

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Perkembangan kecerdasan buatan, khususnya pemrosesan bahasa alami (NLP), meningkat pesat sejak munculnya *Large Language Model* (LLM). Model berukuran besar ini meningkatkan kemampuan sistem dalam memahami konteks kalimat dan menghasilkan teks yang menyerupai bahasa manusia. Meskipun memiliki kinerja yang tinggi, LLM modern membutuhkan infrastruktur komputasi yang memadai, terutama *Graphics Processing Unit* (GPU). Ketergantungan pada perangkat keras kelas atas menimbulkan kendala ketika model dijalankan pada perangkat dengan spesifikasi standar, terutama dalam bentuk kegagalan alokasi memori (*Out of Memory/OOM*). Studi (Laskaridis et al., 2024) menunjukkan bahwa inferensi LLM pada platform seluler bersifat *memory-bound*, sehingga risiko OOM meningkat pada perangkat dengan kapasitas RAM terbatas.

Di sisi lain, kebutuhan mobilitas digital mendorong pengembangan asisten virtual yang mampu bekerja secara mandiri dan responsif tanpa bergantung pada koneksi jaringan. Kebutuhan praktis ini terwujud pada berbagai implementasi sektoral yang krusial, mulai dari sistem informasi rumah sakit (Fadillah Ahmad et al., 2024) hingga integrasi simpul LLM dalam infrastruktur pemantauan jaringan nirkabel pada ekosistem kawasan urban (Sevim et al., 2024). Ketergantungan AI pada layanan cloud komersial melalui API dapat menimbulkan risiko keamanan, terutama ketika sistem memproses data pribadi pengguna. Untuk mengurangi risiko kebocoran data sensitif, seperti rekam medis dan informasi pribadi, pemrosesan AI perlu diarahkan ke

perangkat lokal melalui pendekatan *Edge Intelligence* agar dapat bekerja secara mandiri dan *offline*, misalnya pada *smartphone* Android (Zhan et al., 2025).

Untuk menjawab kendala komputasi dan kebutuhan privasi, *Small Language Model* (SLM) dirancang dengan jumlah parameter yang lebih ringkas agar lebih sesuai dengan kapasitas perangkat personal. Namun, kendala struktural timbul dari keterbatasan fondasi arsitektur fisik gawai konsumen. Sebagian besar *smartphone* kelas menengah hanya memiliki RAM sekitar 8 GB, yang digunakan bersama oleh sistem operasi Android dan layanan latar belakang. Arsitektur tersebut tidak dirancang untuk menahan beban komputasi matriks model AI yang berada dalam bentuk presisi murni *Floating-Point* 16-bit (FP16). Jika model asli tetap dimuat, sistem Android dapat menghentikan proses secara paksa (*force close*) untuk mencegah kehabisan memori. Demi menghindari kegagalan operasional ini, penerapan teknik optimasi pada aspek *Edge Intelligence* menjadi prosedur yang mutlak diperlukan (Zhang et al., 2024).

Salah satu metode yang umum digunakan untuk mengurangi ukuran model agar dapat dijalankan pada RAM perangkat seluler adalah *Post-Training Quantization* (PTQ). Berbagai kajian menunjukkan bahwa pemadatan bobot ke presisi 3-bit hingga 5-bit dapat menurunkan kebutuhan memori hingga kurang dari sepertiga ukuran awalnya tanpa melalui proses pelatihan ulang (*retraining*) yang membutuhkan biaya komputasi tinggi (Dettmers et al., 2023; Frantar et al., 2023). Meskipun PTQ meningkatkan efisiensi komputasi, pengurangan presisi pada bobot dapat menimbulkan gangguan pada representasi model (*perturbation*). Namun, pengurangan presisi berisiko menghilangkan parameter penting (*outliers*) yang berperan dalam menjaga kualitas representasi model. Akibatnya, kemampuan penalaran model dan kualitas teks yang dihasilkan dapat menurun (Gong et al., 2024).

Berdasarkan kondisi tersebut, penelitian ini berfokus pada pengujian performa PTQ dengan pendekatan *mixed-precision k-quants* pada varian Q3_K_M, Q4_K_M, dan Q5_K_M. Pengujian ini menguji secara komparatif dua arsitektur SLM kontemporer dengan ukuran parameter yang berbeda, yaitu LFM 2.5 (1,2 miliar parameter) dan Qwen 3.5 (2 miliar parameter). Eksperimen ini bertujuan menjalankan proses inferensi secara lokal di lingkungan terminal Android (Termux) tanpa antarmuka grafis. Penelitian ini bertujuan untuk membuktikan secara empiris batas efektivitas kompresi yang paling optimal untuk mempertahankan stabilitas memori dari risiko *Force Close*, sekaligus menginvestigasi akselerasi kecepatan inferensi (*Tokens per Second* / TPS) pada CPU ARM serta signifikansi tingkat degradasi *Perplexity* (PPL) yang terjadi pasca-kuantisasi.

