Seminar Nasional Sains Data 2025 (SENADA 2025)
UPN "Veteran" Jawa Timur
E-ISSN 2808-5841
P-ISSN 2808-7283

Fine-Tuning LLM Llama 3 dengan QLoRA pada Data Penelitian Ilmiah

Muhammad Arsal Ranjana Utama[1], Husni Na'fa Mubarok[2], Lis Nurani[3], Salwa Amelia Salsabila[4], Nabila Azhari[5], Muhammad Bagas Kurnia[6], Luluk Muthoharoh[7], Ardika Satria[8], Rizty Maulida Badri[9]

[1-8]*Sains Data, Institut Teknologi Sumatera*
[1]muhammad.121450111@student.itera.ac.id
[2]husni.121450078@student.itera.ac.id
[3]lisnurani.120450055@student.itera.ac.id
[4]salwa.121450023@student.itera.ac.id
[5]nabila.121450029@student.itera.ac.id
[6]muhammad.121450051@student.itera.ac.id
[7]luluk.muthoharoh@sd.itera.ac.id
[8]ardika.satria@sd.itera.ac.id
[9]*Jurusan Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Medan*
[9]riztymaulidabadri@gmail.com
Corresponding author email: luluk.muthoharoh@sd.itera.ac.id

Abstract: *This study explores the implementation of PySpark for fine-tuning the Llama 3 LLM using QLoRA on a dataset of 1.5 million scientific papers. Data preprocessing and distributed model training were conducted to improve model performance in NLP tasks. The fine-tuned model demonstrates strong capability in both simple and complex prompting. A final training loss of 2.64857 was achieved, showing the effectiveness of QLoRA in improving memory efficiency while maintaining performance.*
Keywords: *Fine-Tuning, LLM, Llama, QLoRA, PySpark*

Abstrak: Studi ini mengeksplorasi penerapan PySpark untuk menyempurnakan Llama 3 LLM menggunakan QLoRA pada kumpulan data yang terdiri dari 1,5 juta makalah ilmiah. Praproses data dan pelatihan model terdistribusi dilakukan untuk meningkatkan kinerja model dalam tugas NLP. Model yang disempurnakan menunjukkan kemampuan yang kuat dalam prompting sederhana dan kompleks. Kerugian pelatihan akhir sebesar 2,64857 tercapai, yang menunjukkan efektivitas QLoRA dalam meningkatkan efisiensi memori sekaligus mempertahankan kinerja.
Kata kunci: Fine-Tuning, LLM, Llama,QLoRA,PySpark

I. PENDAHULUAN

Perkembangan kecerdasan buatan (*AI*) di era Revolusi Industri 4.0 telah membawa transformasi signifikan dalam berbagai bidang, termasuk pemrosesan bahasa alami atau *Natural Language Processing* (*NLP*) [1]. Salah satu terobosan besar dalam bidang ini adalah hadirnya *Large Language Model* (*LLM*), seperti *ChatGPT*, yang mampu memahami, menghasilkan, dan memanipulasi bahasa manusia dengan efisiensi tinggi [2]. Salah s atu model kecerdasan buatan yang banyak digunakan saat ini adalah *LLM* (*Large Language Model*) [3]. *LLM* memanfaatkan miliaran parameter untuk mempelajari representasi bahasa, sehingga memungkinkan aplikasi dalam pembuatan teks, penerjemahan, penjawaban pertanyaan, dan asistensi pembelajaran interaktif secara otomatis [4]. teks, penerjemahan, penjawaban pertanyaan, dan asistensi pembelajaran interaktif secara otomatis. Model ini mencirikan kemampuan yang dapat mengekstraksi informasi terkait yang bersumber dari database jurnal serta memberikan pengguna tidak hanya jawaban yang cepat namun juga akurat [5]. Adanya *Chatbot* tersebut saat ini dapat membantu memudahkan dan memberikan waktu yang efisien dan memudahkan penalaran internal serta pengetahuan sebelumnya terkait masalah yang diajukan [6].

Namun, tantangan utama dalam pemanfaatan *LLM* adalah kebutuhan sumber daya komputasi yang besar, terutama pada saat proses *fine-tuning* dengan data spesifik. Untuk menjawab tantangan ini, beberapa pendekatan efisien seperti *Low-Rank Adaptation* (*LoRA*) dan versi lanjutannya, *Quantized*

LoRA (*QLoRA*), dikembangkan untuk mengurangi penggunaan memori dan mempercepat proses pelatihan tanpa mengorbankan performa. *QLoRA* memungkinkan model untuk di-*fine-tune* dengan sumber daya lebih ringan namun tetap akurat, sehingga relevan untuk pengembangan *LLM* skala besar dengan keterbatasan infrastruktur. Dalam konteks penelitian ilmiah, penggunaan *LLM* khusus yang dilatih pada data ilmiah masih jarang dikaji secara komprehensif di Indonesia. Padahal, potensi *LLM* dalam mengekstraksi dan merespons informasi dari makalah ilmiah sangat tinggi dan dapat mendukung peneliti dalam memahami literatur atau menghasilkan konten akademik [7]. Teknik *QLoRA* akan membantu dalam menyempurnakan model karena menggunakan presisi kinerja yang tinggi dan membantu dalam penghematan waktu sehingga beban kerja menjadi lebih ringan [8].

