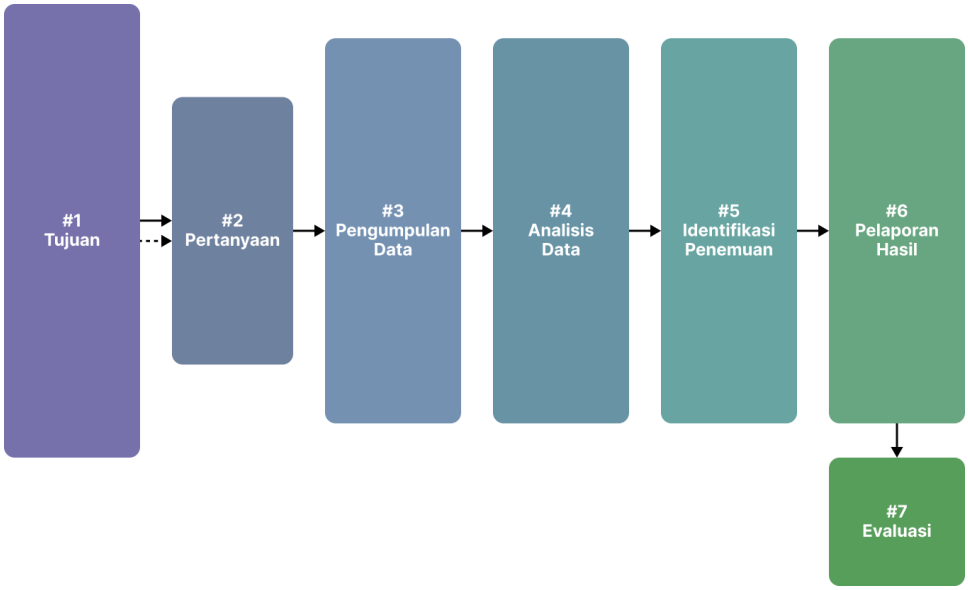
Sesuai dengan desain penelitian yang telah dijabarkan di atas beserta desain pengembangan dan desain pengujian, maka secara garis besar penelitian akan dijalankan seperti pada bagan berikut:



Gambar 3.7. Garis Besar dari Prosedur Penelitian

3.5 Analisis Data

Analisis data merupakan sebuah cara pengolahan data menjadi informasi yang menyangkut hubungan, keputusan, dan ide yang bertujuan untuk membantu mencapai tujuan dari pekerjaan maupun penelitian. Berdasarkan (Richmond, 2006) mengenai analisis data, terdapat tahapan yang perlu diikuti sebagaimana menjadi tahapan terstruktur dan terorganisasi pada proses analisis data.



Gambar 3.8. Tahap Analisis Data dalam Linear

Gambar 3.8 memerlihatkan tahapan dari analisis data dalam linear yang digunakan dalam analisis data di penelitian ini.

3.5.1 Tujuan

Tujuan dari dilakukannya analisis data adalah sebagaimana tujuan yang ingin dicapai pada penelitian ini yang telah dijabarkan pada poin 1.2.

3.5.2 Pertanyaan

Dalam penelitian ini, pengujian empiris dilakukan atas tujuan untuk mengetahui bagaimana performa antara penerapan arsitektur MVP dan MVVM pada aplikasi *chatbot*. Maka dari itu, terdapat beberapa hal yang perlu diketahui terkait tujuan tersebut sebagaimana metrik yang telah dijabarkan pada pon 3.3.3, yaitu:

