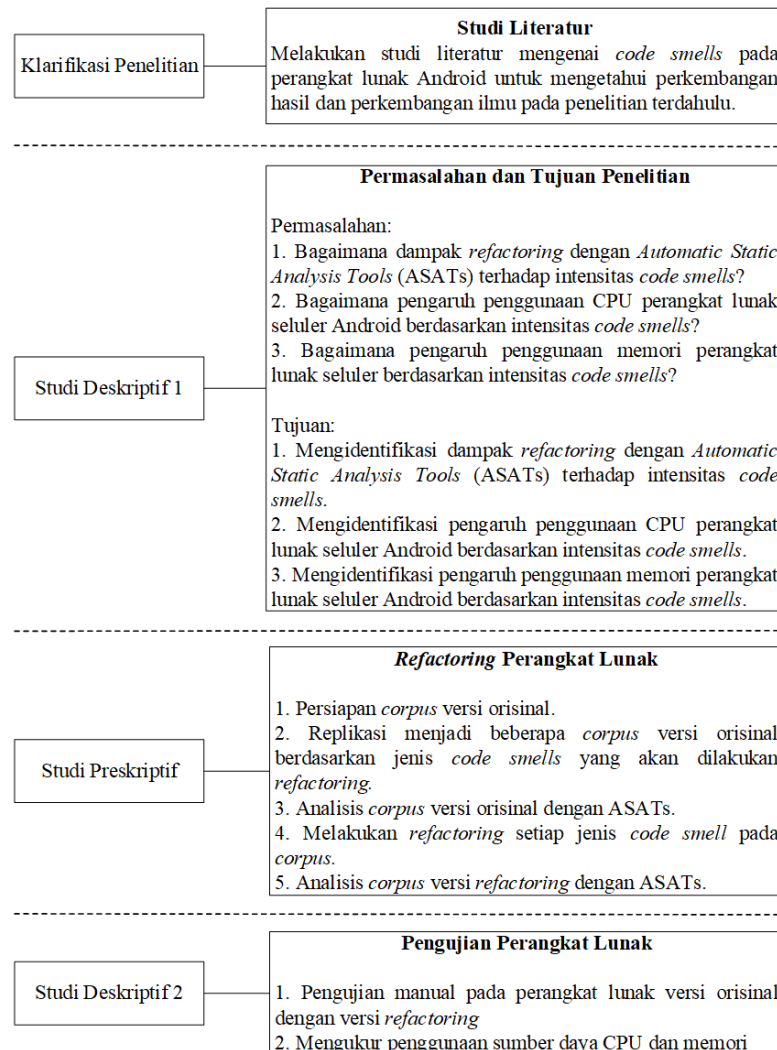


BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian yang digunakan pada penelitian adalah dengan metode *Design Research Methodology* (DRM). Dalam DRM ini terdiri dari empat langkah yaitu Klarifikasi Penelitian atau *Research Clarification* (RC), Studi Deskriptif I atau *Descriptive Study I* (DS-I), Studi Preskriptif atau *Perscriptive Study* (PS), dan Studi Deskriptif II atau *Descriptive Study II* (DS-II) (Blessing & Chakrabarti, 2009). Skema penelitian yang akan dilakukan dalam penelitian dengan berdasarkan kerangka kerja DRM seperti pada **Gambar 3.1**.



Gambar 3.1 Skema Penelitian

Berikut adalah penjelasan dari skema penelitian sebagai berikut:

3.1.1 Klarifikasi Penelitian

Tahapan ini melakukan studi literatur yang relevan untuk mengetahui perkembangan ilmu dan hasil penelitian terdahulu, memfokuskan dan memperjelas tujuan penelitian sesuai dengan topik yang akan diteliti mengenai *code smells*. Kajian artikel penelitian digunakan dalam penelitian sebagai referensi yang relevan. Selain itu buku yang relevan digunakan untuk memperluas landasan teori.

3.1.2 Studi Deskriptif I

Tahapan selanjutnya yaitu mengidentifikasi sebuah permasalahan dan tujuan penelitian dengan melihat permasalahan dan tujuan penelitian yang pernah dilakukan sebelumnya. Permasalahan tersebut merupakan pertanyaan yang akan dipecahkan oleh penelitian berdasarkan latar belakang yang telah dibuat. Selain itu, tujuan pada DS-I adalah tujuan diharapkan dapat dicapai pada penelitian yang dilakukan.