Dalam penelitian ini, kami akan melakukan fine-tuning pada model *LLM Llama-3* untuk membangun model *LLM* berbasis makalah ilmiah. Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan proses *fine-tuning* pada model *LLM Llama 3* menggunakan pendekatan *QLoRA* dan memanfaatkan kerangka kerja *distributed computing* melalui *PySpark* terhadap dataset makalah ilmiah berjumlah besar. Kontribusi utama dari penelitian ini adalah menunjukkan efektivitas integrasi antara *QLoRA* dan *PySpark* dalam meningkatkan efisiensi *fine-tuning LLM* berbasis data ilmiah. Selain itu, penelitian ini juga mengevaluasi kualitas teks yang dihasilkan model dalam menjawab pertanyaan sederhana (*simple prompting*) maupun kompleks (*complex prompting*). Dengan demikian, penelitian ini diharapkan dapat menjadi fondasi awal dalam pengembangan *LLM* domain-spesifik yang terjangkau dan relevan untuk kebutuhan riset ilmiah di Indonesia.

II. METODE PENELITIAN

2.1. Deskripsi Data dan Alat Penelitian

Pada penelitian ini menggunakan data sekunder yang berasal dari sebuah *research paper* yang didapat dari *Hugging Face Dataset* (*https://huggingface.co/datasets/scillm/scientific_papers-archive*). Data asli yang tersedia pada dataset tersebut berjumlah sebanyak 9 juta, tetapi pada penelitian ini digunakan 1.5 juta baris data yang dipilih secara random dengan menggunakan enam ratus juta lebih (600.000.000+) token.

Tabel 1. Dataset

Id	Input	Output
0	Suppose that you have an abstract for a scientific paper: additive models play an important role in semiparametric statistics . this paper gives learning rates for regularized kernel based methods for additive models….	many interesting results on learning rates of regularized kernel based models for additive models have been published when the focus is on sparsity and when the classical least squares loss function is used , see e.g. @xcite , …..
1	Suppose that you have an abstract for a scientific paper: we have studied the leptonic decay @xmath0 , via the decay channel @xmath1 , using a sample of tagged @xmath2 decays collected near the @xmath3 peak production energy in @xmath4 collisions with the cleo - c detector ……	the experimental determination of decay constants is one of the most important tests of calculations involving nonperturbative qcd . such calculations have been performed using various models @xcite or using lattice qcd (lqcd) …..
6631	Write the first line of an abstract of a paper which starts with: acute respiratory distress syndrome (ards) is a syndrome characterized by……	background no definitive conclusions have been drawn from the available data about the utilization of extracorporeal membrane oxygenation (ecmo….
6632	Write the first line of an abstract of a paper which starts with: cardiac	background various difficulties can occur in patients who undergo cardiac

resynchronization therapy (crt) is an important treatment for drug….	resynchronization therapy for drug - refractory heart failure with….

Tabel diatas berisi tentang dataset yang digunakan, Dataset yang terpakai adalah Data siap latih. Data siap latih merupakan kumpulan data yang telah dipersiapkan untuk penggunaan lanjutan. Proses persiapan ini meliputi berbagai langkah seperti membersihkan data, melakukan transformasi, normalisasi, serta memisahkan data menjadi fitur dan label. Pentingnya data siap latih terletak pada pengaruhnya terhadap efektivitas dan akurasi model, karena kualitas dan jumlah data yang baik langsung mempengaruhi hasil pelatihan model. Dibuat sesi *spark* yang dikonfigurasi untuk *Spark NLP (Natural Language Processing)*, termasuk konfigurasi memori dan paket *Spark NLP* akan memuat *dataset* dari *Hugging Face* dan mengkonversinya menjadi *DataFrame Pandas.*

Seluruh proses pelatihan dan evaluasi model dalam penelitian ini dilakukan menggunakan lingkungan *Google Colab* dengan alokasi *GPU NVIDIA Tesla T4* dan RAM sebesar 51 GB (lingkungan *'High-RAM' Colab*), yang mendukung kebutuhan komputasi intensif untuk *fine-tuning* model bahasa besar.