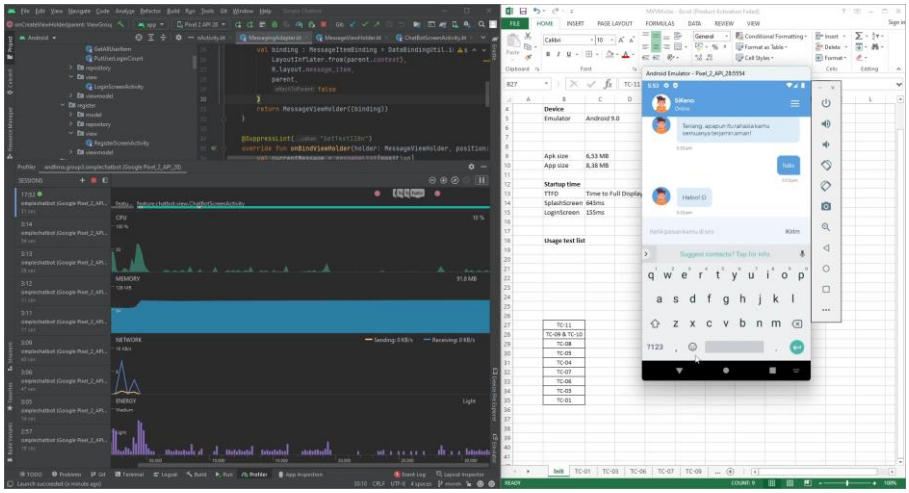
1. Bagaimana kinerja CPU pada arsitektur MVP dan MVVM?
2. Bagaimana penggunaan memori pada arsitektur MVP dan MVVM?
3. Bagaimana komsumsi energi pada arsitektur MVP dan MVVM?
4. Bagaimana kecepatan membuka aplikasi pada arsitektur MVP dan MVVM?

3.5.3 Pengumpulan Data

Tahap ini akan berjalan sejalan dengan tahapan analisis atribut khusus pada poin 3.3.1.4. Komponen data yang akan dikumpulkan pada penelitian ini sesuai dengan metrik pengujian pada poin 3.3.3. Atas hal tersebut, terdapat data yang ingin didapatkan yaitu:

- Penggunaan sumber daya setiap interval waktu tertentu
- Rata-rata penggunaan sumber daya setiap skenario
- Rata-rata total dari penggunaan sumber daya setiap metrik
- Perbandingan antara durasi membuka aplikasi antar arsitektur

Untuk mendapatkan data-data tersebut, diperlukan data awal yang dihasilkan dari pengujian. Data untuk metrik penggunaan sumber daya seperti penggunaan CPU, memori, dan energi dilihat menggunakan alat yang telah tersedia pada IDE Android Studio yaitu Android Profiler. Dengan Android Profiler, peneliti dapat melihat penggunaan sumber daya secara *real time* saat aplikasi sedang digunakan.



Gambar 3.9. Pengumpulan Data Penggunaan Sumber Daya

Gambar 3.9 memerlihatkan contoh saat pengujian sedang berlangsung. Penggunaan cara ini untuk mendapatkan data penggunaan sumber daya dapat menghasilkan data dalam interval waktu 200 detik sehingga kerincian penggunaan sumber saya dapat dilihat dengan baik.

Data yang telah didapatkan akan dikumpulkan dengan teratur menggunakan tabel. Tabel dibuat sebagai dokumen dari setiap komponen metrik yang diteliti. Setiap variabel akan direpresentasikan dalam simbol yang disesuaikan, seperti *timestamp* (T), *CPU usage* (CU), *Memory usage* (MU), *Energy* (E), dan *Energy's score* (ES).

Tabel 3.4. Tabel Data Pengujian Pengguaan Sumber Daya

Activity	Timestamp (ms)	CPU (%)	Memori (MB)	Energi	Energi (skor)
$Activity_1$	T_1	CU_1	MU_1	E_1	ES_1
$Activity_2$	T_2	CU_2	MU_2	E_2	ES_2
$Activity_{...}$	$T_{...}$	$CU_{...}$	$MU_{...}$	$E_{...}$	$ES_{...}$
$Activity_n$	T_n	CU_n	MU_n	E_n	ES_n

Tabel 3.4 di atas merupakan tabel yang akan digunakan dalam pengumpulan data penggunaan sumber daya aplikasi dengan arsitektur yang akan diuji. Dengan tabel

tersebut, ketiga metrik penggunaan sumber daya dari CPU, memori, dan energi langsung diambil salam satu skenario pengujian.

