**MIND (Multimedia Artificial Intelligent Networking Database) Journal**

MIND Journal | ISSN (p): 2528-0015 | ISSN (e): 2528-0902 | Vol. 9 | No. 2 | Halaman 248 - 260
DOI : https://doi.org/10.26760/mindjournal.v9i2.248-260 Desember 2024

# *Fine-Tuning* LLaMA-2-Chat untuk ChatBot Penerjemah Bahasa Gaul menggunakan LoRA dan QLoRA

**ANDRI SUSILO, VINY CHRISTANTI, MANATAP DOLOK LAURO**

Program Studi Teknik Informatika, Universitas Tarumanagara
Email: andri.535210074@stu.untar.ac.id

*Received* 18 November 2024 **|** *Revised* 25 November 2024 **|** *Accepted* 27 Desember 2024

**ABSTRAK**

*Bahasa gaul, yang berkembang pesat di kalangan generasi Z dan Alpha, sering kali sulit dipahami oleh generasi lain atau dalam konteks formal. Bahasa ini memiliki variasi yang tidak terstruktur dan terus berubah, memerlukan model bahasa yang adaptif untuk memahaminya. Penelitian ini bertujuan untuk mengukur kualitas hasil terjemahan fine-tuning model LLaMA-2 dalam menerjemahkan bahasa gaul ke bahasa formal, dengan menggunakan metrik evaluasi BLEU Score sebagai alat utama. Selain itu, pendekatan LoRA dan QLoRA digunakan untuk meningkatkan efisiensi fine-tuning dengan mengurangi kebutuhan komputasi dan memori. Dataset yang digunakan terdiri dari data media sosial dan data buatan yang diformat dalam bentuk percakapan untuk menangkap konteks secara lebih baik. Hasil evaluasi menunjukkan skor BLEU terbaik sebesar 0.0369, yang menegaskan bahwa model masih perlu disempurnakan untuk menghasilkan terjemahan bahasa gaul yang optimal.*

***Kata kunci**: bahasa gaul, LLaMA-2, LoRA, QLoRA*

**ABSTRACT**

*Slang, which is growing rapidly among generations Z and Alpha, is often difficult for other generations to understand or in formal contexts. This language has unstructured variations and is constantly changing, requiring adaptive language models to understand it. This research aims to measure the quality of the translation results of fine-tuning the LLaMA-2 model in translating slang into formal language, using the BLEU Score evaluation metric as the main tool. Additionally, LoRA and QLoRA approaches are used to improve fine-tuning efficiency by reducing computing and memory requirements. The dataset used consists of social media data and artificial data formatted in conversational form to better capture context. The evaluation results show the best BLEU score of 0.0369, which confirms that the model still needs to be refined to produce optimal slang translations.*

***Kata Kunci**: slang language, LLaMA-2, LoRA, QloRA*

## 1. PENDAHULUAN

Bahasa gaul telah berkembang menjadi fenomena sosial yang dinamis dan terus berubah, terutama dikalangan generasi z dan generasi alpha, sebagai refleksi dari budaya populer dan kreativitas individu **(Dewi, 2023)**. Bahasa ini sering kali sulit dipahami oleh generasi lain atau dalam konteks formal. Tantangan ini semakin kompleks dengan munculnya variasi bahasa yang tidak memiliki struktur baku, sehingga penerjemahan dan pemahaman terhadap bahasa gaul menjadi tantangan tersendiri **(Satriani, 2023).** Dalam bidang pemrosesan bahasa alami (NLP), pengembangan model yang mampu memahami dan menerjemahkan bahasa gaul membutuhkan pendeketan yang lebih adaptif. Salah satu tantangan utama adalah dinamika kosakata yang terus berkembang dan keberagaman pola komunikasi yang tidak mengikuti kaidah tata bahasa standar.

Model bahasa konvensional seperti LSTM, Naive Bayes, RNN, dan BERT umumnya mengalami kesulitan dalam menangani tantangan tersebut, karena keterbatasan dalam menangani konteks jangka panjang, variasi bahasa yang cepat berubah, serta ketidakmampuan model-model tersebut dalam menangani data dengan struktur yang tidak teratur atau tidak baku. Model-model ini sering kali membutuhkan struktur bahasa yang lebih baku dan konsisten untuk dapat menghasilkan prediksi yang akurat. Dalam konteks bahasa gaul, yang sangat dinamis dan tidak mengikuti tata bahasa standar, model konvensional sulit beradaptasi dengan cepat terhadap perubahan dan keberagaman bahasa **(Sun, 2021)**. Sehingga diperlukan metode yang lebih spesifik untuk tugas penerjemahan bahasa gaul tersebut.

Model bahasa besar (LLM) seperti LLaMA-2, yang dikembangkan oleh Meta AI, sebagai model generatif *open source*, model ini menawarkan potensi besar dalam menyelesaikan berbagai tugas NLP, termasuk pembuatan teks, percakapan, dan penerjemahan bahasa yang kompleks. Sehingga LLaMA-2 lebih relevan untuk tugas tugas yang melibatkan transformasi teks karena dapat menghasilkan teks baru berdasarkan input yang diberikan **(Roumeliotis, 2023)**. Sebagai model generatif, LLaMA-2 dapat menciptakan teks berdasarkan input yang diberikan, karena model ini memiliki miliaran parameter dan pelatihan pada triliunan token.

