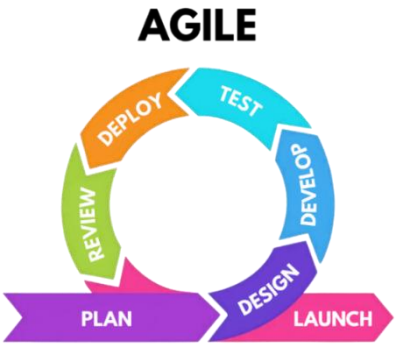
BAB III
METODE PENELITIAN

Penelitian ini merupakan *R&D* untuk *Large Vision Language Model (LVLM)* pada bahasa daerah berdaya rendah. Kerangka *life cycle* mengikuti standar *ISO/IEC/IEEE* 15288:2023 dan 24748-1:2024 yang menekankan proses iteratif, konkuren, *rollbackable*, dan *tailorable* [75], [76]. Sebagaimana ditunjukkan pada Gambar 3.1, siklus kerja diorganisasikan dalam putaran *Plan* → *Design* → *Develop* → *Test* → *Deploy* → *Review* → *Launch*; setiap putaran diawasi *quality gates* agar keputusan *go/no-go* terukur dan replikabel.

Pada bab ini, dibahas: alur iteratif pada Gambar 3.1 beserta peran *quality gates*; rancangan dan pembangunan model berbasis dua *backbone*, *LLaMa 3.2 11B Vision* dan *Gemma 3 12B (PT)*, dengan pendekatan *Parameter-Efficient Fine-Tuning (PEFT)* (*QSBoRA-FA*); *pipeline* data dan tahapan *training* (*Continued Pretraining (CP)* → *Instruction Tuning (IT)* → *Knowledge Distillation (KD, opsional)* → *Direct Preference Optimization (DPO)*); lingkungan komputasi (*Google Colab* untuk pelatihan dan *Hugging Face Spaces* untuk *staging/review*); serta strategi evaluasi yang mencakup *benchmark NusaX—MT* (*chrF++*) dan *Senti* (*Weighted F1-score*, *Accuracy*)—dan *human evaluation* oleh panel pakar (8 orang: 4 Sunda, 4 Jawa) dengan skala *Likert* terinspirasi *MMStar*, *VALOR-Eval*, dan *HarmonicEval*. Selain itu, dipaparkan pemantauan energi–emisi menggunakan *CodeCarbon* sebagai praktik *Green AI* [54], [55].



Gambar 3. 1 Diagram pendekatan *Agile* tingkat *life-cycle*

Ramadhirra Azzahra Putri, 2025
PENDEKATAN BERKELANJUTAN BERBASIS PARAMETER-EFFICIENT FINE-TUNING UNTUK MEMBANGUN SISTEM PERCAKAPAN BERBASIS AI DENGAN DUKUNGAN BAHASA DAERAH PADA LARGE VISION LANGUAGE MODEL
Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

3.1 Perencanaan *(Planning)*

Tahap *planning* menetapkan arah metodologis pengembangan *LVLM* untuk bahasa *low-resource*, dengan fokus pada bahasa Sunda dan bahasa Jawa sebagai bahasa daerah berpenutur terbesar di Indonesia [86], [87]. Kegiatan utama meliputi kajian literatur untuk memetakan *state of the art (SoTA)* dan masalah ekosistem Indonesia [1], [3], [4]; perencanaan data melalui pemilihan *benchmark NusaX* dan pengayaan terkontrol termasuk *knowledge distillation* [59], [43], [44]; rancangan strategi *benchmarking* kuantitatif *(chrF++, Weighted F1-score, Accuracy)* dan kualitatif berbasis panel pakar [62], [65], [68]–[70]; serta penetapan siklus iteratif dengan *quality gates* berurutan sesuai standar *ISO/IEC/IEEE 15288:2023* dan 24748-1:2024 [75], [76].

3.1.1 Tinjauan Literatur dan Identifikasi Masalah

Subbagian ini memetakan *state of the art LLM* (model teks) dan *LVLM* (teks dan visual) dalam konteks bahasa Indonesia *low-resource* untuk menetapkan ruang penelitian pada tahap perencanaan, tanpa mengunci pilihan teknis terlebih dahulu [1], [3], [4]. Fokus bahasa pada Sunda dan Jawa karena jumlah penuturnya terbesar (*SP2020*: Jawa 42,19%; Sunda 18,68%) [86], [87]. Tinjauan dilakukan pada empat klaster:

(1) Bahasa dan sumber daya: menilai kecukupan korpus serta variasi dialek/*register* dan menetapkan *NusaX* sebagai tolok ukur kuantitatif lintas iterasi [59];

(2) Arsitektur dan *multimodality*: mengidentifikasi keluarga *LVLM* yang terdokumentasi baik dan sesuai batas komputasi, mengingat karya lokal masih didominasi *text-only* sehingga dibutuhkan pemrosesan visual–teks [3], [4], [22], [29];

(3) Penyesuaian dan efisiensi: menghimpun opsi kandidat seperti *PEFT* [30], kuantisasi [34], *knowledge distillation* [43], [44], serta *Preference-based Post-training* untuk tahap lanjut [24];

(4) Evaluasi dan tata nilai: merancang pengukuran berimbang dengan *NusaX*

untuk kuantitatif [59], validasi pakar berbasis *MMStar/VALOR-Eval/HarmonicEval* [68]–[70], serta pemantauan energi–emisi *(CodeCarbon)* sebagai praktik *Green AI* [54], [55].

3.1.2 Perencanaan Strategis Pengembangan

Perencanaan strategis menetapkan teknologi, garis besar arsitektur sistem, serta metodologi pengujian dan validasi yang akan diterapkan. Tahap ini juga menentukan kriteria keberhasilan, parameter evaluasi, dan metrik pengukuran untuk menilai efektivitas pengembangan, selaras dengan *life cycle iterative*, *rollbackable*, dan *tailorable* sesuai standar *ISO/IEC/IEEE* [75], [76]. Penetapan mempertimbangkan keterbatasan sumber daya, timeline, dan kompleksitas teknis, sehingga pendekatan modular dipilih agar pengembangan bertahap, *debugging*, dan *maintenance* lebih mudah serta fleksibel terhadap perubahan dan iterasi selanjutnya.

Perencanaan data mencakup pemilihan *dataset* pelatihan dan evaluasi yang relevan untuk Sunda–Jawa, dengan *benchmark NusaX* untuk pemantauan lintas iterasi [3], [4], [59]. Kajian juga menilai cakupan dan kelayakan *dataset* terhadap tugas *multimodal* target. Opsi pengayaan data disiapkan bila diperlukan, termasuk pembuatan *dataset* baru secara terkontrol melalui *Knowledge Distillation* dari *teacher* ke *student*, agar efisien dalam penggunaan sumber daya tanpa harus melakukan *full fine-tuning* [43], [44].

3.1.3 Strategi *Benchmarking*

Strategi benchmarking mencakup perencanaan pengukuran kuantitatif dan kualitatif untuk menilai kemajuan pada setiap putaran iteratif, selaras dengan tugas yang diuji, yaitu terjemahan dan klasifikasi sentimen, serta mengacu pada praktik rujukan dari karya Cendol dan *Komodo-7B* [3], [4]. Pendekatan ini dirancang agar evaluasi konsisten lintas iterasi dan relevan dengan tujuan penelitian, memastikan bahwa setiap langkah pengembangan *LVLM* dapat dilacak secara jelas dan progresnya dapat dibandingkan secara sistematis antar bahasa.

Untuk evaluasi kuantitatif, tugas terjemahan menggunakan *chrF++* sebagai

metrik utama karena ketangguhannya terhadap variasi tokenisasi/morfologi dan korelasi yang baik dengan penilaian manusia [62], [61]. Tugas klasifikasi sentimen menggunakan *Weighted F1-score* agar ketidakseimbangan kelas tercermin adil, dengan *Accuracy* sebagai metrik pendamping, sesuai praktik evaluasi *LLM* terkini [60], [63]–[65]. Pelaporan dilakukan per bahasa (Indonesia, Sunda, Jawa) serta agregat lintas bahasa agar tren kemajuan mudah ditelusuri [3], [4].

Tolok ukur kuantitatif mengacu pada *dataset benchmark NusaX*, yang menyediakan pasangan tugas *machine translation* dan *sentiment analysis* untuk bahasa Indonesia, Inggris, dan sepuluh bahasa daerah Indonesia dengan skema *train/test* seragam. *Subset NusaX-MT* digunakan untuk terjemahan dan *NusaX-Senti* untuk sentimen [59]. Untuk evaluasi kualitatif, validasi pakar dilakukan melalui kuesioner skala *Likert* dengan pedoman yang mengintegrasikan prinsip-prinsip kerangka *LVLM* kontemporer, menilai keselarasan teks–*visual, faithfulness–coverage,* dan koherensi [68]–[70].