Terkhusus untuk EC, untuk memudahkan pengumpulan dan analisis data, hasil pengkategorian akan dikonversikan menjadi skor kategori konsumsi energi yang ditampilkan di Android Profiler. Pada *tool* tersebut, konsumsi energi tidak disebutkan nilai variabel spesifik karena berdasarkan dokumentasi pengembang Android oleh Google penggunaan energi merupakan nilai yang relatif terhadap perangkat keras penyusun perangkat yang digunakan terutama besar kapasitas baterai pada perangkat. Atas hal tersebut, setiap kategori pada Tabel 3.5 diterjemahkan menjadi skor supaya hasilnya dapat dilakukan perhitungan.

Tabel 3.5. Nilai Skor Kategori *Energy Consumption*

Kategori	Skor
None	0
Light	1
Medium	2
Heavy	3

Tabel 3.5 di atas merupakan nilai konversi dari kategori yang dihasilkan dari profiler menjadi skor. Dengan konversi tersebut diharapkan pengumpulan dan analisis data menjadi lebih mudah dan dapat memerlihatkan hasil yang ingin dicapai.

```
mainHandler.postAtFrontOfQueueAsync {
  reportFullyDrawn()
}
```

Gambar 3.10. Metode Dalam Pengumpulan Data Waktu *Startup*

Untuk mendapatkan waktu *startup* yang tepat dan akurat, diperlukan cara yang sesuai dan dapat diimplementasikan pada kode. Gambar 3.10 adalah cara yang digunakan untuk mengetahui waktu *startup* saat mencapai TTFD untuk setiap arsitektur MVP dan MVVM.

```
2022-12-21 08:32:08.193 2092-3184/? I/ActivityManager: Fully drawn andlima.group3.simplechatbot/.feature.chatbot.view.ChatBotScreenActivity: +1s462ms
```

Gambar 3.11. Contoh Hasil Data Waktu *Startup*

Gambar 3.11 di atas memerlihatkan hasil dari dilakukannya proses pengujian waktu *startup*. Perlu diperhatikan bahwa pada sistem operasi Android, aplikasi dikatakan terbuka sepenuhnya saat TTFD tercapai. Waktu tersebut dapat dicapai saat tampilan masih pada halaman *splash*. Namun, pengartian *startup* untuk pengguna seringkali dikatakan saat aplikasi sudah dapat digunakan oleh penggunam, sehingga untuk pengujian ini TTFD juga disertakan pada tampilan halaman kedua setelah *splash* yang mengartikan bahwa aplikasi sudah dapat digunakan oleh pengguna saat total waktu TTFD kedua halaman sudah tercapai. Data tersebut akan diambil dan dikumpulkan pada Tabel 3.7.

Tabel 3.6. Tabel Data Waktu *Startup* untuk Satu Perangkat Pengujian

Activity	Cold Start		Warm Start	
	MVP	MVVM	MVP	MVVM
$Activity_1$	$Durasi_1$	$Durasi_1$	$Durasi_1$	$Durasi_1$
$Activity_2$	$Durasi_2$	$Durasi_2$		
$Activity_3$			$Durasi_2$	$Durasi_2$
Total	$Total_1$	$Total_2$	$Total_3$	$Total_4$

Tabel 3.6 di atas memerlihatkan tabel untuk pengumpulan data dalam pengujian kecepatan membuka aplikasi untuk setiap arsitektur dan pada jenis *startup* yang berbeda. Dengan menggunakan tabel tersebut setiap data yang dikumpulkan dilakukan untuk setiap perangkat yang berbeda dari setiap tipe *startup* yang ada.