Model ini memiliki kapasitas untuk memahami dan menghasilkan teks dalam berbagai gaya bahasa, termasuk bahasa gaul, dan dapat menyesuaikan konteks yang diberikan (Minaee, 2024). Namun, untuk tugas mengoptimalkan model ini pada tugas spesifik seperti penerjemah bahasa gaul, diperlukan *fine-tuning*. Dalam penelitian ini, pendekatan *Low-Rank Adaptation* (LoRA) dan *Quantied Low-Rank Adaptation* (QLoRA) akan digunakan untuk *fine-tuning* model besar seperti LLaMA-2 **(Dettmers, 2024).** Kedua metode tersebut menawarkan solusi efektif untuk *fine-tuning*, metode ini memungkinkan penyesuain parameter model secara efisien, mengurangin kebutuh komputasi, dan meningkatkan kinerja pada tugas spesifik tanpa memodifikasi keseluruhan model. Penelitian ini berfokus pada pemanfaatan model LLaMA-2 dengan tujuan mengevaluasi efektivitas *fine-tuning* dengan metode LoRA dan QLoRA dalam menerjemahkan bahasa gaul ke bahasa formal.

## 2. METODOLOGI PENELITIAN

LLaMA-2 merupakah keluarga model bahasa besar (LLM) yang dikembangkan oleh Meta AI pada tahun 2023. Model ini tersedia dengan beberapa ukuran parameter, yaitu 7B, 13B, 34B, dan 70B. Selain itu LLaMA-2 tersedia dengan versi chat, versi chat adalah model LLaMA-2 yang sudah disesuaikan untuk tugas dialog. Pada penelitian ini, akan menggunakan versi chat dengan parameter yang digunakan adalah 7B dengan 7 miliar parameter.

## 2.1. Arsitektur LLaMA-2

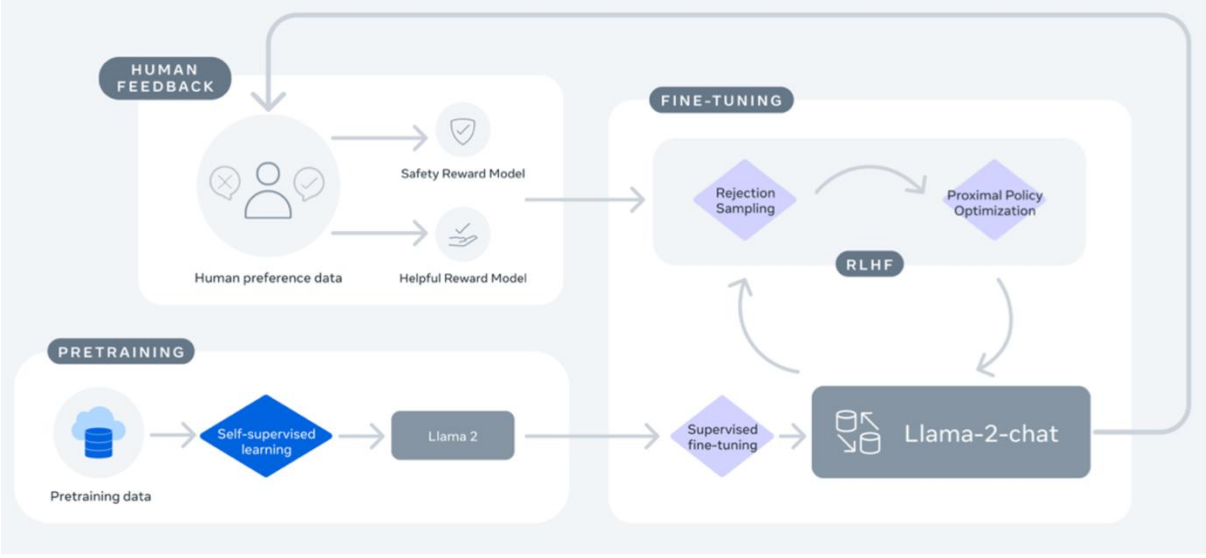
LLaMA-2 adalah model berbasis *Transformer* yang dirancang untuk efisiensi dan skalabilitas, dengan arsitektur *auto-regressive* yang memprediksi token berikutnya berdasarkan konteks sebelumnya. Model ini menggunakan inovasi seperti *pre-normalization* dengan RMSNorm, aktivasi SwiGLU, dan *rotary positional embeddings* (RoPE) untuk menjaga urutan token dalam sekuens **(Jayaseelan, 2023)**. Fitur unggulan LLaMA-2 adalah *Grouped-Query Attention* (GQA), yang meningkatkan efisiensi dalam memproses input dan memperbaiki performa model dalam tugas-tugas yang kompleks i, terutama untuk model berskala besar seperti 34B dan 70B parameter. Fitur LLaMA-2 dapat dilihat pada Tabel 1.

**Table 1. Fitur LLaMA-2**

| ***Training Data*** | ***Params*** | ***Context Length*** | **GQA** | ***Tokens*** | **LR** |
|---|---|---|---|---|---|
| *A new mix of publicly available online data* | 7B | 4K | No | 2.0T | $3.0 \times 10^{-4}$ |
| | 13B | 4K | No | 2.0T | $3.0 \times 10^{-4}$ |
| | 34B | 4K | Yes | 2.0T | $1.5 \times 10^{-4}$ |
| | 70B | 4K | Yes | 2.0T | $1.5 \times 10^{-4}$ |

Setiap lapisan LLaMA-2 mencakup transformasi input menjadi *embeddings*, stabilisasi distribusi bobot dengan RMSNorm, proses *self-attention* menggunakan *KV Cache* untuk konteks panjang, dan *feed-forward layers* dengan SwiGLU untuk efisiensi pemilihan informasi (Zhang, 2019). Peningkatan lainnya meliputi dukungan panjang konteks hingga 4.000 token dan pelatihan menggunakan 2 triliun token, memastikan pemahaman mendalam terhadap berbagai tugas.