1.2. Identifikasi Permasalahan

Berdasarkan latar belakang yang telah diuraikan, dapat diidentifikasi beberapa permasalahan yang menjadi fokus dalam penelitian ini sebagai berikut:

1. Ketergantungan yang tinggi terhadap penyedia layanan awan komersial dalam pemrosesan instruksi *Large Language Model* (LLM) menimbulkan risiko keamanan berupa potensi serangan siber dan kebocoran data sensitif pada arsitektur komputasi awan (*cloud computing*). Kondisi tersebut mendorong urgensi pergeseran paradigma menuju sistem kecerdasan buatan (*Artificial Intelligence/AI*) otonom yang mampu mengeksekusi proses inferensi secara lokal (*offline*) langsung pada perangkat tepi (*edge device*).
2. Dimensi ukuran fisik berkas dari dua skala *Small Language Model* (SLM) berspesifikasi tinggi, yaitu LFM 2.5 yang memiliki 1,2 miliar parameter dan Qwen 3.5 yang memiliki 2 miliar parameter, dalam format presisi murni

Floating-Point 16-bit (FP16) terlalu besar sehingga membebani kapasitas media penyimpanan lokal (*storage footprint*) perangkat seluler serta membatasi efisiensi distribusi model pada ekosistem komputasi tepi.

3. Kapasitas RAM perangkat Android kelas menengah yang rata-rata berukuran 8 GB memiliki risiko tinggi mengalami krisis memori (*Out of Memory / OOM*) pada penerapan di dalam perangkat (*on-device deployment*). Hal ini disebabkan oleh topologi *Unified Memory* yang memicu terjadinya terminasi proses secara paksa (*Force Close*) saat memuat model berpresisi murni *Floating-Point* 16-bit (FP16), sehingga membatasi landasan empiris mengenai tingkat kompresi ideal pada lingkungan *shared memory* Android tersebut.
4. Belum adanya dokumentasi yang komprehensif mengenai tingkat akselerasi pemrosesan teks pada arsitektur perangkat keras seluler, baik pada fase *Prompt Processing* maupun *Generation Tokens per Second* (TPS), ketika algoritma *Post-Training Quantization* (PTQ) dieksekusi secara *bare-metal* menggunakan CPU berbasis ARM seperti MediaTek Helio G99 melalui emulator lingkungan terisolasi (*Termux*) tanpa antarmuka grafis.
5. Pemadatan matriks parameter *Small Language Model* (SLM) ke dalam format kuantisasi presisi campuran (*mixed-precision k-quants*) yang ketat seperti varian Q3_K_M, Q4_K_M, dan Q5_K_M berpotensi memperkenalkan gangguan matematis (*perturbation*) berupa fenomena halusinasi model akibat kompresi ekstrem. Terdapat kesenjangan penelitian (*research gap*) yang mendalam mengenai persentase degradasi nalar kecerdasan model dan pembengkakan nilai *Perplexity* (PPL) ketika aspek *Edge Intelligence* diterapkan pada dua skala model yang berbeda ukuran parameternya, yaitu LFM 2.5 (1,2B) dan Qwen 3.5 (2B).

Berpijak pada identifikasi permasalahan di atas, diperlukan sebuah analisis komprehensif untuk mengevaluasi sejauh mana penerapan metode *Post-Training Quantization* (PTQ) berformat GGUF *K-Quants* mampu mengatasi kendala keterbatasan sumber daya sistem operasi Android. Tujuan utamanya adalah mencari titik keseimbangan (*trade-off*) yang paling pas. Sistem harus bisa berjalan dengan memori yang lebih irit dan cepat, tetapi tidak boleh sampai merusak kemampuan model dalam menjawab perintah pengguna. Rangkaian identifikasi masalah ini selanjutnya menjadi landasan operasional dalam merumuskan batasan serta arah evaluasi eksperimental pada tahapan penelitian berikutnya.