3.5.4 Metode Analisis Data

Menggunakan data yang didapatkan dari tahap sebelumnya, data dapat segera dibuat visualisasi untuk membantu analisis data yang lebih dalam. Visualisasi data tidak menggantikan analisis, tetapi menjadi landasan yang efektif untuk memandu analisis selanjutnya (Richmond, 2006). Selanjutnya dilakukan analisis komparatif terhadap data yang didapatkan dari pengujian pada arsitektur MVP dengan MVVM. Sebagian data yang didapatkan sudah bisa dilakukan komparasi secara langsung untuk memiliki data yang ingin di dapatkan pada poin 3.5.3. Namun, untuk rata-rata penggunaan dan perbandingan durasi riil dengan target dibutuhkan perhitungan seperti di bawah.

1. Rata-rata Penggunaan

$$\bar{x}_{arsitektur} = \frac{\sum_{n=1}^{\infty} X_{data}}{N_{data}} \quad (1)$$

Dengan $\bar{x}$ merupakan rata-rata arsitektur dari satu skenario yang diuji, X_{data} merupakan total jumlah data yang ada pada satu skenario, dan N_{data} merupakan total banyaknya data yang didapatkan pada satu skenario.

2. Persentase Perbedaan Hasil Rata-rata

$$\%D_{(ID\ Uji)} = \frac{|\bar{x}_{ars1} - \bar{x}_{ars2}|}{\bar{x}_{uji}} \times 100 \quad (2)$$

Dengan *%D* merupakan *percent difference* atau persentase perbedaan, $\bar{x}_{ars}$ merupakan rata-rata dari satu arsitektur dalam satu skenario, dan $\bar{x}_{uji}$ merupakan rata-rata keseluruhan di skenario.

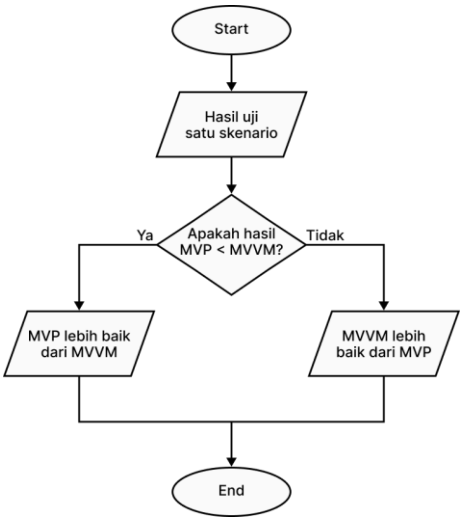
3. Uji T

Uji T merupakan pengujian statistik untuk mencari bagaimana perbedaan yang didapatkan dari hasil pengujian yang telah dilakukan. Uji T dapat memerlihatkan keberadaan perbedaan signifikan antara dua kelompok data. Hasil pengujian komparasi memiliki kemungkinan bahwa hasil pada pengamatan tidak signifikan karena adanya kesalahan pengambilan sampel. Namun, jika uji T memerlihatkan hasil yang signifikan maka hasil pengujian memerlihatkan karakteristik yang sebenarnya daripada kesalahan atau kebetulan pengambilan sampel. Pada penelitian ini, uji T akan dilakukan untuk melihat keberadaan perbedaan signifikan dari hasil pengujian komparasi arsitektur MVP dan MVVM. Uji T akan dilakukan menggunakan analisa data dari Microsoft Excel untuk mempermudah mendapatkan hasil dan menghindari adanya kesalahan dari kalkulasi data secara manual.

3.5.5 Identifikasi Penemuan

Setelah data didapatkan, dilakukan identifikasi. Identifikasi pada tahap ini akan bersamaan dengan tahap identifikasi *tradeoffs* pada poin 3.3.1.6 dan juga dapat menjawab dari hipotesis yang telah dibuat pada poin 2.5. Untuk bisa mengetahui arsitektur mana yang lebih unggul pada setiap pengujian, diperlukan hasil

komparasi yang sesuai. Dalam penentuan hasil komparasi, digunakan alur penentuan sebagai berikut.