Tahapan kerja LLaMA-2-chat melibatkan *pretraining* dengan data publik untuk membangun representasi yang besar, diikuti oleh *fine-tuning* menggunakan *supervised fine-tuning* (SFT) dan *Reinforcement Learning with Human Feedback* (RLHF). SFT menyelaraskan model dengan instruksi manusia melalui dataset teranotasi, sementara RLHF meningkatkan performa dengan memanfaatkan umpan balik manusia (Touvron, 2023). Pelatihan LLaMA-2-chat dengan RLFHF dapat dilihat pada Gambar 1.




**Gambar 1. Pelatihan LLaMA-2-chat dengan RLHF**

## 2.2. *Fine-Tuning*

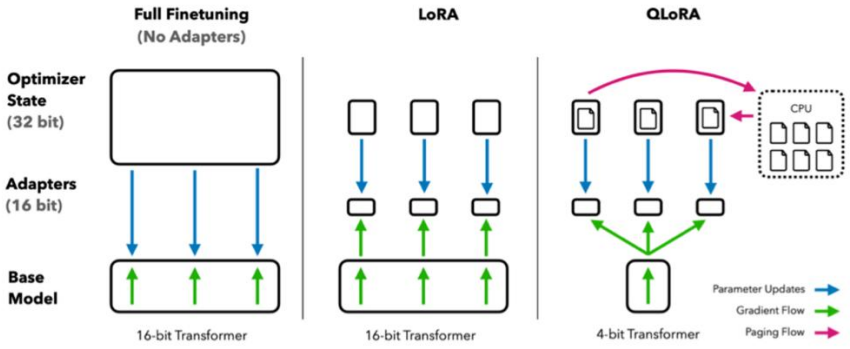
*Fine-tuning* adalah proses penyesuaian model yang telah dilatih sebelumnya (*pre-trained* model) agar lebih optimal untuk tugas atau data tertentu. *Pre-trained* model biasanya dilatih menggunakan dataset besar dan beragam, sehingga model tersebut memiliki kemampuan untuk memahami data secara umum **(Li Z. a.-l., 2023)**. *Fine-tuning* bertujuan untuk mengadaptasi model agar lebih sesuai dengan data spesifik. Keunggulan dari *fine-tuning* meliputi peningkatan kinerja pada tugas tertentu, seperti terjemahaan bahasa gaul, kebutuhan dataset yang lebih kecil, serta efisiensi waktu dan sumber daya komputasi dibandingkan dengan melatih model dari awal **(Church, 2021).** Salah satu pendekatan dalam *fine-tuning* adalah menggunakan teknik *Parameter-Efficient Fine-Tuning* (PEFT), yang memungkinkan penyesuaian model besar dengan mengubah hanya sebagian kecil dari parameter **(Ding, 2023)**. Dalam penelitian ini, metode PEFT yaitu LoRA dan QLoRA, akan digunakan untuk membantu mengurangi kebutuhan memori dan komputasi, memungkinkan adaptasi yang cepat dan hemat sumber daya untuk tugas-tugas yang sangat spesifik **(Han, 2024).**

## 2.3. LoRA dan QLoRA

*Low-Rank Adaptation* (LoRA) adalah teknik *fine-tuning* yang efisien untuk model bahasa besar (LLM). LoRA tidak memperbarui semua bobot modelnya, akan tetapi LoRA mencari perubahan pada bobot model selama proses *fine-tuning*, LoRA memfokuskan pada penyesuaian perubahan bobot dengan mendekomposisi matriks bobot menjadi bentuk *low-rank*. Pendekatan ini memungkinkan penyesuaian model dengan jumlah parameter yang lebih sedikit, sehingga mengurangi kebutuhan memori dan komputasi. Pemilihan *rank* yang tepat sangat penting dalam LoRA, karena menentukan ukuran matriks yang akan diadaptasi, yang pada akhirnya mempengaruhi kinerja model **(Li Y. a., 2023)**.

*Quantization* adalah proses mengonversi bobot model dari presisi tinggi (misalnya, 32-bit *floating point*) ke presisi lebih rendah (seperti 8-bit atau 4-bit). Tujuannya adalah untuk mempercepat proses inferensi dan mengurangi jejak memori model. Namun, *quantization* dapat menyebabkan penurunan akurasi karena hilangnya informasi presisi tinggi **(Nagel, 2021).**

*Quantize* LoRA (QLoRA) menggabungkan keunggulan LoRA dan *quantization*. Dalam QLoRA, model pra-terlatih dikonversi ke format 4-bit untuk mengurangi penggunaan memori, memungkinkan *fine-tuning* model besar pada perangkat dengan sumber daya terbatas. Selama *fine-tuning*, gradien dihitung melalui model yang telah di *quantize*, dan hanya lapisan LoRA yang diperbarui. Pendekatan ini mempertahankan kinerja model yang sebanding dengan *fine-tuning* penuh 16-bit, sambil mengurangi kebutuhan memori secara signifikan **(Dettmers, 2024)**. Perbandingan LoRA dan QLoRA dapat dilihat pada Gambar 2.