1.3. Perumusan Masalah

Berdasarkan identifikasi permasalahan yang telah diuraikan, maka perumusan masalah dalam penelitian mengenai analisis performa *Post-Training Quantization* (PTQ) pada *Small Language Model* untuk implementasi *Edge Computing* berbasis Android ini diformulasikan ke dalam bentuk pertanyaan ilmiah sebagai berikut:

1. Bagaimana performa awal LFM 2.5 dan Qwen 3.5 dalam format FP16 pada platform Edge Computing berbasis Android?
2. Bagaimana pengaruh *Post-Training Quantization* (PTQ) berformat GGUF *K-Quants* pada varian Q3_K_M, Q4_K_M, dan Q5_K_M terhadap ukuran model dan penggunaan RAM pada perangkat seluler?
3. Sejauh mana *Post-Training Quantization* (PTQ) meningkatkan kecepatan inferensi yang diukur melalui *Prompt Processing* dan *Generation* TPS pada CPU ARM MediaTek Helio G99?
4. Bagaimana dampak penerapan berbagai tingkat kuantisasi tersebut terhadap kualitas keluaran linguistik dan retensi akurasi kognitif model, yang dievaluasi

secara kuantitatif melalui pengujian *Perplexity* (WikiText-2), keragaman leksikal (MT-Bench TTR), serta instrumen *benchmark* logika standar (MMLU, GSM8K, dan HumanEval)?

5. Berdasarkan analisis pertukaran multi-dimensi (*multi-dimensional trade-off*) yang meliputi efisiensi ukuran, beban memori RAM, kecepatan TPS, dan metrik retensi logika, konfigurasi varian model serta tingkat kompresi manakah yang terbukti paling optimal untuk diimplementasikan secara stabil pada perangkat Android berkapasitas RAM 8 GB?

Perumusan masalah tersebut menjadi dasar fundamental dalam melakukan analisis dan evaluasi eksperimental terhadap efektivitas *Post-Training Quantization* (PTQ) sebagai teknik optimisasi model AI. Rangkaian pertanyaan ilmiah ini dirancang secara sistematis untuk memetakan batasan komputasi perangkat gawai secara presisi, sekaligus memberikan landasan empiris yang kuat guna mendukung kelayakan implementasi *Small Language Model* otonom pada lingkungan *Edge Computing* bersumber daya terbatas.

1.4. Tujuan dan Manfaat

1.4.1. Tujuan Penelitian

Berdasarkan judul "Analisis Performa *Post-Training Quantization* (PTQ) pada *Small Language Model* untuk Implementasi *Edge Computing* Berbasis Android", tujuan yang ingin dicapai melalui pelaksanaan penelitian eksperimental ini adalah:

1. Menganalisis kinerja performa arsitektur *Small Language Model* (SLM), khususnya pada varian LFM 2.5 (1,2B) dan Qwen 3.5 (2B), sebelum penerapan

algoritma kompresi atau berada dalam kondisi *baseline* presisi murni *Floating-Point* 16-bit (FP16) pada platform *Edge Computing* berbasis Android.

2. Menganalisis pengaruh penerapan metode *Post-Training Quantization* (PTQ) berformat GGUF *K-Quants* dengan tingkat kepadatan Q3_K_M, Q4_K_M, dan Q5_K_M terhadap persentase reduksi ukuran fisik berkas model (*storage footprint*) yang digunakan pada perangkat Android.
3. Mengevaluasi pengaruh kuantisasi presisi campuran (*mixed-precision*) tersebut terhadap efisiensi penggunaan sumber daya fisik perangkat, meliputi pemakaian memori (*RAM processes*), utilitas CPU ARM MediaTek Helio G99, dan kapasitas penyimpanan guna memitigasi risiko penghentian paksa (*Force Close*) akibat limitasi arsitektur *Unified Memory* Android.
4. Menganalisis pengaruh berbagai varian kompresi PTQ terhadap kecepatan inferensi model pada perangkat Android yang diukur secara terperinci melalui metrik *Prompt Processing* dan *Generation Tokens per Second* (TPS).
5. Mengevaluasi pengaruh pemotongan presisi numerik tersebut terhadap kualitas keluaran teks dan retensi kapasitas kognitif model melalui pengukuran nilai *Perplexity* (PPL) serta pencapaian skor akurasi pada instrumen *benchmark* standar (MMLU, GSM8K, dan HumanEval).
6. Membandingkan secara komparatif kinerja *Small Language Model* sebelum dan sesudah penerapan PTQ untuk memetakan titik kesetimbangan operasional yang paling optimal (*trade-off* terbaik) antara efisiensi fisik dengan kualitas nalar bahasa pada lingkungan perangkat gawai berkapasitas RAM 8 GB.

Rangkaian tujuan tersebut dirancang sebagai kerangka kerja empiris untuk menguji kelayakan teknis dari metode kuantisasi pasca-pelatihan. Melalui pencapaian

poin-poin tujuan ini, penelitian diharapkan dapat memberikan visualisasi data yang akurat mengenai batas kemampuan komputasi tepi dalam mengakomodasi model kecerdasan buatan secara lokal.