3.1.3 Studi Preskriptif

Tahapan berikutnya yaitu dilakukan *refactoring* perangkat lunak Android. Tahapan yang dilakukan pada PS mengacu pada penelitian terdahulu yang kemudian dimodifikasi untuk mendapatkan nilai penggunaan CPU dan memori yang baik.

3.1.4 Studi Deskriptif II

Tahapan terakhir, pengujian manual perangkat lunak pada perangkat Android dilakukan dengan membandingkan *corpus* orisinal dan *corpus* yang telah dilakukan *refactoring*. DS-II dilanjutkan dengan mengukur penggunaan sumber daya menggunakan perangkat Android dengan ADB dan ObreusDroid untuk mengumpulkan data sumber daya, data yang dikumpulkan tersebut berupa angka nilai yang akan dikelola dan diukur secara manual. Selain itu DS-II akan disimpulkan dari hasil pengukuran sumber daya.

3.2 Corpus Perangkat Lunak Android

Corpus atau kumpulan perangkat lunak Android yang diteliti bersumber dari F-Droid berdasarkan penelitian terdahulu oleh Anwar dkk., (2019) terkait konsumsi energi dalam sebuah artikel ilmiah *conference*. Perangkat lunak tersebut

belum diteliti terkait objek penggunaan sumber daya. Perangkat lunak Android dipilih karena mudah untuk ditemukan *code smells*, karena terdapat kompleksitas kode program. Semua *corpus* dikembangkan dengan bahasa pemrograman Java yang terdiri dari tiga. Walaupun terdapat beberapa bahasa pemrograman Kotlin dalam *corpus*, namun *code smells* pada bahasa pemrograman Java telah didefinisikan lebih tersebar. Fitur perangkat lunak yang diteliti menggunakan sumber daya terhadap komponen non-fungsional yaitu penggunaan kinerja prosesor dan penggunaan memori yang belum diteliti. Perangkat lunak ini telah diunduh lebih dari 10.000 di Google Play Store dengan ulasan ‘4’ (Anwar dkk., 2019). Selain itu terdapat ringkasan mengenai karakteristik perangkat lunak yang diteliti pada **Tabel 3.1**.

Tabel 3.1 Karakteristik perangkat lunak yang diteliti menurut Anwar dkk., 2019

Karakteristik	Calculator	Todo-List	Openflood
Kategori	<i>Tools</i>	<i>Tools</i>	<i>Puzzle</i>
<i>Line of Code</i>	7758	6145	1236
Hijau <i>Line of Code</i>	5935	5466	948
Umur proyek (tahun)	6.8	3	3.1
Unduhan (Google Play Store)	Lebih dari 1.000	Lebih dari 10.000	Lebih dari 10.000
Ulasan (Rerata ulasan pengguna)	4.5	4	4.6

Hijau: mempresentasikan hasil pemindaian dengan SonarQube yang tidak berkaitan dengan menurut Anwar dkk., 2019

3.3 Instrumen Penelitian

Penelitian ini berfokus pada observasi hasil pengujian yang dilakukan pada sistem. Hal ini dilakukan untuk memudahkan proses mengumpulkan, sehingga perlu adanya perangkat seluler dan alat pendukung. Selain itu, terdapat penjelasan secara rinci mengenai perangkat seluler dan alat pendukung serta metrik penelitian yang digunakan:

3.3.1 Perangkat Seluler

Perangkat seluler digunakan untuk pengujian perangkat lunak, perangkat seluler menggunakan sistem operasi Android. Perangkat telah dilakukan proses *rooting* untuk proses mengumpulkan sumber daya secara otomatis. Selain itu, terdapat spesifikasi perangkat secara ringkas pada **Tabel 3.2**.

Tabel 3.2 Spesifikasi perangkat Android Redmi S2 yang digunakan

Spesifikasi Perangkat	Deskripsi
Nama Perangkat	Redmi S2 (YSL)
<i>Central Processing Unit</i> (CPU)	2 GigaHertz (GHz) Octa Core
Chipset	Qualcomm MSM8953 Snapdragon 625
Memori	3 GigaBytes (GB)
Sistem Operasi	Android 10
Status Root	Iya

3.3.2 Alat Pendukung

Alat yang digunakan dalam penelitian ini berupa perangkat lunak dan pustaka (*library*). Alat tersebut bertujuan membantu mengumpulkan dan mengelola hasil penggunaan sumber daya selama pengujian manual dan membantu mendeteksi *code smells*. Selain itu, penjelasan mengenai alat terdapat pada **Tabel 3.3**, sedangkan penjelasan mengenai pustaka terdapat pada **Tabel 3.4**.