Dengan perencanaan tersebut, *benchmarking* menyediakan cara ukur yang konsisten dan relevan terhadap tujuan penelitian, yaitu peningkatan kemampuan kebahasaan *low-resource* serta validasi kemampuan *multimodal* dalam konteks Indonesia [59], [60]–[65], [68]–[70]. Pendekatan gabungan kuantitatif dan kualitatif memungkinkan pengembangan *LVLM* yang adaptif dan akurat, sekaligus memantau aspek yang tidak sepenuhnya terwakili oleh metrik otomatis, sehingga hasil evaluasi dapat digunakan sebagai dasar iterasi dan optimasi model berikutnya.

3.1.4 Panel Validasi Pakar

Panel validasi akan melibatkan delapan pakar: empat penutur asli (*native speaker*) Sunda dan empat penutur asli Jawa, dengan penguasaan menyeluruh atas *register/tingkat tutur* (Sunda: *kasar–loma–lemes*; Jawa: *ngoko–madya–krama/krama alus–krama inggil*) dan pengalaman mengajar menggunakan bahasa daerah sebagai bahasa pengantar. Komposisi ini dipilih untuk memastikan reliabilitas *human evaluation* dan kepekaan budaya dalam penilaian keluaran model [81]–[85]. Profil *evaluator* dapat dilihat pada Lampiran 1 dan Lampiran 2. Panel

per bahasa ditetapkan berjumlah empat orang untuk menjaga reliabilitas dan keberagaman sudut pandang. [82], [83].

Peran umum panel: memvalidasi kemampuan model, membandingkan kedua model, melakukan cek koherensi dan kelancaran bahasa, serta kesetiaan isi pada konteks visual (untuk keluaran *multimodal*) melalui kuesioner skala *Likert* dan komentar kualitatif singkat pada *Google Form* [81]–[84].

3.1.5 Rencana Iterasi, *Quality Gates*, dan Kriteria Keberhasilan

Setiap putaran *Plan* → *Design* → *Develop* → *Test* → *Deploy* → *Review* → *Launch* menerapkan *quality gates* berurutan agar keputusan *go/no-go* terukur dan replikabel, selaras dengan *life cycle* yang iteratif dan dapat di-*tailor* menurut *ISO/IEC/IEEE* 15288:2023 dan 24748-1:2024 [75], [76].

1. *Monitor*, Memantau *Training Telemetry*

   Training telemetry seperti *optimization metrics* (*loss*, *grad_norm*, *learning_rate*), *progress* (*epoch*), dan *auxiliary diagnostics* pada *Post-training* seperti *DPO*, yakni *preference diagnostics* (*rewards/chosen, rewards/rejected, rewards/accuracies rewards/margins*) serta, *likelihood diagnostics (logps/chosen, logps/rejected, logits/chosen, logits/rejected)*, dipantau untuk mencegah terjadinya *underfitting, overfitting* maupun *stalling* performa di mana model sudah mencapai *diminishing return*.

2. *Test*, Uji *gibberish* (wajib lulus)

   Setelah memantau *training telemetry,* pemeriksaan awal sebagai *smoke test* pada tugas teks dan visi–teks dilakukan untuk memastikan tidak ada keluaran tak bermakna, salah rujuk visual, atau ketidaksesuaian tingkat tutur. Jika gagal, dilakukan perbaikan terarah dan pengulangan iterasi (tidak berlanjut/*step back*).

3. *Deploy*, *Staging* Privat (Non-publik)

   Model yang lolos *smoke-test* di-*deploy* ke lingkungan *staging* privat untuk "membekukan" artefak dan menghasilkan keluaran konsisten bagi evaluasi berikutnya. Tahap ini tidak dimaknai sebagai rilis.

4. *Review*, Evaluasi kuantitatif dan kualitatif
   a. *Benchmark* kuantitatif menggunakan *NusaX* per bahasa (Indonesia, Sunda, Jawa) dengan metrik *chrF++* untuk terjemahan serta *Weighted F1-score* dan *Accuracy* untuk klasifikasi; capaian dibandingkan *baseline* dan tren lintas putaran [59], [60]–[65].
   b. Validasi kualitatif oleh panel pakar berbasis skala *Likert* dengan pedoman evaluasi *LVLM* kontemporer (koherensi, kesesuaian *register*/tingkat tutur, serta *faithfulness–coverage*) [68]–[70].

      Kriteria lulus *Review*: (i) melampaui ambang *benchmark* yang ditetapkan per bahasa; (ii) tidak ada indikasi *gibberish* residu; (iii) konsistensi penilaian pakar pada dimensi bahasa dan *multimodal*.
5. *Launch*, Rilis terbatas/publik

   Rilis terbatas/publik hanya jika seluruh kriteria *Review* terpenuhi. Jika gagal pada *Review*, lakukan *rollback* terarah (penataan data, penyesuaian kurikulum/*parameter*) dan, bila relevan, aktifkan opsi *Post-training* berbasis preferensi (seperti *DPO*) pada iterasi berikutnya [24].
6. Rangkaian ini memastikan setiap *increment* yang keluar dari *staging* telah melewati uji *smoke-test*, *benchmark* NusaX, dan validasi pakar secara berurutan, sesuai prinsip *Agile* yang menekankan *continuous feedback* dan *re-prioritization* antarputaran [59], [60]–[65], [68]–[70], [75], [76].

3.2 Perancangan (*Designing*)

Tahap perancangan menyusun spesifikasi arsitektur, modul, dan lingkungan eksekusi sebagai dasar implementasi. Hasilnya adalah *blueprint* yang *reproducible*, hemat sumber daya, dan siap diterapkan pada model *LVLM* untuk bahasa Sunda/Jawa.

3.2.1 Alat dan Bahan

Pemilihan komponen didasarkan pada kebutuhan teknis untuk membangun dan menguji *LVLM multimodal* bagi bahasa Sunda/Jawa berdaya rendah. Lingkungan komputasi *Google Colab (GPU A100)* dipilih untuk kompatibilitas

PyTorch/*Transformers* dan efisiensi pelatihan; *backbone base/pretrained* (*LLaMa 3.2 11B Vision* dan *Gemma 3 12B PT*) dipilih sebagai titik awal netral yang mudah diadaptasi; dan *Hugging Face Spaces* serta *Gradio* disiapkan untuk inferensi serta umpan balik pakar/publik. Seluruh komponen dipilih menurut kriteria performa, stabilitas, reprodusibilitas (*version pinning*), dan kemudahan integrasi *pipeline*. *Dataset* digunakan untuk melatih model dan sebagai *benchmark* kuantitatif (terpisah, berbeda). Ringkasan spesifikasi tersedia pada Lampiran 3-6.

3.2.1.1 *Environment* pada *Google Colaboratory*

Pada *Google Colaboratory*, seluruh eksperimen dijalankan di lingkungan *notebook* berbasis *Jupyter* dengan akselerasi *GPU NVIDIA A100-SXM4-40GB*. *Platform* ini dipilih karena ketersediaan *GPU* kelas *data center* dan ekosistem *driver/toolkit* yang stabil *(Driver 550.54.15, CUDA Runtime 12.4, CUDA Toolkit 12.5.82, cuDNN 9.3.0, NCCL 2.23.4)*, sehingga kompatibel dengan *PyTorch 2.6.0+cu124* beserta komponen *vision/audio* terkait. Tumpukan pustaka inti untuk pengembangan dan evaluasi mencakup *Transformers*, *Accelerate*, *PEFT*, *TRL*, *Unsloth*, *Datasets, scikit-learn, NumPy serta Pandas*. Pemantauan konsumsi daya/telemetri *GPU* dilakukan melalui *CodeCarbon*.

3.2.1.2 *Environment* pada *Hugging Face Spaces*

Lingkungan *Hugging Face Spaces* digunakan untuk tahap *Deploy* dan *Launch*: model ”dipaketkan” untuk inferensi dengan *UI Gradio* sehingga publik (terutama *human evaluator*) dapat mencoba keluaran secara interaktif, sementara artefak rilis utamanya adalah model *weight*s pada repositori *Hugging Face*. Versi pustaka (*Transformers*, Accelerate, *PEFT*, dsb.) dipertahankan selaras dengan ekosistem *Colab* guna menjaga paritas lingkungan dan reproduktibilitas. Fokusnya adalah pemuatan bobot secara efisien, eksekusi inferensi *multimodal* (teks–gambar), serta integrasi *I/O* yang ringan tanpa komponen pelatihan. Lampiran 4 mencantumkan paket dan versi pada *Hugging Face Spaces (deploy/launch)* yang diselaraskan dengan lingkungan *Google Colab*.