1.4.2. Manfaat Penelitian

Hasil penelitian eksperimental ini diharapkan dapat memberikan kontribusi dan manfaat yang luas bagi beberapa pihak terkait, yang diuraikan sebagai berikut:

1. Manfaat Bagi Penulis

Penelitian ini berfungsi sebagai salah satu syarat akademik untuk memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi S1 Teknologi Informasi, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika. Di samping itu, penelitian ini memberikan kesempatan bagi penulis untuk memperdalam pemahaman mengenai teknologi *Artificial Intelligence* (AI), khususnya *Small Language Model* (SLM), *Edge Computing*, dan teknik optimasi model menggunakan *Post-Training Quantization* (PTQ). Lebih lanjut, penelitian ini menjadi sarana untuk mengembangkan kemampuan analisis, implementasi, dan evaluasi model AI pada perangkat Android sehingga dapat meningkatkan kompetensi akademik maupun profesional di bidang kecerdasan buatan dan komputasi bergerak.

2. Manfaat Bagi Objek Penelitian

Penelitian ini menyediakan referensi teknis, dokumentasi empiris, serta purwarupa operasional bagi pengembang dalam merancang infrastruktur kecerdasan buatan yang berjalan secara lokal (*on-device*) di perangkat Android. Hasil pengujian ini dapat dijadikan pedoman dalam memilih varian bit kuantisasi yang tepat sehingga mampu menghasilkan model aplikasi yang ringan, responsif, dan sepenuhnya menjaga privasi

data pengguna melalui proses inferensi luring (*offline*). Data kuantitatif dalam riset ini juga memberikan gambaran realistis mengenai batas kemampuan gawai kelas menengah berspesifikasi RAM 8 GB, sehingga membantu pengembang mengambil keputusan arsitektur perangkat lunak yang presisi serta mengurangi ketergantungan pada API *cloud* komersial.

3. Manfaat Bagi Pembaca

Penelitian ini diharapkan dapat menjadi sumber referensi ilmiah, sitasi, dan bahan pembelajaran bagi mahasiswa, peneliti, akademisi, maupun praktisi industri yang tertarik pada bidang *Artificial Intelligence*, *Small Language Model*, dan *Edge Computing*. Hasil analisis dalam dokumen ini memberikan wawasan komprehensif mengenai anatomi mekanis teknik kompresi *Post-Training Quantization* (PTQ), serta tata laksana penyebaran (*deployment*) model bahasa secara mandiri menggunakan *framework llama.cpp* di lingkungan emulator Termux Android.

1.5. Metode Penelitian

Penelitian ini menggunakan pendekatan empiris kuantitatif eksperimental dengan kerangka komparasi bertingkat (*multi-level comparative performance analysis*). Pendekatan ini difokuskan untuk menguji dan membandingkan performa arsitektur *Small Language Model* (SLM) sebelum dan sesudah diterapkannya teknik *Post-Training Quantization* (PTQ) melalui pengukuran parameter komputasi secara objektif. Varian tingkat presisi bobot model yang dikomparasikan meliputi presisi murni *Floating-Point* 16-bit (FP16) sebagai kondisi *baseline*, serta tiga varian terkompresi berformat GGUF *K-Quants*, yaitu Q5_K_M, Q4_K_M, dan Q3_K_M.

Tahapan operasional penelitian diselenggarakan di atas dua jenis *host* (*host-to-target*) eksekusi yang saling melengkapi. *Host* pertama adalah perangkat *Personal Computer* (PC) berbasis WSL2 Ubuntu dengan akselerasi GPU NVIDIA RTX 3060. Lingkungan ini digunakan untuk tahap prapemrosesan konversi berkas menggunakan utilitas `huggingface-cli`, `convert_hf_to_gguf.py`, dan `llama-quantize`, serta evaluasi metrik intrinsik kognitif model yang memerlukan akselerasi paralel komputasi tinggi. *Host* kedua adalah *smartphone* Tecno Pova 5 (MediaTek Helio G99, RAM 8 GB) yang diisolasi menggunakan emulator terminal *non-root* Termux. Pada *host* target ini, mesin inferensi *framework llama.cpp* dikompilasi ulang secara natif (*bare-metal*) guna mengukur metrik performa langsung pada lingkungan *Edge Computing* sesungguhnya.

Pengujian dieksekusi melalui skenario terstandarisasi yang ketat, meliputi keseragaman parameter inferensi, anggaran alokasi token, konfigurasi jumlah *threads*, serta penggunaan dataset evaluasi yang identik. Standarisasi ini diterapkan agar perbedaan performa yang terekam murni merupakan implikasi teknis dari tingkat kompresi bobot, bukan akibat anomali atau variabel eksternal. Parameter performa yang diukur secara objektif dalam penelitian ini meliputi:

1. Efisiensi Penyimpanan (*Storage Footprint*)

Mengukur perubahan dimensi ukuran fisik berkas model `.gguf` sebelum dan sesudah proses kuantisasi untuk mengetahui persentase efisiensi penyimpanan lokal yang berhasil direduksi.