Tabel 3.3 Perangkat lunak yang digunakan dalam penelitian

No.	Nama	Deskripsi
1	Android Studio	Android Studio digunakan untuk editor kode program java pada perangkat lunak Android.
2	Android Debug Bridge (ADB)	ADB digunakan untuk komunikasi dengan perangkat Android melalui antarmuka baris perintah.
3	Bash scripting	Bash scripting digunakan untuk mengembangkan alat ObreusDroid dengan tujuan mengumpulkan dan menentukan data sumber daya CPU dan memori dengan pengujian manual.

4	Python 3	Python 3 digunakan untuk menganalisis data sumber daya yang telah dikumpulkan setelah proses pengujian manual.
5	Java 11.0.17	Bahasa pemrograman Java 11.0.17 digunakan untuk menjalankan perangkat lunak Android.
6	SonarQube 9.5	SonarQube 9.5 digunakan untuk meninjau kode program secara otomatis untuk mendeteksi <i>code smells</i> pada perangkat lunak Android.

Tabel 3.4 Pustaka yang digunakan dalam penelitian

No.	Nama	Deskripsi
1	Pandas	Pandas adalah pustaka Python 3 yang digunakan untuk mengolah data.
2	Numpy	Numpy adalah pustaka Python 3 yang digunakan untuk memanipulasi data yang berbentuk array atau matriks.
3	Matplotlib	Matplotlib adalah pustaka Python 3 yang digunakan untuk membuat grafik atau visualisasi data.
4	Scipy	Scipy adalah pustaka Python 3 yang digunakan untuk mengelola data dalam bentuk statistik.

3.3.3 Metrik Penelitian

Metrik penelitian digunakan untuk mengukur dan mengevaluasi hasil modifikasi perangkat lunak dan pengujian perangkat lunak. Pengukuran dan evaluasi hasil *refactoring code smells* yang digunakan dalam penelitian ini menyesuaikan dengan salah satu metrik penelitian milik (Etemadi Someoliayi dkk., 2022). Untuk pengukuran dan evaluasi sumber daya sesuai dengan pedoman ISO/IEC 25023 (Sutino dkk., 2018), dan perbandingan dengan perangkat lunak menyesuaikan dengan metrik penelitian milik (Alkandari dkk., 2021). Selain itu, seluruh metrik penelitian tercantum pada **Tabel 3.5**.

Tabel 3.5 Metrik penelitian sesuai aspek

Aspek	Kode	Metrik	Satuan
<i>Code smells</i>	M1	Rasio <i>refactoring code smells</i>	Persentase (%)
<i>Resource</i>	M2	Rata-rata penggunaan prosesor	Persentase (%)
<i>utilization</i>	M3	Rata-rata penggunaan memori	Persentase (%)
Perbandingan	M4	Perubahan relatif penggunaan prosesor atau penggunaan memori	Persentase (%)

- 1) Rasio *refactoring code smells* (**M1**), menghitung rasio, dengan memperbaiki seberapa banyak *code smells* yang terdeteksi dan diperbaiki. Terdapat formula yang digunakan sebagai berikut:

$$FDR = \frac{|All\ Detected\ Before\ Repair| - |All\ Detected\ After\ Repair|}{|All\ Detected\ Before\ Repair|} \quad (1)$$

$|All\ Detected\ Before\ Repair|$ = *Code smells* yang terdeteksi sebelum diperbaiki.

$|All\ Detected\ After\ Repair|$ = *Code smells* yang telah dilakukan *refactoring* dan dideteksi kembali.

- 2) Rata-rata penggunaan prosesor (**M2**), seberapa besar atau kecil penggunaan prosesor yang digunakan selama proses pengujian manual berdasarkan durasi waktu tertentu. Terdapat formula yang digunakan sebagai berikut:

$$x = \sum_{i=1\ to\ n} (A_i / B_i) / n \quad (2)$$

A_i = Waktu prosesor yang digunakan untuk menjalankan serangkaian tugas yang diberikan dalam observasi

B_i = Waktu operasi untuk melakukan tugas di observasi i

n = Jumlah observasi.

