Fine-Tuning LLM dengan QLoRA untuk Chatbot Berbasis Pengetahuan Organisasi di PT. United Tractors Tbk

(Skripsi)

Oleh

ANNISA QURROTA A'YUN

NPM 2115061103

FAKULTAS TEKNIK

UNIVERSITAS LAMPUNG

BANDAR LAMPUNG

2025

Fine-Tuning LLM dengan QLoRA untuk Chatbot Berbasis Pengetahuan Organisasi di PT. United Tractors Tbk

Oleh

ANNISA QURROTA A'YUN

Skripsi

Sebagai Salah Satu Syarat Untuk Mendapat Gelar

SARJANA TEKNIK

Pada

Program Studi S-1 Teknik Informatika

FAKULTAS TEKNIK

UNIVERSITAS LAMPUNG

BANDAR LAMPUNG

2025

ABSTRAK

Fine-Tuning LLM dengan QLoRA untuk Chatbot Berbasis Pengetahuan Organisasi di PT. United Tractors Tbk

Oleh

Annisa Qurrota A'yun

Akses cepat terhadap informasi perusahaan penting untuk meningkatkan efektivitas kerja, mendukung pengambilan keputusan, dan mendorong inovasi. Di United Tractors, karyawan masih menghadapi kesulitan menemukan informasi relevan karena pengetahuan perusahaan tersebar di berbagai sumber. Untuk menjawab tantangan tersebut, penelitian ini menerapkan *Large Language Model* (LLM) yang di-*fine-tune* pada domain khusus dan mengevaluasi performanya dibandingkan dengan *base model*. Proses *fine-tuning* dilakukan menggunakan dataset *annual report* tahun 2023 dalam bahasa Indonesia. Seluruh data digunakan untuk pelatihan, sedangkan 20% di antaranya dipilih secara acak dan diparafrase untuk evaluasi. Model yang digunakan adalah Llama 3.1 8B dengan pendekatan QLoRA, dijalankan di Google Colab berbasis GPU T4. Proses dilakukan secara iteratif sebanyak delapan belas kali untuk memperoleh konfigurasi terbaik. Evaluasi dilakukan melalui dua pendekatan, yaitu penilaian subjektif berbasis GPT-4 dengan metriks keramahan, keringkasan, kebergunaan, dan akurasi serta pengukuran objektif menggunakan BERTScore (precision, recall, dan F1-score). Hasil menunjukkan bahwa model hasil fine-tuning mengalami peningkatan performa pada hampir seluruh metriks, dengan skor rata-rata meningkat dari 2,09 menjadi 2,95. Metriks keramahan dan keringkasan memperoleh skor tertinggi masing-

masing sebesar 4,09 dan 4, sementara kebergunaan dan akurasi memperoleh skor 3 dan 2,38. Nilai F1 BERTScore juga meningkat dari 0,659 menjadi 0,818, yang menandakan peningkatan kesamaan semantik. Temuan ini membuktikan bahwa fine-tuning efektif dalam meningkatkan kualitas respons model, khususnya dalam aspek keringkasan, keramahan, dan kesesuaian semantik dengan jawaban referensi. Namun, model masih memiliki keterbatasan dalam menghasilkan informasi faktual atau angka kuantitatif yang akurat sebagaimana tercantum dalam annual report.

Kata kunci: LLM, Chatbot, QLoRA, BERTScore, Evaluasi GPT-4

ABSTRACT

Fine-Tuning LLM with QLoRA for Organization Knowledge-Based Chatbots at PT. United Tractors Tbk

By

Annisa Qurrota A'yun

Quick access to important company information is essential for improving work effectiveness, supporting decision-making, and encouraging innovation. At United Tractors, employees still face difficulties finding relevant information because company knowledge is scattered across various sources. To address this challenge, this study applies a Large Language Model (LLM) that is fine-tuned to a specific domain and evaluates its performance compared to the base model. The fine-tuning process was carried out using the 2023 annual report dataset in Indonesian. All data was used for training, while 20% of it was randomly selected and paraphrased for evaluation. The model used was Llama 3.1 8B with the QLoRA approach, run on Google Colab based on GPU T4. The process was carried out iteratively eighteen times to obtain the best configuration.

The evaluation was conducted using two approaches, namely subjective assessment based on GPT-4 with metrics of friendliness, conciseness, usefulness, and accuracy, as well as objective measurement using BERTScore (precision, recall, and F1-score). The results show that the fine-tuned model experienced improved performance across almost all metrics, with the average score increasing from 2.09 to 2.95. The friendliness and conciseness metrics received the highest scores of 4.09 and 4, respectively, while usefulness and accuracy received scores of 3 and 2.38.

The BERTScore F1 value also increased from 0.659 to 0.818, indicating an increase in semantic similarity. These findings prove that fine-tuning is effective in improving the quality of model responses, particularly in terms of conciseness, friendliness, and semantic similarity to the reference answers. However, the model still has limitations in generating accurate factual information or quantitative figures as stated in the annual report.

Keywords: LLM, Chatbot, QLoRA, BERTScore, GPT-4 Evaluation

Judul Skripsi	: **Fine-Tuning LLM dengan QLoRA untuk Chatbot Berbasis Pengetahuan Organisasi di PT. United Tractors Tbk**
Nama Mahasiswa	: Annisa Qurrota A'yun
Nomor Pokok Mahasiswa	: 2115061103
Program Studi	: S-1 Teknik Informatika
Jurusan	: Teknik Elektro
Fakultas	: Teknik

MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama	Pembimbing Pendamping
Puput Budi Wintoro, S.Kom., M.T.I	**Rio Ariestia Pradipta, S.Kom.,M.T.I**
NIP. 198410312019031004	NIP. 198603232019031013

2. Mengetahui

Ketua Jurusan Teknik Elektro	Ketua Program Studi Teknik Informatika
Herlinawati, S.T., M.T.	**Yessi Mulyani, S.T., M.T.**
NIP 197103141999032001	NIP. 197312262000122001

MENGESAHKAN

1. Tim Penguji

   Ketua : **Puput Budi Wintoro, S.Kom., M.T.I** .....

   Sekretaris : **Rio Ariestia Pradipta, S.Kom.,M.T.I** .....

   Penguji : **Yessi Mulyani, S.T., M.T.** .....

2. Dekan Fakultas Teknik

Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.

NIP 197509282001121002

Tanggal Lulus Ujian Skripsi: **13 Oktober 2025**

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya dengan judul "Fine-Tuning LLM dengan QLoRA untuk Chatbot Berbasis Pengetahuan Organisasi di PT. United Tractors Tbk" dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 18 November 2025

Pembuat Pernyataan,

Annisa Qurrota A'yun

NPM 2115061103

RIWAYAT HIDUP

Penulis dilahirkan di Bandar Lampung, pada tanggal 18 Mei 2002. Penulis merupakan anak pertama dari dua bersaudara pasangan Bapak Amruzi Setiagama dan Ibu Titin Apriyanti. Penulis menyelesaikan pendidikannya di SDIKT Robbi Rodhiya pada tahun 2014, SMPN 2 Bandar Lampung pada tahun 2017, dan MAN 1 Bandar Lampung pada tahun 2020. Pada Tahun 2021, penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik Universitas Lampung melalui jalur SBMPTN. Selama menjalani kuliah, penulis telah memfokuskan diri pada bidang kecerdasan buatan dengan mengambil konsentrasi Sistem Cerdas. Selama menjadi mahasiswa, penulis telah menjalani berbagai kegiatan antara lain:

1. Menjadi anggota divisi Pengabdian Masyarakat di Himpunan Mahasiswa Teknik Elektro Universitas Lampung (Himatro) pada tahun 2022-2023.
2. Mengikuti program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengikuti program Bangkit Academy Batch 2 2023 dengan learning path Machine Learning pada tahun 2023
3. Mengikuti program Magang Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan menjadi IT and Data Science Developer Intern di PT. United Tractors Tbk. pada tahun 2024
4. Mengikuti program pelatihan Big Data using Python dari Kementerian Komunikasi dan Informatika Republik Indonesia pada akademi Fresh Graduate Academy (FGA) Digital Talent Scholarship 2024.
5. Mengikuti Program Technopreneurship Mahasiswa Proteksi Unila 2024 yang diselenggarakan oleh Universitas Lampung pada tahun 2024.

MOTTO

Stay hungry, stay foolish.

(Steve Jobs)

Maka, sesungguhnya beserta kesulitan ada kemudahan.
Sesungguhnya beserta kesulitan ada kemudahan.

(Q.S. Al Insyirah: 5-6)

PERSEMBAHAN

Segala puji bagi Allah SWT, Tuhan semesta alam, atas rahmat dan karunia-Nya penulis dapat menyelesaikan skripsi ini. Sholawat dan salam semoga senantiasa tercurah kepada Nabi Muhammad SAW, beserta keluarga, sahabat, dan para pengikutnya hingga akhir zaman.

Dengan penuh rasa syukur, saya persembahkan skripsi ini kepada:
Kedua orang tua saya yang tercinta, Ayah dan Bunda,
yang selalu memberikan doa, dukungan dan kasih
sayang tiada henti, serta Adik saya Zafran yang selalu
mendukung dan menghibur penulis.

Dan seluruh pihak yang terlibat dan berkontribusi dalam
penelitian serta penyusunan skripsi ini.

SANWACANA

Alhamdulillahirabbil'alamin, segala puji dan syukur atas kehadirat Allah SWT, yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini. Sholawat serta salam penulis sanjungkan kepada Nabi Muhammad SAW, yang penulis harapkan syafaatnya di hari akhir kelak.

Skripsi dengan judul "Fine-Tuning LLM dengan QLoRA untuk Chatbot Berbasis Pengetahuan Organisasi di PT. United Tractors Tbk" ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika, Universitas Lampung. Dalam proses penelitian dan penyusunan skripsi ini, banyak pihak yang telah memberikan dukungan dan kontribusi kepada penulis dalam pelaksanaannya. Oleh sebab itu, dengan rasa hormat, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua dan keluarga yang selalu memberikan doa, dukungan, dan kasih sayang yang tiada henti;
2. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung;
3. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung
4. Ibu Yessi Mulyani, S.T., M.T., selaku Ketua Program Studi Teknik Informatika Universitas Lampung dan dosen penguji yang telah membantu proses kelancaran penelitian dan memberikan masukkan serta apresiasi terhadap penelitian ini;

5. Bapak Puput Budi Wintoro, S.Kom, M.T.I selaku pembimbing utama yang telah bersedia membimbing penulis serta telah banyak memberikan masukan dan saran selama melaksanakan proses penelitian ini ;
6. Bapak Rio Ariestia Pradipta, S.Kom, M.T.I selaku pembimbing pendamping dan dosen pembimbing akademik yang telah bersedia membimbing penulis selama melaksanakan proses penelitian ini dan juga melaksanakan proses perkuliahan;
7. Seluruh jajaran dosen dan staf Jurusan Teknik Informatika Universitas Lampung yang memberikan dan dukungan untuk proses penelitian ini;
8. Mbak Rika yang telah membantu penulis dalam menyiapkan segala hal administratif;
9. Mas Ciptahadi Nugraha, S.IP, MBA selaku Team Leader UT CORPU Knowledge, Infrastructure, and Facilitator yang telah membantu penulis dalam proses penelitian ini;
10. Teman-teman *server* Wibu Stress dan grup Keluarga Bapak Nopal yang telah menemani penulis dalam suka dan duka selama proses perkuliahan penulis serta memberikan saran dan secara tidak langsung terlibat dalam tahapan-tahapan pembuatan skripsi ini;
11. Jazilatul Funun, Nur Muhammad Naufal, dan Fujita Rahmah, sahabat-sahabat saya ketika masa sekolah yang tetap mendukung dan memberikan bantuan kepada penulis meskipun sudah terpisah oleh jarak.
12. Ni Putu Tiara, Chelly Sabrina, Anindya Kinarya, serta seluruh teman-teman PSTI 2021 yang telah memberikan kebersamaan, dukungan, dan bantuan kepada penulis selama masa perkuliahan ini.