3.2.1.3 *Backbone* Model

Penelitian ini mengadopsi pendekatan strategis dengan menggunakan model dasar (*base/preTrained*) sebagai fondasi, alih-alih model yang telah disesuaikan untuk instruksi (*instruction-tuned*). Keputusan ini bertujuan untuk memulai proses dari titik awal yang netral, yakni model yang belum terselaraskan oleh instruksi dan preferensi manusia. Titik awal yang netral ini krusial untuk meminimalkan bias perilaku generatif dan mengurangi risiko interferensi atau degradasi performa saat model diadaptasi ke domain atau bahasa baru [11], [23], [40], [21].

Sejalan dengan pendekatan tersebut, dua *LVLM* dipilih sebagai *backbone*: *LLaMa 3.2 11B Vision* dan *Gemma 3 12B PT*. Pemilihan keduanya didasarkan pada beberapa faktor kunci: dukungan ekosistem dan dokumentasi yang matang, kompatibilitas tinggi dengan *pipeline* pelatihan *PEFT* di lingkungan *Python*, serta rekam jejak penggunaannya dalam riset berbahasa Indonesia [3], [4], [22], [29], [37]. Pilihan ini secara langsung mendukung alur metodologi yang dirancang, yaitu: *Continued Pretraining* → *Instruction Tuning* → *Knowledge Distillation* (opsional) → *Preference-based Post-training/DPO* (opsional).

3.2.1.4 *Dataset*

Pemilihan *dataset* dilakukan berdasarkan pertimbangan metodologis yang ketat. Karena sumber daya bahasa daerah Indonesia tergolong *low-resource*, hanya *dataset* dengan format terstandar, anotasi konsisten, dan distribusi jelas yang efektif digunakan. Selain itu, keterbatasan *dataset multimodal* membuat penelitian ini memfokuskan pelatihan pada lapisan bahasa *(language layers)* model *multimodal*, tanpa *fine-tuning* pada *vision layers*. Dengan cara ini, kemampuan pengolahan *multimodal* dari pra-pelatihan tetap terjaga, sementara adaptasi bahasa daerah dilakukan sepenuhnya melalui *text-only fine-tuning*.

Dataset utama yang digunakan mencakup *Cendol Collection*, *Javanese Alpaca Cleaned*, dan *Sundanese Alpaca Cleaned*. *Cendol Collection* dipakai pada tahap *Continued Pretraining* untuk memperluas representasi bahasa daerah. Sementara *dataset Javanese* dan *Sundanese Alpaca Cleaned* digunakan pada tahap

Instruction Tuning agar model mampu mengikuti perintah dalam bahasa daerah. *Dataset* hasil *Knowledge Distillation* dipertimbangkan, namun bersifat opsional. Untuk *benchmarking* (kuantitatif), *NusaX-MT* dan *NusaX-Senti* digunakan sebagai tolok ukur standar kinerja model pada tugas *machine translation* dan *sentiment analysis* multibahasa, memastikan evaluasi konsisten dan relevan.

Keterbatasan utama penelitian adalah minimnya *dataset multimodal* yang relevan untuk bahasa daerah. Sebagai pelengkap, *Knowledge Distillation* dieksplorasi dari model *closed-source* seperti *Google Gemini 2.5 Pro* untuk menghasilkan data instruksi tambahan dan *preference-based* dalam bahasa Sunda dan Jawa. Distilasi ini difokuskan pada korpus teks murni *(text-only)*, guna memperkaya pemahaman semantik dan memperluas cakupan instruksi lintas domain, sekaligus mempertahankan efisiensi komputasi tanpa menambah beban pada lapisan *vision* model.

3.2.1.5 *Parameter-Efficient Fine-Tuning* (*PEFT*) Berbasis *Green AI*

Pendekatan *fine-tuning* dalam penelitian ini menggunakan *Parameter-Efficient Fine-Tuning* (*PEFT*) untuk efisiensi sumber daya dan *Green AI*, di mana hanya sebagian kecil parameter strategis yang disesuaikan, bukan seluruh bobot model. Keunggulannya meliputi efisiensi komputasi (*FLOPs* dan konsumsi daya lebih rendah), jejak memori ringan (*optimizer states* hanya untuk parameter terlatih), serta iterasi *Agile* dengan artefak ringan *(~GB)* yang memungkinkan eksperimen, validasi, dan *rollback* cepat. Dari berbagai varian *PEFT*, *QSBoRA-FA* dipilih karena matriks A dibekukan dan difokuskan pada matriks B yang dapat dilatih, sehingga pembelajaran menekankan kompetensi bahasa target (Sunda/Jawa) sembari menjaga *vision-language alignment*; konfigurasi ini terbukti memberikan *trade-off* akurasi-biaya terbaik dibanding *LoRA, DoRA,* atau *Frozen-B*.

Secara teknis, pelatihan diorkestrasi dengan *Transformers, BitsandBytes, Accelerate, PEFT* (untuk *adapter QSBoRA-FA*), *Unsloth,* dan *CodeCarbon* untuk telemetri energi serta *TRL* untuk *optional DPO*, dengan fokus hanya pada *language*

layers tanpa mengubah *vision encoder/projector*. Strategi *PEFT* dianggap berhasil bila: (i) meningkatkan performa metrik *(Weighted F1-score/Accuracy* pada *NusaX-Senti* dan *chrF++* pada *NusaX-MT* untuk bahasa Sunda dan bahasa Jawa), (ii) stabil, lolos *smoke test* dan *quality gate* tanpa *gibberish*, dan (iii) menghasilkan *adapter* ringan dan *reproducible* untuk *rollback* dan *branching* cepat pada siklus pengembangan berikutnya.

3.2.2 Strategi Adaptasi Arsitektur

Strategi adaptasi arsitektur dalam penelitian ini difokuskan secara eksklusif pada lapisan bahasa *(language layers)*, sementara seluruh komponen pada jalur pemrosesan visual, termasuk *vision encoder* dan *vision-language projector*, dibekukan *(frozen)*. Keputusan ini didasarkan pada dua justifikasi utama. Pertama, praktik terbaik dan efisiensi: pendekatan ini selaras dengan praktik modern pelatihan *LVLM* untuk menjaga alignment *multimodal* hasil pra-pelatihan, seperti pada model *Gemma 3* [86], sekaligus menekan biaya komputasi sesuai prinsip *Green AI*.

Kedua, keterbatasan sumber daya: hingga Agustus 2025, belum tersedia korpus *VLM* berpasangan gambar-teks yang terstandar untuk bahasa Sunda atau Jawa. Sumber daya yang ada terbatas pada format teks, sehingga *text-only fine-tuning* menjadi pilihan paling tepat dan berbasis bukti [1], [59]. Adaptasi ini diimplementasikan menggunakan metode *QSBoRA-FA*, sebagaimana dijelaskan pada §3.2.2.

3.2.2.1 Konfigurasi *Adapter* dan Penargetan Modul

Meskipun kedua model *backbone* menggunakan hiperparameter *adapter* yang sama secara *default* (*rank* r=128, α=128), berdasarkan pada *paper Cendol* [3], strategi penargetan modulnya dibedakan secara sengaja untuk mengoptimalkan adaptasi berdasarkan karakteristik arsitektur masing-masing.

1. *LLaMa* 3.2 *11B Vision*: Adaptasi Leksikal Inklusif

   Pada *backbone LLaMa* 3.2 *11B Vision*, *adapter PEFT* diterapkan secara luas untuk memaksimalkan kapasitas adaptasi ke bahasa daerah.

   a. Modul yang Diadaptasi: *q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj,*

down_proj, serta *embed_tokens* dan *lm_head*.

b. Rasional: Menyertakan *embed_tokens* dan *lm_head* adalah keputusan strategis yang didukung oleh lini riset NLP Indonesia sebelumnya, yang menyoroti pentingnya penyesuaian pada level leksikal untuk adaptasi bahasa *low-resource* yang efektif [3], [4].

2. *Gemma 3 12B PT*: Adaptasi Fokus pada Representasi Internal

Untuk *Gemma 3 12B PT*, strategi yang lebih konservatif diterapkan dengan fokus pada lapisan pemrosesan internal, sementara lapisan *input* dan *output* dibekukan.

a. Modul yang Diadaptasi: *q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj.*

b. Rasional Metodologis: Keputusan ini didasarkan pada dua pertimbangan utama:

1) Stabilitas Leksikal: Studi seperti *Half Fine-tuning (HFT)* menunjukkan bahwa membekukan lapisan *embedding* dan *lm_head* adalah strategi yang efisien dan efektif dalam skenario data terbatas (*data-scarce*), karena menjaga stabilitas representasi leksikal yang sudah dipelajari [87].