- 3) Rata-rata penggunaan memori (**M3**), seberapa besar atau kecil penggunaan memori yang digunakan selama proses pengujian manual berdasarkan durasi waktu tertentu. Terdapat formula yang digunakan sebagai berikut:

(3)

$$x = \sum_{i=1 \text{ to } n} (A_i / B_i) / n$$

A_i = Ukuran memori aktual yang digunakan untuk melakukan serangkaian tugas pada pemrosesan sampel ke- i

B_i = Ukuran memori yang tersedia untuk melakukan tugas pemrosesan sampel ke- i

n = Jumlah sampel yang diproses

- 4) Perubahan relatif penggunaan prosesor atau penggunaan memori (**M4**), seberapa besar atau kecil perubahan relatif penggunaan prosesor atau penggunaan memori berdasarkan rata-rata penggunaan prosesor atau penggunaan memori. Terdapat formula yang digunakan sebagai berikut:

(4)

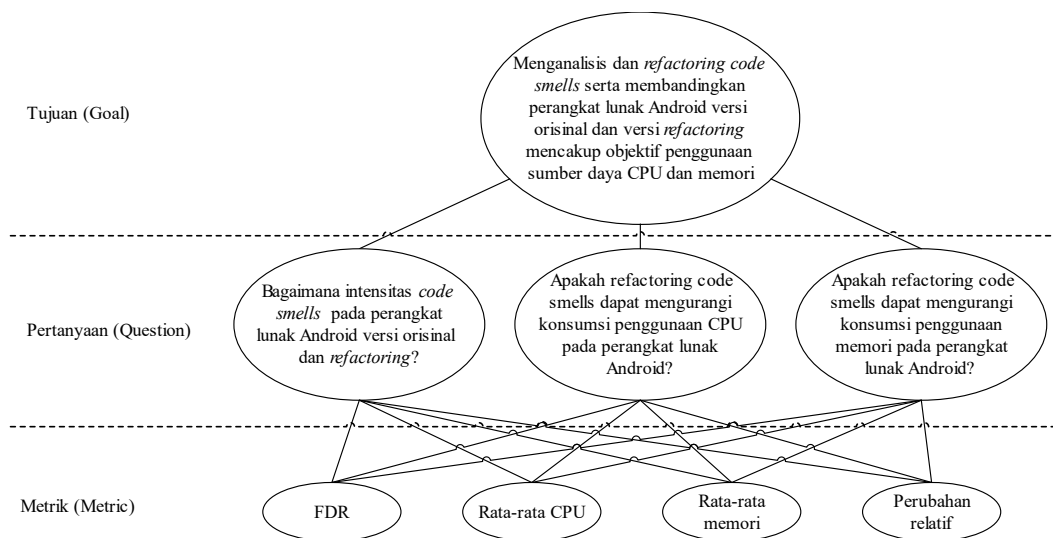
$$C = \frac{x_{refactored} - x_{original}}{x_{original}}$$

$x_{original}$ = Nilai rata-rata milik perangkat lunak versi orisinal

$x_{refactored}$ = Nilai rata-rata milik versi *refactoring*

3.4 Goal Question Metric (GQM)

Metrik penelitian yang diperoleh dari instrumen penelitian (lihat **Subbab 3.3.3**), maka dipetakan berdasarkan tujuan, rumusan masalah, dan metrik yang berkaitan dengan pendekatan GQM. Sehingga dengan GQM dapat menetapkan fokus penelitian. Selain itu, terdapat rancangan GQM yang divisualisasikan dalam bentuk **Gambar 3.2**.