**Gambar 2. Perbandingan LoRA dan QLoRA**

## 2.4 BLEU Score

BLEU (*Bilingual Evaluation Understudy*) adalah sebuah algoritma yang awalnya dikembangkan untuk mengevaluasi seberapa akurat teks yang diterjemahkan oleh mesin. BLEU adalah salah satu metrik evaluasi pertama yang mengklaim memiliki korelasi tinggi dengan penilaian kualitas dari manusia. BLEU menggunakan pendekatan pemodelan n-gram untuk membandingkan teks respons dari model dengan teks referensi dalam data uji kebenaran (*ground truth*) **(Ghassemiazghandi, 2024).**

Dalam evaluasi ini, *n-gram* mengacu pada urutan kata yang digunakan untuk mengukur kesesuaian antara output yang dihasilkan dan teks referensi. Pada level paling dasar, unigram mengukur kesesuaian kata per kata, sementara bigram mengukur kesesuaian pada pasangan kata yang berurutan. Evaluasi kemudian berlanjut ke trigram, 4-gram, yang memperluas cakupan analisis ke urutan kata yang lebih panjang. Metrik BLEU menghitung skor berdasarkan perbandingan *n-gram* ini antara output model dan referensi, dengan mempertimbangkan presisi dari kecocokan *n-gram* pada berbagai level.

Hasil perhitungan ini kemudian diubah menjadi skor geometris yang dirata-ratakan, dan penalti panjang (*brevity penalty*) diterapkan untuk menghindari bias pada output yang lebih pendek dari teks referensi. Dengan pendekatan ini, BLEU memberikan gambaran objektif mengenai seberapa baik output model dalam merefleksikan struktur linguistik yang ada pada teks referensi **(Dutta & Klakow, 2019).**

Keunggulan BLEU Score adalah kemudahan perhitungannya dan penggunaannya yang luas dalam berbagai penelitian, sehingga memungkinkan perbandingan antar model berdasarkan tolok ukur yang sama. Namun, BLEU juga memiliki beberapa kekurangan. BLEU Score tidak mempertimbangkan aspek semantik dan memiliki kesulitan dalam menangani bahasa non-Inggris. Selain itu, kelemahan lain dari BLEU adalah asumsinya bahwa terjemahan manusia telah melalui proses tokenisasi, hal ini dapat menyulitkan saat membandingkan model dengan tokenizer yang berbeda **(Mathur, 2020).**

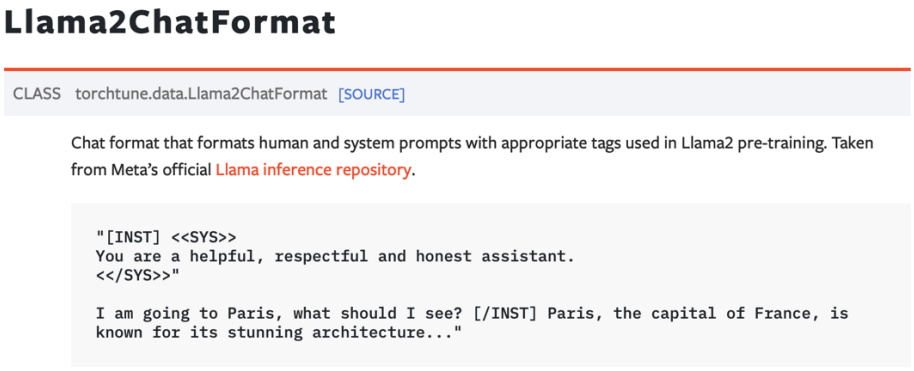
$$BLEU = BP * exp \sum_{n=1}^{N} w_n \log p_n \quad (1)$$

Dimana:

1. BP (Brevity Penalty) mengukur panjang keluaran.
2. w_n adalah bobot dari n-gram.
3. p_n adalah probabilitas kemunculan n-gram dalam keluaran dibandingkan dengan referensi.

## 2.5 Deskripsi Dataset

Dataset yang digunakan dalam proses *fine-tuning* LLaMA-2-chat terdiri atas dua jenis data, yaitu data yang dikumpulkan dari media sosial dan data yang dibuat secara manual. Dataset pertama dikumpulkan melalui proses scraping dari platform YouTube dengan menggunakan YouTube Scraper API dengan pustaka Python googleapiclient. Data yang diambil adalah komentar yang mengandung frasa-frasa bahasa gaul, dengan bantuan daftar kata kunci sebagai filter untuk memastikan relevansi. Setiap komentar yang terkumpul kemudian diproses dan diformat agar sesuai dengan struktur percakapan (*conversation*) yang diperlukan untuk *fine-tuning* LLaMA-2-chat. Format dataset dalam bentuk percakapan ini bertujuan agar model dapat menangkap konteks yang lebih luas dari setiap kalimat bahasa gaul yang ada, sehingga diharapkan mampu menghasilkan terjemahan yang lebih akurat. Format dataset untuk *finetuning* LLaMA-2-chat dapat dilihat pada Gambar 3.




**Gambar 3. Format Dataset**

Dataset kedua dibuat secara manual, baik untuk kalimat bahasa gaul maupun terjemahannya, sehingga konteks penggunaan kata dapat dipertimbangkan dengan lebih baik. Tidak seperti dataset pertama, yang diterjemahkan sebagian besar secara otomatis menggunakan program dengan kamus bahasa gaul, dataset manual ini merujuk pada contoh penggunaan dari sumber berita dan media sosial, sehingga hasil terjemahannya lebih kontekstual.

