

BAB III

METODOLOGI PENELITIAN

3.1. Tahapan Penelitian

Penelitian eksperimental ini dilaksanakan melalui tahapan yang tersusun secara sistematis agar pengujian tetap terarah dan menghasilkan data yang objektif. Mengacu pada kerangka evaluasi kuantisasi yang mencakup dimensi kapasitas, kelugasan, dan efisiensi (Jin et al., 2024), alur penelitian ini terbagi menjadi lima tahapan utama sebagai berikut:

1. Studi Pendahuluan dan Perumusan Masalah.

Tahap awal berfokus pada pemetaan literatur terkini terkait *Mobile Edge Computing* (Sevim et al., 2024; Zhang et al., 2024), *Post-Training Quantization* (Dettmers et al., 2023; Frantar et al., 2023), serta urgensi pemrosesan AI secara *offline* untuk menjamin privasi pengguna (Zhan et al., 2025). Hasil studi pustaka menjadi dasar perumusan hipotesis bahwa kompresi *mixed-precision* berformat GGUF dapat mengatasi keterbatasan RAM 8 GB pada perangkat Android kelas menengah, sebagaimana dikuatkan oleh studi kuantisasi ramah-mobile (Tan et al., 2024) dan analisis *kernel* eksekusi LLM terkuantisasi pada CPU ARM (Gope et al., 2024).

2. Persiapan Lingkungan Komputasi (Environment Setup).

Lingkungan pengujian (*testbed*) dikonfigurasi pada dua perangkat *host* yang saling melengkapi. Perangkat *host* pertama adalah unit komputer personal (PC) yang menjalankan *Windows Subsystem for Linux 2* (WSL2) berbasis sistem operasi Ubuntu dan dilengkapi unit pemrosesan grafis NVIDIA RTX 3060. Komputer ini digunakan

untuk persiapan berkas model dan evaluasi akurasi yang membutuhkan akselerasi komputasi paralel tinggi. Pada PC ini, dilakukan instalasi setelan perangkat pengembangan bahasa C++ seperti build-essential dan cmake, lingkungan Python, CUDA Toolkit, serta pustaka huggingface_hub. Repositori kode *llama.cpp* dikloning dan dikompilasi dengan mengaktifkan opsi DGGML_CUDA=ON agar utilitas llama-quantize dan llama-perplexity dapat memanfaatkan akselerasi GPU secara penuh. Perangkat *host* kedua adalah *smartphone* Android Tecno Pova 5 yang menjalankan aplikasi terminal emulator *non-root* Termux. Gawai ini difungsikan untuk pengukuran performa komputasi secara langsung pada lingkungan sasaran penyebaran. Pada Termux, dilakukan pembaruan paket dasar serta instalasi perangkat pengembangan clang, cmake, dan make. Selanjutnya, pustaka *llama.cpp* dikompilasi ulang secara natif agar instruksinya selaras dengan arsitektur ARM (Ray & Pradhan, 2026).

3. Persiapan Objek Model dan Kuantisasi (Ablation Setup).

Bobot parameter dari model LFM 2.5 berkapasitas 1,2 miliar parameter dan model Qwen 3.5 berkapasitas 2 miliar parameter diunduh dari repositori HuggingFace ke dalam PC menggunakan utilitas terminal huggingface-cli (Touvron et al., 2023). Pemilihan model keluarga Qwen sebagai objek uji kedua didasarkan pada kompetensi pemahaman bahasa Indonesia yang telah terbukti kemampuannya (Nurohim et al., 2025). Bobot mentah dalam format .safetensors tersebut dikonversi menjadi format FP16.gguf melalui skrip bantuan convert_hf_to_gguf.py. Untuk mengeksekusi tahapan uji evaluasi yang memuat gradasi tingkat kompresi, berkas FP16.gguf dikuantisasi menggunakan utilitas ./llama-quantize pada PC dengan bantuan akselerasi prosesor grafis. Proses ini menghasilkan tiga varian presisi campuran (*k-quants*), yaitu Q5_K_M.gguf, Q4_K_M.gguf, dan Q3_K_M.gguf (Dettmers et al., 2023). Keempat