2. Konsumsi Memori (*Peak RAM Usage*)

Mengukur jumlah beban alokasi puncak memori utama (RAM) yang ditarik oleh model selama proses inferensi berlangsung guna memastikan batas keamanan dari ancaman *Force Close* oleh pengawas memori sistem Android.

3. Kecepatan Inferensi (*Tokens per Second / TPS*)

Mengukur tingkat akselerasi pemrosesan model di atas CPU arsitektur ARM, yang dipecah ke dalam dua fase perhitungan komputasi, yaitu fase evaluasi perintah (*Prompt Processing TPS*) dan fase pembangkitan teks (*Generation TPS*).

4. Utilitas Prosesor (*CPU Load*)

Mengukur distribusi tingkat pemanfaatan daya komputasi unit prosesor saat mesin inferensi dijalankan untuk mengetahui sejauh mana dampak PTQ meringankan atau membebani arsitektur perangkat seluler.

5. Retensi Kecerdasan Kognitif (*Inference Quality*)

Mengukur fluktuasi degradasi logika dan kualitas keluaran teks model pasca-kuantisasi menggunakan instrumen evaluasi matematis *Perplexity* (berdasarkan dataset WikiText-2) serta skor *benchmark* standar (MMLU, GSM8K, HumanEval, dan MT-Bench TTR).

Data log sistem yang terekam selama pengujian kemudian dianalisis menggunakan metode statistik deskriptif untuk memperoleh sebaran data rerata, margin persentase perubahan, serta skor komposit (*compound score*). Selanjutnya, pembuktian signifikansi perbedaan performa komparatif antara model *baseline* dan

seluruh varian kompresinya divalidasi secara matematis melalui uji statistik inferensial *Welch's t-test* dan *One-way ANOVA*.

Hasil pengujian analitis ini pada akhirnya bermuara pada evaluasi konklusif mengenai sejauh mana teknik *Post-Training Quantization* mampu menekan batas penggunaan sumber daya *smartphone* tanpa mengorbankan kualitas keluaran secara fatal, sekaligus menetapkan rekomendasi konfigurasi kompresi yang paling ideal untuk skenario peluncuran *Small Language Model* otonom di ekosistem *Edge Computing*.

1.6. Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian mengenai "Analisis Performa *Post-Training Quantization* (PTQ) pada *Small Language Model* untuk Implementasi *Edge Computing* Berbasis Android" ini dilakukan untuk memperoleh data yang valid dan relevan terkait pengaruh penerapan kompresi terhadap kinerja model. Adapun teknik pengumpulan data yang digunakan adalah sebagai berikut:

a. Observasi Eksperimental

Observasi dilakukan melalui ekstraksi data *log* sistem secara objektif guna memantau dinamika perilaku perangkat keras maupun perangkat lunak saat siklus inferensi berlangsung. Proses observasi ini didistribusikan ke dalam dua lingkungan kerja (*host*) yang saling terintegrasi. Lingkungan pertama menggunakan perangkat PC (WSL2 Ubuntu) berspesifikasi unit pemrosesan grafis NVIDIA RTX 3060, yang diobservasi khusus untuk mengukur metrik kognitif intrinsik model yang membutuhkan daya komputasi masif (seperti akurasi MMLU, GSM8K, dan HumanEval). Lingkungan kedua adalah perangkat *smartphone* Android Tecno Pova 5

yang diakses melalui emulator terminal Termux. Pada perangkat ini, observasi difokuskan pada pemantauan beban kerja unit pemrosesan pusat (CPU) melalui perintah `ps`, pemantauan penggunaan memori utama tertinggi (*peak RAM*) melalui pembacaan nilai `VmRSS` pada direktori `/proc/pid/status`, serta pengukuran kecepatan teks (*Tokens per Second/TPS*) dari keluaran statistik mesin inferensi `llama-cli`. Penggunaan format `.gguf` pada kedua lingkungan merujuk pada satu sumber data tunggal (*source of truth*) guna memastikan perbandingan data tetap setara (*apple-to-apple*).

b. Studi Pustaka

Studi pustaka dilakukan dengan menghimpun dan memetakan referensi dari berbagai karya tulis ilmiah, baik jurnal nasional maupun internasional, yang dipublikasikan pada rentang 2023–2026. Fokus telaah literatur mencakup teori fundamental arsitektur LLM/SLM (Touvron et al., 2023), teori gangguan matematis pada kuantisasi (Gong et al., 2024), pedoman *Edge Intelligence* (Zhang et al., 2024), serta validasi empiris penerapan *llama.cpp* di perangkat ARM dengan sumber daya terbatas (Ray & Pradhan, 2026). Referensi pilar dari (Dettmers et al., 2023; Frantar et al., 2023; Jin et al., 2024; Lin et al., 2026) digunakan sebagai instrumen landasan teori sekaligus pemandu standar parameter kalibrasi pada tahap evaluasi.