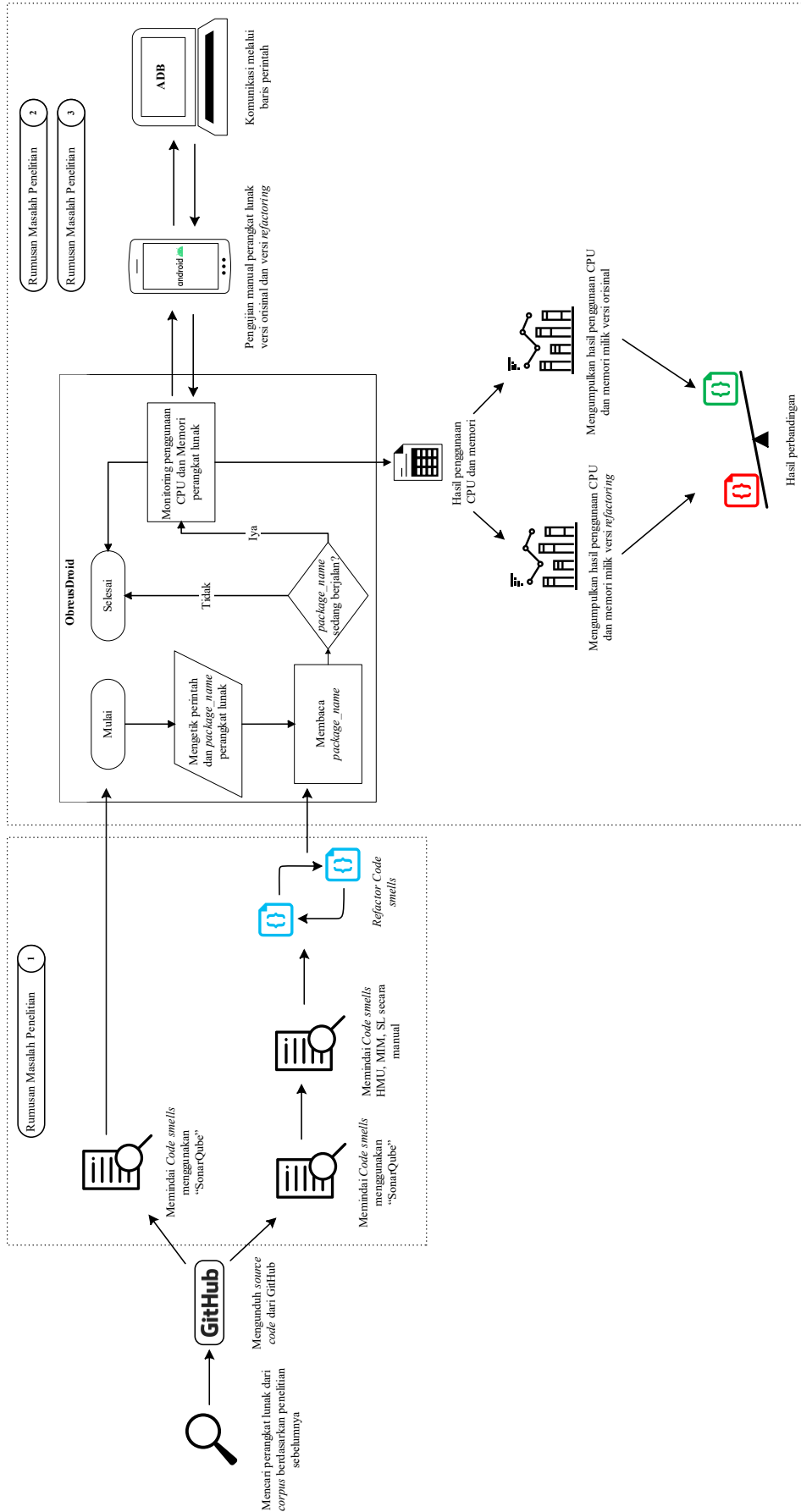


Gambar 3.2 Goal Question Metric (GQM)

Penelitian ini akan mengukur dan membandingkan perangkat lunak versi orisinal dan versi *refactoring* secara kuantitatif. Hasil pengukuran **M1** ditujukan untuk melihat jumlah persentase *code smells* yang diperbaiki (Etemadi Someoliayi dkk., 2022). Pengukuran nilai **M2** dan **M3** dilakukan selama pengujian manual pada perangkat lunak untuk mengetahui nilai rata-rata pada hasil penggunaan CPU dan memori berdasarkan formula ISO/IEC 25023 : 2016. Kemudian dilanjutkan dengan pengukuran **M4**, jika nilai yang dihasilkan pada **M4** bernilai negatif, maka kinerja CPU dan memori meningkat (Alkandari dkk., 2021).

3.5 Prosedur Penelitian

Penelitian ini menggunakan kerangka kerja yang mengacu pada beberapa penelitian terdahulu yang dimodifikasi terkait proses *refactoring code smells*. Modifikasi kerangka kerja dilakukan sesuai beberapa *future work* pada penelitian sebelumnya yaitu mengeksplorasi *code smells* lainnya, pengumpulan data secara otomatis, dan menggunakan perangkat seluler (tidak menggunakan emulator). Kerangka kerja dalam penelitian ini tercantum pada **Gambar 3.3**.