2) Kecukupan *Tokenizer*: *Gemma 3 12B PT* menggunakan *tokenizer* dengan ~262.000 entri yang cakupannya sudah sangat multibahasa. Temuan pada eksplorasi awal menunjukkan bahwa model *Gemma 3 12B PT* dapat *mengetik token* bahasa Sunda/Jawa, namun gagal *menyusunnya* menjadi kalimat yang koheren. Hal ini mengindikasikan bahwa masalah utama terletak pada pemodelan relasi antar *token* di lapisan representasi internal, bukan pada cakupan kosakata. Oleh karena itu, adaptasi difokuskan pada blok atensi dan MLP [90].

3.2.2.2 Kebijakan *Freezing / Unfreezing* per Tahap

Kebijakan pembekuan *(freezing)* dan adaptasi modul berbeda-beda pada setiap tahap pengembangan *LVLM*, yaitu *Continued Pretraining (CP), Instruction*

Tuning (IT), dan *Post-training* bila diperlukan, sebagaimana ditunjukkan pada Lampiran 7 dan Lampiran 8. Untuk *LLaMa 3.2 11B Vision* dan *Gemma 3 12B PT*, seluruh komponen *vision encoder/tower* dan *projector* tetap dibekukan, sedangkan *embed_tokens* dan *lm_head* diadaptasi atau disimpan sesuai model. Proyeksi *q/k/v/o, gate/up/down* (MLP), dan Matriks B diadaptasi menggunakan *QSBoRA-FA*, sedangkan *Norm* dan Matriks A tetap dibekukan kecuali diperlukan stabilitas, sehingga setiap tahap menyeimbangkan efisiensi komputasi dan kemampuan adaptasi.

3.2.2.3 Kebijakan *Rank-alpha Adapter*: *Default* dan *Fallback*

1. **Setelan *Default***

   Untuk kedua *backbone*, proyek menetapkan r = 128 dan α = 128, ini adalah praktik umum untuk *adapter* pada model berskala besar karena menjaga skala pembaruan netral (rasio α/r ≈ 1): cukup kuat untuk belajar, namun tidak berlebihan, seperti pada kasus Cendol [15], [30], [58].

2. **Penanganan Instabilitas**

   Jika muncul gejala seperti *over-steering*, *overfit*, atau keluaran *gibberish*, penyesuaian dilakukan dengan menurunkan *rank* dan menskalakan α secara proporsional agar amplitudo pembaruan tetap terkendali:

   a. Opsi moderat: α = 2r (contoh: $r = 16 \rightarrow \alpha = 32$). Rasio 2× lazim di praktik dan cenderung stabil saat koreksi ringan sudah memadai [30], [58].

   b. Opsi seimbang untuk konteks *low-resource* (dipilih sebagai *fallback* proyek): α = 3r (contoh: $r = 16 \rightarrow \alpha = 48$). Karena bahasa Sunda/Jawa berdaya rendah, *rank* kecil membantu menekan *overfit*, sementara α yang sedikit lebih besar menjaga kekuatan sinyal pembaruan agar model tetap belajar pola bahasa secara wajar. Angka 3× adalah heuristik proyek yang selaras dengan arah bukti bahwa skema penskalaan harus dikencangkan saat *rank* berubah, meski bukan kaidah baku [30], [58], [88].

3. **Mengapa Konfigurasi r = 16, α = 64 Dihindari?**

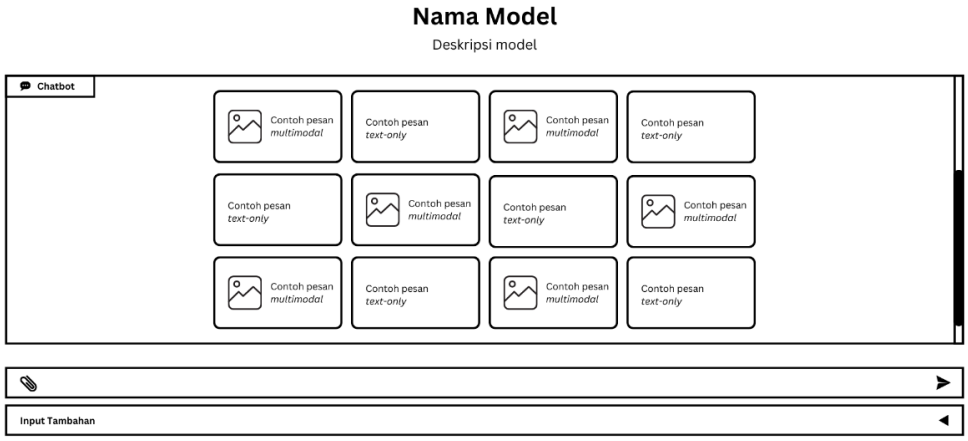
Rasio 4× lebih agresif dan meningkatkan risiko *over-steering* pada data terbatas. Literatur tentang stabilisasi penskalaan *adapter* menekankan sensitivitas terhadap *rank* dan menganjurkan skala yang lebih konservatif; karena itu konfigurasi 4× dihindari, rasio 3× pada *rank* kecil dipilih sebagai kompromi yang lebih aman apabila rasio 2× tidak memenuhi, seperti pada kasus *language learning* yang dinilai lebih kompleks daripada *style-transfer* dan kasus serupa [88].

4. **Dampak Komputasi dan Energi**

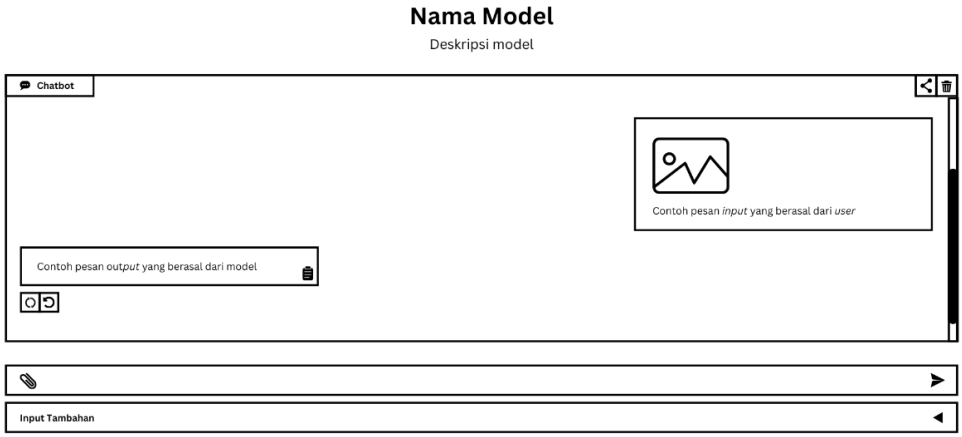
   Menu*run*kan *rank* otomatis mengurangi jejak memori dan waktu pelatihan, sehingga iterasi cepat, *rollback* ringan, dan pengendalian emisi (selaras *Green AI*) lebih mudah dicapai [30], [58].

3.2.3 *Wireframe* Antarmuka Pengguna

Perancangan antarmuka pengguna *(User Interface)* bertujuan menciptakan pengalaman yang intuitif, efisien, dan mudah dipelajari saat berinteraksi dengan sistem asisten virtual berbasis *LVLM*. *Wireframe* berfungsi sebagai *blueprint* visual untuk memvalidasi arsitektur informasi, alur interaksi, dan hierarki komponen sebelum implementasi berfidelitas tinggi. Antarmuka diimplementasikan pada *Hugging Face Spaces* menggunakan *Gradio* agar dapat diakses melalui peramban tanpa instalasi lokal, sekaligus mendukung *Deployment* cepat dan replikasi uji. *Wireframe dari* antarmuka pengguna dapat dilihat pada Gambar 3.2 dan Gambar 3.3.



Gambar 3. 2 *Wireframe* tampilan awal



Gambar 3. 3 *Wireframe chatbox*

Peran pada tahapan metodologi:

1. Tahap *Deploy* dan *Review*: *UI* menjadi *demo* interaktif yang dapat digunakan oleh pakar untuk menguji model.
2. Tahap *Launch*: *UI* tetap dipertahankan sebagai demo interaktif di *Spaces* agar publik dapat mencoba model tanpa mengunduh bobot. Fokus utama tahap ini adalah merilis bobot (*weight*s) dan model *card* ke repositori *Hugging Face* publik (termasuk versi, lisensi, dan petunjuk penggunaan). Dengan demikian, *UI* bertindak sebagai etalase penggunaan, sementara artefak rilis resmi adalah repositori model.