Gambar 3.12. *Flowchart* Penentuan Hasil Komparasi

Setelah mendapatkan hasil komparasi dari setiap pengujian skenario, hasil akhir akan ditentukan dengan cara yang sama dengan Gambar 3.12 berdasarkan jumlah total hasil positif dari setiap pengujian.

3.5.6 Pelaporan Hasil

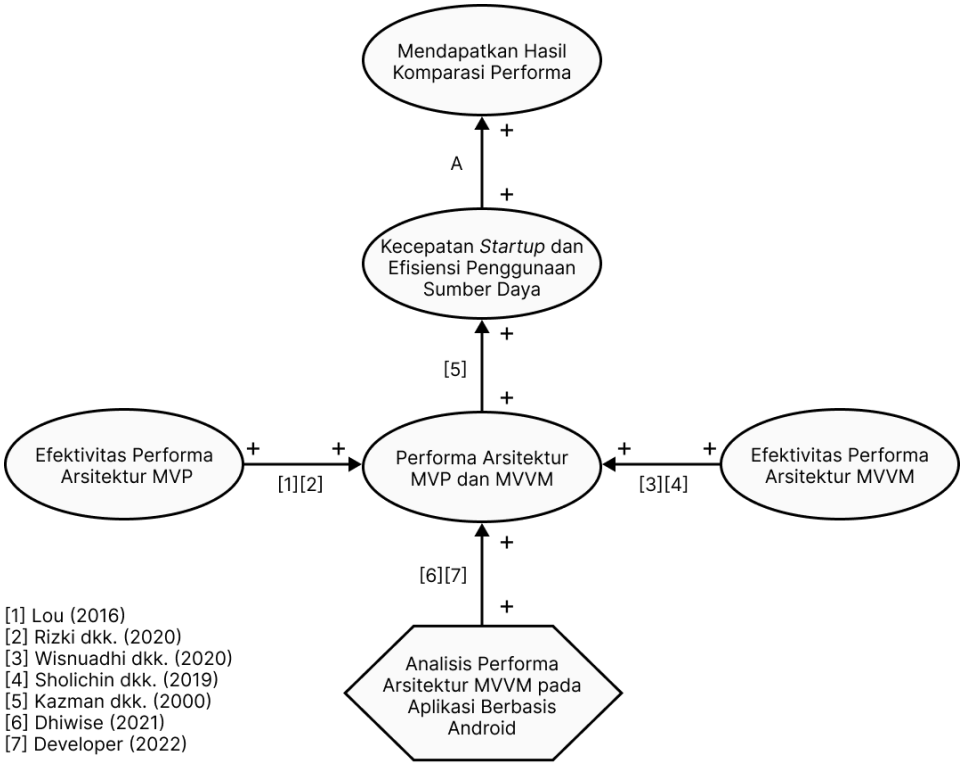
Hasil penemuan akan dilaporkan sebagaimana hasil uji skenario untuk mengetahui performa antara MVVM dan MVP telah didapatkan. Mengikuti panduan pada (Richmond, 2006), terdapat tiga komponen utama dalam penulisan laporan yang mendasar. Pertama, laporan yang baik disusun untuk memberikan informasi dalam urutan yang logis. Kedua, memasukan audiends atau target penelitian yang sudah ditentukan. Ketiga, menuliskan data yang telah dikumpukan, dianalisis, dan ditafsirkan. Dengan pola tersebut, tulisan dapat memberi tahu target peneliti maupun orang lain tentang apa yang baru saja ditemukan.

3.5.7 Evaluasi

Tahap terakhir dari proses analisis data adalah evaluasi. Pada tahap ini, dilakuakan evaluasi terhadap penelitian yang dijalankan beserta data yang ditemukan sebagai penemuan. Hasil dari tahap ini akan dipaparkan bersamaan dengan rekomendasi pada poin 5.2.

3.6 Model Dampak

Berdasarkan model referensi hasil studi literatur pada penelitian terkait beserta metode penelitian yang akan dilakukan, didapatkan model dampak seperti pada Gambar 3.13.



Gambar 3.13. Model Dampak