Penulis menyadari bahwa penelitian ini masih jauh dari sempurna dan masih banyak kekurangannya dikarenakan keterbatasan pengetahuan dan wawasan penulis. Harapan penulis, semoga penelitian ini dapat memberikan manfaat bagi semua.

Bandar Lampung, 18 November 2025

Penulis

Annisa Qurrota A’yun

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

I. PENDAHULUAN

1.1 Latar Belakang

Pengetahuan adalah aset esensial bagi United Tractors dalam upaya peningkatan yang keberlanjutan. Hal ini sesuai dengan penelitian oleh Jarrahi [1], pengetahuan yang tersedia dapat dikembangkan menjadi solusi atas permasalahan yang terjadi, yang pada akhirnya mendorong inovasi. Knowledge management akan berperan sebagai pengelola pengetahuan. Proses knowledge management berfokus pada ekstraksi pengetahuan, memperjelasnya, dan menyimpannya secara sistematis untuk penggunaan di masa depan. Akses yang mudah ke knowledge database dapat meningkatkan efektivitas karyawan dalam mengerjakan tugas, decision making, dan berinovasi. Selain itu, mudahnya akses ke knowledge database mampu mempromosikan lingkup pengetahuan yang tumbuh bersama perusahaan.

Namun, karyawan United Tractors kerap menghadapi kesulitan ketika mencari informasi yang tepat untuk mendukung pekerjaan mereka. Hal ini berefek pada terhambatnya produktivitas karyawan karena sumber informasi yang dibutuhkan tersebar dalam berbagai sumber dan format. Oleh karena itu, karyawan memerlukan akses cepat dan mudah ke organizational knowledge.

AI berpotensi untuk membantu menyimpan dan mengorganisasi pengetahuan yang bersifat eksplisit, mengintegrasikan dan menghubungkan pengetahuan dari beberapa tempat, serta menawarkan antarmuka sistem yang lebih natural dan intuitif [1]. LLM menjadi solusi yang potensial untuk mengatasi masalah ini. LLM adalah language model yang telah dilatih sebelumnya dengan general knowledge dalam jumlah besar (pre-trained language model), seperti data teks yang bersumber dari Wikipedia, buku, website, artikel, dan dataset publik. Namun, LLM yang hanya

dilatih dengan general knowledge tidak optimal untuk tugas yang sangat spesifik, sehingga diperlukan proses fine-tuning dengan data khusus sesuai kebutuhan organisasi.

LLM yang telah difine-tune dengan data perusahaan memungkinkan pengembangan chatbot untuk memudahkan karyawan dalam mengakses informasi secara lebih efisien, yang dapat mendukung inisiatif knowledge management system perusahaan. Dengan ini, karyawan United Tractors memiliki akses cepat dan akurat ke organizational knowledge sesuai kebutuhan mereka.

Penelitian ini bertujuan untuk melakukan fine-tuning model LLM dengan metode QLoRA menggunakan dataset annual report perusahaan dalam bahasa Indonesia dan mengevaluasi performanya berdasarkan berbagai metriks. Hasil penelitian ini diharapkan dapat memberikan arah, masukan, dan kontribusi dalam pengembangan chatbot di masa depan, khususnya dalam hal fine-tuning model LLM untuk domain annual report perusahaan.

Mengacu pada penelitian sebelumnya [2], [3], [4] metode fine-tuning ideal untuk tugas yang spesifik pada domain tertentu dan sangat mudah beradaptasi untuk keperluan personalisasi serta memberikan respons yang lebih ringkas dan akurat. Sementara RAG sangat bergantung pada kualitas kemampuan pengambilan pengetahuan (retrieval) dan basis pengetahuannya berpotensi mengandung dokumen yang tidak relevan. RAG dibatasi oleh kualitas vector database dan kapabilitas pengambilan pengetahuan. Respons yang diberikan terbatas pada similarity search yang tidak berkaitan dengan kapabilitas reasoning LLM.

Dalam penelitian ini, LLM Llama 3 dipilih karena berdasarkan penelitian-penelitian sebelumnya, keluarga Llama merupakan LLM open source dengan performa baik untuk tugas question answering, sehingga sesuai untuk dieksplorasi dalam konteks fine-tuning dataset annual report perusahaan ini.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah penilitian ini adalah sebagai berikut:

1. Bagaimana proses pelatihan fine-tuning LLM menggunakan dataset *annual report* perusahaan?
2. Bagaimana performa LLM jika di-fine-tune menggunakan domain khusus, yakni dataset annual report perusahaan dalam bahasa Indonesia?
3. Bagaimana perbandingan performa model LLM yang di-fine-tune menggunakan dataset annual report perusahaan berdasarkan hasil evaluasi GPT-4 dan BERTScore?

1.3 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mendeskripsikan proses pelatihan fine-tuning LLM untuk chatbot menggunakan dataset *annual report* PT. United Tractors
2. Mengevaluasi performa model LLM yang di-fine-tune untuk chatbot menggunakan dataset annual report perusahaan dalam bahasa Indonesia
3. Membandingkan performa model LLM yang di-fine-tune untuk chatbot menggunakan dataset annual report perusahaan berdasarkan hasil evaluasi GPT-4 dan BERTScore.

1.4 Manfaat

Manfaat yang diharapkan dari penelitian ini sebagai berikut:

1. Menjadi referensi dalam pengembangan dan evaluasi fine-tuning LLM khususnya pada domain yang spesifik dalam hal ini annual report dalam bahasa Indonesia, serta menambah literatur terkait performa model LLM pada data yang mengandung informasi faktual kuantitatif.
2. Memberikan gambaran mengenai performa LLM hasil fine-tuning pada domain annual report, yang dapat menjadi dasar pertimbangan dalam

mengembangkan model serupa untuk mendukung pengelolaan pengetahuan perusahaan di masa depan.

3. Menyediakan hasil evaluasi yang dapat dijadikan acuan untuk mengimplementasikan LLM guna mendukung akses informasi dan knowledge management di lingkungan kerja.

1.5 Batasan

Penelitian ini terbatas pada lingkup pengembangan model, di mana data yang digunakan hanya berasal dari Annual Report perusahaan tahun 2023 dengan mempertimbangkan aspek privasi. Dataset yang digunakan untuk melatih model juga tidak mencakup laporan keuangan yang dilampirkan oleh auditor. Selain itu, penelitian ini tidak mencakup tahap deployment model menjadi aplikasi siap pakai atau integrasi dalam knowledge management system secara menyeluruh. Seluruh proses penelitian dilakukan menggunakan resource tingkat konsumen.

1.6 Sistematika

BAB I Pendahuluan

Bab ini membahas mengenai latar belakang mengapa perlu dilakukannya penelitian, rumusan masalah yang memuat masalah yang akan diteliti di penelitian, tujuan dilakukannya penelitian, dan batasan masalah yang dibahas dalam penelitian

BAB II Tinjauan Pustaka

Bab ini berisi mengenai teori-teori yang digunakan sebagai referensi dalam penelitian

BAB III Metodologi Penelitian

Bab ini membahas terkait waktu dan tempat penelitian dilakukan, alat dan bahan yang digunakan dalam mengerjakan penelitian, tahapan dari pengerjaan penelitian.

BAB IV Hasil dan Pembahasan

Bab ini membahas tahapan pengumpulan dan pemrosesan data, pemilihan LLM, training setup, fine-tuning, dan evaluasi, serta menganalisis hasil evaluasi BERTScore dan GPT-4

BAB V Kesimpulan dan Saran

Bab ini memuat kesimpulan yang diperoleh dari pembahasan hasil melakukan penelitian dan saran-saran untuk pengembangan penelitian lebih lanjut.

II. TINJAUAN PUSTAKA

2.1 Knowledge Management System

Knowledge management system memanfaatkan pengetahuan perushaan yang sudah dikumpulkan, bertujuan untuk meningkatkan efisiensi operasional. Sistem ini didukung oleh penggunaan basis pengetahuan (knowledge base). Basis pengetahuan biasanya berperan penting terhadap kesuksesan knowledge management yang menyediakan tempat yang terpusat untuk menyimpan informasi dan bisa diakses kapan saja.

Perusahaan yang menerapkan strategi knowledge management mencapai business outcomes lebih cepat, meningkatkan organizational learning dan kolaborasi antar anggota tim, serta mempercepat proses pengambilan keputusan pada bisnis. Selain itu, strategi ini juga mempercepat proses organisasional, seperti pelatihan, on-boarding, yang menyebabkan kepuasan dan retensi pegawai yang lebih tinggi

Salah satu alat yang digunakan oleh sebuah lembaga untuk mendapatka manfaat dari knowledge management adalah dengan membuat data warehouse. Data warehouse menyatukan data dari berbagai sumber ke dalam satu tempat, tersentral, untuk mendukung kegiatan analisis data maupun artificial intelligence. Data diambil dari berbagai sumber supaya perusahaan dapat membuat insight, memberdayakan pegawai untuk membuat keputusan berdasarkan data yang ada [5]. Dalam hal ini, penggunaan LLM sebagai chatbot mampu membantu strategi knowledge management system dalam perusahaan karena meningkatkan efisiensi karyawan dalam mencari informasi.

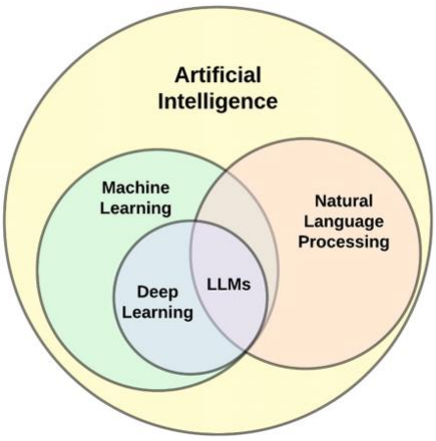
2.2 Chatbot

Chatbot adalah sebuah program komputer yang menggunakan artificial intelligence dan natural language processing (NLP) untuk memahami pertanyaan pengguna dan merespons secara otomatis seperti percakapan manusia.

Chatbot memudahkan pengguna untuk mencari informasi yang dibutuhkan dengan merespons pertanyaan dan permintaan melalui input teks, audio, ataupun keduanya, tanpa intervensi manusia. Chatbot saat ini menggunakan natural language understanding (NLU) untuk memenuhi kebutuhan pengguna. Teknologi ini bergantung pada machine learning dan deep learning, elemen dari AI, untuk mengembangkan basis pengetahuan yang semakin terperinci tentang pertanyaan dan tanggapan yang didasarkan pada interaksi pengguna. Hal ini meningkatkan kemampuan mereka untuk memprediksi kebutuhan pengguna secara akurat dan merespons dengan benar dari waktu ke waktu. [6]

2.3 Large Language Model (LLM)

Large Language Model (LLM) [7] adalah bidang yang dihasilkan dari natural language processing, konsep deep learning, dan generative AI sebagaimana yang dapat dilihat pada Gambar 1.



Gambar 1. Posisi LLM dalam Artificial Intelligence [7]

Large Language Models (LLM) adalah model AI yang umumnya merupakan turunan dari arsitektur Transformer dan didesain untuk memahami dan menghasilkan (generate) bahasa manusia, kode, dan lain-lain. Model ini dilatih menggunakan teks data dalam jumlah yang sangat besar, memungkinkan model untuk memahami kompleksitas dan nuance dari bahasa manusia. LLM mampu menjalankan banyak tugas berkenaan dengan bahasa, dari kalsifikasi teks sampai text generation dengan akurasi tinggi.

LLM sendiri berangkat dari penyempurnaan dari neural language model. Neural language model adalah model bahasa tingkat lanjut yang digunakan dalam NLP, menggunakan neural networks untuk mempelajari pola statistik dan hubungan antarkata dalam corpus text yang besar. Neural language model bisa memproses sekuens kata dengan panjang yang berbeda-beda, lebih efektif dalam memahami konteks dan menghasilkan teks yang koheren dan relevan. Neural language model didasarkan pada dua arsitektur utama: recurrent neural network dan model transformer.