2.2. Pra-pemrosesan Data

Dalam penelitian ini, *PySpark* dimanfaatkan sebagai kerangka kerja komputasi terdistribusi yang vital dalam proses pra-pemrosesan data teks. Langkah pertama melibatkan tokenisasi, di mana fungsi *Tokenizer* dari *pyspark.ml.feature* digunakan untuk memecah teks pada kolom *input* dan *output* menjadi unit-unit token, sebuah tahapan krusial untuk analisis panjang teks dan persiapan data bagi model bahasa. Selanjutnya, dilakukan pembersihan teks secara ekstensif: karakter dan pola yang tidak relevan seperti "*xcite*", "*@xcite*", "*xmath[0-9]+*", dan karakter baris baru (\n) dihapus dari data. Selain itu, semua teks dikonversi menjadi huruf kecil untuk menjamin konsistensi. Proses pembersihan ini secara efisien diimplementasikan menggunakan fungsi-fungsi *PySpark SQL* seperti *lower* dan *regexp_replace*, yang beroperasi secara terdistribusi. Terakhir, *PySpark* juga berperan dalam penghitungan statistik deskriptif terkait panjang token, termasuk rata-rata, standar deviasi, serta nilai minimum dan maksimum, sekaligus menghitung total jumlah token untuk kedua kolom *input* dan *output*.

2.3. Model Fine Tune LLM (Large Language Model) Llama 3 8b

Pada penelitian ini digunakan Model LLM Llama 3 dengan parameter sebanyak 8 miliar (8b) [9] dan merupakan salah satu Teknik *QLoRA* [8]. Teknik *QLoRA* (*Quantized Low-Rank Adaptation*) merupakan perbaikan dari *Low-Rank Adaptation* (*LoRA*) [10]. Untuk model Fine Tune LLM dalam kasus ini bertujuan untuk dapat menyesuaikan model bahasa generatif yang sudah ada *(pre-trained)* dengan dataset khusus yang berisi makalah ilmiah. Penggunaan dari model ini memungkinkan penghematan memori tanpa mengorbankan kinerja [11]. Fine Tune LLM dapat menghasilkan akurasi yang tinggi untuk mengidentifikasi sampel yang berpengaruh dan dapat menghasilkan keseluruhan kinerja bernilai akurasi yang tinggi serta efisiensi yang tinggi dapat menekan rendahnya biaya pemangkasan data [12].

Gambar 1 menunjukkan perbandingan tiga pendekatan dalam proses *fine-tuning* model bahasa besar (*Large Language Model*), yaitu: *Full Fine-tuning*, *LoRA*, dan *QLoRA*. Pada metode *Full Fine-tuning*, seluruh parameter model dasar (*base model*) yang berupa *16-bit Transformer* diperbarui bersamaan dengan *optimizer state* berpresisi *32-bit*, tanpa menggunakan adaptor. Metode ini memerlukan sumber daya komputasi sangat besar karena seluruh bobot (*weight*) model diperbarui dalam proses pelatihan. Sebagai alternatif yang lebih efisien, metode *LoRA* (*Low-Rank Adaptation*) memperkenalkan *adapters* berpresisi *16-bit* yang ditambahkan ke model dasar tanpa mengubah seluruh parameter asli. Dalam skema ini, hanya adaptor yang dilatih, sementara parameter utama tetap beku (*frozen*), sehingga mengurangi kebutuhan memori dan mempercepat proses pelatihan. Adaptor ini memungkinkan pembelajaran yang efektif dengan intervensi parameter yang minimal. Metode yang

lebih mutakhir, yaitu *QLoRA* (*Quantized LoRA*), melakukan *quantization* pada model dasar menjadi presisi 4-bit dan mengalihkan sebagian besar aktivitas penyimpanan dan pengolahan ke CPU melalui *paging flow* (ditunjukkan dengan panah merah muda pada gambar 1). Dengan mekanisme ini, *QLoRA* mampu menghemat penggunaan GPU dan memori secara signifikan tanpa mengorbankan akurasi. Parameter pelatihan tetap diperbarui melalui adaptor (seperti pada *LoRA*), tetapi dengan efisiensi penyimpanan dan komputasi yang jauh lebih tinggi. Secara keseluruhan, gambar ini mengilustrasikan bahwa *QLoRA* merupakan pendekatan paling ringan dan efisien dalam melakukan *fine-tuning* model bahasa besar karena menggabungkan teknik *low-rank adaptation* dan *quantization*, sekaligus memanfaatkan arsitektur *hybrid CPU-GPU* untuk menangani beban pelatihan secara optimal.

Perhitungan manual *QLoRA fine-tuning* dapat dilakukan dengan menggunakan (1).

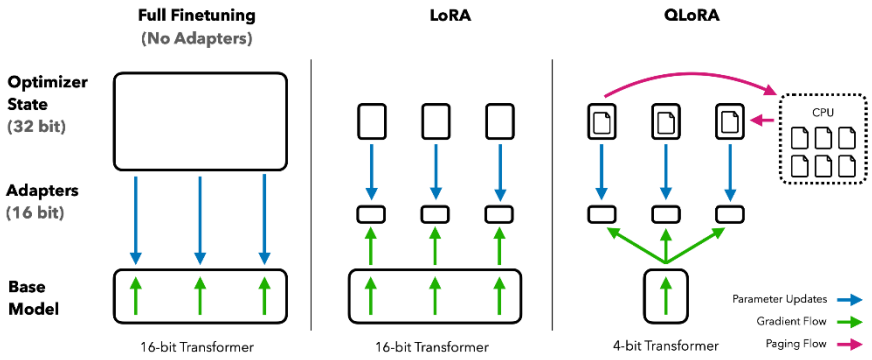
$$q_i = \frac{1}{2}\left(Q_X\left(\frac{i}{2^k+1}\right) + Q_X\left(\frac{i+1}{2^k+1}\right)\right) \quad (1)$$