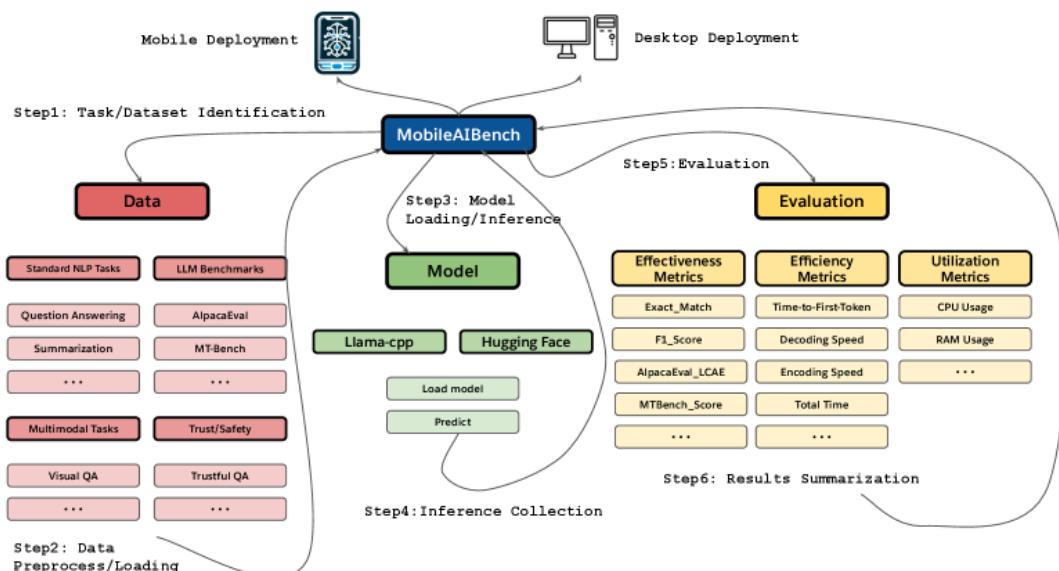
varian berkas dari masing-masing model yang terdiri dari satu berkas dasar FP16 dan tiga berkas varian kompresi kemudian ditransfer ke dalam direktori penyimpanan internal gawai pintar Android sebagai objek uji yang seragam.

4. Eksekusi Eksperimen (System Benchmarking).

Pengujian beban kerja dilaksanakan pada kedua perangkat sesuai dengan pembagian metrik observasi. Pada PC dengan spesifikasi NVIDIA RTX 3060, metrik *Perplexity* dieksekusi melalui modul `./llama-perplexity` dengan menggunakan dataset teks WikiText-2 (Gong et al., 2024). Sementara itu, tingkat akurasi pada tolok ukur MMLU, GSM8K, HumanEval, dan MT-Bench dijalankan melalui skrip Python yang dirancang khusus. Pada perangkat gawai pintar Tecno Pova 5 melalui lingkungan Termux, pengukuran dilakukan terhadap lonjakan penggunaan memori utama tertinggi atau *VmRSS*, *CPU peak*, *Prompt TPS*, *Generation TPS*, dan total waktu eksekusi menggunakan skrip `benchmark.sh`. Parameter inferensi dikontrol pada *temperature* 0,35, *top-p* 0,9, *min-p* 0,05, *repeat penalty* 1,1, jumlah token generasi 1024, *context size* 2048, dan enam *thread* aktif (Zhang et al., 2024). Proses evaluasi dijalankan secara berurutan mulai dari varian FP16 sebagai variabel kontrol, dilanjutkan dengan varian Q5, Q4, dan varian Q3. Setiap varian model diuji tiga kali, dengan jeda *cooldown* manual 3–5 menit antar eksekusi untuk menekan *thermal throttling*. Jeda manual 3–5 menit pada suhu ruangan 25°C ini dimaksudkan untuk menyimulasikan skenario *cold start inference*. Pendekatan ini selaras dengan skenario penggunaan nyata (*use case*) asisten AI pada *smartphone*, di mana pengguna umumnya berinteraksi singkat (*burst workload*) pada kondisi suhu termal yang stabil, bukan dalam kondisi *stress-test* berkepanjangan.