Terjemahan otomatis pada dataset pertama memiliki keterbatasan, di mana kata atau frasa cenderung diterjemahkan secara tetap. Misalnya, kata "fomo" selalu diterjemahkan menjadi "takut ketinggalan zaman," tanpa memperhitungkan variasi makna yang mungkin berbeda sesuai konteks. Pendekatan ini mengurangi fleksibilitas model dalam mengenali makna yang lebih dinamis. Sebaliknya, terjemahan manual pada dataset kedua mampu memberikan variasi makna yang sesuai dengan konteks, sehingga dapat meningkatkan kualitas terjemahan model secara keseluruhan.

## 3. ANALISIS DAN PEMBAHASAN

*Fine-tuning* dilakukan pada model LLaMA-2-7b-Chat menggunakan teknik LoRA dan QLoRA untuk mengoptimalkan performa model dalam penerjemahan bahasa gaul ke bahasa formal. Proses *fine-tuning* memanfaatkan parameter yang dapat dilihat pada Tabel 2.

**Tabel 2. Konfigurasi Parameter QLoRA**

| **Parameter QLoRA** | | |
|---|---|---|
| **Parameter** | **Nilai** | **Penjelasan** |
| lora_r | 64 | Dimensi perhatian (*attention*) untuk LoRA. |
| lora_alpha | 16 | Parameter alpha untuk skala LoRA. |
| lora_dropout | 0.1 | Probabilitas dropout untuk lapisan LoRA. |

Parameter QLoRA digunakan untuk mengatur efisiensi *fine-tuning* model besar dengan fokus pada pengurangan kebutuhan sumber daya komputasi. Nilai lora_r menentukan dimensi perhatian, sedangkan lora_alpha berfungsi untuk mengontrol skala peningkatan parameter LoRA. Dengan mengatur lora_dropout, potensi *overfitting* pada lapisan tambahan dapat diminimalkan, sehingga meningkatkan generalisasi model. Selanjutnya adalah konfigurasi parameter *bitsandbytes* yang dapat dilihat pada Tabel 3.

**Tabel 3. Konfigurasi Parameter *bitsandbytes***

| **Parameter *bitsandbytes*** | | |
|---|---|---|
| **Parameter** | **Nilai** | **Penjelasan** |
| use_4bit | True | Mengaktifkan pemuatan model dengan presisi 4-bit. |
| bnb_4bit_compute_dtype | "float16" | Jenis komputasi untuk model 4-bit. |
| bnb_4bit_quant_type | "nf4" | Jenis kuantisasi yang digunakan (NF4). |
| use_nested_quant | False | Mengaktifkan kuantisasi bersarang untuk model 4-bit. |

Parameter bitsandbytes memungkinkan pemrosesan model besar dengan menggunakan kuantisasi presisi rendah, seperti 4-bit. Aktivasi use_4bit mengurangi konsumsi memori model. bnb_4bit_compute_dtype dan bnb_4bit_quant_type menentukan presisi komputasi dan jenis kuantisasi untuk efisiensi yang lebih tinggi. Namun, opsi use_nested_quant dinonaktifkan untuk menghindari kompleksitas tambahan yang dapat memengaruhi performa pelatihan. Selanjutnya konfigurasi parameter *TrainingArguments* pada Tabel 4.

**Tabel 4. Konfigurasi Parameter *TrainingArguments***

| **Parameter *TrainingArguments*** | | |
|---|---|---|
| **Parameter** | **Nilai** | **Penjelasan** |
| output_dir | "./results" | Direktori keluaran untuk hasil prediksi dan checkpoint model. |
| num_train_epochs | 4 | Jumlah epoch pelatihan. |
| FP16 | False | Mengaktifkan pelatihan dengan presisi float 16-bit. |
| bf16 | False | Mengaktifkan pelatihan dengan presisi bfloat16 (True jika menggunakan A100). |
| per_device_train_batch_size | 4 | Ukuran batch untuk pelatihan per GPU |
| per_device_eval_batch_size | 4 | Ukuran batch size untuk evaluasi per GPU |
| gradient_accumulation_steps | 1 | Jumlah langkah pembaruan untuk akumulasi gradien. |
| gradient_checkpointing | True | Mengaktifkan checkpointing gradien untuk efisien memori |
| max_grad_norm | 0.3 | Batas normalisasi gradien maksimum. |
| learning_rate | 2e-4 | Tingkat pembelajaran awal untuk optimizer. |
| weight_decay | 0.001 | Penurunan bobot untuk semua lapisan kecuali bias/LayerNorm. |
| optim | "paged_adamw_32bit" | Optimizer yang digunakan. |
| lr_scheduler_type | "cosine" | Jadwal tingkat pembelajaran |
| max_steps | -1 | Jumlah langkah pelatihan maksimum (*override epoch*). |
| warmup_ratio | 0.03 | Rasio langkah untuk warmup linier tingkat pembelajaran. |
| group_by_length | True | Mengelompokkan urutan ke dalam batch dengan panjang yang sama untuk efisiensi. |
| save_steps | 0 | Langkah pembaruan untuk menyimpan checkpoint. |
| logging_steps | 25 | Langkah pembaruan untuk mencatat log. |

Parameter *TrainingArguments* mengatur proses pelatihan model, termasuk jumlah epoch, ukuran batch, dan tingkat pembelajaran.

gradient_accumulation_steps dan gradient_checkpointing memungkinkan pelatihan pada perangkat dengan memori terbatas dengan mengurangi kebutuhan memori pada setiap langkah. Parameter lr_scheduler_type dan warmup_ratio memastikan pengaturan tingkat pembelajaran yang optimal selama pelatihan. Parameter ini dirancang untuk efisiensi dan stabilitas pelatihan, terutama pada model besar. Terakhir adalah konfigurasi parameter *Supervised Fine-Tuning* yang dapat dilihat pada Tabel 5.