3.3 Pengembangan (*Development*)

Tahap pengembangan mengeksekusi *CP* dan *IT* (serta *KD* dan *DPO* bila diperlukan) pada *language layers* menggunakan *PEFT QSBoRA-FA*, di lingkungan *Colab* yang dipakukan versinya. Stabilitas dijaga melalui *smoke test*, *quality gate*, dan kontrol eksperimen (satu *run* = satu *session*, *checkpoint* privat), sehingga hasil pelatihan efisien, *reproducible*, dan siap dievaluasi lanjut.

3.3.1 Rencana Pengembangan Model

1. **Tahap Persiapan Pengembangan**
   a. Menetapkan ruang kerja: *text-only fine-tuning* pada *language layers*; *vision path* dibekukan.
   b. Aset data pengembangan:
      1) *CP* (*Continued Pretraining*): *Cendol Collection*.
      2) *IT* (*Instruction Tuning*): *Javanese Alpaca Cleaned* dan *Sundanese Alpaca Cleaned*.
      3) Augmentasi *dataset* (jika diperlukan): *Knowledge Distillation* (*KD*) berbasis *Gemini 2.5 Pro* untuk menambah pasangan instruksi–jawaban dan *DPO-style* dalam bahasa target.
   c. Lingkungan: *Google Colab* (A100) dengan *version pinning* pustaka; *CodeCarbon* diaktifkan pada saat proses *training*.
2. **Perancangan Teknis**
   a. Mekanisme *PEFT*: *QSBoRA-FA* (matriks A dibekukan/terstruktur; B terlatih), model di-*load* dalam kuantisasi *4-bit*.
   b. *Target modules* dan *freezing*:
      1) *LLaMa 3.2 11B Vision: q/k/v/o, gate/up/down (MLP), embed_tokens, lm_head → adapted; vision path → freeze.*
      2) *Gemma 3 12B PT*: *q/k/v/o, gate/up/down → adapted; embed_tokens/lm_head → freeze; vision path → freeze.*
   c. *Hyperparameters adapter* awal: r = 128, α = 128, *dropout* 0,05.
   d. Model di-*load* dalam kuantisasi *4-bit*, mengikuti prinsip *QSBoRA-FA*

3. **Implementasi *CP* (*Continued Pretraining*)**
    a. *Pre-processing*: normalisasi, *tokenization*, *length filtering*, deduplikasi ringan, *flattening*, dan *pre-processing* lainnya, menyesuaikan kebutuhan.
    b. *Sampling* bahasa: campuran *subset* terpilih untuk bahasa Sunda dan bahasa Jawa dengan rasio yang menjaga keseimbangan dan representasi bahasa Sunda/Jawa, yakni rasio 50:50 dengan *time-boxing*.
    c. *Hyperparameter Trainer* disesuaikan dengan *training telemetry* dan penggunaan memori untuk mencegah terjadinya *OOM (Out of Memory)*.
    d. Pelatihan: *mixed precision*, *gradient accumulation*, *early stopping* berbasis *validation loss*, menggunakan *trainer*.
    e. Artefak: *Adapter weights CP*, *training logs*, profil energi (*CodeCarbon*).
    f. *Rollback* cepat: jika tidak stabil → *fallback* ke r = 16, α = 48 (aturan α = 3r untuk konteks *low-resource*) lalu ulang putaran singkat.
4. **Implementasi *IT* (*Instruction Tuning*)**
    a. Menggunakan *tokenizer* berasal dari model versi *instruct* dikarenakan kebutuhan *chat_template* yang hanya dimiliki *tokenizer* model versi *instruct,* berikut detailnya:
        1) *LLaMa* 3.2 *11B Vision* menggunakan *tokenizer LLaMa 3.2 11B Vision Instruct.*
        2) *Gemma 3 12B PT* menggunakan *tokenizer Gemma 3 12B IT.*
    b. *Pre-processing*: normalisasi, *tokenization*, *length filtering*, deduplikasi ringan, *chat-formatting*, *assistant-masking*, dan *pre-processing* lainnya, menyesuaikan kebutuhan.
    c. Kurikulum instruksi: rasio 50:50 per bahasa (bahasa Sunda dan bahasa Jawa).
    d. *Hyperparameter SFTTrainer* disesuaikan dengan *training telemetry* dan penggunaan memori untuk mencegah terjadinya *OOM (Out of Memory)*.
    e. Pelatihan: *mixed precision*, *gradient accumulation*, *early stopping* berbasis *validation loss*, menggunakan *SFTTrainer*.

f. Kontrol stabilitas: pantau *over-steering/gibberish* pada *test set*.
g. *Rollback* cepat: jika tidak stabil → *fallback* ke r = 16, α = 48 (aturan α = 3r untuk konteks *low-resource*) lalu ulang putaran singkat.
h. Artefak: *adapter IT* dan atau *DPO*, *config final*, *changelog*.

5. **Opsi *Post-training* (jika diperlukan)**
   a. *DPO*/*Preference-based Post-training*: diaktifkan hanya bila keluaran instruksi masih belum memadai setelah *IT*.
   b. Sumber data preferensi/instruksi tambahan: *KD* berbasis *Gemini 2.5 Pro* (teks) untuk menghasilkan pasangan instruksi–jawaban/demonstrasi yang relevan dengan bahasa Sunda dan bahasa Jawa.
   c. Kebijakan *freezing* mengikuti §3.2.2 (tetap fokus pada *language layers*; *vision path* beku).
   d. *Guardrail*: mulai dari konfigurasi ringan (seperti *learning rate* rendah, *batch* kecil) dan evaluasi cepat pada *test set* sebelum dilanjutkan.
6. ***Smoke Test* dan *Quality Gates***
   a. Sebelum dilaksanakan *smoke test*, *monitoring training telemetry* dilakukan terlebih dahulu.
   b. Pelaksanaan: *smoke test* dilakukan langsung di *Colab* dengan *notebook* inferensi yang memuat model terbaru.
   c. Cakupan cepat:
      1) Teks: *subset* kecil per bahasa (bahasa Sunda dan bahasa Jawa) untuk memeriksa adanya *gibberish* dan kelancaran secara kasat mata.
      2) Visi–teks *(sanity check)*: *subset* kecil pasangan gambar–teks internal (jika ada) untuk memastikan *grounding* dasar, tanpa klaim evaluasi *multimodal* formal.
   d. *Gate* minimal: lulus *smoke test* (tanpa *gibberish*/kegagalan fatal) → lanjut; jika gagal → perbaiki (tu*run rank*, sesuaikan α, atau perketat *regularization*), ulang *smoke test*.
7. **Pengendalian Eksperimen**

a. Satu *run* = satu *Colab session* lengkap dengan *CodeCarbon*.
b. *Temp checkpoint* setiap *run* didorong ke *Hugging Face Hub* sebagai repositori privat untuk keperluan *rollback* dan replikasi.
c. *Seed* dan *config* tersimpan di *notebook* (termasuk versi pustaka) untuk reproduksibilitas.

8. Rencana Iterasi Pengembangan (*CP* → *IT* → *Refine* / PT opsional)
   a. Satu siklus: *CP* singkat → *IT* → *smoke test* (*Colab*) → *gate* minimal
   b. Jika stabil: lakukan *refine* kecil (seperti *LR sweep* singkat). Jika tidak stabil: aktifkan *fallback* r=16, α=48, ulang segmen *CP*/*IT* singkat.
   c. *Post-training* (*DPO*) hanya diaktifkan bila diperlukan setelah *IT*, dengan *KD Gemini 2.5 Pro* sebagai sumber preferensi/instruksi tambahan.
9. Artefak dan Dokumentasi
   a. Simpan: *adapter weights CP*/*IT*/*DPO, training dan eval logs*, profil energi (*CodeCarbon*), *notebook Colab*, serta *config (.json)* yang terkait.

3.4 Pengujian (*Testing*)

Tahap pengujian berfungsi sebagai *quality gate* ringan untuk memverifikasi kelayakan keluaran sebelum evaluasi menyeluruh. Cakupan dibatasi pada bahasa Sunda dan bahasa Jawa dengan tiga *smoke test* berjenjang, *CP* (kelanjutan konteks), *IT* (kepatuhan instruksi untuk teks dan citra), dan *DPO* (*alignment* preferensi). Pengujian bersifat operasional non-terdokumentasi, fokus pada *smoke test*/*quality gate* (koherensi, relevansi lokal, ketiadaan *gibberish*, serta konsistensi bahasa tanpa menilai *register*). Hasilnya dipakai sebagai keputusan *Go/No-Go* cepat.