2.3.1 Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), dan Gated Recurrent Unit (GRU)

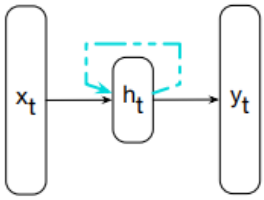
RNN [7], [8] adalah neural network didesain untuk menghandle data sekuensial, cocok untuk memproses sekuens kata dalam bahasa natural. RNN memiliki recurrent structure yang memungkinkan untuk maintain hidden states, menangkap informasi tentang konteks dari kata-kata sebelumnya. Konteks berperan penting dalam model bahasa sebab arti dari sebuah kata seringkali bergantung pada kata yang mendahuluinya. Salah satu varian RNN yang banyak digunakan dalam model bahasa adalah long short-term memory (LSTM) network, yang didesain untuk mengatasi masalah vanishing gradient dan menghandle dependensi long-range.

RNN adalah artificial neural network dua arah, memungkinkan output dari node mempengaruhi input setelahnya pada node yang sama. Adanya internal state (memory) memungkinkan untuk memproses sekuens input apapun (arbitrary

sqeuences), membuatnya cocok untuk data sekuensial, membuatnya efektif dalam menangkap dependensi sementara dan konteks pada bahasa nautral.

Ide utama dari RNN adalah hidden states yang berperan sebagai memory, menangkap informasi dari beberapa step sebelumnya dan pass ke langkah selanjutnya. Hal ini membuat RNN bisa menghandle sekuens dengan panjang berapapun dan mempertahankan konteks selama memproses kata pada sebuah kalimat. RNN memiliki keuntungan dalam menangkap long-range dependency dalam sekuens, efektif untuk memahami konteks dari sebuah kata dalam kalimat. Namun, beberapa ketertbatasannya seperti masalah vanishing gradients yang menghambat kemampuannya untuk menangkap long-term dependnensi secara efektif.

Struktur dari RNN sederhana dapat dilihat pada Gambar 2.



Gambar 2. Recurrent Neural Network (RNN) sederhana [8]

Seperti halnya jaringan feedforward biasa, sebuah vektor input yang merepresentasikan input saat ini, yaitu x_t, dikalikan dengan sebuah weight matrix dan kemudian dilewatkan melalui fungsi aktivasi non-linier untuk menghitung nilai-nilai pada lapisan tersembunyi (hidden layer). Lapisan tersembunyi ini kemudian digunakan untuk menghasilkan output yang sesuai, yaitu y_t

Perbedaan utama dari jaringan feedforward terletak pada adanya recurrent link (ditunjukkan dengan garis putus-putus pada Gambar 2). Link ini menambahkan nilai dari lapisan tersembunyi pada waktu sebelumnya ke perhitungan lapisan tersembunyi saat ini.

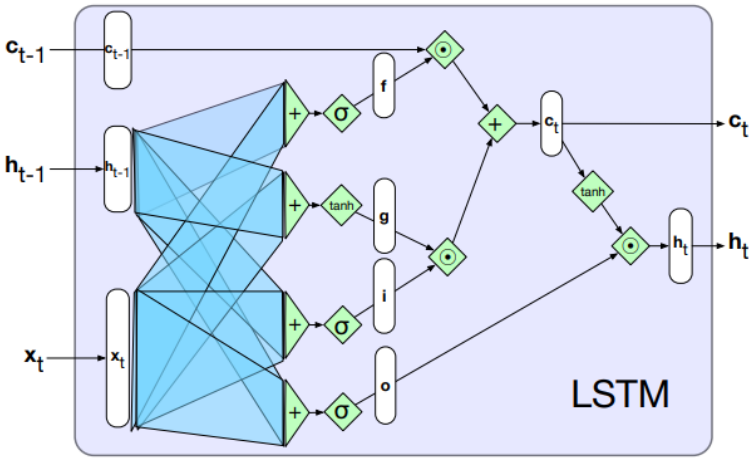
Dengan kata lain, input untuk menghitung h_t tidak hanya berasal dari x_t , tetapi juga dari h_{t-1} (aktivasi dari waktu sebelumnya). Lapisan tersembunyi dari langkah waktu

sebelumnya membawa semacam memori atau konteks, yang membantu jaringan dalam memproses urutan serta memengaruhi keputusan yang diambil di waktu-waktu berikutnya.

Pendekatan ini tidak memberlakukan batas panjang tetap pada konteks sebelumnya; konteks yang tersimpan dalam hidden layer sebelumnya dapat mencakup informasi hingga ke awal urutan.

Permasalahan vanishing gradient adalah permasalahan yang muncul ketika training RNN, terutama network dengan layer yang banyak atau sekuens yang panjang. Hal ini timbul karena algoritma backpropagation dan mengupdate weight dari model selama proses training. Vanishing gradient menjadi masalah yang besar dalam deep RNN (yang memiliki banyak layer) atau ketika memproses sekuens yang panjang. Ketika gradien menghilang (vanish) proses pembelajaran model melambat secara signifikan dan bahkan terhenti. Vanishing gradient ini disebabkan oleh kesulitan network dalam menangkap ketergantungan jangka panjang, karena gradien yang dihitung secara berulang dapat mengalami perubahan secara eksponensial. Gradien dapat menjadi sangat kecil (vanishing) seiring dengan bertambahnya jumlah lapisan.

Untuk mengatasi masalah ini, berbagai variasi RNN dengan arsitektur tertentu diperkenalkan, seperti long short-term memory (LSTM) [7], [8] dan gated recurrent unit (GRU) [7]. LSTM dan GRU memiliki mekanisme gating yang secara seletif mengontrol aliran infromasi pada network. Mekanisme ini membantu RNN mempertahankan dan mengupdate informasi yang relevan dalam waktu yang lebih panjang, secara efektif mengatasi permasalahan vanishing gradient dan meningkatkan kemampuan model untuk mempelajari dependensi jangka panjang dalam data sekuensial.



Gambar 3. Satu unit LSTM tunggal [8]

Sebuah unit LSTM tunggal yang ditampilkan sebagai grafik komputasi dapat dilihat pada Gambar 3. Input ke setiap unit terdiri dari input saat ini ($x_{t)}$, state tersembunyi sebelumya ($h_{t-1)}$, dan konteks (cell state) sebelumnya ($c_{t-1)}$. Output dari unit ini adalah state tersembunyi yang baru ($h_{t)}$ dan konteks (cell state) yang diperbarui (c_t).

LSTM mengelompokkan masalah pengelolaan konteks ini menjadi dua masalah utama, yakni menghapus informasi yang tidak lagi dibutuhkan dari konteks dan enambahkan informasi yang kemungkinan akan dibutuhkan untuk pengambilan keputusan di masa mendatang. Kunci dari pendekatan ini adalah strategi pengelolaan konteks tidak ditentukan secara eksplisit (hard-coded), melainkan dipelajari langsung oleh jaringan. LSTM melaukan hal ini dengan menambahkan layer konteks eksplisit ke dalam arsitektur (selain recurrent hidden layer biasa) san enggunakan unit saraf khusus yang memiliki gerbang (gates) untuk mengatur aliran informasi masuk dan keluar dari unit-unit dalam layer.

Setiap gerbang dalam LSTM mengikuti pola desain yang sama, yaitu:

a. Sebuah lapisan feedforward,
b. Diikuti oleh fungsi aktivasi sigmoid
c. Diikuti oleh perkalian elemen demi elemen (pointwise multiplication) dengan layer yang sedang dikendalikan (gated).

Pemilihan sigmoid sebagai fungsi aktivasi karena sifatnya yang mendorong output mendekati 0 atau 1, sehingga bila dikombinasikan dengan perkalian pointwise, efeknya mirip dengan binary mask. Nilai-nilai pada layer yang bertepatan dengan nilai mendekati 1 di dalam masker akan dilewatkan (*passed thorugh*) hampir tanpa perubahan, sementara nilai-nilai yang berkaitan dengan nilai rendah (mendekati 0) akan dihapus.

GRU [7] adalah varian lain dari RNN yang juga mengatasi masalah vanishing gradient. GRU menggunakan mekanisme gate untuk mengontrol aliran informasi secara selektif melalui hidden state, supaya bisa secara efektif retaining relevant context dalam sekuens yang panjang.

GRU cell terdiri atas beberapa komponen, termasuk reset gate dan update gate. Gate ini adalah neural network dengan fungsi aktivasi sigmoid yang menghasilkan value antara 0 dan 1. Reset gate menentukan seberapa banyak hidden state sebelumnya yang harus dilupakan (forgot) atau direset, memungkinkan GRU secara selektif mengupdate hidden state berdasarkan input terkini dan hidden state sebelumnya. Update gate menentukan seberapa banyak informasi baru yang harus di-retain dan merge ke hidden state.

Pada setiap time step, GRU cell mengambil current word embedding dan hidden state sebelumnya sebagai input. GRU menghitung value dari reset gate dan update gate menggunakan fungsi aktivasi sigmoid berdasarkan input dan hidden state sebelumnya. Kemudian, GRU menghitung candidate activation, hidden state baru yang incorporates informasi dari input terkini dan output dari reset gate. Candidate activation dikombinasikan dengan hidden state sebelumnya, weighted by output dari update gate, untuk menghitung hidden state baru di time step terkini. Output dari GRU cell (hidden state) kemudian digunakan untuk memprediksi distribusi probabilitas terhadap kata selanjutnya dalam sekuens.

GRU menunjukkan performa yang baik dalam menangkap dependensi jangka panjang dan konteks pada data sekuensial. GRU menjadi alternatif populer dari LSTM karena arstitektur yang lebih sederhana dan training prosess yang efisien.

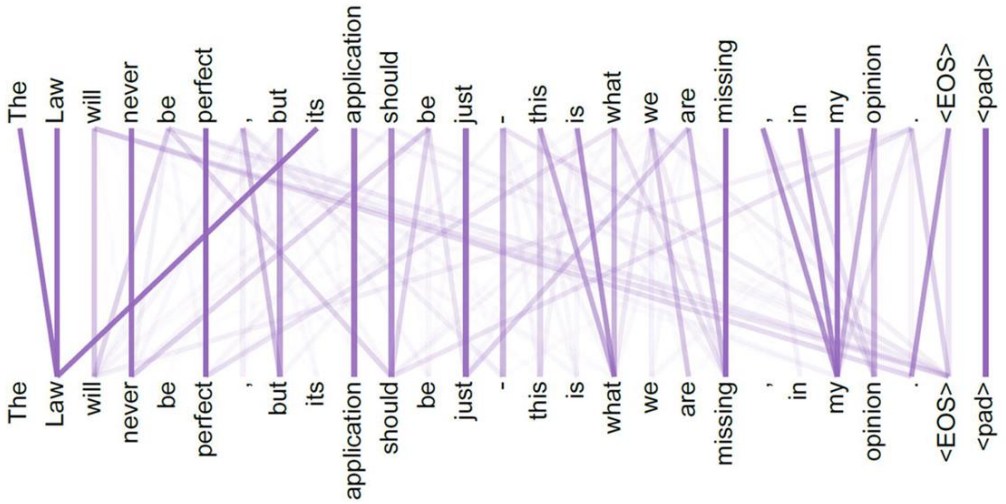
LSTM dan GRU dua-duanya efektif dalam mengatasi masalah vanishing gradient dan menangkap dependensi jangka panjang dalam data sekuensial. Arsitektur LSTM yang kompleks, dengan mekanisme yang menggunakan tiga gate menyediakan kontrol yang lebih detail atas aliran informasi yang cocok untuk task yang membutuhkan memori menajamen yang detail. Di sisi lain, arsitektur sederhana dari GRU membuat GRU menjadi alternatif yang efisien daripada LSTM.

Meskipun model bahasa berbasis RNN telah memberikan kontribusi yang signifikan pada NLP, kemampuannya sudah dilampaui oleh arsitektur terbaru, seperti transformers. Transformers telah menunjukkan performa superior dalam menangkap dependensi jangka panjang.