**Tabel 5. Konfigurasi Parameter *Supervised Fine-Tuning***

| **Parameter *Supervised Fine-Tuning*** | | |
|---|---|---|
| **Parameter** | **Nilai** | **Penjelasan** |
| max_seq_length | None | Panjang maksimum urutan yang digunakan. |
| packing | False | Mengemas beberapa contoh pendek dalam urutan input yang sama. |
| device_map | {"": 0} | Memuat seluruh model pada GPU 0. |

Parameter SFT (*Supervised Fine-Tuning*) membantu mengatur batasan urutan input selama pelatihan, seperti max_seq_length untuk panjang maksimum input. Dengan packing dinonaktifkan, setiap contoh diproses secara individu, yang meminimalkan kompleksitas. device_map memastikan model sepenuhnya dimuat ke GPU, memungkinkan pengolahan yang terpusat dan efisien selama pelatihan.

Pada penelitian ini, beberapa parameter akan diuji untuk menentukan hasil terbaik dari *fine-tuning* model LLaMA-2-7b-chat. Parameter yang akan diuji adalah nilai dari num_train_epoch, learning_rate, dan bnb_4bit_compute_dtype. Dilakukan 4 eksperimen dengan masing-masing nilai parameter yang dapat dilihat pada Tabel 6.

**Tabel 6. Eksperimen Nilai Parameter**

| **Eksperimen ke-** | **Nilai Parameter** | | |
|---|---|---|---|
| | **bnb_4bit_compute_dtype** | **num_train_epoch** | **learning_rate** |
| 1 | "float16" | 1 | 2e-4 |
| 2 | "bfloat16" | 2 | 2e-5 |
| 3 | "float16" | 3 | 2e-4 |
| 4 | "bfloat16" | 4 | 1e-5 |

Eksperimen ke-1 menghasilkan data metrik berupa *training loss* yang tercatat pada setiap langkah. Hasil perhitungan *training loss* pada eksperimen ke-1 dapat dilihat pada Tabel 7.

**Tabel 7. *Training Loss* Eksperimen ke-1**

| **Step** | ***Training Loss*** |
|---|---|
| 25 | 3.9462 |
| 50 | 2.3222 |
| 75 | 2.1144 |
| 100 | 1.913 |
| 125 | 1.8257 |
| 150 | 2.0187 |

Berdasarkan hasil pelatihan, metrik *loss* menunjukkan penurunan bertahap selama proses training. Proses *fine-tuning* berlangsung selama 3 menit 57 detik, dengan pemakaian VRAM GPU sebesar 5,6 GB. Pada step awal, nilai *loss* tercatat sebesar 3.9462. Nilai ini terus menurun hingga mencapai 1.8257 pada step 125, sebelum mengalami sedikit kenaikan menjadi 2.0187 pada step akhir. Rata-rata *loss* keseluruhan (*train_loss*) adalah 2.3395, menunjukkan bahwa model berhasil mempelajari pola dasar dari dataset dalam jumlah iterasi yang terbatas. Penurunan *learning_rate* bertahap sesuai mekanisme *scheduler* menunjukan bahwa proses optimasi berjalan dengan baik dan sesuai dengan desain. Eksperimen ke-1 ini memberikan gambaran awal tentang bagaimana model merespons parameter yang digunakan.

Selanjutnya, eksperimen ke-2 menghasilkan data metrik berupa *training loss* yang tercatat pada setiap langkah. Hasil perhitungan *training loss* pada eksperimen ke-2 dapat dilihat pada Tabel 8.

**Tabel 8. *Training Loss* Eksperimen ke-2**

| Step | *Training Loss* |
|---|---|
| 25 | 5.5158 |
| 50 | 5.6523 |
| 75 | 5.0705 |
| 100 | 3.8994 |
| 125 | 3.6849 |
| 150 | 3.5197 |
| 175 | 2.7925 |
| 200 | 2.8059 |
| 225 | 2.6713 |
| 250 | 2.5275 |
| 275 | 2.5813 |
| 300 | 2.5876 |

Pada eksperimen ke-2, Proses *fine-tuning* berlangsung selama 19 menit 32 detik, dengan pemakaian VRAM GPU sebesar 5,7 GB. Hasil pelatihan menunjukkan bahwa nilai *training loss* menurun secara bertahap selama proses fine-tuning. Pada iterasi awal (Step 25), nilai *loss* berada di angka 5.5158, kemudian turun secara konsisten hingga mencapai nilai terendah 2.5275 pada Step 250. Namun, terdapat sedikit fluktuasi pada iterasi berikutnya, dengan *loss* meningkat tipis menjadi 2.5876 pada Step 300. Hasil ini menunjukkan bahwa model berhasil mempelajari pola dari dataset dengan lebih baik dibandingkan eksperimen pertama, meskipun peningkatan *loss* pada akhir pelatihan dapat mengindikasikan adanya kebutuhan untuk menyesuaikan parameter lebih lanjut.