3.4.1 Ruang Lingkup dan Pengecualian

Pengujian pada bagian ini berfungsi sebagai *quality gate* ringan sebelum evaluasi menyeluruh. Cakupan dibatasi pada bahasa Sunda dan bahasa Jawa. Uji bersifat operasional non-terdokumentasi (hasil tidak diarsipkan; hanya pemeriksaan cepat). Pengecualian: evaluasi *register* berada di luar lingkup (*out of scope*).

3.4.2 *Smoke Test* untuk *Continued Pretraining (CP)*

Smoke test pada tahap *Continued Pretraining* bertujuan untuk

memverifikasi bahwa model menghasilkan kelanjutan yang wajar, terbentuk baik, dan relevan secara lokal terhadap masukan teks maupun citra berbahasa Sunda dan Jawa, tanpa menuntut *instruction following* atau *formatting* spesifik. Keluaran *CP* boleh tampak "*rogue*" (tidak terformat rapi/kurang *task-aware*), namun tidak *gibberish*.

1. Rancangan Uji (*Micro CP, Context+Image-8*, non-terdokumentasi)
   a. Bahasa dan Modus: bahasa Sunda dan bahasa Jawa; *teks→teks* dan *image→teks*.
   1) Sampel: total 8 percobaan per *run*:
      (a) Teks→teks (4): 2 konteks bahasa Sunda + 2 konteks bahasa Jawa (±15–50 *token*) dari narasi/eksposisi ringkas.
      (b) Gambar→teks (4): 2 gambar umum (objek/adegan sederhana) dipasangkan dengan jangkar bahasa minimal (lihat di bawah) untuk bahasa Sunda dan bahasa Jawa.
   2) Inferensi: *greedy decoding* (*temperature* = 0,0), *max_new_tokens* = 50, *seed* = 42.
   3) Jangkar bahasa untuk *image* (tanpa instruksi tugas):
      (a) Bahasa Sunda: *"Gambar ieu nembongkeun:"* + ⟨*image*⟩ → lanjutkan generasi.
      (b) Bahasa Jawa: *"Gambar iki nerangake:"* + ⟨*image*⟩ → lanjutkan generasi.
      (Fungsinya hanya sebagai penanda bahasa; bukan *prompt* tugas.)

2. Kriteria Kualitatif (visual, non-kuantitatif)

Sampel dinilai *OK* apabila seluruh butir berikut terpenuhi:

1. Bahasa konsisten (bahasa Sunda untuk jangkar bahasa Sunda; bahasa Jawa untuk jangkar bahasa Jawa).
2. Relevansi lokal terjaga: kelanjutan membahas konteks/citra yang sama, tidak menyimpang total.
3. Kalimat terbentuk baik: struktur dasar dan tanda baca memadai; tidak terjadi

looping/karakter acak.

Keputusan *CP (soft gate)*: mayoritas sampel (≥ 6/8) *OK* → *CP-OK* (lanjut). Jika tidak, tandai risiko dan periksa *tokenizer/detokenization/data mix*; *CP* tetap *non-blocking* terhadap fase berikutnya.

3.4.3 *Smoke Test* untuk *Instruction Tuning* (*IT*)

Smoke test pada tahap *Instruction Tuning* bertujuan untuk memverifikasi kepatuhan instruksi dasar dan minimnya *mixing* pada keluaran bahasa Sunda dan bahasa Jawa untuk teks dan citra-berkondisi, tanpa menilai *register*.

1. Rancangan Uji (*Micro IT, Text+Image-8*, non-terdokumentasi)
   a. Set *prompt* (tetap lintas *run*): total 8 (4 bahasa Sunda + 4 bahasa Jawa), terdiri atas 2 *text-only* dan 2 *image-grounded* per bahasa:
      1) *Text-only* (2 per bahasa):
         (a) *QA* singkat (1 kalimat tanya umum),
         (b) *Paraphrase*/ringkas paragraf pendek (≤60 kata).
      2) *Image-grounded* (2 per bahasa):
         (c) *Descriptive grounding* (uraikan objek utama, relasi sederhana *subjek–aksi–atribut*),
         (d) *Robust continuity* (pertahankan bahasa target meski *prompt* mengandung 1–2 kata non-target).
   b. Inferensi: *seed* = 42; *temperature* = 0,2; *top_p* = 0,9; *max_new_tokens* = 256; gunakan *chat template* standar.
   c. Contoh formulasi (inti instruksi dalam bahasa target):
      1) Bahasa Sunda, deskripsi citra: *"Jelaskeun eusi gambar dina basa Sunda, sing jelas tur rinci."*
      2) Bahasa Jawa, deskripsi citra: *"Terangna isine gambar nganggo basa Jawa kanthi cetha lan ringkes."*
2. Kriteria Kualitatif (visual)

Sampel dinilai *OK* apabila:

1. Instruksi dipatuhi (format jawaban sesuai:

daftar/ringkas/parafrasa/deskripsi),

2. Tidak *gibberish* (tidak kosong, tidak *looping*),
3. Tidak terjadi *mixing* kasat mata (jawaban konsisten dalam bahasa Sunda dan bahasa Jawa),
4. Untuk *image-grounded*: uraian sejalan dengan isi *image* pada level objek/aksi/atribut wajar (tanpa menuntut *OCR* atau fakta rinci).

   Keputusan *IT* (*gate* utama): mayoritas sampel (≥ 6/8) OK → *Go*. Jika tidak → *No-Go* dan lakukan *triage* singkat (tu*run*kan *temperature*, tambahkan *few-shot anchor* "Tetap gunakan bahasa [X]", verifikasi *eos_token_id* dan *repetition_penalty*).

3.4.4 *Smoke Test* untuk *Post-training* (*DPO*), *Alignment Gate*

Smoke test pada tahap *Post-training* bertujuan untuk memverifikasi *preference alignment* dasar pada keluaran bahasa Sunda dan bahasa Jawa menggunakan pasangan kecil *chosen vs rejected*, tanpa metrik probabilistik penuh; keputusan bersifat kualitatif dan cepat.

1. Rancangan Uji (*Micro DPO, Qual-12*, non-terdokumentasi)
   a. Set: 12 pasangan *non-Train* $(x,\ y_{chosen},\ y_{rejected})$ → 6 bahasa Sunda + 6 bahasa Jawa.
   b. Modus: *text-only* (opsional *image-grounded* bila tersedia, namun tidak diwajibkan).
   c. Inferensi: *seed* tetap (42), *temperature* = 0,2, *top_p* = 0,9, *max_new_tokens* = 256; gunakan *chat template* standar.
2. Kriteria Kualitatif (visual) dan Keputusan *Go/No-Go*

Sebuah pasangan dinilai OK jika keluaran model lebih dekat ke *ychosen (bentuk/struktur/kepatuhan instruksi) dibanding yrejected*, tanpa *over-refusal* pada *benign prompts*, tanpa *gibberish*, dan tanpa *mixing* kasat mata (tetap bahasa Sunda dan bahasa Jawa).

Keputusan: mayoritas pasangan *OK* (≥ 8/12) → *Go*; selain itu *No-Go*.

Triage singkat (bila *No-Go*): kurasi ulang pasangan ambigu/*noisy*,

penyesuaian hiperparameter β, validasi *reference policy*, dan uji ulang dengan *parameter* inferensi yang diketatkan.

3.5 *Deployment*

Tahap ini bertujuan untuk menyajikan model yang telah lolos *smoke test* ke tahap *Review* sehingga pakar dapat menguji interaktif (teks maupun citra) via *Hugging Face Space* berbasis *Gradio*, dengan penentuan bahasa implisit (model membalas mengikuti bahasa masukan pengguna; tidak ada pengalih bahasa di *UI*). *Register* berada di luar cakupan.

3.5.1 Alur *Deployment*

Berikut ini merupakan alur dari tahap *Deployment*:

1. *Gate → Review*. Setelah lulus *smoke test* (Bab 3.4), model di-*deploy* untuk *Review* pakar.
2. Simpan dan *push adapter* (*PEFT*) ke *Hugging Face (private repo)* sebagai titik *rollback*.
3. *Merge adapter → base* model (*LLaMa*/*Gemma*) menjadi *merged weights* siap inferensi.
4. *Push merged weights* ke *Hugging Face* (*private repo* terpisah) dengan *tag* versi.
5. Bangun *Hugging Face Space* (publik) berbasis *Gradio* yang memuat model *(merged)*, mengikuti *Wireframe* halaman awal dan *Chatbox* (Gambar 3.2–3.3).