Gambar 3.3 Kerangka Kerja Penelitian

Alur kerangka kerja pada penelitian ini secara umum terdiri dari tiga tahapan besar yaitu persiapan perangkat lunak, *refactoring* perangkat lunak, dan pengujian perangkat lunak. Seluruh tahapan tersebut mengacu pada model DRM yang telah dideskripsikan pada **Gambar 3.1**. Dalam tahapan pertama, persiapan perangkat lunak akan dilakukan analisis menggunakan ASATs. Tahapan kedua yaitu *refactoring* perangkat lunak dimulai dengan memodifikasi perangkat lunak berdasarkan motivasi pada **Gambar 2.5** hingga *code smells* dapat menghilang. Tahapan terakhir yaitu pengujian perangkat lunak dilakukan pengujian manual berdasarkan skenario pengujian (lihat **Lampiran 3**) yang disertai pengumpulan data penggunaan sumber daya CPU dan memori menggunakan ADB dan ObreusDroid. Setelah itu, dilanjutkan dengan menggunakan formula untuk mengukur *code smells* dan penggunaan sumber daya serta perubahan relatif yang dihasilkan.

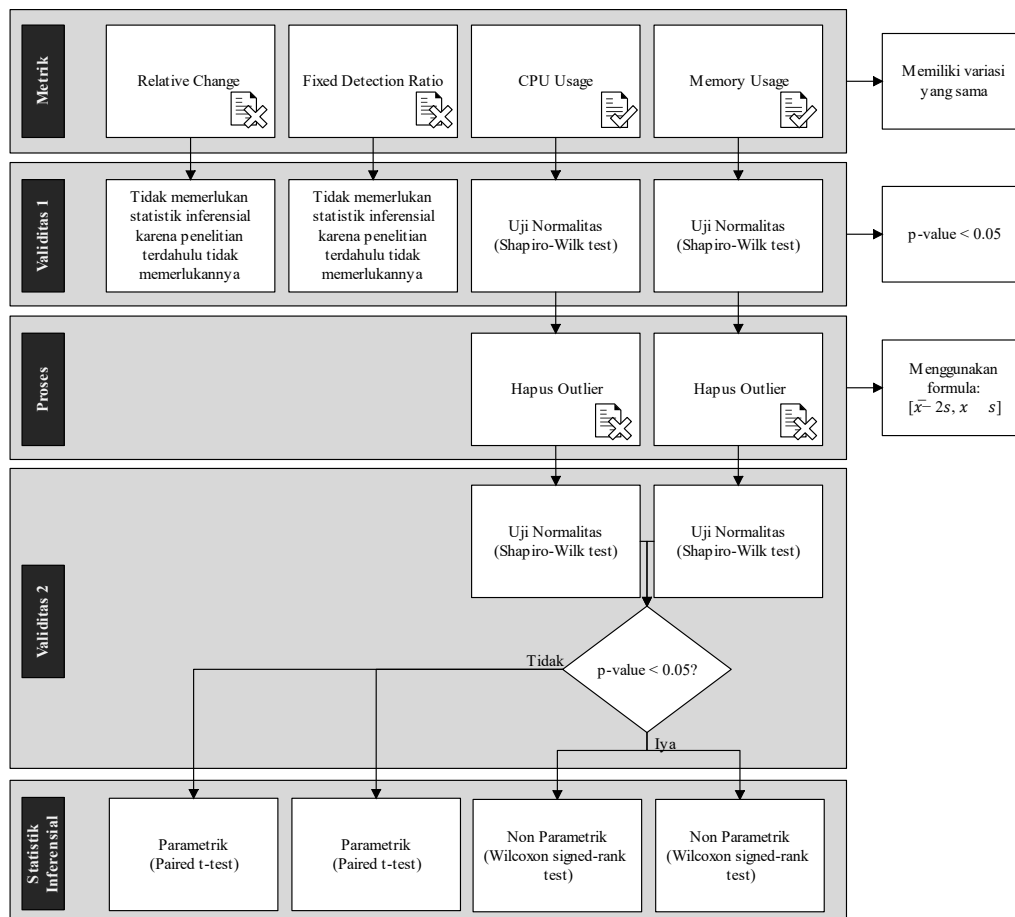
3.6 Teknik Analisis Data

Hasil pengujian yang telah dilakukan sesuai prosedur penelitian di atas, kemudian dilakukan analisis lebih lanjut untuk menguji secara statistik. Data yang dikumpulkan akan dilakukan normalisasi data dengan uji shapiro wilk untuk pengecekan prasyarat uji secara parametrik (Shapiro & Wilk, 1965). Jika hasil nilai uji shapiro wilk signifikan $p = 0,00$ menunjukkan bahwa data tidak terdistribusi dengan normal, yang berarti melanggar uji parametrik (Amrit & Meijberg, 2018). Sehingga perlu dilakukan uji secara non-parametrik, namun jika $p > 0,05$ menunjukkan bahwa data terdistribusi dengan normal dan perlu dilakukan uji parametrik. Sedangkan pada **RQ1** tidak dilakukan analisis statistik, karena tidak diperlukan dan menyesuaikan dengan penelitian terdahulu. Selain itu, terdapat teknik yang berpotensi untuk digunakan dan prasyaratnya.