Penggunaan *learning rate* yang lebih kecil (2e-5) dibandingkan eksperimen pertama memungkinkan model untuk melakukan pembelajaran secara lebih stabil, tanpa perubahan parameter yang terlalu drastis pada setiap iterasi. Sementara itu, peningkatan jumlah *epoch* dari 1 menjadi 2 memberikan model kesempatan untuk mempelajari pola dataset dengan lebih mendalam. Penurunan *loss* yang signifikan pada beberapa iterasi, terutama dari Step 100 (3.8994) ke Step 175 (2.7925), menunjukkan bahwa model mulai menangkap korelasi dalam dataset dengan baik. Namun, fluktuasi kecil pada nilai *loss* setelah Step 250 dapat disebabkan oleh berbagai faktor, seperti ukuran batch atau variasi dalam dataset.

Durasi pelatihan yang lebih lama dan penggunaan VRAM yang lebih besar pada eksperimen kedua ini sejalan dengan peningkatan jumlah *epoch*. Hal ini mengindikasikan adanya peningkatan kompleksitas pemrosesan model, yang seharusnya menghasilkan output yang lebih berkualitas dibandingkan eksperimen pertama. Namun, untuk menilai efektivitas keseluruhan, hasil keluaran model pada eksperimen kedua perlu dianalisis dan dibandingkan dengan eksperimen lainnya.

Eksperimen ke-3 menghasilkan data metrik berupa *training loss* yang tercatat pada setiap langkah. Hasil perhitungan *training loss* pada eksperimen ke-3 dapat dilihat pada Tabel 9.

**Tabel 9. *Training Loss Eksperimen* Ke-3**

| Step | *Training Loss* |
|---|---|
| 25 | 4.4155 |
| 50 | 2.4879 |
| 75 | 2.1227 |
| 100 | 1.8751 |
| 125 | 1.7446 |
| 150 | 1.8910 |
| 175 | 1.4710 |
| 200 | 1.4215 |
| 225 | 1.4359 |
| 250 | 1.4001 |
| 275 | 1.4086 |
| 300 | 1.3617 |
| 325 | 1.3228 |
| 350 | 1.2235 |
| 375 | 1.1819 |
| 400 | 1.1353 |
| 425 | 1.1929 |
| 450 | 1.1307 |

Pada eksperimen ke-3, dilakukan *fine-tuning* berlangsung selama 7 menit 44 detik, dengan penggunaan VRAP GPU 5,6 GB. Hasil pelatihan menunjukkan bahwa nilai *training loss* terus menurun secara konsisten seiring bertambahnya jumlah langkah. Pada Step 25, nilai *loss* berada di angka 4.4155, kemudian turun menjadi 2.4879 di Step 50, dan terus menurun hingga mencapai nilai terendah 1.1307 pada Step 450. Penurunan ini mencerminkan bahwa model berhasil mempelajari pola dataset dengan lebih baik dibandingkan eksperimen sebelumnya. Stabilitas dalam penurunan *loss*, khususnya pada *epoch* akhir, mengindikasikan bahwa model lebih efektif dalam melakukan generalisasi terhadap data.

Selanjutnya, eksperimen ke-4 menghasilkan data metrik berupa *training loss* yang tercatat pada setiap langkah. Hasil perhitungan *training loss* pada eksperimen ke-4 dapat dilihat pada Tabel 10.

**Tabel 10. *Training Loss* Eksperimen ke-4**

| Step | *Training Loss* |
|---|---|
| 25 | 5.5925 |
| 50 | 6.0769 |
| 75 | 6.0420 |
| 100 | 4.9599 |

| | |
|---|---|
| 125 | 5.0448 |
| 150 | 4.7894 |
| 175 | 3.7308 |
| 200 | 3.9058 |
| 225 | 3.6275 |
| 250 | 3.3070 |
| 275 | 3.3412 |
| 300 | 3.1090 |
| 325 | 2.8831 |
| 350 | 2.8335 |
| 375 | 2.6272 |
| 400 | 2.5094 |
| 425 | 2.7413 |
| 450 | 2.4717 |
| 475 | 2.5248 |
| 500 | 2.5982 |
| 525 | 2.3867 |
| 550 | 2.4708 |
| 575 | 2.5852 |
| 600 | 2.3806 |

*Fine-tuning* pada eksperimen ke-4 berlangsung selama 41 menit 57 detik, dengan penggunaan VRAM GPU 5,7 GB. Hasil pelatihan menunjukkan tren penurunan *training loss* secara bertahap seiring bertambahnya jumlah langkah, meskipun terdapat beberapa fluktuasi. Pada Step 25, *training loss* tercatat sebesar 5.5925, kemudian menurun hingga 2.3806 pada Step 600. Penurunan ini mengindikasikan bahwa model mampu mempelajari pola dataset dengan cukup baik, meskipun prosesnya lebih lambat dibandingkan eksperimen sebelumnya karena *learning rate* yang lebih kecil.