5. Analisis Komparatif dan Penarikan Kesimpulan.

Data mentah yang diperoleh dari pengujian di Termux diekstraksi ke dalam format tabular. Selanjutnya, metrik efisiensi perangkat keras (RAM dan TPS) serta metrik kualitas model (PPL dan akurasi *benchmark*) dianalisis menggunakan kerangka evaluasi tiga dimensi menurut (Jin et al., 2024) untuk mengidentifikasi titik operasional yang paling optimal (*sweet spot*). Alur penelitian lima tahap yang digunakan yaitu studi literatur, penyiapan lingkungan kerja, kuantisasi, pengujian pada perangkat, dan analisis sejalan dengan alur kerja (*workflow*) evaluasi *Post-Training Quantization* (PTQ) pada perangkat lokal yang dipaparkan oleh (Jin et al., 2024) serta (Murthy et al., 2024). Pemetaan antara tahapan penelitian ini dengan enam langkah alur kerja MobileAIBench (Murthy et al., 2024), sedangkan ilustrasi alur kerjanya diadaptasi pada Gambar III.1.



Sumber: (Murthy et al., 2024)

Gambar III. 1
Arsitektur workflow benchmarking SLM/LLM on-device.

3.2. Instrumen Penelitian

Penelitian ini menitikberatkan pada evaluasi kinerja infrastruktur ujung dengan sumber daya terbatas (*resource-constrained edge*) (Ray & Pradhan, 2026). Instrumen yang digunakan terdiri atas dua himpunan perangkat keras (PC sebagai *host* persiapan model dan evaluasi akurasi, serta *smartphone* sebagai *host target deployment*) dan ekosistem perangkat lunak *open-source*, sebagaimana dirinci pada Tabel III.1 dan Tabel III.2.

Tabel III. 1
Spesifikasi Perangkat Keras (Hardware)

Peran	Komponen	Spesifikasi	Keterangan
<i>Host</i> persiapan & evaluasi akurasi	PC + GPU	NVIDIA RTX 3060 (WSL2 Ubuntu)	Eksekusi kuantisasi (<i>llama-quantize</i>), evaluasi <i>Perplexity</i> , dan akurasi MMLU/GSM8K/HumanEval/MT-Bench.
<i>Host target deployment</i>	Perangkat	Tecno Pova 5	<i>Smartphone</i> kelas menengah.
<i>Host target deployment</i>	SoC	MediaTek Helio G99 (ARM big.LITTLE)	<i>CPU-bound inference</i> .
<i>Host target deployment</i>	Memori Utama (RAM)	8GB LPDDR4x	<i>Shared-memory</i> dengan <i>Host OS</i> , alat ukur sekaligus <i>bottleneck force close</i> (Zhang et al., 2024).
<i>Host target deployment</i>	Penyimpanan Internal	UFS 2.2 256GB	Menampung seluruh variasi berkas .gguf.

Sumber: (Penulis)

Tabel III. 2
Spesifikasi Perangkat Lunak (Software)

Peran	Komponen	Versi/Tooling	Keterangan
PC persiapan	Sistem Operasi	Windows 11 + WSL2 Ubuntu	Lingkungan persiapan berkas .gguf dan eksekusi evaluasi akurasi.

Peran	Komponen	Versi/Tooling	Keterangan
PC persiapan	Akselerator	CUDA Toolkit + NVIDIA RTX 3060	Mempercepat <i>llama-quantize</i> dan <i>llama-perplexity</i> .
PC persiapan	Akuisisi model	huggingface_hub	Mengunduh bobot resmi (.safetensors) dari HuggingFace.
PC persiapan	Konversi & kuantisasi	convert_hf_to_gguf.py, ./llama-quantize	Konversi ke FP16.gguf lalu kuantisasi ke Q3/Q4/Q5_K_M
PC persiapan	Evaluasi Kognitif	./llama-perplexity	Mengkalkulasi degradasi linguistik (PPL) pada <i>dataset</i> WikiText-2 (Gong et al., 2024).
PC persiapan	Evaluasi Akurasi	Skrip Python kustom (MMLU, GSM8K, HumanEval, MT-Bench)	Menjalankan 100 sampel acak per <i>benchmark</i> .
Android <i>target</i>	Sistem Operasi Dasar	Android 13	Lapisan manajemen memori inti (<i>Host OS</i>).
Android <i>target</i>	Lingkungan Simulasi Terminal	Termux <i>non-root</i>	Menyediakan fondasi paket Linux murni tanpa membuka enkripsi partisi sistem.
Android <i>target</i>	Mesin Inferensi	<i>llama.cpp</i> (C/C++ <i>bare-metal</i> , ARM-native)	Dikompilasi natif untuk instruksi CPU ARM (Ray & Pradhan, 2026).
Android <i>target</i>	Skrip Otomatisasi	benchmark.sh (Lampiran B)	Menjalankan inferensi, memantau VmRSS + %CPU, dan mencatat TPS ke CSV.