2.3.2 Transformer

Transformer adalah model deep learning tingkat lanjut yang didesain dengan mekanisme attention untuk memproses data sekuens. LLM dibangun berdasarkan arsitektur Transformer, maka LLM mengimplementasikan prinsip deep learning dalam skala parameter dan data yang sangat besar. Layers yang saling terhubung melakukan transformasi data berulang untuk memahami pola bahasa. Arsitektur Transformer pertama kali diperkenalkan oleh Vaswani et al. [9] yang menjadi landasan bagi perkembangan LLM. Untuk menjelaskan konsep dasar Transformer secara lebih sistematis, penelitian ini juga merujuk pada buku oleh Amaratunga [7] dan Ozdemir [10].

Mekanisme attention adalah inti dari arsitektur transformer. Secara sederhana, mekanisme attention memungkinkan sebuah model untuk berfokus pada bagian spesifik yang berisi informasi relevan dari data input. Mekanisme ini memungkinkan model untuk menangkap hubungan dan dependensi antar elemen. Visualisasi konsep ini dapat dilihat pada Gambar 4.



Gambar 4. Visualisasi bagaimana sebuah contoh kalimat mempelajari dependensi dari modul attention sebuah model transformer [7].

Mekanisme attention memiliki tiga komponen utama yakni queries (Q), keys (K), dan values (V). Vektor query merepresentasikan elemen mana yang ‘sedang diperhatikan’ dan merupakan vektor yang berisi properti dari elemen saat ini, vektor key merepresentasikan elemen lain dalam sekuens, dan vektor value berisi informasi yang berhubungan dengan setiap elemen dalam sekuens.

Mekanisme attention sendiri bisa dijelaskan sebagai memetakan (mapping) sebuah query dan sebuah set pasangan dari key-value ke sebuah output, di mana query, keys, values, dan output, semuanya berupa vektor. Output dikalkulasi sebagai weighted sum dari value. Weight yang disematkan pada setiap value dihitung dengan compatibility function dari query dengan key yang sesuai.

Selanjutnya, attention scores dihitung menggunakan dot product dari vektor query dan key. Attention scores menunjukkan seberapa relevan sebuah elemen dengan elemen lain dalam sekuensnya. Fungsi softmax diterapkan pada attention scores untuk mengonversi skor ke dalam distribusi probabilitas sehingga weight berjumlah 1. Hasil dari fungsi softmax digunakan untuk menghitung weighted sum dari vektor value. Vektor value ini menjadi vektor konteks, yang berisi “nilai kontribusi” dari setiap elemen kepada elemen saat ini.

Arsitektur Transformer menggunakan mekanisme self-attention, yakni versi lebih umum dari mekanisme attention tradisional yang menghubungkan posisi berbeda dalam satu sekuens untuk membuat representasi dari sekuens tersebut. Arsitektur Transformer dapat dilihat pada Gambar 5.

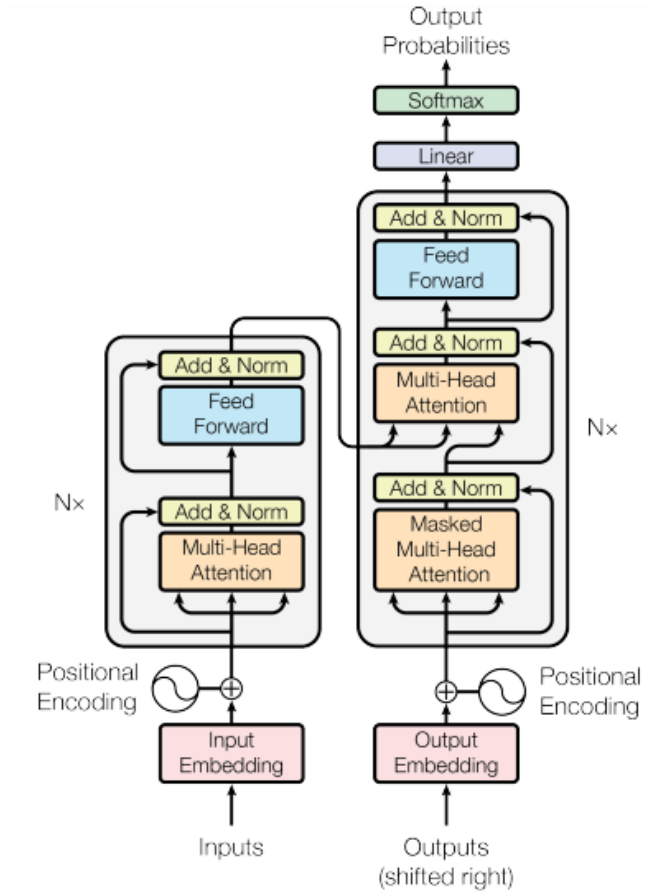
Arsitektur asli dari Transformer adalah sequence-to-sequence model yang memiliki komponen sebagai berikut:

1. Tokenizer untuk mengonversi teks ke token
2. Layer embedding untuk mengonversi token menjadi representasi semantik
3. Layer transformer yang memiliki kapabilitas penalaran dan terdiri atas layer attention dan multilayer perceptron.

Layar transformer bisa menjadi dua tipe:

1. Encoder adalah tersusun atas lapisan identik sejumlah N. Encoder yang bertugas mengambil teks asli (raw text), memecahnya ke dalam komponen-komponen inti, mengubahnya ke dalam vector, dan menggunakan attention untuk memahami konteks dari teks. Setiap lapisan memiliki dua sub-lapisan. Pertama adalah multi-head self-attention mechanism, dan yang kedua adalah positionwise feed-forward network (multilayer perceptron), terdiri atas dua transformasi linear dengan aktivasi rectified linear unit (ReLU) di antaranya. Masing-masing dari dua sub-layer memiliki residual connection di sekitarnya, diikuti dengan layer normalisasi. Output dari masing-masing sub-layer adalah LayerNorm(x + Sublayer(x)), dengan Sublayer(x) adalah fungsi yang diterapkan oleh sub-layernya sendiri. Semua sub-layer pada model, termasuk embedding layer, menghasilkan output dengan dimensi $d_{model} = 512$. Informasi mengenai posisi relatif sebuah kata dalam sekuens dimasukkan ke dalam input embeddings, yang dilakukan dengan positional encoding.
2. Decoder tersusun atas layer identik sejumlah N. Decoder bertugas untuk menghasilkan teks menggunakan tipe attention yang dimodifikasi untuk memprediksi token selanjutnya yang paling mungkin muncul. Decoder terdiri atas tiga sub-layer, dua sub-layer serupa dari encoder dan satu sub-

layer tambahan yang melakukan multi-head attention kepasa output dari susunan encoder. Residual connection juga diterapkan di masing-masing sub-layer, diikuti dengan layer normalisasi. Layer self-attention pada decoder dimodifikasi supaya posisi tidak memberikan attention kepada posisi berikutnya. Ketika model memproses kata ke-i, model hanya boleh memberikan attention pada kata sebelumnya (posisi kurang dari i) dan tidak boleh memberikan attention pada kata setelahnya. Hal ini dilakukan dengan masking (menutupi posisi yang belum boleh dilihat) dan menggeser input satu posisi ke kanan agar model membuat prediksi hanya berdasarkan kata-kata yang sudah diketahui..



Gambar 5. Arsitektur Transformer [9]

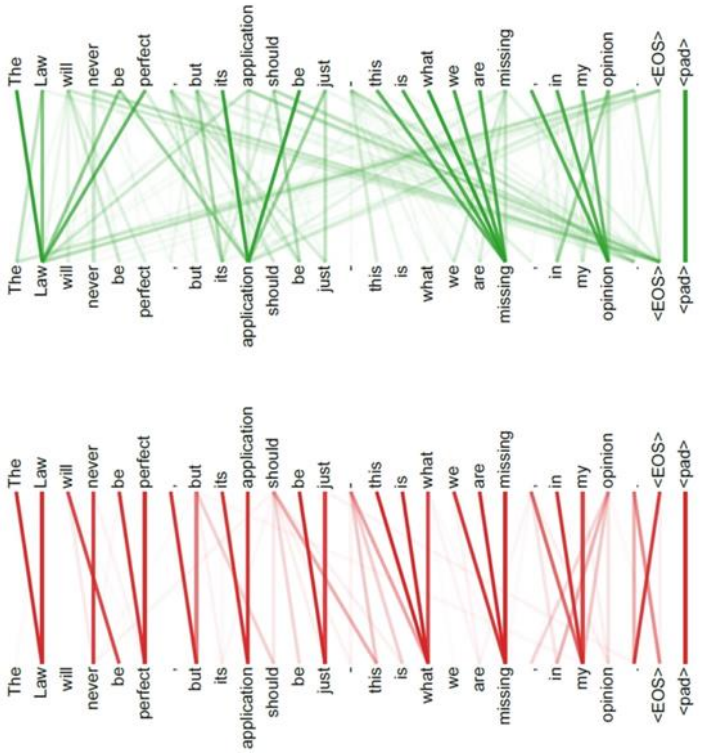
Selain arsitektur transformer, terdapat dua konsep penting lainnya, yakni scaled dot product dan multi head attention.

Scaled Dot-Product menerima input yang terdiri atas query dan key dengan dimensi d_k dan value dengan dimensi d_v. Dot products dari query dihtiung dengan semua key dan masing-masing dibagi $\sqrt{d_k}$ dan menerapkan fungsi softmax untuk mendapat weight dari value.

Dalam praktiknya, fungsi attention dihitung pada sebuah set dari query secara simultan, digabung menjadi satu ke dalam matriks Q. Key dan value juga digabung ke dalam matriks K dan V. Matriks output dihitung dengan cara:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

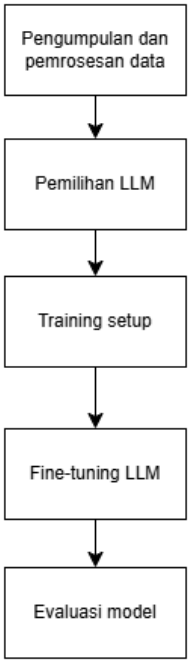
Multi Head Attention secara linear memproyeksikan query, key, dan value sebanyak h kali dan menggunakan bobot matriks yang berbeda. Masing-masing set h kemudian melalui perhitungan attention secara independen, memungkinkan model untuk berfokus pada bagian-bagian yang berbeda dari input secara bersamaan. Output dari attention head ini kemudian digabungkan dan melewati proyeksi akhir untuk mendapatkan hasil akhir. Metode ini membantu model untuk mempelajari berbagai hubungan antarkata, misalnya dependensi lokal (kata terdekat) dan dependensi jarak jauh (kata-kata yang jauh). Visualisasi dari dua head dari layer yang sama yang telah mempelajari representasi yang berbeda dapat dilihat pada Gambar 6.



Gambar 6. Contoh visualisasi dari dua head dari layer yang sama, telah mempelajari representasi yang berbeda [9]

2.4 Alur Kerja Fine-Tuning LLM

Metodologi penelitian ini didasarkan pada penelitian oleh Jeong [11] dan Parthasarathy [12]. Proses fine-tuning secara garis besar dijelaskan pada Gambar 7.



Gambar 7. Alur kerja fine-tuning LLM

Pada tahap pengumpulan data, data yang relevan dipilih dari berbagai sumber. Kemudian, data diproses dan dibersihkan, termasuk mengatasi nilai yang kosong, serta diformat untuk memenuhi kebutuhan yang spesifik. Teknik penambahan data (data augmentation) juga dilakukan untuk memperluas dataset. Salah satu teknik yang digunakan adalah parafrase data, baik secara manual maupun menerjemahkan teks ke bahasa lainnya lalu diterjemahkan kembali ke bahasa asal. Tahap ini berfungsi untuk memastikan data yang digunakan untuk fine-tuning berkualitas tinggi.

Tahap kedua adalah pemilihan LLM. LLM dipilih sesuai kebutuhan dan mempertimbangkan kriteria berikut:

a. Ukuran model: Secara umum, model yang lebih besar memiliki kemampuan yang lebih baik. Untuk mengatasi konsumsi memori yang tinggi, versi quantized bisa digunakan.
b. Kegunaan dan kemampuan model: Tentukan kebutuhan model, misalnya text summarization, sentiment analysis, dan sebagainya, sebab model memiliki spesialisasinya masing-masing. Pilih model dengan performa

terbaik dan paling sesuai dengan kebutuhan. Masing-masing arsitektur model memiliki kelebihan dan kekurangannya tersendiri.

c. Ketersediaan model: Pertimbangkan ketersediaan model dengan memperhatikan dokumentasi model, license, maintenance, dan frekuensi pembaharuan untuk menghindari masalah yang mungkin terjadi.
d. Ketersediaan sumber daya: perhatikan kesesuaian antara kebutuhan komputasi LLM dan sumber daya yang dimiliki.