**Tabel 11. Hasil Keluaran Model**

| **Eksperimen ke-** | **Hasil Keluaran Teks** | | |
|---|---|---|---|
| | **Input** | **Output** | **Bleu Score** |
| 1 | shees mobil itu sigma banget | sekali mobil itu sangat unik banget<br><br>everybody loves this car so much 😍 | 0.0 |
| 2 | | kok kita tau, sih mobil itu sigma banget. everybody loves it! 😍 | 0.0 |
| 3 | | wow mobil itu keren sekali banget kamu penggemar mobil keren kamu mobil itu sangat unik banget kamu mobil itu sangat keren sekali banget kamu mobil itu sangat bagus banget kamu mobil itu sangat baik sekali banget kamu mobil itu sangat indah banget kamu mobil itu sangat impresif banget kamu mobil itu sangat keren sekali banget kamu mobil itu sangat baik sekali banget kamu mobil itu sangat indah banget kamu mobil itu sangat impresif banget kamu mobil | 0.0369 |
| 4 | | keren mobil itu sigma banget, tapi saya lebih suka mobil yang lebih sederhana dan mudah digunakan. everybody has their own preferences, right? 😊 | 0.0 |

Berdasarkan hasil eksperimen 1 hingga 4, dapat dianalisis bahwa fine-tuning model LLaMA-2 7b-chat dengan berbagai kombinasi parameter memberikan dampak yang signifikan terhadap kualitas keluaran teks. Setiap hasil dari eksperimen, diuji dengan inputan yang sama. Hasil dapat dilihat pada Tabel 11.

## 4. KESIMPULAN

Dari keempat eksperimen *fine-tuning* model LLaMA-2-7b chat yang dilakukan, ditemukan bahwa kombinasi parameter *num_train_epochs*, *learning rate*, dan presisi aritmatika secara signifikan memengaruhi hasil pelatihan. Penurunan nilai *training loss* terlihat pada semua eksperimen, dengan eksperimen ketiga menghasilkan *loss* akhir paling rendah (1.13) serta proses pembelajaran yang lebih stabil. Durasi pelatihan meningkat secara signifikan seiring dengan bertambahnya jumlah *epoch*, dari 3 menit 57 detik pada eksperimen pertama hingga 41 menit 57 detik pada eksperimen keempat. Namun, eksperimen keempat menunjukkan bahwa penggunaan *learning rate* yang terlalu kecil memperlambat konvergensi tanpa memberikan peningkatan kualitas keluaran yang signifikan. Meskipun nilai *training loss* menurun, kualitas teks yang dihasilkan model masih menunjukkan keterbatasan, seperti pengulangan frasa yang berlebihan dan koherensi kalimat yang kurang baik. Skor BLEU pada eksperimen pertama dengan skor 0.0, eksperimen kedua dengan skor 0.0, eksperimen ketiga dengan skor 0.0369, dan ekperimen keempat dengan skor 0.0 menegaskan bahwa model belum mampu menghasilkan terjemahan bahasa gaul yang optimal. Eksperimen ketiga memberikan kombinasi parameter terbaik dalam hal efisiensi waktu dan kualitas pelatihan, sehingga dapat dijadikan *baseline* untuk eksperimen lanjutan. Optimasi pada proses decoding serta perbaikan dataset menjadi langkah penting berikutnya untuk meningkatkan kemampuan model dalam memahami dan menerjemahkan bahasa gaul dengan lebih baik. Serta evaluasi seperti kuesioner dengan mengevaluasi secara langsung pengguna bahasa gaul perlu dipertimbangkan agar dapat memperoleh masukan dan penilaian secara langsung dengan pengguna.

## UCAPAN TERIMA KASIH

Terima kasih saya sampaikan kepada semua pihak yang telah memberikan dukungan, bantuan, dan arahan selama proses penelitian ini. Kepada dosen pembimbing, yang selalu memberikan bimbingan yang berharga dan memberikan insight yang mendalam dalam setiap tahap pengerjaan penelitian ini. Saya juga mengucapkan terima kasih kepada teman-teman yang telah memberikan masukan dan motivasi. Semoga apa yang telah saya pelajari dan hasilkan dapat memberikan kontribusi yang bermanfaat, baik di dunia akademik maupun di dunia praktis.

## DAFTAR RUJUKAN

Church, K. W. (2021). Emerging trends: A gentle introduction to fine-tuning. *Natural Language Engineering, 27*, 763-778.

Dettmers, T. a. (2024). Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems, 36*.

Dewi, A. C. (2023). Penggunaan Bahasa Gaul di Kalangan Remaja. *Nusantara Journal of Multidisciplinary Science, 1*, 1032-1043.

Ding, N. a.-M. (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence, 5*, 220-235.

Dutta, S., & Klakow, D. (2019). Evaluating a neural multi-turn chatbot using BLEU score. *Dutta, Sourav and Klakow, Dietrich*, 1-12.

Ghassemiazghandi, M. (2024). An Evaluation of ChatGPT's Translation Accuracy Using BLEU Score. *Theory and Practice in Language Studies*, 985-994.

Han, Z. a. (2024). Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Jayaseelan, N. (2023). LLaMA 2: The New Open Source Language Model. *Journal of Machine Learning Research, 24*, 1-15.

Li, Y. a. (2023). Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*.

Li, Z. a.-l. (2023). Label supervised llama finetuning. *arXiv preprint arXiv:2310.01208*.

Mathur, B. N. (2020). Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. *arXiv preprint arXiv:2006.06264*.

Minaee, S. a. (2024). Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Nagel, M. a. (2021). A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*.

Roumeliotis, K. I. (2023). Llama 2: Early Adopters' Utilization of Meta's New Open-Source Pretrained Model.

Satriani, A. D. (2023). Dampak dan transformasi perkembangan bahasa gaul dalam bahasa Indonesia modern. *Jurnal Pengabdian West Science, 2*, 421-426.

Sun, Z. a. (2021). A computational framework for slang generation. *Transactions of the Association for Computational Linguistics, 9*, 462-478.

Touvron, H. a. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Zhang, B. a. (2019). Root mean square layer normalization. *Advances in Neural Information Processing Systems, 32*.