3.5.2 Lingkungan *Serving*

Deployment memanfaatkan *Hugging Face Spaces* (Gradio) dengan *GPU NVIDIA H200 (ZeroGPU)*, menjaga paritas versi pustaka inti (*Transformers, Accelerate, PEFT*, dll.) terhadap ekosistem *Colab* sesuai yang telah ditetapkan di §3.2.1.2 dan Lampiran 3. Fokusnya adalah pemuatan bobot efisien, inferensi *multimodal* (teks–gambar), dan I/O ringan tanpa komponen pelatihan.

3.5.3 Perilaku *UI*

Di bawah ini merupakan perilaku dari antarmuka pengguna:

1. Kartu contoh (*text-only* dan *multimodal*)

   Fungsi kartu contoh hanya mengisi kolom *input* sesuai contoh (serta melampirkan gambar jika kartu *multimodal*). Tidak mengubah bahasa apa pun.

   Contoh: klik kartu "Contoh pesan *multimodal*" → kolom teks terisi contoh kalimat, gambar contoh terlampir → pengguna menekan Kirim.

2. Penentuan bahasa (implisit, tanpa kontrol di *UI*)

   Aturan tunggal: balasan mengikuti bahasa teks terakhir dari pengguna pada percakapan (Sunda atau Jawa).

   a. Contoh 1: pengguna menulis dalam Sunda → balasan Sunda.
   b. Contoh 2: pada giliran berikutnya pengguna menulis dalam Jawa → balasan Jawa.
   c. *Pesan pertama hanya gambar: UI* menampilkan *helper text* agar pengguna mengetik jangkar singkat terlebih dahulu.
      1) Sunda: *"Tulis hiji frasa pondok dina basa Sunda."*
      2) Jawa: *"Tulisen ukara cekak nganggo basa Jawa."*

      Kirim dinonaktifkan sampai ada teks jangkar.

3. Modus *Multimodal (image→text)*

   Jika ada gambar, bahasa keluaran mengikuti bahasa teks yang diketik bersama gambar (jangkar). Bila pengguna menambahkan gambar tanpa teks pada pesan pertama, berlaku aturan pada butir B (wajib jangkar singkat dulu).

   Contoh: pengguna mengetik "*Jelaskeun eusi gambar sing basajan.*" lalu melampirkan gambar → keluaran dalam Sunda yang menjelaskan isi gambar.

3.5.4 Operasional dan *Rollback*

Berikut ini merupakan praktik operasional dan *rollback* yang dilakukan:

1. Keamanan: *HF token* disimpan sebagai *secret*; repositori model tetap *private* sebelum *launch.*

2. Monitoring ringan: amati *error/latency/queue*; lakukan perbaikan apabila diperlukan.
3. *Rollback*: alihkan *Space* ke *tag weights* stabil sebelumnya, *rebuild* dari *adapter-only,* atau *rollback* ke versi sebelumnya.

3.5.5 Batasan

Deployment ini digunakan untuk tahap *review* kualitatif (Sunda/Jawa) dan bukan *benchmark* kuantitatif; artefak rilis resmi adalah model *weight*s di repositori *HF*, sementara *HF Space* berperan sebagai etalase uji interaktif.

3.6 *Review*

Bagian *Review* mengonsolidasikan bukti kinerja melalui dua jalur: (1) evaluasi kuantitatif dengan metrik terstandar, *chrF++* untuk *MT* serta *Weighted F1-score* dan *Accuracy* untuk klasifikasi, dan (2) evaluasi kualitatif *live* oleh penutur asli bahasa Sunda dan bahasa Jawa menggunakan kuesioner skala *Likert*. Hasil kuantitatif memberi dasar perbandingan dan replikasi, sementara umpan balik pakar memotret kegunaan praktis; penilaian *register* berada di luar lingkup.

3.6.1 Evaluasi Kuantitatif

Evaluasi kuantitatif menggunakan metrik numerik terstandar untuk mengukur kinerja model secara objektif. Setiap keluaran model, misalnya terjemahan (*machine translation/MT*) atau prediksi *sentiment analysis*, diubah menjadi skor melalui metrik yang mapan. Pendekatan ini memudahkan perbandingan model, pelacakan kemajuan eksperimen, serta replikasi.

3.6.1.1 Instrumen dan Landasan

Berikut ini merupakan metrik yang digunakan di tahap evaluasi kualitatif:

1. *chrF++ (Machine Translation)*, menghitung *F-score* atas tumpang tindih *n-gram* di tingkat karakter dan kata; tahan variasi tokenisasi/morfologi. Rumus presisi, *recall*, dan *F-score* merujuk Persamaan (2.6)–(2.8) [61–62].

2. *Weighted F1-score (Sentiment Analysis),* menghitung *F1* per kelas, lalu rata-rata tertimbang menurut *support* kelas; meredam bias kelas mayoritas (Persamaan 2.9) [63–64].
3. *Accuracy (Sentiment Analysis)*, proporsi prediksi benar (Persamaan 2.10); sederhana namun rentan bias pada *class imbalance*, sehingga dilaporkan bersama *Weighted F1-score* [65].

3.6.1.2 Prosedur Pengujian

Berikut ini merupakan prosedur pengujian di tahap evaluasi kualitatif:

(A) *Machine Translation (MT) , chrF++*

1. Arah uji: *sun↔ind, jav↔ind* dan *sun↔jav*.
2. Pra-proses: penghapusan *formatting*, normalisasi spasi dan *generation cutoff.*
3. Inferensi: *greedy* atau *beam search* (tetap konsisten antar model).
4. Skoring: laporkan *chrF++* korpus per arah.

(B) *Sentiment Analysis*, *Weighted F1-score* dan *Accuracy*

1. Label: {positif, netral, negatif}.
2. Inferensi: *text-only*; *thresholding* standar (kelas *argmax*).
3. Skoring: *Weighted F1-score* (utama) dan *Accuracy* (pendamping).

3.6.1.3 Batasan

Berikut ini merupakan batasan di tahap evaluasi kualitatif:

1. Ruang tugas terbatas pada *MT* dan sentimen (teks); evaluasi *multimodal* (teks–gambar) dinilai pada *live review*.
2. Akurasi konteks budaya tidak sepenuhnya tertangkap metrik otomatis; oleh karena itu hasil kuantitatif melengkapi *review* pakar, bukan menggantikannya.
3. *Pre-/post-processing* yang berbeda dapat memengaruhi skor; konfigurasi dikunci agar paritas antarmodel terjaga.

3.6.2 Evaluasi Kualitatif

Evaluasi kuantitatif ini adalah evaluasi langsung (*live evaluation*) untuk mengukur performa *Large Vision Language Model (LVLM)* pada bahasa Sunda dan bahasa Jawa dengan tugas teks dan teks–gambar yang mewakili konteks komunikasi umum.

3.6.2.1 Komposisi *Evaluator*/Pakar

Sebanyak delapan penutur asli dilibatkan, empat untuk bahasa Sunda dan empat untuk bahasa Jawa. Partisipan tidak seluruhnya berprofesi sebagai guru; komposisi mencakup guru, staf sekolah/pabrik, dan pengurus yayasan (lihat Lampiran 1 dan 2). Pemilihan berfokus pada keberagaman sudut pandang, dialek *(registers)*, dan kepekaan budaya.

3.6.2.2 Instrumen dan Landasan

Berikut ini merupakan instrumen dan landasan yang digunakan di tahap evaluasi kuantitatif:

1. Instrumen: kuesioner *Likert* 5 poin pada *Google Form* (empat bagian; butir lengkap dicantumkan di bawah).
2. Landasan: *MMStar*, *VALOR-Eval*, dan *HarmonicEval* digunakan untuk merancang butir dan fokus evaluasi (seperti *dependency visual*, *coverage*, *faithfulness*, *multikriteria*). Skoring akhir tidak memakai perhitungan kerangka tersebut, hanya hasil kuesioner *Likert* pada *Google Form*.

3.6.2.3 Prosedur *Live*

Evaluasi kualitatif dilaksanakan secara tatap muka (luring*, in-person*) untuk menilai performa model pada bahasa Sunda dan Jawa. Sesi dilakukan di lokasi uji yang telah ditentukan, dengan alur berikut:

1. **Instrumen**

   Kuesioner *Likert* beserta kolom komentar/saran pada *Google Form* yang akan diisi oleh *evaluator.*

2. **Landasan perancangan butir**

   Item evaluasi dirancang berdasarkan kerangka *MMStar, VALOR-Eval,* dan *HarmonicEval* (*dependency visual, coverage, faithfulness,* multikriteria). Skoring akhir hanya menggunakan hasil kuesioner *Likert*, tanpa perhitungan formula asli kerangka tersebut.