Tahap ketiga adalah training setup. Secara garis besar, terdapat tiga langkah dalam training setup:

a. Mengatur training environment: Pastikan environment sistem atau cloud memiliki hardware yang dibutuhkan. Untuk penggunaan TPU, umumnya bisa menggunakan environment Google Cloud dengan TPU. Dari sisi software, framework deep learning dibutuhkan, seperti PyTorch dan TensorFlow . Selain itu, penggunaan library seperti transformer dari Hugging Face bisa menyederhanakan proses untuk me-load pre-trained model dan tokenizer.
b. Mendefinisikan hyperparameter: Hyperparameter seperti learning rate, batch size, epoch, dan dropout rate adalah hal penting untuk meningkatkan kmeampuan model. Learning rate mendikte seberapa cepat model beradaptasi, menentukan seberapa banyak weight yang harus diperbarui. Batch size adalah hyperparameter yang menentukan banyaknya data diproses oleh model di setiap iterasi. Epoch merujuk pada satu putaran penuh dari dataset training secara keseluruhan dan dianggap selesai ketika model sudah memproses semua batch dan memperbarui parameternya berdasarkan loss. Dropout rate menentukan seberapa banyak neuron yang di-dropout secara acak untuk mencegah model melihat pola yang terlalu spesifik.
c. Menginisialisasi optimizer dan loss function.

Selanjutnya tahap fine-tuning. Pertama, akan dilakukan inisialisasi pre-trained tokenizer dan model. Kemudian, penerapan strategi fine-tuning yang tepat. Selanjutnya, mengatur training loop, termasuk memperbarui parameter yang

relevan supaya efisien. Interpretasikan kurva training loss untuk memastikan model belajar secara efektif dan menghindari underfitting atau overfitting. Setelah training epoch, evaluasi model pada data valudasi untuk mengetahui kemampuan generalisasi model. Pantau kemampuan model secara berkala pada data validasi untuk memastikan performa model pada data yang belum pernah dilihat kemudian atur hyperparameter. Secara konsisten, pantau hasil training dan validasi, serta atur hyperparameter untuk mengoptimasi performa model dan menghindari overfitting. Tahap fine-tuning akan dilakukan beberapa kali eksperimen untuk mencari konfigurasi yang paling optimal.

Terakhir adalah tahap evaluasi. Kriteria dan metrik evaluasi adalah hal yang penting untuk menilai kemampuan model secara kuantitatif dan kualitatif. Untuk menilai perfroma secara kuantitatif, berbagai metrik seperti accuracy, precision, recall, dan F1 Score dapat digunakan. BERTScore menghitung precision, recall, dan F1 Score berdasarkan similarity score. Sebagai tambahan untuk evaluasi secara kuantitatif, menggunakan penilaian dengan bantuan GPT-4 juga penting untuk evaluasi secara komprehensif. Hal ini dilakukan untuk menilai seberapa baik model telah mempelajari domain knowledge. Dengan menggunakan metrik kuantitatif dan kualitatif, kelayakan model untuk bidang tertentu dapat dinilai. Hasil evaluasi nantinya digunakan untuk peningkatan lebih lanjut dari model.

2.5 LLM Meta Llama 3.1

Berdasarkan model card pada Hugging Face [13], Meta Llama 3.1 adalah koleksi Large Language Model (LLM) multilingual yang terdiri atas pretrained model dan instruction-tuned dalam ukuran 8B, 70B, dan 405B. Llama 3.1 dirilis pada 23 Juli 2024 dan menerima modal input berupa teks. Llama 3.1 dilatih menggunakan 15 triliun token dari data yang tersedia secara publik dan terbatas pada informasi yang tersedia sebelum Desember 2023. Secara arsitektur, Llama 3.1 adalah auto-regressive language model yang menggunakan arsitektur transformer yang telah dioptimasi. Llama 3.1 dapat digunakan untuk keperluan komersial dan riset dalam berbagai bahasa. Model ini telah dilatih dengan koleksi bahasa yang lebih luas, tidak terbatas pada delapan bahasa yang didukung (bahasa Inggris, bahasa Jerman,

bahasa Perancis, bahasa Italia, bahasa Portugis, bahasa Hindi, bahasa Spanyol, dan bahasa Thailand). Rata-rata hasil evaluasi Llama 3.1 pada benchmark umum menunjukkan kemampuan yang lebih baik dibandingkan pendahulunya, Llama 3 8B.

2.6 Fine Tuning

Fine-tuning adalah proses melatih sebuah pre-trained large language model untuk mengerjakan sebuah tugas atau domain yang spesifik, menggunakan general knowledge yang sudah dipelajari sebelumnya oleh model dan mengaitkannya ke data baru supaya bisa mengerjakan tugas yang lebih spesifik [7]. Meskipun pre-trained models memiliki pengetahuan bahasa yang luas, kemampuan dalam area yang spesifik masih perlu dilatih. Oleh karena itu, fine-tuning mengatasi batasan ini dengan memungkinkan model untuk mempelajari data dari domain yang spesifik supaya pengaplikasiannya semakin akurat dan efektif [14].

LLM awalnya dilatih pada corpus text yang besar dan beragam. Pada fase ini, model mempelajari struktur bahasa, tata bahasa, fakta, kemampuan penalaran, dan bahkan bias yang ada pada data. Training yang bersifat umum ini menghasilkan model yang berpengetahuan tetapi tidak terspesialisasi dalam suatu tugas.

Setelah pre-training, model dapat dilatih lebih lanjut (fine-tuned) pada dataset yang lebih kecil yang spesifik [7]. Salah satu metode fine-tuning adalah supervised fine-tuning.

Pada metode supervised fine-tuning, diperlukan labeled dataset untuk tugas terkait, yakni masing-masing input dari data point terasosiasi dengan jawaban. Model akan menyesuaikan parameternya untuk memprediksi label-label ini seakurat mungkin. Supervised fine tuning bisa meningkatkan kemampuan model secara signifikan, efektif dan efisien untuk mengkustomisasi LLM [14].

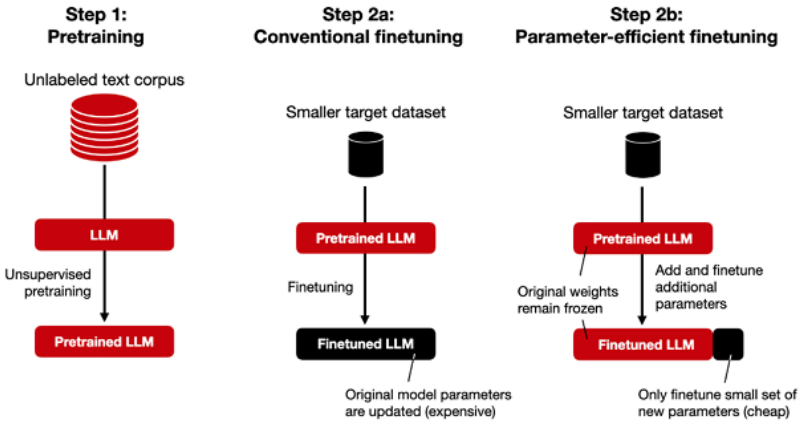
Keberhasilan proses fine-tuning dipengaruhi beberapa hal:

1. Pastikan kualitas dan kuantitas data. Data harus bersih, relevan, dan dalam jumlah besar.

2. Fine tuning adalah proses panjang yang berulang. Lakukan eksperimen terhadap berbagai settings untuk learning rates, batch size, dan jumlah epoch untuk mencari tahu setup terbaik untuk project.
3. Evaluasi model secara berkala ketika proses training untuk melacak efektivitas dan mengimplementasikan modifikasi yang diperlukan.

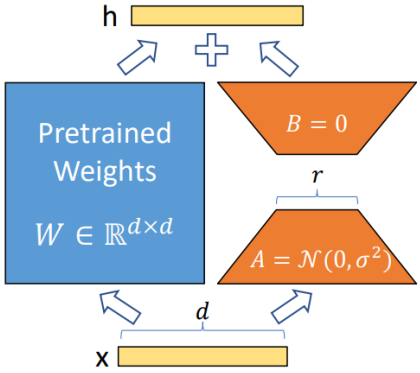
2.7 Parameter-Efficient Fine Tuning (PEFT)

Full fine tuning, atau fine tuning secara tradisional, membutuhkan sumber daya komputasi dan waktu yang besar. Sebagai contoh, untuk fine-tune model Llama 16-bit dengan parameter sebanyak 65B membutuhkan lebih dari 780 GB memori GPU [15]. Oleh karena itu, pendekatan Parameter-Efficient Fine Tuning dilakukan sebagai solusi dari biaya komputasi yang besar. PEFT bekerja dengan mengurangi banyaknya parameter yang dapat dilatih (trainable parameter) pada LLM. Hal ini dilakukan dengan menambahkan sejumlah parameter baru atau mem-fine tune sejumlah parameter yang ada pada LLM. Teknik PEFT dapat mempertahankan pengetahuan yang sudah dimiliki LLM sembari menggabungkan pengetahuan bidang khusus yang baru. Pendekatan ini juga dapat mengurangi kemungkinan untuk terjadinya overfitting sebab dataset baru biasanya berukuran lebih kecil daripada data yang digunakan ketika pre-training [16]. Perbandingan fine-tuning yang dilakukan untuk untuk melatih pre-trained LLM, full fine-tuning, dan PEFT dapat dilihat pada Gambar 8.



Gambar 8. Perbandingan pre-training LLM, fine tuning tradisional, dan PEFT [11]

Pendekatan LoRa (Low-rank adaptations), memungkinkan proses pembelajaran atau fine tune untuk bobot matriks tertentu pada LLM. LoRA membekukan bobot dari pre-trained model dan memasukkan trainable rank decomposition matrices ke dalam setiap layer arsitektur Transformer, mengurangi jumlah parameter yang dapat dilatih [17]. Di dalam setiap lapisan transformer, LoRa menentukan target fine tuning dengan memilih bobot matriks tertentu. Matriks ini kemudian direplikasi dan dilakukan fine tuning [11]. Secara spesifik, LoRA memasukkan dua feed-forward layers yang berdekatan dengan setiap feed-forward layer dalam transformer model. Layer pertama memproyeksikan input ke dalam ruang dengan dimensi lebih rendah dan layer kedua mengembalikannya ke dimensi yang asli. Perubahan secara inkremental ini, direpresntasikan sebagai delta h, ditambahkan ke representasi asli yang tersembunyi sehingga menghasilkan representasi yang telah diperbarui atau h' [4]. Gambaran cara kerja LoRA dapat dilihat pada Gambar 9.



Gambar 9. Reparametrization LoRA [17]

Metode pengembangan dari LoRA, yakni QLoRA, dapat melakukan finetuning yang dengan memori yang lebih sedikit. QLoRA [18] mampu mem-finetune model dengan 65B parameter pada sebuah 48 GB GPU tunggal dengan mempertahankan performa yang sama jika difine-tune 16-bit secara utuh. Efisiensi metode ini memungkinkan untuk mem-finetune model yang harusnya tidak mungkin dilakukan jika menggunakan finetuning tradisional.