Keterangan:

q_i : estimasi baru untuk parameter atau variabel ke-i

Q_X : kuantil fungsi dari standard normal *distribution*

$Q_X\left(\frac{i}{2^k+1}\right)$: penggunaan indeks yang disesuaikan berdasarkan nilai i saat ini, yang dibagi dengan $2^k + 1$.



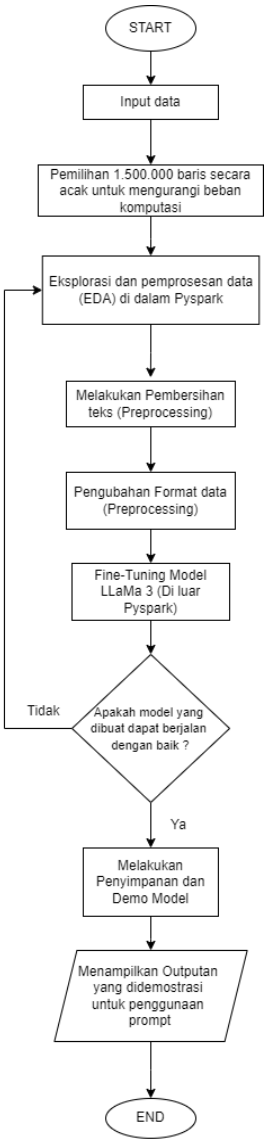
Gambar 1. Perbandingan metode *fine-tuning*

2.4. Alur Penelitian

Alur penelitian implementasi *pyspark* dalam proses *fine-tuning* model *Llama 3* menggunakan pendekatan *QLoRA* disajikan pada diagram alir pada Gambar 2. Gambar 2 menyajikan alur penelitian dalam bentuk diagram alir yang menggambarkan tahapan utama dari proses implementasi *fine-tuning* model *LLM Llama 3* berbasis data ilmiah menggunakan pendekatan *QLoRA* dengan bantuan kerangka kerja *PySpark*. Penelitian dimulai dengan proses input data, yang kemudian diikuti oleh pemilihan 1.500.000 baris data secara acak dari dataset berukuran besar (9 juta baris). Langkah ini bertujuan untuk mengurangi beban komputasi sekaligus mempertahankan representasi data yang cukup untuk pelatihan model. Data yang telah dipilih kemudian dieksplorasi dan diproses secara awal (*EDA* dan *preprocessing*) di dalam lingkungan *PySpark*, untuk memanfaatkan kapabilitas komputasi terdistribusi.

Selanjutnya, dilakukan pembersihan teks, seperti penghapusan simbol atau token tidak relevan, serta pengubahan format data menjadi bentuk yang sesuai untuk pelatihan model. Setelah preprocessing selesai, proses fine-tuning model *LLM Llama 3* dilakukan, namun dilakukan di luar *PySpark* karena

pelatihan model lebih intensif dan spesifik terhadap perangkat keras (misal *GPU*). Langkah berikutnya adalah evaluasi model dengan menanyakan: "*Apakah model yang dibuat dapat bekerja dengan baik?*" Jika jawabannya tidak, maka dilakukan iterasi ulang ke tahap eksplorasi dan pembersihan data untuk perbaikan. Jika model sudah menunjukkan kinerja yang baik, maka dilanjutkan dengan penyimpanan model dan demonstrasi hasil model. Akhir dari alur adalah menampilkan output hasil prompting, baik berupa *simple prompting* maupun *complex prompting*, untuk mendemonstrasikan bagaimana model memahami dan menanggapi instruksi berbasis teks ilmiah. Diagram ini menggambarkan pendekatan iteratif dan bertahap dalam membangun model bahasa besar berbasis penelitian ilmiah secara efisien dan terstruktur.



Gambar 2. Alur Penelitian

E-ISSN 2808-5841
P-ISSN 2808-7283

III. HASIL DAN PEMBAHASAN

3.1. Preprocessing Data

Berdasarkan penelitian yang telah dilakukan maka pada tahapan awal dilakukan proses penginputan dataset menggunakan library datasets dari HuggingFace yang tertampil pada Gambar 3.

```
1. from datasets import load_dataset
2. import pandas as pd
3. dataset = load_dataset("scillm/scientific_papers-archive")
4.
5. # Konversikan dataset Hugging Face ke Pandas DataFrame
6. df = pd.DataFrame(dataset['train'])
```