c. Eksperimen

Eksperimen merupakan teknik utama dalam pengumpulan data penelitian ini. Proses eksperimen dilakukan dengan mengimplementasikan dua arsitektur *Small Language Model* (LFM 2.5 dan Qwen 3.5) dalam dua kondisi utama, yaitu model dengan presisi murni *Floating-Point* 16-bit (FP16) sebagai *baseline*, dan model yang telah dikompresi menggunakan metode PTQ berformat GGUF *K-Quants* (Q5_K_M,

Q4_K_M, dan Q3_K_M). Skenario inferensi distandardisasi secara ketat (meliputi kesamaan parameter konfigurasi, anggaran *token*, dan jumlah *threads*) agar data yang diperoleh objektif.

Data kuantitatif yang dikumpulkan selama eksperimen meliputi:

- 1) Ukuran fisik berkas model (*Storage Footprint / Model Size*).
- 2) Konsumsi memori utama maksimal (*Peak RAM Usage*).
- 3) Tingkat utilitas prosesor (*CPU Load*).
- 4) Akselerasi kecepatan inferensi (*Prompt & Generation TPS*).
- 5) Kualitas retensi kecerdasan model (*Perplexity* dan Skor Akurasi Logika).

d. Dokumentasi

Dokumentasi dilakukan dengan mengumpulkan dan menyimpan berbagai data hasil pengujian yang terekam selama proses eksperimen. Data dokumentasi ini berupa tangkapan layar (*screenshot*) terminal eksekusi, *log* sistem keluaran dari skrip otomatisasi (*csv files*), rekam jejak konfigurasi kompilasi *llama.cpp*, serta tabel tabulasi matriks metrik dari pengujian *benchmark*. Seluruh dokumentasi ini digunakan sebagai bukti autentik pelaksanaan penelitian sekaligus sebagai bahan analisis statistik dalam penyusunan laporan hasil akhir.

e. Pengukuran Kinerja Sistem (*Performance Measurement*)

Pengumpulan data juga dieksekusi melalui pengukuran langsung terhadap parameter kinerja sistem menggunakan alat bantu (*tools*) pengukuran perangkat lunak internal. Pengukuran metrik kognitif dilakukan menggunakan modul komputasi *./llama-perplexity* dan kerangka evaluasi metrik akurasi eksternal, sedangkan

pengukuran beban arsitektur fisik dilakukan memanfaatkan utilitas diagnostik bawaan subsistem Linux/Android. Data pengukuran performa ini divalidasi dengan pengulangan eksekusi (*run*) sebanyak tiga kali guna memastikan konsistensi luaran numerik sebelum diolah menjadi analisis komparatif atas dampak PTQ terhadap *Small Language Model*.

Melalui kombinasi teknik observasi, studi pustaka, eksperimen, dokumentasi, dan pengukuran kinerja sistem yang terisolasi tersebut, penelitian ini diharapkan dapat menghasilkan himpunan data yang komprehensif, terukur, dan objektif untuk mengevaluasi efektivitas *Post-Training Quantization* (PTQ) dalam implementasi kecerdasan buatan otonom pada platform *Edge Computing* berbasis Android.

1.7. Ruang Lingkup

Agar trajektori analisis dalam penelitian mengenai analisis performa *Post-Training Quantization* (PTQ) pada *Small Language Model* untuk implementasi *Edge Computing* berbasis Android ini tetap fokus pada sumbu permasalahan utama dan tidak melebar, maka ruang lingkup penelitian dibatasi pada poin-poin sebagai berikut:

1. Lingkungan komputasi eksperimental didistribusikan menggunakan skema *host-to-target deployment* yang melibatkan dua ekosistem perangkat keras spesifik, yaitu perangkat evaluasi *host* berupa unit PC dengan GPU NVIDIA RTX 3060 berbasis WSL2 Ubuntu untuk mengeksekusi pengujian metrik kognitif intrinsik melalui akselerasi paralel CUDA, serta perangkat *target deployment* berupa *smartphone* Tecno Pova 5 ditenagai *System-on-Chip* (SoC) MediaTek Helio G99, RAM LPDDR4x 8 GB, dan ROM UFS 2.2 256 GB untuk menguji metrik efisiensi infrastruktur fisik secara riil.