Sumber: (Penulis)

3.3. Metode Pengumpulan Data

Penelitian terapan ini berfokus pada ekstraksi data komputasional yang objektif (*system logging benchmarking*), tanpa bergantung pada instrumen opini subjektif. Sejalan dengan karakteristik kuantitatif eksperimental tersebut, metode pengumpulan data yang diaplikasikan dibatasi pada dua teknik utama: pengamatan langsung (observasi eksperimental terhadap *log* sistem) dan studi pustaka (telaah referensi ilmiah).

1. Observasi dilakukan melalui *system benchmarking logging* untuk memantau indikator performa perangkat keras dan lunak pada kedua *host* saat SLM menjalankan proses inferensi. Pada *host* PC NVIDIA RTX 3060, peneliti mencatat nilai *Perplexity* WikiText-2 dari keluaran `./llama-perplexity` serta skor akurasi MMLU, GSM8K, HumanEval, dan MT-Bench dari skrip Python kustom. Pada *host* perangkat Android Tecno Pova 5 (Termux), konsumsi RAM proses (VmRSS dari `/proc/<pid>/status`), *CPU peak* (%CPU dari `ps`), *Prompt* TPS, dan *Generation* TPS dicatat secara periodik melalui *resource monitor* di dalam skrip `benchmark.sh`. Pencatatan dilakukan pada setiap transisi resolusi kompresi (FP16, Q5_K_M, Q4_K_M, dan Q3_K_M) sehingga dapat dianalisis lintas-*host* untuk menemukan kombinasi optimal antara retensi kualitas kognitif (PC) dan efisiensi *runtime* (Android) (Jin et al., 2024; Zhang et al., 2024). Pemantauan konsumsi memori (*Resident Set Size*/VmRSS) pada perangkat Android dilakukan dengan interval waktu 0,5 detik. Interval ini dipilih sebagai *trade-off* metodologis; interval yang terlalu singkat (misalnya 0,1 detik) berisiko menambah *CPU overhead* dari *daemon* pemantau sehingga dapat mendistorsi data TPS pada `llama.cpp` di lingkungan *non-root*.
2. Studi Pustaka Pengumpulan data sekunder dilakukan dengan menghimpun dan memetakan kajian literatur dari jurnal akademik mutakhir (periode 2023–2026) yang membahas PTQ (Dettmers et al., 2023; Frantar et al., 2023), arsitektur model bahasa terbuka (Touvron et al., 2023), teknik kompresi berbasis aktivasi (Lin et al., 2026), serta fenomena *Mobile Edge Computing* (Zhang et al., 2024) (Sevim et al., 2024). Urgensi lokalisasi pemrosesan AI untuk menjamin privasi data (Zhan et al., 2025) dan kebutuhan sistem respons cerdas yang instan pada sektor layanan publik (Fadillah Ahmad et al., 2024) turut memperkuat justifikasi eksperimental

penelitian ini. Studi kepustakaan ini menjadi landasan dalam penentuan parameter kalibrasi pengujian, sehingga skenario *stress-test* yang diterapkan sejalan dengan standar evaluasi performa AI pada perangkat dengan sumber daya terbatas (Ray & Pradhan, 2026).

3.4. Metode Analisis Data

Data metrik mentah (*raw data log*) yang diekstraksi dari perangkat uji selanjutnya diolah menggunakan Analisis Komparatif Bertingkat. Metode ini mengkaji selisih kinerja (*performance gap*) antara model presisi murni (FP16 sebagai *baseline*) dengan level kompresi di bawahnya, yaitu Q5_K_M, Q4_K_M, dan Q3_K_M (Dettmers et al., 2023; Frantar et al., 2023).