3.6.1 Teknik Analisis dan Pengujian

Teknik analisis yang digunakan menggunakan teknik *paired t-test* dan *wilcoxon signed-rank test*. Untuk uji *wilcoxon signed-rank*, prasyarat teknik ini menguji jumlah baris data sampel yang sama dan pengujian tidak memerlukan normalisasi (Bluman, 2012, hlm. 672), sebaliknya *paired t-test*, data tersebut harus dilakukan uji normalisasi agar tidak melanggar (Bluman, 2012, hlm. 492). Upaya

yang dilakukan untuk mendapatkan nilai yang terdistribusi normal perlu dilakukan penghapusan *outlier* (Cruz & Abreu, 2017). Selain itu, alur statistik data tercantum pada **Gambar 3.4** formula mengenai *paired t-test*, *wilcoxon signed-rank*, dan formula *outlier* tercantum pada **Lampiran 11**.

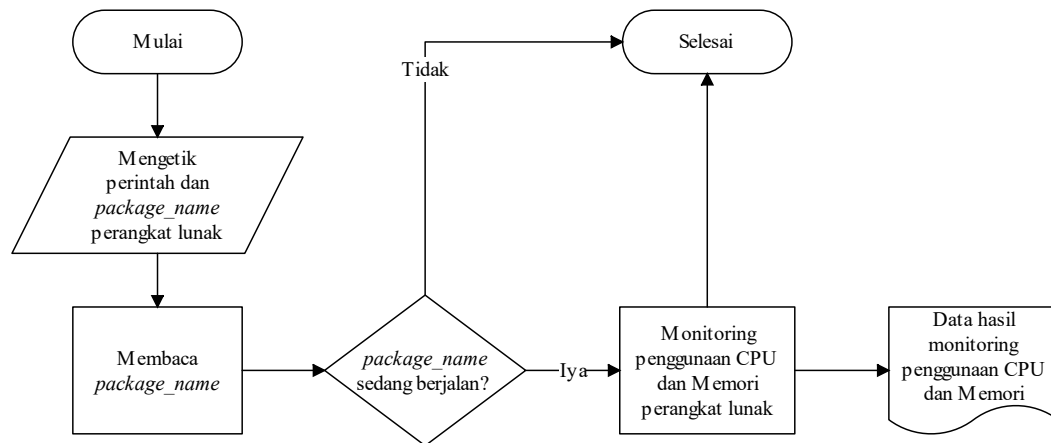


Gambar 3.4 Alur Statistik Data

3.7 Rancangan Sistem ObreusDroid

Penelitian ini menggunakan program sebagai alat untuk membantu observasi sumber daya perangkat lunak terkait prosesor dan memori yang berjalan di perangkat Android bernama ObreusDroid (**Observer resource usage AnDroid**) (Al-Hijri, 2023). Utamanya rancangan sistem ini dibuat dengan Bash Scripting dengan perintah yang disediakan linux, karena sistem operasi Android didasari dari Linux. ObreusDroid disimpan di penyimpanan perangkat seluler dan dijalankan melalui ADB. Sebagai catatan, alat ini digunakan untuk pengujian manual

perangkat lunak berdasarkan skenario pada perangkat lunak, sehingga alat ini digunakan sebagai *testbed* penelitian. Selain itu, diagram rancangan sistem ObreusDroid dapat dilihat pada **Gambar 3.5**.



Gambar 3.5 Rancangan sistem ObreusDroid