2. Lingkungan eksekusi pada perangkat target dibatasi pada sistem operasi Android 13 melalui aplikasi emulator terminal *non-root* Termux dengan memanfaatkan pustaka *framework llama.cpp* yang dikompilasi secara mandiri (*native compilation*) agar instruksi pada CPU ARM dapat dieksploitasi secara maksimal tanpa menggunakan antarmuka grafis.
3. Objek penelitian dibatasi pada dua varian *Small Language Model* (SLM) berbobot ringan, yaitu LFM 2.5 dengan kapasitas 1,2 miliar parameter sebagai representasi arsitektur ultra-ringan, dan Qwen 3.5 dengan kapasitas 2 miliar parameter sebagai representasi arsitektur penalaran (*reasoning*) yang lebih berat, guna memvalidasi generalisasi temuan lintas-arsitektur.
4. Teknik optimasi model yang diterapkan dibatasi murni pada metode *Post-Training Quantization* (PTQ) berformat GGUF dengan pendekatan presisi campuran (*mixed-precision k-quants*) melintasi rentang 3-bit hingga 5-bit yang meliputi varian Q3_K_M, Q4_K_M, dan Q5_K_M dengan menggunakan format *Floating-Point* 16-bit (FP16) sebagai tolok ukur dasar (*baseline*), tanpa membahas metode optimasi lainnya seperti *Quantization-Aware Training* (QAT), *pruning*, *knowledge distillation*, atau *model parallelism*.
5. Penelitian difokuskan secara eksklusif pada fase proses inferensi (*inference*) model secara lokal dan otonom di dalam perangkat bergerak tanpa ketergantungan terhadap API layanan *cloud*, serta tidak membahas proses pelatihan (*training*) maupun penyetelan halus (*fine-tuning*) langsung pada perangkat Android.
6. Pengukuran metrik kognitif dan kualitas linguistik yang dievaluasi pada perangkat *host* PC RTX 3060 dibatasi pada pengukuran nilai *Perplexity* (PPL) menggunakan dataset WikiText-2, serta pengujian tingkat akurasi penalaran terstandarisasi

menggunakan instrumen *benchmark* MMLU, GSM8K, HumanEval, dan pengukuran keragaman leksikal melalui *MT-Bench Type-Token Ratio* (TTR).

7. Pelaksanaan uji *benchmark* akurasi kognitif dibatasi secara metodologis sebanyak 100 sampel acak per jenis instrumen sebagai bentuk kompromi atas keterbatasan daya komputasi perangkat keras (*hardware constraint*), di mana setiap instrumen dieksekusi secara repetitif sebanyak tiga kali perulangan demi menekan bias variansi acak serta menjamin stabilitas statistik pada skor akhir.
8. Pengukuran metrik efisiensi infrastruktur fisik pada perangkat target Android Helio G99 dibatasi pada parameter ukuran fisik berkas model (.gguf), alokasi konsumsi memori utama tertinggi (*peak RAM processes*) melalui pembacaan nilai VmRSS pada direktori /proc/pid/status, beban kerja puncak unit prosesor (*CPU Peak*), serta kecepatan *throughput* inferensi token yang terdiri atas *Prompt Speed* dan *Generation Speed* dalam satuan *Tokens per Second* (TPS).
9. Analisis hasil penelitian diposisikan sebagai studi komparatif relatif untuk mengukur persentase rasio degradasi (*delta*) kinerja antar-varian pada himpunan 100 sampel yang identik, serta difokuskan secara ketat pada pengaruh penerapan PTQ terhadap efisiensi penggunaan sumber daya perangkat keras dan perubahan performa model, tanpa membahas aspek keamanan, privasi data, ataupun optimalisasi sistem operasi Android secara mendalam.

Dengan batasan ruang lingkup tersebut, penelitian ini difokuskan pada evaluasi, pemetaan, dan analisis dampak penerapan *Post-Training Quantization* terhadap kinerja dan tingkat efisiensi *Small Language Model* yang dijalankan secara lokal pada

perangkat Android sebagai platform *Edge Computing*. Hasil analisis komparatif ini secara krusial ditujukan untuk menghasilkan rekomendasi ilmiah mengenai konfigurasi teknik kompresi yang paling kompeten dalam mencapai titik kesetimbangan operasional paling optimal (*trade-off* terbaik) pada lingkungan gawai bersumber daya terbatas.

1.8. Hipotesis

Hipotesis penelitian disusun berdasarkan tujuan dan rumusan masalah untuk menguji pengaruh *Post-Training Quantization* (PTQ) terhadap kinerja *Small Language Model* (SLM) pada platform *Edge Computing* berbasis Android. Hipotesis ini menggambarkan prediksi ilmiah mengenai parameter komputasi fisik dan kualitas kognitif yang diharapkan terjadi setelah penerapan teknik kompresi pada model yang diuji.