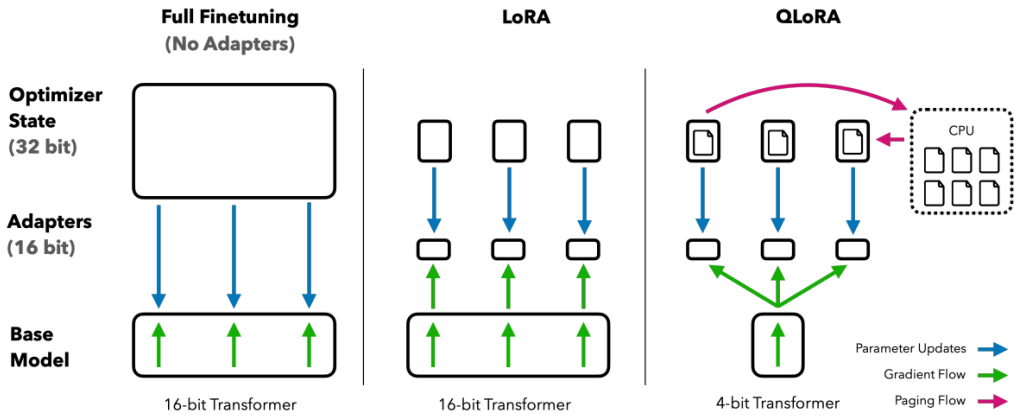
Tidak seperti LoRA yang menggabungkan data tambahan dengan menyimpan network model dasar secara utuh, QLoRA menggunakan tipe data 4-bit NormalFloat (NF4) yang optimal untuk bobot yang terdistribusi secara normal, double quantization untuk mengurangi memory footprint dengan mengkuantisasi konstanta kuantisasi (quantizing the quantization constants), dan paged optimizers untuk mengatur lonjakan memori. NF4 meningkatkan performa secara signifikan jika dibandingkan dengan FP4 dan Int4, serta double quantization mengurangi memory footprint tanpa mengurangi performa. Perbandingan antara LoRA dan QLoRA dapat dilihat pada Gambar 10.

Seperti yang telah dipaparkan sebelumnya, model yang difine-tune dengan QLoRA dapat menyamai performa model yang difine-tune secara penuh apabila dilakukan dengan praktik terbaik. Namun, jika QLoRA dilakukan menggunakan FP4, performa dapat turun sebanyak ~1%. Dalam berbagai praktiknya juga, QLoRA menunjukkan sedikit penurunan performa jika dibandingkan dengan fine-tune

secara penuh. Secara umum, penurunan kemampuan model masih dapat diterima mengingat pengurangan penggunaan daya komputasi dan memori yang signifikan.

Komponen dari QLoRA adalah sebagi berikut:

1. 4-bit NormalFloat Quantization digunakan supaya model menggunakan memori yang lebih sedikit. 4-bit NormalFloat Quantization mengompres angka yang sebelumnya adalah 16 atau 32 bits menjadi 4 bits dengan cara mengelompokkan nilai atau value ke dalam "keranjang" berdasarkan seberapa umum nilai tersebut.
2. Double Quantization merupakan proses mengkuantisasi konstanta kuantisasi untuk penghematan memori. Ketika blocksize kecil digunakan untuk 4-bit quantization yang presisi, terjadi memory overhead. Misalkan, menggunakan konstan 32 bit dan blocksize 64 bits, konstanta quantization menambah 32/64 = 0,5 bits per parameter. Double quantization membantu mengurangi memory footprint pada konstanta quantization.
3. Paged Optimizers menggunakan NVIDIA unified memory feature yang melakukan transfer otomatis page-to-page antara CPU dan GPU untuk GPU processing yang bebas error jika GPU kehabisan memori. Fitur ini digunakan untuk mengalokasikan paged memory untuk state dari optimizer, yang secara otomatis dialihkan CPU RAM ketika GPU kehabisan memori dan dikembalikan ke GPU ketika memori dibutuhkan untuk meng-update optimizer.



Gambar 10. Perbandingan LoRA dan QLoRA [18]

2.8 Unsloth

Gambar 11. Logo Unsloth

Untuk eksperimen ini, digunakan framework Unsloth [19]. Framework ini didasarkan pada bahasa Triton dari OpenAI. Framework ini mengakselerasi proses training LLM secara signifikan dengan menulis ulang kernel. Versi open source dari Unsloth bisa mempercepat proses fine-tuning dua kali lipat sembari mengurangi penggunaan memory sebanyak 50%. Unsloth mengoptimasi proses training LoRA melalui diferensiasi matriks manual. Dengan weight matriks LoRA yang lebih kecil, performa ditingkatkan dengan penempatan parentheses yang benar ketika mengombinasikan multi-level matrix multiplication dari wieght matrices Llama 3. Unsloth juga menggunakan Triton untuk mengimplementasikan manual automatic differentiation dan optimasi chain matrix multiplication, mengurangi operasi floating-point dan mempercepat training LoRA.

2.9 Torch

Gambar 12. Logo Torch

Torch [20] adalah framework scientific computing dengan dukungan yang luas untuk algoritma machine learning yang mengutamakan penggunaan GPU. Framewrok ini menggunakan bahasa LuaJIT dan mengimplementasikan C/CUDA sehingga mudah digunakan dan efisien.

Torch bertujuan untuk memiliki fleksibilitas maksimum serta performa yang cepat dalam membangun scientific algorithms dan membuat prosesnya menjadi sederhana. Torch tersedia dengan ekosistem besar dari packages yang disediakan oleh komunitas dalam bidang ML, computer vision, signal processing, parallel processing, image, video, audio and networking.

Fitur utama dari Torch adalah library neural network dan optimization yang mudah untuk digunakan, dan fleksibel dalam megimplementasikan topologi neural network yang kompleks

2.10 TRL

TRL [21] adalah library cutting edge yang didesain untuk post-training dari model dasar menggunakan teknik advance seperti Supervised Fine-Tuning (SFT), Proximal Policy Optimization (PPO), and Direct Preference Optimization (DPO). Library ini dibangun di atas ekosistem library Transformers dan mendukup berbagai arsitektur model dan modalities, dan bisa scaled-up di berbagai hardware setup.

Pada TRL, berbagai metode fine-tuning mudah diakses via trainers seperti SFTTrainer, GRPOTrainer, DPOTrainer, RewardTrainer dan lain-lain. Library ini juga sudah terintegrasi full dengan PEFT memungkinkan training pada model yang besar melalui LoRA/QLoRA. Library ini juga terintegrasi dengan unsloth untuk mengakselerasi training menggunakan kernel yang telah dioptimasi.

2.11 Transformers

Transformers [22] adalah library dari pretrained natural language processing, computer vision, audio, dan model multimodal untuk inference dan training.

Transformers dapat digunakan untuk elatih model pada data, membangun inference application, dan menggenerate text dengan LLM. Beberapa fitur utama Transformers adalah sebagai berikut:

a. Pipeline: kelas inference yang sederhana dan teroptimasi untuk berbagai tugas machine learning seperti text generation, image segmentation, automatic speech recognition, document question answering, dll.
b. Trainer: Trainer yang komprehensif dan mendukung mixed precision, torch.compile, dan FlashAttention untuk training dan distributed training untuk model PyTorch
c. Generate: Text generation dengan LLM dan VLM dengan cepat, termasuk dukungan untuk streaming and multiple decoding strategies.

2.12 Evaluasi berbasis GPT-4

Evaluasi dengan memanfaatkan bantuan GPT-4 telah menjadi pendekatan yang umum digunakan di kalangan peneliti. Hal ini disebabkan oleh kemampuan GPT-4 yang menunjukkan performa tinggi dalam menilai model secara konsisten dan akurat. Beberapa studi menyatakan bahwa evaluasi berbasis GPT-4 sering kali memiliki korelasi yang tinggi dengan penilaian manusia. Keunggulan lainnya adalah efisiensi dalam waktu dan biaya, mengingat proses evaluasi manual oleh pakar umumnya memerlukan sumber daya yang besar dan memakan waktu. Oleh karena itu, GPT-4 tidak hanya digunakan sebagai objek penelitian, tetapi juga sebagai alat bantu untuk validasi hasil. Dengan demikian, penggunaan GPT-4 dalam proses evaluasi memberikan alternatif yang praktis dan scalable bagi penelitian dan pengembangan LLM.

2.14 BERTScore

BERTScore [23] adalah metode evaluasi yang terdiri F1, Precision, dan Recall. BERTScore menghituing skor yang merepresentasikan similaritas antara teks yang dihasilkan dan dan teks yang dijadikan rujukan (gold standard reference) untuk menilai similaritas semantik. Similaritas antara dua kalimat dihitung sebagai sum

dari cosine similarities antara token embedding, yang membuat BERTScore memiliki kapasitas untuk mendeteksi parafrase. Metrik ini didesain supaya sederhana dan mudah digunakan serta bisa diterapkan pada berbagai kebutuhan (task) tanpa memerlukan penyesuaian secara spesifik. Menurut penelitian sebelumnya, BERTScore berkorelasi baik dengan penilaian manusia.
Arsitektur dari BERTScore adalah sebagai berikut:

a. Contextual embeddings: Kalimat referensi dan kalimat yang dihasilkan (kalimat kandidat) direpresentasikan menggunakan contextual embeddings berdasarkan kata-kata di sekitarnya, dikomputasi oleh model sseperti BERT, Roberta, XLNET, dan XLM.
b. Cosine similarity: Similaritas antara contextual embeddings dari kalimat referensi dan kalimat kandidat dihitung menggunakan cosine similarity.
c. Token matching untuk precision dan recall: Setiap token dalam kalimat kandidat dicocokkan dengan token yang paling mirip dengan token pada kalimat referensi, begitu pula sebaliknya, untuk menghitung recall dan precision. Recall dan precision kemudian akan dikombinasikan untuk menghitung F1 Score
d. Importance weighting: Tingkat kepentingan kata-kata yang langka dipertimbangkan dengan Inverse Document Frequency (IDF), yang bisa dimasukkan ke dalam BERTScore equation, meskipun sifatnya opsional dan bergantung pada domain.
e. Baseline Rescaling: Nilai BERTScore secara linear diskalakan ulang supaya mudah dibaca oleh manusia.

2.15 Penelitian Terdahulu

Terdapat sepuluh penelitian terkait yang dijadikan rujukan untuk mlekaukan penelitian ini.

Penelitian pertama oleh Shahrukh Azhar Ahsan [16] yang berjudul Developing a Cybersecurity Domain Chatbot based on an Open Source Large Language Model bertujuan untuk mengedukasi masyarakat luas tentang pentingnya cybersecurity.

Terdapat dua dataset yang digunakan, yakni OWASP 2023 Top 10 Mobile and API vulnerabilities, NVD Data Collection yang kemudian diubah ke dalam format yang dibutuhkan dengan GPT-4 Turbo dan menghasilkan masing-masing 273 row dan 18,861 row question and answer pair. Model yang digunakan adalah Falcon serta Llama2, di-finetune menggunakan metode QloRA. Evaluasi dilakukan menggunakan bantuan GPT-3,5 Turbo dengan cara membandingkan jawaban asli dan jawaban yang dihasilkan model. Hasilnya, Falcon-7B dan Llama2 dapat digunakan secara efektif pada topik cybersecurity.

Penelitian berjudul Efficient Finetuning Large Language Models For Vietnamese Chatbot oleh Vu-Thuan Doan, dkk [24] bertujuan untuk mem-fine tune LLM untuk chatbot berbahasa Vietnam yang berfokus pada domain umum dan medis. Data yang digunakan adalah dataset publik seperti Alpaca, GPT4All, dan ChatDoctor yang diterjemahkan ke dalam bahasa Vietnam. Peneliti melakukan parameter efficient fine-tuning LoRA terhadap model dasar, Bloomz-mt-7B dan GPTJ-6B, dan menghasilkan dua model: Bloomz-Chat dan GPTJ-Chat (chatbot dengan domain umum) serta Bloomz-Doctor dan GPTJ-Doctor (chatbot dengan domain medis). Hasilnya, performa dari Bloomz-Chat lebih baik daripada GPTJ-Chat, begitu pula performa Bloomz-Doctor dibandingkan GPTJ-Doctor.