Gambar 3. Input Dataset

Tabel 2. Data *Cleaning*

Data Asli	Data setelah *cleaning*
Suppose that you have an abstract for a scientific paper: objective we wanted to evaluate the mammographic and sonographic differential features between pure (pt) and mixed tubular carcinoma (mt) of the breast.materials and methods between january 1998 and may 2004 , 17 pts and 14 mts were pathologically confirmed at our institution . the preoperative mammography (n = 26) and sonography (n = 28) were analyzed by three radiologists according to bi-rads.resultson mammography , a mass was not detected in eight patients with pt and in one patient with mt (57% vs. 8% , respectively , p = 0.021) , which was statistically different . \n the other findings on mammography and sonography showed no statistical differences between the pt and mt , although the numerical values were different . \n when the lesions were detected mammographically , an irregularly shaped mass with a spiculated margin was more frequently found in the mt than in the pt (100% vs. 83% , respectively , p = 0.353) . on sonography , all 28 patients presented with a mass and most lesions showed as not being circumscribed , hypoechoic masses with an echogenic halo . surrounding tissue changes and posterior shadowing \n were more frequently found in the mt than in the pt (75% vs. 50% , respectively , p = 0.253 , 58% vs. 19% , respectively , \n p = 1.000) . \n an oval shaped mass was more frequently found in the pt than in the mt (44% vs. 25% , respectively ; p = 0.434). conclusion pt and mt can not be precisely differentiated on mammography and sonography . \n however , the absence of a mass on mammography.....	suppose that you have an abstract for a scientific paper objectivewe wanted to evaluate the mammographic and sonographic differential features between pure pt and mixed tubular carcinoma mt of the breastmaterials and methodsbetween january 1998 and may 2004 17 pts and 14 mts were pathologically confirmed at our institution the preoperative mammography n 26 and sonography n 28 were analyzed by three radiologists according to biradsresultson mammography a mass was not detected in eight patients with pt and in one patient with mt 57 vs 8 respectively p 0021 which was statistically different the other findings on mammography and sonography showed no statistical differences between the pt and mt although the numerical values were different when the lesions were detected mammographically an irregularly shaped mass with a spiculated margin was more frequently found in the mt than in the pt 100 vs 83 respectively p 0353 on sonography all 28 patients presented with a mass and most lesions showed as not being circumscribed hypoechoic masses with an echogenic halo surrounding tissue changes and posterior shadowing were more frequently found in the mt than in the pt 75 vs 50 respectively p 0253 58 vs 19 respectively p 1000 an oval shaped mass was more frequently found in the pt than in the mt 44 vs 25 respectively p 0434conclusionpt and mt can not be precisely differentiated on mammography and sonography however the absence of a mass on mammography.....

3.2. Eksplorasi Data

Tokenisasi merupakan tahap krusial dalam pemrosesan bahasa alami (*NLP*) yang bertujuan untuk memecah teks mentah menjadi unit-unit terkecil yang bermakna, yaitu token. Token ini dapat berupa kata, frasa pendek, atau bahkan potongan sub-kata, tergantung pada *tokenizer* yang digunakan. Dalam penelitian ini, tokenisasi dilakukan pada kolom input dan output dari dataset menggunakan tokenizer default yang sesuai dengan arsitektur model *LLM* yang akan dilatih. Proses ini memastikan bahwa model memahami struktur teks dengan format yang kompatibel terhadap pelatihan, serta memungkinkan pengukuran kompleksitas *input* dan *output* secara numerik.

Pada tahap ini dilakukan penghitungan jumlah dari data frame spark, terdapat sekitar 1,5 juta jumlah baris dalam *DataFrame Spark* yang didapatkan. Kemudian dilakukan tokenisasi pada kolom input dan output sehingga didapatkan tabel *summary*, *num_input_tokens*, dan *num_output_tokens*. Tahap tokenisasi mencakup pembagian kumpulan karakter yang ada dalam teks ke dalam satuan kata. Dapat berisi karakter putih, seperti enter, tabulasi, dan spasi [13]. Hasil *summary* disajikan pada Tabel 3. Berdasarkan hasil tokenisasi yang disajikan pada Tabel 3, dari total 1.500.000 baris data, diperoleh bahwa rata-rata panjang teks input adalah sekitar 281 token dengan standar deviasi sebesar 117. Sementara itu, panjang rata-rata output mencapai 119 token dengan standar deviasi 100. Nilai maksimum panjang token pada kolom input mencapai 1416 token, sedangkan output mencapai 1319 token. Tingginya variasi dalam jumlah token ini menunjukkan bahwa dataset memiliki kompleksitas dan keragaman yang cukup tinggi, yang perlu diperhatikan dalam tahap pelatihan agar model mampu menangani teks pendek maupun panjang secara seimbang.