Hipotesis Umum

H₁: Penerapan *Post-Training Quantization* (PTQ) berformat GGUF memberikan pengaruh yang signifikan terhadap karakteristik efisiensi fisik dan kualitas inferensi *Small Language Model* pada platform *Edge Computing* berbasis Android.

H₀: Penerapan *Post-Training Quantization* (PTQ) berformat GGUF tidak memberikan pengaruh yang signifikan terhadap karakteristik efisiensi fisik dan kualitas inferensi *Small Language Model* pada platform *Edge Computing* berbasis Android.

Hipotesis Khusus

H1a

Penerapan *Post-Training Quantization* (PTQ) mampu mengurangi ukuran fisik berkas *Small Language Model* secara signifikan jika dibandingkan dengan model sebelum kuantisasi (*baseline* FP16).

H1b

Penerapan *Post-Training Quantization* (PTQ) mampu menurunkan penggunaan alokasi puncak memori utama (*peak RAM usage*) selama proses inferensi berlangsung pada perangkat Android.

H1c

Penerapan *Post-Training Quantization* (PTQ) memberikan pengaruh yang signifikan terhadap tingkat utilitas kerja unit pemrosesan pusat (CPU) pada saat *Small Language Model* dijalankan di perangkat Android.

H1d

Penerapan *Post-Training Quantization* (PTQ) mampu menghasilkan akselerasi kecepatan generasi teks yang diukur melalui metrik *Tokens per Second* (TPS) jika dibandingkan dengan model sebelum kompresi.

H1e

Penerapan *Post-Training Quantization* (PTQ) tidak menyebabkan penurunan kualitas hasil keluaran bahasa atau degradasi akurasi kognitif yang fatal secara menyeluruh dibandingkan dengan model asli (*baseline*).

H1f

Penerapan *Post-Training Quantization* (PTQ) mampu meningkatkan efisiensi implementasi *Small Language Model* secara komprehensif sehingga layak dioperasikan secara lokal pada perangkat Android berspesifikasi memori terbatas.

Hipotesis Operasional

Secara operasional, berdasarkan landasan teoretis arsitektur dan konfigurasi eksperimental, penelitian ini memproyeksikan bahwa model bahasa yang telah melalui proses *Post-Training Quantization* (PTQ) presisi campuran (*mixed-precision*) melintasi rentang 3-bit hingga 5-bit akan menunjukkan indikator empiris sebagai berikut:

- a) Penurunan ukuran dimensi fisik berkas .gguf model LFM 2.5 dan Qwen 3.5 minimal mencapai kisaran 50% hingga 68% dibandingkan dengan ukuran berkas asli berpresisi murni *Floating-Point* 16-bit (FP16).
- b) Pengurangan alokasi konsumsi memori RAM proses fase inferensi secara masif, sehingga mampu mengamankan margin memori utama gawai dan mengeliminasi risiko intervensi perlindungan sistem operasi Android berupa *Out of Memory* (OOM) Killed.
- c) Optimalisasi lebar pita memori (*memory bandwidth efficiency*) yang berimplikasi langsung pada peningkatan kecepatan pemrosesan perintah (*Prompt Processing*) dan pembangkitan kata (*Generation Tokens per Second* / TPS) secara signifikan pada CPU berbasis ARM MediaTek Helio G99.
- d) Terjadinya perubahan kualitas keluaran linguistik yang bersifat minor pada tingkat kompresi menengah yaitu varian Q5_K_M dan Q4_K_M, yang

ditandai dengan pelebaran nilai *Perplexity* (PPL) kurang dari satu poin serta rata-rata penurunan skor akurasi di bawah lima persen terhadap *baseline*.

- e) Terjadinya degradasi akurasi kognitif dan meluruhnya kapabilitas penalaran secara signifikan pada tingkat kompresi ekstrem yaitu varian Q3_K_M, khususnya pada dimensi nalar matematis (*GSM8K*) dan logika pemrograman (*HumanEval*).
- f) Varian tingkat kompresi Q4_K_M terbukti secara kuantitatif mampu menjadi titik kesetimbangan operasional paling optimal (*trade-off* terbaik) untuk melokalisasi *Small Language Model* secara otonom pada perangkat Android dengan spesifikasi RAM sebesar 8 GB.

Dengan demikian, hipotesis utama penelitian ini menetapkan bahwa *Post-Training Quantization* (PTQ) dengan format GGUF *K-Quants* mampu mengeksplorasi efisiensi penggunaan sumber daya perangkat keras Android secara radikal dan mempercepat kecepatan inferensi token tanpa menyebabkan penyusutan kualitas penalaran yang destruktif, sehingga model menjadi sangat optimal untuk diterapkan pada lingkungan *Edge Computing*.