Berlandaskan kerangka evaluasi tiga dimensi (*Three-Dimensional Evaluation Framework*) (Jin et al., 2024), data diolah untuk memproduksi tiga matriks kalkulasi utama berdasarkan kompartemen perangkat kerasnya:

1. Analisis Akurasi Kognitif dan Linguistik (Eksplorasi *Host* PC RTX 3060)

Evaluasi akurasi tugas kognitif (MMLU, GSM8K, dan HumanEval) serta nilai *Perplexity* (WikiText-2) difokuskan murni pada perolehan skor logika, bukan pada metrik kecepatan (*throughput*). Guna mengeliminasi variansi acak, nilai akurasi untuk kedua model uji (LFM 2.5 dan Qwen 3.5) merupakan nilai rerata matematis (*mean*) yang diperoleh dari 3 kali perulangan pengujian (*3 evaluation runs*) berbasis 100 sampel acak per instrumen. Sementara itu, metrik keragaman leksikal (*MT-Bench Type-Token Ratio / TTR*) dianalisis secara independen dan dikeluarkan dari kalkulasi rata-rata akurasi komposit untuk menjaga konsistensi data.

2. Analisis Efisiensi Perangkat Keras dan Infrastruktur (Eksplorasi *Target Edge* Android)

Data *runtime* dirangkum sebagai rata-rata dari tiga kali pengujian yang dilakukan langsung pada *smartphone*. Data *peak* RAM diolah menjadi persentase reduksi untuk mengevaluasi apakah sistem Android berisiko mengalami *Force Close* (Zhang et al., 2024). Formula reduksi memori yang digunakan adalah:

$$\text{Reduksi Memori (\%)} = \frac{RAM_{FP16} - RAM_{Kuantisasi}}{RAM_{FP16}} \times 100\%$$

Bagi varian yang presisi FP16-nya gagal dimuat akibat kehabisan ruang memori (*Out of Memory / OOM*), data dicatat sebagai *hardware limit*. Selain itu, data waktu pemrosesan dikonversi menjadi kecepatan inferensi atau TPS (*total token* dibagi durasi eksekusi dalam detik).

3. Analisis Degradasi Kualitas Kognitif

Perubahan kualitas bahasa model dianalisis melalui nilai *Perplexity* (PPL) dan margin peningkatannya pascakompresi (Gong et al., 2024). Nilai PPL bersifat *inverse*, artinya semakin tinggi PPL setelah kompresi dibandingkan *baseline*, semakin besar penurunan kualitas model akibat hilangnya *outliers* (Gong et al., 2024). Risiko ini dapat ditekan melalui pendekatan kompresi berbasis perlindungan bobot *salient weights* (Lin et al., 2026).

Data akhir dari matriks efisiensi (RAM dan TPS) selanjutnya dikorelasikan secara grafis dengan matriks kognitif (PPL dan akurasi *benchmark*). Pemilihan metrik *peak* RAM, *prompt/generation speed*, dan total waktu eksekusi sebagai instrumen utama

konsisten dengan kerangka MobileAIBench yang dirancang khusus untuk *benchmarking* LLM dan LMM pada ekosistem *on-device* (Murthy et al., 2024).

Melalui analisis kuantitatif ini, peneliti menetapkan titik *sweet spot*, yaitu varian *k-quants* yang paling optimal untuk dijalankan pada *smartphone* Android berkapasitas RAM 8 GB (Jin et al., 2024; Ray & Pradhan, 2026). Untuk memvalidasi signifikansi perbedaan antar-varian secara inferensial, digunakan dua uji statistik: *Welch's t-test* untuk perbandingan pasangan dan *One-way ANOVA* untuk perbandingan simultan keempat varian pada taraf signifikansi $\alpha = 0,05$.

Untuk menentukan varian terbaik secara kuantitatif, penelitian ini menggunakan formula *Compound Score* yang menggabungkan *Generation Speed*, persentase retensi akurasi, dan efisiensi RAM menjadi satu skor komposit:

$$\text{Compound Score} = \frac{\text{Gen TPS} \times \text{Retensi Akurasi}}{\text{RAM (GB)}}$$