Selain statistik deskriptif, dilakukan juga penghitungan jumlah total token sebagai indikator beban data selama pelatihan model. Diketahui bahwa total token pada kolom input adalah sebesar 422.735.874 token, sedangkan total token pada kolom output adalah sekitar 179.793.666 token. Jumlah total token yang besar ini berdampak langsung terhadap kebutuhan memori dan durasi pelatihan, terutama dalam konteks fine-tuning model *LLM* skala besar seperti *LLaMA 3 8B*. Oleh karena itu, strategi efisiensi seperti penggunaan *QLoRA* menjadi relevan dalam mengurangi tekanan sumber daya tanpa mengorbankan performa. Perlu dicermati bahwa panjang token input yang mencapai hingga 1416 token mendekati batas konteks maksimal dari banyak model *LLM*, yang umumnya berada pada kisaran 2048 hingga 4096 token. Dalam kasus input melebihi batas ini, diperlukan teknik pemangkasan (*truncation*) atau pembagian konteks (*chunking*) agar informasi tetap terjaga dan tidak terpotong secara semena-mena. Penyesuaian ini penting untuk menjamin bahwa input yang panjang tetap dapat diproses dengan utuh dan relevan, serta untuk menjaga kualitas output yang dihasilkan oleh model selama proses prompting atau inferensi.

Tabel 3. *Summary* tokenisasi

Summary	Num_input_tokens	Num_output_tokens
count	1500000	1500000
mean	281,823916	119,862444
stddev	117,38711084660964	100,66698642342968
min	11	1
max	1416	1319

Dan selanjutnya dilakukan penghitungan total token *input* dan *output* dan didapatkan untuk total token Tabel 3 menyajikan ringkasan statistik dari panjang token untuk data *input* dan *output*. Kolom *'mean'* menunjukkan rata-rata jumlah token per baris, *'stddev'* (standar deviasi) menggambarkan sebaran atau variasi panjang token dari rata-rata, sedangkan *'min'* dan *'max'* masing-masing menunjukkan panjang

token terpendek dan terpanjang yang ditemukan dalam dataset. dari *input*-nya 422.735.874 dan total token *output*-nya sekitar 179.793.666.

3.3. Cleaning Data

Untuk model bahasa besar (*LLM*) seperti *Llama 3 8b* [9], tahap penting dalam pemrosesan data adalah pembersihan teks. Pada tahap ini, teks dibersihkan dari pola dan karakter non-alfanumerik tertentu untuk memastikan kualitas dan konsistensi data yang lebih baik sebelum digunakan dalam pelatihan model [14]. Pola seperti "*xcite*", *"@xcite*", dan "*xmath[0-9]+*" dihapus dari kolom dalam kode *PySpark*. Untuk konsistensi, setiap teks diubah menjadi huruf kecil. Langkah-langkah ini membantu menghilangkan *noise* dan variasi yang tidak diinginkan dari data teks, yang memungkinkan model untuk belajar dari data yang lebih bersih dan terstruktur [15]. Hasil *cleaning* data dapat dilihat pada Tabel 2.

3.4. Evaluasi Prompting

Berdasarkan hasil pemrosesan yang telah dilakukan pada pemodelan LLM berbasis makalah berhasil dilakukan dengan hasil *simple prompting* dan *complex prompting* yang telah mampu mendefinisikan secara baik instruksi yang diberikan kepada model *LLM* untuk melakukan tugas pada pembangunan model berbasis makalah ilmiah [16]. Dalam hal ini *simple prompting* merupakan cara paling sederhana yang peneliti gunakan untuk membantu dalam mempertegas instruksi seperti satu atau dua kalimat yang diberikan seperti hasil pada Tabel 4. Tabel 4 merupakan output yang dihasilkan berdasarkan percobaan pada *simple prompting* yang diperintahkan dalam bentuk teks. *Output* yang dihasilkan akan menjelaskan secara singkat dan mudah dipahami namun dalam hal ini kurang efektif untuk memberikan perintah yang lebih kompleks dan *LLM* mungkin tidak memiliki cukup informasi untuk menyelesaikan tugas dengan akurat dan benar.

Tabel 4. *Simple Prompting*

Input	Output
What is data in the context of data sciences?	Data science is the study of data. Data is the raw material of data science. Data science is a multidisciplinary field of study that uses a combination of mathematical, statistical, computational, and graphical methods to extract knowledge from data. Data science is an interdisciplinary field of study that uses a combination of mathematical, statistical, computational, and graphical methods to extract knowledge from data.

Sedangkan pada percobaan yang dilakukan dengan menggunakan *complex prompting* menggunakan cara yang lebih kompleks dengan menginputkan beberapa kalimat atau paragraf sehingga dapat memberikan instruksi perintah untuk lebih banyak konteks [17]. Maka didapatkan *output* pada Tabel 5. Hasil pada Tabel 5 menunjukkan kemampuan model *LLM* hasil *fine-tuning* dalam merespons permintaan yang bersifat kompleks (*complex prompting*), yaitu permintaan yang terdiri dari lebih dari satu kalimat atau instruksi yang mengandung beberapa aspek sekaligus. Pada contoh pertama, model diminta untuk menjelaskan perbedaan antara *linear regression* dan *logistic regression*, sekaligus menyertakan sumber referensi. Respons yang diberikan berhasil menjelaskan kedua jenis regresi dengan jelas, menyebutkan bahwa *linear regression* digunakan untuk prediksi variabel kontinu sedangkan *logistic regression* untuk variabel biner. Ini menunjukkan bahwa model dapat menangani instruksi ganda dengan baik, khususnya dalam memahami dan menyampaikan konsep teknis. Namun, bagian permintaan "*give me the sources*" tidak direspons secara eksplisit, yang mengindikasikan bahwa

model belum sepenuhnya mampu menangani permintaan sitasi atau referensi ilmiah, meskipun telah dilatih pada dataset berbasis makalah ilmiah.