Penelitian yang dilakukan oleh Ting Fang Tan, dkk. yang berjudul Fine-tuning Large Language Model (LLM) Artificial Intelligence Chatbots in Ophthalmology and LLM-based evaluation using GPT-4 [25] bertujuan untuk meneliti efektivitas fine-tuning terhadao berbagai LLM untuk merespons pertanyaan pasien terkait ophthalmology. Penelitian ini menggunakan dataset yang terdiri atas 400 record pasangan soal dan jawaban tentang ophthalmology yang dibuat oleh ophthalmologist. Dilakukan fine-tune pada lima model dasar, yakni GPT-3.5, LLAMA2-7b, LLAMA2-7b-Chat, LLAMA2-13b, and LLAMA2-13b-Chat. Evaluasi dilakukan menggunakan bantuan GPT-4 untuk mengukur akurasi, relevansi, keamanan pasien, dan seberapa mudah jawaban bisa dipahami. Kemudian, hasil evaluasi GPT-4 dibandingkan dengan ranking dari human clinician. Hasilnya, tiga model terbaik adalah GPT 3.5 (87,1%), Llama2-13B (80,9%), dan

Llama2 13B Chat (75,5%). Hasil evaluasi GPT-4 selaras dengan pemeringkatan yang dilakukan oleh clinician.

Rasha Ragab dan Abdulrahman Altahhan dalam penelitiannya yang berjudul Fine-Tuning Of Small/Medium Llms For Business Qa On Structured Data [26] meneliti dan meningkatkan kemampuan LLM skala kecil dan menengah dalam merumuskan query SQL yang akurat dan bisa menghasilkan respons yang tepat dari database, terutama dalam konteks bisnis, yakni sales and supply chain management. Data yang digunakan adalah kombinasi dari dataset Hugging Face b-mc2/sql-create-context dataset (gabungan data WikiSQL dan datasep Spider) dan dataset baru yang berfokus pada topik sales and supply chain operations. Model dasar yang digunakan adalah meta-llama/Llama-2-7b-chat-hf, defog/sqlcoder-7b, bugdaryan/Code-Llama-2-13B-instruct-text2sql, serta gaussalgo/T5-LM-Large-text2sql-spider dan dilakukan meenggunakan metode fine-tuning QLoRA. Hasil evaluasi menunjukkan bahwa SQL Coder 2 yang telah di-fine tune mampu merespons 54% jawaban benar dan Llama 2 13B Instruct Text2SQL menghasilkan 47% jawaban benar.

Penelitian yang dilakukan oleh Sunil Rufus yang berjudul Empowering Emotional Support Chatbots with Large Language Models [27] bertujuan untuk meneliti potensi dan penggunaan LLM untuk meningkatkan bantuan secara emosional bagi pengguna melalui layanan chatbot. Data yang digunakan yakni Extensible Emotional Support Dialogue (untuk training dan evaluation), emotional support conversation dataset (untuk evaluation). Model dasar, Mistral 7B Instruct, Llama2 7B dan Phi-3-Mini-4K, di-fine tune dengan teknik LoRA. Proses evaluasi dilakukan dengan metriks kuantitatif yakni perplexity, BLEU, ROUGE, dan BERTScore, penilaian GPT-4, serta penilaian manusia berdasarkan beberapa kriteria, yaitu engagement, fluency, comforting, kelayakan jawaban, dan saran. Hasilnya, Mistral memiliki performa yang baik berdasarkan evaluasi otomatis. Sementara itu, performa Llama2-7B dinilai baik oleh evaluasi GPT-4. Kedua model yang di-fine tune memiliki kemampuan lebih baik daripada model dasarnya berdasarkan berbagai metriks evaluasi.

Owen Christian Wijaya dan Ayu Purwarianti dalam penelitiannya yang berjudul An Interactive Question-Answering System using Large Language Model and Retrieval-Augmented Generation in An Intelligent Tutoring System on the Programming Domain [28] mengembangkan Question-Answering System berbasis web untuk membantu siswa untuk mempelajari pemrograman. Dataset yang digunakan adalah modul pembelajaran pemrograman. Model yang digunakan dalam eksperimen ini adalah LLM yang dikuantisasi ke 4-bit wieghts, yakni CodeLlama-7B Instruct, CodeGemma-7B Instruct, DeepSeek-Coder-6.7B Instruct, Zephyr-7B, Notus 7B, dan Llama3-8B. Sistem menggunakan metode RAG. Evaluasi dilakukan dengan penilaian subjektif secara internal dan penyebaran kuesioner secara online untuk penilaian eksternal. Untuk mengevaluasi proses retrieval dokumen, mean average precision (MAP) metric digunakan, serta evaluasi subjektif untuk membandingkan hasil retrieval sistem untuk ukuran chunk yang berbeda.

Penelitian berjudul A Comparison of LLM Fine-tuning Methods and Evaluation Metrics with Travel Chatbot Use Case oleh Sonia Meyer, dkk [2] bertujuan untuk membandingkan metode fine-tuning LLM serta metode evaluasi LLM menggunakan kasus penggunaaan chatbot travel. Dataset yang digunakan bersumber dari Reddit API dengan subreddit yang berhubungan dengan travel. Metode QLoRA dan RAFT diterapkan pada dua model yang digunakan, yakni Llama2 7B dan Mistral 7B. Evaluasi dilakukan dengan End to End (E2E) benchmark method of “Golden Answers”, metriks tradisional NLP, RAG Assessment (Ragas), evaluasi dengan OpenAI GPT-4, dan penilaian manusia. Model dengan kemampuan terbaik adalah Mistral RAFT. Penelitian ini juga menunjukkan bahwa metriks kuantitatif serta Ragas tidak linear dengan penilaian manusia, evaluasi menggunakan OpenAI GPT-4 adalah yang paling linear.

Anggun Tri Utami Br. Lubis dalam penelitiannya Question Answering System pada Chatbot Telegram Menggunakan Large Language Models (LLM) dan Langchain (Studi Kasus UU Kesehatan) [29] mengembangkan Question Answering System berbasis chatbot dengan studi kasus Undang-Undang Nomor 17 Tahun 2023 tentang

Kesehatan. Dataset yang digunakan adalah UU No 17 Tahun 2023 tentang Kesehatan dari laman peraturan.bpk.go.id yang berbentuk file PDF. Model GPT-3.5 Turbo dan RAG diterapkan dalam sistem ini. Evaluasi dilakukan menggunakan BERTScore dan ROUGE Score. Hasil evaluasi dari BERTScore mendapatkan rata-rata nilai precision, recall, f1-score masing-masing sebesar 76%, 80%, 78%. Sedangkan untuk ROUGE-1 sebesar 60%, 45%, 50%, untuk ROUGE-2 sebesar 34%, 25%, 28%, dan untuk ROUGE-L sebesar 45%,34%,38%.

Penelitian yang dilakukan oleh Joonas Hakkarainen [30] dalam penelitiannya Fine-Tuning An Open Source Chatbot To Translate Code From Python To Java Using Qlora: Translating For More Energy Efficient Code berfokus pada eksplorasi kelayakan fine-tuning LLM untuk chatbot penerjemah dari python ke Java dengan QLoRA. Dataset dihasilkan dari mengombinasikan dataset GeeksForGeeks, Avatar, dan Alpaca. Dataset yang digunakan adalah dataset translation besar dengan 12.000 sampel pasangan kode yang akan digunakan untuk melatih model besar (large), dataset translation kecil dengan 2000 sampel untuk melatih model kecil (small) dan dataset instruction yang juga dibuat dari Alpaca dataset dengan 10.000 sampel. Model yang digunakan adalah Llama2-7B, Llama2-7B Chat, Llama2-13B Chat. Evaluasi dilakukan menggunakan BLEU, CodeBLEU, pass@1, dan pass@5. Hasilnya, P2JLlama Instruct (Llama2 13 B Chat yang di-finetune dengan dataset instruction) menunjukkan performa terbaik.

Andri Susilo, dkk dalam penelitiannya yang berjudul Fine-Tuning LLaMA-2-Chat untuk ChatBot Penerjemah Bahasa Gaul menggunakan LoRA dan QLoRA [31] mengukur kualitas hasil jawaban dari model Llama2 7B yang telah di-finetune untuk menerjemahkan bahasa gaul ke bahasa formal. Dataset dihasilkan dari scraping komentar YouTube yang mengandung bahasa gau serta data yang dibuat secara manual dengan merujuk sumber berita dan media sosial. Metode LoRA dan QLoRA digunakan untuk mem-finetune model Llama2 7B Chat. Evaluasi dilakukan menggunakan BLEU Score dengan hasil terbaik sebesar 0,0369.

Tabel 1. Penelitian Terdahulu

No	Judul Penelitian	Dataset	Model dan Metode	Hasil Evaluasi
1.	Developing a Cybersecurity Domain Chatbot based on an Open Source Large Language Model	OWASP 2023 Top 10 Mobile and API vulnerabilities (273 QA pair), NVD Data Collection (18,861 QA pair). Keduanya diubah ke dalam format yang dibutuhkan menggunakan GPT-4 Turbo	Model:Falcon, Llama2 Metode: QLoRA	Evaluasi dilakukan dengan cara memasukkan question, true answer, dan model answer ke GPT-3.5-Turbo. Falcon-7B dapat menghasilkan lebih banyak respons akurat. Llama2-7B dapat menghasilkan lebih banyak contoh kode. Oleh karena itu, keduanya bisa digunakan secara efektif untuk bidang cybersecurity.
2.	Efficient Finetuning Large Language Models For Vietnamese Chatbot	Alpaca (52k samples), GPT4All (150k samples), ChatDoctor (200k samples). Semuanya diterjemahkan ke dalam bahasa vietnam.	Model: Bloomz-mt-7B. GPTJ-6B Metode: LoRA	Evaluasi dilakukan menggunakan penilaian dari GPT-4. Untuk ranah umum, model Bloomz-Chat menunjukkan performa yang baik dibandingkan dengan GPT-J-Chat. Sementara untuk ranah medis, Bloomz-Doctor menunjukkan performa yang baik; jawaban yang

				dihasilkan GPTJ-Doctor dapat diterima.
3.	Empowering Emotional Support Chatbots with Large Language Models	Extensible Emotional Support Dialogue (untuk training dan evaluation), emotional support conversation dataset (untuk evaluation)	Model: Mistral 7B Instruct, Llama2 7B, Phi-3-Mini-4K Metode: LoRA	Evaluasi dilakukan menggunakan penilaian dari GPT-4. Hasilnya, Llama2 menghasilkan jawaban yang paling baik, namun Mistral memiliki inference time tercepat.
4	Fine-Tuning Of Small/Medium LLMs For Business QA on Structured Data	Hugging Face b-mc2/sql-create-context dataset dan dataset baru untuk keperluan bisnis.	Llama2 7B Chat HF, Defog SQLCoder 7B, Bugdaryan Code-Llama2 13B Instruct Text2SQL, Zero-shot evaluation, QLoRA	Evaluasi dilakukan menggunakan Rouge N-grams dengan n=1. Model SQL Coder 2 yang difine-tune meraih 54% jawaban benar, model Llama2 13B Instruct Text2SQL meraih 47% jawaban benar
5.	Fine-tuning Large Language Model (LLM) Artificial Intelligence	Dataset berisi 400 pasangan soal dan jawaban yang berkaitan dengan ophthalmology,	Model: GPT 3.5, Llama2 7B, Llama2 7B Chat, Llama2-13B,	Evaluasi dilakukan menggunakan bantuan GPT-4 dan penilaian dari human clinician, yang kemudian kedua penilaian tersebut

	Chatbots in Ophthalmology and LLM-based evaluation using GPT-4	meliputi berbagai kondisi.	dan Llama2 13B Chat. Metode: Fine-tune	dievaluasi menggunakan Cohen's Kappa, Spearman and Kendall Tau correlation coefficients. Tiga model terbaik adalah GPT 3.5 (87,1%), Llama2-13B (80,9%), dan Llama2 13B Chat (75,5%).
6.	An Interactive Question-Answering System using Large Language Model and Retrieval-Augmented Generation in An Intelligent Tutoring System on the Programming Domain	Modul pembelajaran pemrograman	Model: LLM yang dikuantisasi ke 4-bit wieghts, yakni CodeLlama-7B Instruct, CodeGemma-7B Instruct, DeepSeek-Coder-6.7B Instruct, Zephyr-7B, Notus 7B, dan Llama3-8B. Metode: RAG	Evaluasi dilakukan dengan melakukan subjective scoring untuk penilaian internal dan online questionnaire untuk penilaian eksternal. Selain itu, Mean Average Precision (MAP) metric juga digunakan untuk mengevaluasi sistem retrieval dokumen. Penilaian secara subjektif juga dilakukan untuk membandingkan hasil retrieval sistem untuk ukuran chunk yang berbeda. Model Llama3 meraih nilai tertinggi pada penilaian internal dan eksternal.