3. **Skenario uji**

   - **Teks**: pertanyaan budaya, ringkasan, terjemahan singkat.
   - ***Multimodal (image→text)***: deskripsi objek/aksi/atribut.

     Prompt diambil dari *bank prompt* pada Lampiran 9-10.

4. **Prosedur sesi**

   a) *Briefing* singkat mengenai tujuan dan aturan.

   b) *Evaluator* menjalankan *prompt* pada *demo Hugging Face Space*.

   c) *Evaluator* mengisi kuesioner *Likert* dan komentar/saran pada *Google Form*.

5. **Kebijakan bahasa**

   Bahasa keluaran mengikuti bahasa teks masukan (Sunda/Jawa). *Register* tidak dinilai (di luar lingkup).

6. **Output sesi**

   Hasil kuesioner pada *Google Form* yang telah diisi oleh *evaluator*.

3.6.2.4 Skema Skoring

Berikut merupakan skema skoring yang berbasis pada hasil di *Google Form*:

1. Skala: *Likert* 5 poin (Sangat Kurang–Sangat Baik) per butir, dengan detail sebagai berikut:
   a. Sangat Kurang
   b. Kurang
   c. Cukup
   d. Baik
   e. Sangat Baik
2. Rekap: hasil diperoleh dari ringkasan *Google Form* (rekap per butir).

3. Perbandingan model: ditarik dari Bagian II (*LLaMa 3.2 11B Vision*) vs Bagian III (*Gemma 3 12B PT*) serta Bagian IV *(Vis-à-vis).*

3.6.2.5 Naskah Butir *Google Form*

Berikut merupakan skema skoring yang berbasis pada hasil di *Google Form*:

Bagian 1 dari 4 , Formulir *Evaluator*: Evaluasi Keseluruhan Model *LVLM*

Bagian I – Identitas *Evaluator*

1. Nama Lengkap (*)
2. Usia (*)
    a. Anak-anak (6–12 tahun) / Remaja (13–17) / Dewasa Muda (18–25) / Dewasa (26–45) / Paruh Baya (46–65) / Lansia (66+)
3. Asal (Kota/Provinsi) (*)
4. Bahasa Ibu (*) → Sunda / Jawa
5. Bahasa yang Dievaluasi (*) → Sunda / Jawa

Bagian 2 dari 4, Bagian II – Penilaian Model *LVLM* Berbasis *LLaMa 3.2 11B Vision (Kenanga 11B DPO)*

Pilih satu jawaban: Sangat Kurang / Kurang / Cukup / Baik / Sangat Baik untuk tiap pernyataan.

1. Apakah model memanfaatkan informasi visual dengan tepat?
    a. Penjelasan: Apakah informasi pada gambar digunakan dengan benar untuk menjawab *prompt*?
    b. Contoh: Jika gambar menunjukkan petani menanam padi, apakah model menjawab tentang aktivitas menanam padi? (*dependency visual*, *MMStar 2024*).
2. Apakah output mencakup semua informasi penting?
    a. Penjelasan: Apakah jawaban menyertakan detail utama dari input (teks/gambar)?
    b. Contoh: Jika *prompt* meminta langkah pembuatan surabi, apakah seluruh tahapan disebutkan? (*coverage*, *VALOR-Eval 2024*).

3. Apakah output setia pada isi input (teks/gambar)?
    a. Penjelasan: Model tidak menambahkan informasi keliru/melewatkan fakta penting.
    b. Contoh: Jika *prompt* menyebut sendok kayu, model tidak menulis sendok *stainless steel* (*faithfulness*, *VALOR-Eval 2024*).
4. Apakah kualitas keseluruhan konsisten di berbagai kriteria?
    a. Penjelasan: Stabil pada aspek visual, cakupan, dan ketepatan tanpa ada satu kriteria sangat buruk (*multikriteria*, *HarmonicEval 2024*).
5. Seberapa layak digunakan di kehidupan nyata?
    a. Penjelasan: Kelayakan untuk *chatbot layanan*, dsb. Apakah jawaban cukup akurat dan natural?

Bagian 3 dari 4 , Bagian III – Penilaian Model *LVLM* Berbasis *Gemma 3 12B PT (Kenanga 12B IT DPO)*

Butir dan skala sama seperti Bagian II, diganti menyebut *Gemma 3 12B PT (Kenanga 12B IT DPO).*

Bagian 4 dari 4 , *Wrap Up* Performa Model, *Vis-à-vis*

1. Model mana yang menurut Anda lebih baik?
    a. Kenanga *11B IT* (*LVLM* Berbasis *LLaMa* 3.2 *11B Vision*)
    b. Kenanga *12B IT DPO* (*LVLM* Berbasis *Gemma 3 12B PT*)
2. Mengapa model tersebut menurut Anda lebih baik? *(isian bebas, wajib)*
3. Komentar atau saran *(*isian bebas, wajib*)*

3.6.2.6 *Bank Prompt*

Bank prompt yang digunakan pada evaluasi kuantitatif mencakup bahasa Sunda dan Jawa dalam format teks maupun *multimodal* (*image→text*), seperti tercantum pada Lampiran 9 hingga Lampiran 10. Untuk bahasa Sunda teks, prompt meliputi pertanyaan budaya, resep, terjemahan, hikmah cerita, dan penjelasan konsep untuk anak-anak dengan fokus evaluasi pada *cultural fit, coverage, fluency, simplicity,* dan *faithfulness*. Versi *multimodal* menambahkan gambar petani, tari, anak dan kucing, guru, serta alat musik dengan target evaluasi serupa. Bank prompt

bahasa Jawa juga terbagi teks dan *multimodal*, mencakup aktivitas menanam padi, candi, anak dan kucing, guru, serta alat musik, dengan fokus evaluasi yang sama, termasuk penggunaan instruksi ringkas dan terjemahan, untuk menilai *faithfulness, coverage, coherence, fluency, simplicity*, dan *cultural fit*.

3.6.3 Catatan Pelaksanaan

Berikut merupakan catatan pelaksanaan evaluasi kualitatif:

1. Non-anonim: identitas *evaluator* dikumpulkan pada *Google Form* (Bagian I).
2. Bahasa keluaran: mengikuti bahasa *input evaluator* (tanpa *toggle*).
3. Cakupan *register*: tetap berada di luar lingkup penilaian.

3.7 *Launch*

Bagian ini mendeskripsikan peluncuran model ke publik dalam bentuk bobot (*weights*). Antarmuka *HF Space* (lihat §3.5) tetap berfungsi sebagai demo interaktif untuk uji coba, sedangkan artefak rilis resmi adalah model *weights* yang tersedia publik.

3.7.1 Artefak yang Diluncurkan

Berikut merupakan artefak yang diluncurkan di tahap *launch*:

1. Dua model final (*Adapter* telah di-*merge* kembali dengan *base* model):
    a. Kenanga *11B IT* (*LVLM* berbasis *LLaMa 3.2 11B Vision*),
    b. Kenanga *12B IT DPO* (*LVLM* berbasis *Gemma 3 12B PT*).
2. Format berkas: **.safetensors* (di-*shard* bila diperlukan), beserta *config* inferensi standar (*config.json*, *generation_config.json*, *tokenizer* dan berkas terkait).

3.7.2 Publikasi di *Hugging Face*

Bagian ini memastikan artefak penelitian didistribusikan secara terbuka, terstruktur, dan mudah direplikasi.

1. Repositori model (publik) untuk masing-masing model (terpisah).
2. Setiap repositori berisi:
3. *README*.md (sekaligus model *card*) dengan:

a. ringkasan model dan ruang lingkup bahasa (Sunda/Jawa),
b. contoh pemakaian (*usage snippets*) untuk *text-only* dan *image-grounded*,
c. catatan *Deployment* singkat (merujuk *HF Space* demo),
d. batasan dan *safety notes* seperlunya,
e. *tags*/*metadata* yang relevan (seperti *language*: *sun, jav, pipeline_tag: text-generation, image-to-text,* dsb.).

4. Berkas bobot model.*safetensors* (atau beberapa *shards*), dan berkas *config/tokenizer* yang konsisten dengan *runtime*.
5. Penamaan versi: *tag* rilis (seperti v1.0) dan *release notes* singkat yang merangkum asal *base* model, status *merge adapter*, serta *compatibility* pustaka.

3.7.3 Relasi dengan *Deployment*

Berikut merupakan relasi tahap ini dengan tahap *Deployment:*

1. *HF Space* (publik): kanal uji coba oleh pengguna/pakar (tanpa *toggle* bahasa; bahasa keluaran mengikuti masukan).
2. Repositori model (publik): sumber rilis resmi untuk *weights* yang dapat di-*pull* dan dijalankan secara lokal/di *server* lain.