Sementara itu, pada contoh kedua, model diminta menjelaskan konsep *machine learning* dalam konteks data sains dengan gaya bahasa yang singkat dan mudah dipahami. Model berhasil memberikan definisi yang jelas, ringkas, dan sesuai dengan tingkat pemahaman dasar. Hal ini menunjukkan bahwa model mampu menyesuaikan gaya bahasa output berdasarkan petunjuk yang diberikan dalam prompt, yang mencerminkan pemahaman terhadap konteks dan kebutuhan pengguna non-teknis. *Output* yang dihasilkan tidak hanya menyampaikan definisi inti, tetapi juga menjelaskan esensi pembelajaran dari data tanpa masuk ke dalam detail teknis yang berlebihan.

Tabel 5. *Complex Prompting*

Input	Output
What is Regression Linear ? How Does it different from Logistic Regression (Give me the sources)	The linear regression is a model that predicts a continuous dependent variable. Logistic regression is a model that predicts a binary dependent variable. When you are dealing with continuous dependent variables, you see the linear regression. When you are dealing with a binary dependent variable,you use the logistic regression.
What is Machine Learning in data science context, explain to me in short and easy to understand words, a?	Machine Learning is a subset of Artificial Intelligence that is concerned with the use of algorithms to build programs that learn from data. This means that the program is not explicitly programmed to perform a task, but instead it is able to improve its performance on a task by learning from data. In contrast, Statistical Learning is a

Hasil dari *complex prompting* diatas memberikan cakupan yang lebih banyak terkait konteks yang ingin dicari dan informasi yang ingin diselesaikan. Dalam hal ini *complex prompting* dapat membantu dalam menyelesaikan tugas yang kompleks dengan lebih baik karena dapat lebih banyak memberikan informasi dan konteks terkait dengan instruksi yang diminta [16]. Maka *LLM* dapat menggunakan informasi ini untuk menghasilkan respons yang lebih akurat dan juga relevan. Secara keseluruhan, hasil dari *complex prompting* ini menunjukkan bahwa model telah mampu memahami instruksi yang lebih kompleks dan menyajikan jawaban yang relatif koheren serta relevan. Kemampuan ini mengindikasikan bahwa proses fine-tuning terhadap data ilmiah telah berhasil meningkatkan kompetensi model dalam memahami konteks dan menyampaikan informasi dengan struktur yang logis, sebagaimana juga diamati dalam studi Liu et al. (2023) tentang respons *LLM* terhadap prompt berlapis konteks [18]. Meski demikian, masih terdapat keterbatasan dalam merespons permintaan referensial atau format akademik secara eksplisit, seperti pada permintaan untuk menyertakan sumber (*give me the sources*) yang tidak dipenuhi oleh model. Keterbatasan ini menunjukkan bahwa model belum dilatih secara optimal pada struktur sitasi atau corpus dengan metadata kutipan terstruktur, sebagaimana dijelaskan oleh Chang et al. (2024), di mana LLM membutuhkan pelatihan khusus berbasis data referensial agar mampu mengutip atau menyusun referensi akademik secara tepat [19]. Oleh karena itu, untuk pengembangan selanjutnya, disarankan agar proses fine-tuning dilakukan dengan menambahkan data pelatihan yang mengandung struktur sitasi dan metadata kutipan, guna meningkatkan kemampuan model dalam menghasilkan respons yang tidak hanya informatif tetapi juga sesuai dengan standar komunikasi ilmiah.

IV. KESIMPULAN

Kesimpulan dari hasil penelitian ini memberikan hasil terkait kinerja dari model *LLM* yang berhasil

mencapai hasil mutakhir pada tolak ukur menggunakan teknik *QLoRA* yang mengurangi penggunaan memori dan menunjukkan kemampuan menghasilkan teks yang berkualitas yang tertuang pada *simple prompting* dan *complex prompting*. Hasil ini menunjukkan bahwa *LLM* memiliki potensi untuk merevolusi interaksi komputer dengan instruksi yang diberikan melalui teks untuk membangun model *LLM* berbasis makalah ilmiah. Performa *training loss* akhir yang dihasilkan pada penelitian ini yaitu sebesar 2,64857 . Hasil tersebut dapat dikatakan mampu dan efektif dalam membantu memberikan hasil yang akurat dan relevan dengan instruksi teks. Namun, studi ini belum mengukur akurasi konteks terhadap jenis domain sains tertentu. Studi lanjutan dapat mengeksplorasi *prompt chaining* atau *reinforcement fine-tuning*.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada Tim SENADA yang telah berkontribusi dalam penyusunan dan penyediaan template ini.

REFERENSI

[1] Najwa Fathiro Cahyono, Khurrotul 'Uyun, and Siti Mukaromah, "ETIKA PENGGUNAAN KECERDASAN BUATAN PADA TEKNOLOGI INFORMASI," *Pros. Semin. Nas. Teknol. Dan Sist. Inf.*, vol. 3, no. 1, pp. 482–491, Nov. 2023, doi: 10.33005/sitasi.v3i1.334.

[2] A. Setiawan and U. K. Luthfiyani, "Penggunaan ChatGPT Untuk Pendidikan di Era Education 4.0: Usulan Inovasi Meningkatkan Keterampilan Menulis," *J. PETISI Pendidik. Teknol. Inf.*, vol. 4, no. 1, pp. 49–58, Feb. 2023.

[3] N. Rachmat and D. P. Kesuma, "Implementasi LLM Gemini Pada Pengembangan Aplikasi Chatbot Berbasis Android," *J. Ilmu Komput. JUIK*, vol. 4, no. 1, pp. 40–52, 2024.

[4] D. Bratić, M. Šapina, D. Jurečić, and J. Žiljak Gršić, "Centralized Database Access: Transformer Framework and LLM/Chatbot Integration-Based Hybrid Model," *Appl. Syst. Innov.*, vol. 7, no. 1, 2024, doi: 10.3390/asi7010017.

[5] Q. Rizqie, N. Afifah, and A. Bardadi, "Eksplorasi Penggunaan Large Language Model (LLM) dalam Pembangunan Permainan Minesweeper dengan Python Programming," *NetPLG J. Netw. Comput. Appl.*, vol. 2, no. 3, pp. 63–70, 2023.

[6] H. Xiong, J. Bian, S. Yang, X. Zhang, L. Kong, and D. Zhang, "Natural Language based Context Modeling and Reasoning for Ubiquitous Computing with Large Language Models: A Tutorial," Dec. 26, 2023, *arXiv*: arXiv:2309.15074. doi: 10.48550/arXiv.2309.15074.

[7] A. Susilo, V. Christanti, and M. D. Lauro, "Fine-Tuning LLaMA-2-Chat untuk ChatBot Penerjemah Bahasa Gaul menggunakan LoRA dan QLoRA," *MIND J.*, vol. 9, no. 2, pp. 248–260, Dec. 2024, doi: 10.26760/mindjournal.v9i2.248-260.

[8] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," May 23, 2023, *arXiv*: arXiv:2305.14314. doi: 10.48550/arXiv.2305.14314.

[9] A. Grattafiori *et al.*, "The Llama 3 Herd of Models," Nov. 23, 2024, *arXiv*: arXiv:2407.21783. doi: 10.48550/arXiv.2407.21783.

[10] H. Rajabzadeh *et al.*, "QDyLoRA: Quantized Dynamic Low-Rank Adaptation for Efficient Large Language Model Tuning," Feb. 16, 2024, *arXiv*: arXiv:2402.10462. doi: 10.48550/arXiv.2402.10462.

[11] A. Mahendra and S. Styawati, "Implementasi Lowk-Rank Adaptation of Large Langauage Model (LoRA) Untuk Effisiensi Large Language Model," *JIPI J. Ilm. Penelit. Dan Pembelajaran Inform.*, vol. 9, no. 4, pp. 1881–1890, Nov. 2024, doi: 10.29100/jipi.v9i4.5519.

[12] X. Lin *et al.*, "Data-efficient Fine-tuning for LLM-based Recommendation," 2024, *arXiv*. doi: 10.48550/ARXIV.2401.17197.

[13] M. S. Amrullah and S. F. Pane, *Analisis Sentimen Masyarakat Terhadap Kebijakan Polisi Tilang Manual Di Indonesia*. Penerbit Buku Pedia, 2023.

[14] M. Siino, I. Tinnirello, and M. L. Cascia, "The Text Classification Pipeline: Starting Shallow going Deeper," *Found. Trends® Inf. Retr.*, vol. 19, no. 5, pp. 557–711, 2025, doi: 10.1561/1500000107.

[15] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc., 2009.

[16] Y. Fu, H. Peng, A. Sabharwal, P. Clark, and T. Khot, "Complexity-Based Prompting for Multi-Step Reasoning," Jan. 30, 2023, *arXiv*: arXiv:2210.00720. doi: 10.48550/arXiv.2210.00720.
[17] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing," *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–35, Sep. 2023, doi: 10.1145/3560815.
[18] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023, doi: 10.1145/3560815.
[19] S. Chang, Y. Liu, J. Eisenstein, "Citation-Aware Language Models," *Transactions of the Association for Computational Linguistics (TACL)*, vol. 12, 2024 (preprint: arXiv:2402.08394).