7.	A Comparison of LLM Fine-tuning Methods and Evaluation Metrics with Travel Chatbot Use Case	Dataset diambil dari Reddit dengan subreddit yang berhubungan dengan travel.	Model: Llama2-7B, Mistral-7B Metode: QLoRA, RAFT, dan RLHF pada model terbaik	Evaluasi dilakukan dengan End to End (E2E) benchmark method of "Golden Answers", metriks tradisional NLP, RAG Assessment (Ragas), evaluasi dengan OpenAI GPT-4, dan penilaian manusia. Model dengan kemampuan terbaik adalah Mistral RAFT.
8.	Question Answering System pada Chatbot Telegram Menggunakan Large Language Models (LLM) dan Langchain (Studi Kasus UU Kesehatan)	UU No 17 Tahun 2023 tentang Kesehatan dari laman peraturan.bpk.go.id yang berbentuk file PDF	Model: GPT-3.5 Turbo Metode: RAG	Evaluasi dilakukan menggunakan BERTScore dan ROUGE Score. Hasil evaluasi dari BERTScore mendapatkan rata-rata nilai precision, recall, f1-score masing-masing sebesar 76%, 80%, 78%. Sedangkan untuk ROUGE-1 sebesar 60%, 45%, 50%, untuk ROUGE-2 sebesar 34%, 25%, 28%, dan untuk ROUGE-L sebesar 45%,34%,38%.

9.	Fine-Tuning An Open Source Chatbot To Translate Code From Python To Java Using Qlora: Translating For More Energy Efficient Code	Dataset dihasilkan dari mengombinasikan dataset GeeksForGeeks, Avatar, dan Alpaca. Dataset translation besar memiliki 12.000 sampel pasangan kode yang akan digunakan untuk melatih model besar (large). Dataset translation kecil memiliki 2000 sampel untuk melatih model kecil (small) . Dataset instruction juga dibuat dari Alpaca dataset dan menghasilkan 10.000 sampel.	Model: Llama2-7B, Llama2-7B Chat, Llama2-13B Chat Metode: QLoRA	Evaluasi dilakukan menggunakan BLEU, CodeBLEU, pass@1, dan pass@5. Hasilnya, P2JLlama Instruct (Llama2 13 B Chat yang di-finetune dengan dataset instruction) menunjukkan performa terbaik.
10.	Fine-Tuning LLaMA-2-Chat untuk ChatBot Penerjemah Bahasa Gaul menggunakan	Dataset dihasilkan dari *scraping* komentar YouTube yang mengandung bahasa gau serta data yang dibuat secara manual dengan merujuk	Model: Llama2-7B Chat Metode: LoRA dan QLoRA	Evaluasi dilakukan dengan BLEU Score. Hasilnya, pada eksperimen pertama, model meraih skor 0, eksperimen kedua 0, eksperimen ketiga 0.0369, dan eksperimen keempat 0. Hal ini

	LoRA dan QLoRA	sumber berita dan media sosial.		menunjukkan kemampuan model yang belum bisa menerjemahkan bahasa gaul. Namun, eksperimen ketiga dapat menjadi *baseline* untuk eksperimen selanjutnya.

Berdasarkan Tabel 1, diketahui bahwa berbagai penelitian mengenai pemanfaatan LLM berbasis chatbot dengan domain yang spesifik telah dilakukan di berbagai bidang, yakni kesehatan, bisnis, maupun domain umum. Penelitian yang akan dilakukan adalah pengembangan model chatbot untuk membantu karyawan PT. United Tractors memperoleh informasi. Dalam proses pengembangannya, model chatbot ini akan menggunakan dataset dengan domain spesifik, yakni *organizational knowledge* perusahaan. Penelitian ini juga akan melakukan proses fine-tuning model dasar Llama 3.1 menggunakan Parameter-Efficient Fine Tuning (PEFT). Metode evaluasi yang akan dilakukan adalah evaluasi kuantitatif menggunakan BERTScore dan evaluasi kualitatif menggunakan evaluasi GPT-4 dengan cara membandingkan jawaban asli dan jawaban yang dihasilkan oleh model.

III. METODE PENELITIAN

3. 1 Waktu dan Tempat

Penelitian dan pembuatan tugas akhir dilakukan selama 5 bulan, dimulai dari bulan Februari 2025 sampai dengan Juli 2025 dengan lokasi penelitian Universitas Lampung. Jadwal Penelitian dapat dilihat pada Tabel 2.

Tabel 2. Jadwal Penelitian

No	Nama Kegiatan	Waktu					
		Feb	Mar	Apr	Mei	Jun	Jul
1.	Studi Literatur						
2.	Pengumpulan dan pemrosesan data						
3.	Pemilihan LLM						
4.	Training setup						
5.	Fine-tuning LLM						
6.	Evaluasi model						
7.	Penyusunan Laporan						

3.2 Alat dan Bahan

3.2.1 Alat

Adapun penelitian ini menggunakan perangkat keras (hardware) dan perangkat lunak (software) dengan spesifikasi yang dapat dilihat pada Tabel 3.

Tabel 3. Alat Penelitian

No	Perangkat	Spesifikasi	Deskripsi
1.	Laptop	Processor AMD Ryzen 5 5500U dengan Radeon Graphics 2.10 GHz, RAM 8 GB, SSD 512 GB, dan Windows 10	Perangkat keras yang digunakan selama melakukan penelitian
2.	Python		Bahasa pemrograman yang digunakan dalam melakukan penelitian, sejak tahap data pre-processing sampai evaluasi.
3.	Google Colab		Perangkat lunak berbasis web yang digunakan untuk menuliskan syntax dalam bahasa pemrograman python.
4.	AI Library		Library yang dibutuhkan untuk mengolah data dan fine-tune model.
	Unsloth		Library yang dirancang khusus untuk membuat proses fine-tuning LLM lebih cepat dan efisien.
	Torch		Digunakan untuk meng-handle dataset, optimisasi parameter, dsb.
	TRL		Mengotomasi dan mempermudah proses Supervised Fine-Tuning (SFT) pada LLM
	Transformers		Menyediakan model dasar dan ekosistem tools yang dibutuhkan untuk fine-tuning

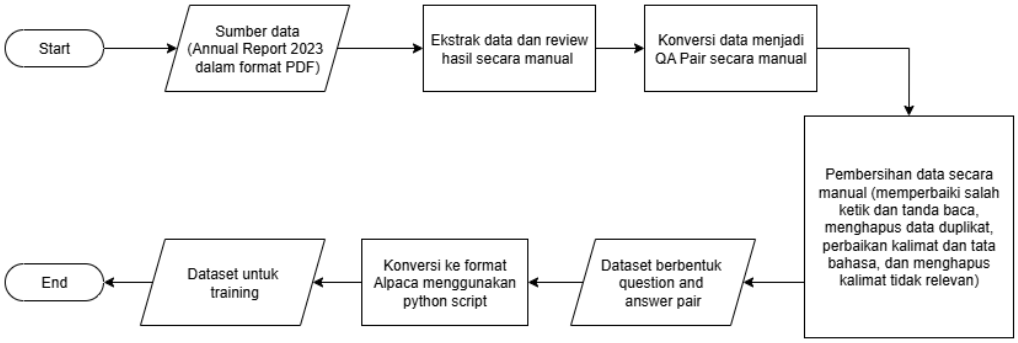
3.2.2 Bahan

Bahan yang digunakan dalam penelitian adalah data berupa pasangan pertanyaan dan jawaban sebanyak 1750 record yang diambil dari Annual Report 2023. Annual Report 2023 tersusun atas 524 halaman, yang terdiri atas ikhtisar utama, laporan manajemen, profil perusahaan, analisis dan pembahasan manajemen, tata kelola perusahaan, tanggung jawab sosial dan lingkungan, dan laporan keuangan 2023. Seperti yang telah dinyatakan pada batasan penelitian, data yang digunakan tidak mencakup bagian laporan keuangan secara rinci. Hal ini disebabkan karena pengolahan data keuangan memerlukan pengetahuan khusus di bidang akuntansi dan keuangan yang berada di luar cakupan kompetensi dan fokus penelitian ini. Ringkasan data finansial sudah ada dalam Annual Report 2023.

3.4 Tahapan Penelitian

Adapun tahapan pada penelitian ini diawali dengan melaksanakan studi literatur untuk mempelajari ilmu yang dibutuhkan untuk pengembangan model dan penelitian yang telah dilakukan sebelumnya yang bersumber dari buku, artikel, serta artikel jurnal. Kemudian, menyediakan alat dan bahan yang dibutuhkan untuk penelitian. Tahap ini termasuk pengumpulan dan pemrosesan data supaya siap untuk dipakai dalam proses selanjutnya. Kemudian, melakukan pemilihan LLM yang akan dipakai. Selanjutnya, mengatur training setup. Lalu, dilakukan proses fine-tuning base model dengan data yang telah disiapkan sebelumnya. Setelah selesai melakukan fine-tuning, dilakukan evaluasi model menggunakan evaluasi kuantitatif dan kualitatif yang meliputi evaluasi BERTScore dan penilaian menggunakan GPT-4 dengan membandingkan jawaban asli sebagai ground truth dengan jawaban yang dihasilkan model. Terakhir, menyusun laporan hasil penelitian.

3.4.1 Pengumpulan dan Pemrosesan Data



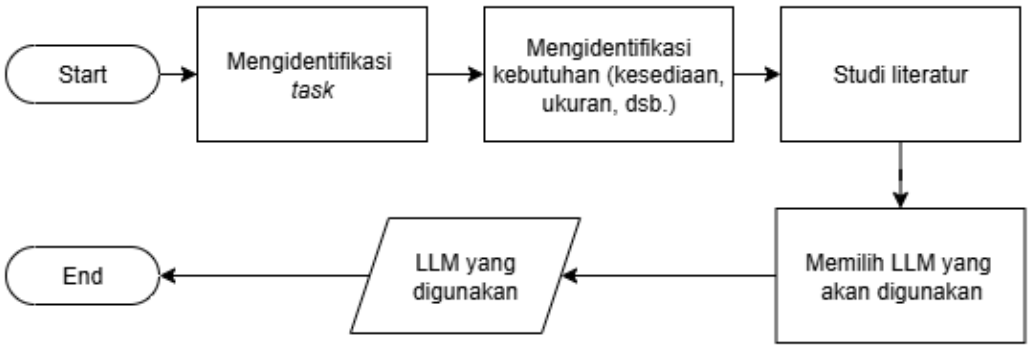
Gambar 13. Alur Pengumpulan dan Pemrosesan Data

Tahap ini dimulai dari pengumpulan data yang dibutuhkan. Data dikumpulkan dari Annual Report perusahaan tahun 2023. Kemudian, data yang telah dikumpulkan dan direview diubah ke dalam bentuk pasangan pertanyaan dan jawaban (question and answer pair) dengan bantuan GPT-4 di bawah pengawasan human annotator untuk menjaga kualitas data. Proses pembersihan (cleaning) turut dilakukan, meliputi memperbaiki salah ketik dan tanda baca, menghapus data duplikat, perbaikan kalimat dan tata bahasa, dan menghapus kalimat tidak relevan. Dataset yang sudah berbentuk question and answer pair diubah ke dalam format Alpaca menggunakan python script. Pembersihan dan pengubahan format terhadap data dilakukan supaya bisa data bisa ditokenisasi. Keseluruhan data digunakan menjadi training dataset. Alur pengumpulan dan pemrosesan data dapat dilihat pada Gambar 13.

Kemudian, untuk dataset evaluasi, dataset diambil dari 20% training dataset yang kemudian diparafrase supaya data yang dilihat oleh model berbeda. Alur pemrosesan dataset evaluasi dapat dilihat pada Gambar 14.

Gambar 14. Pemrosesan Dataset Evaluasi

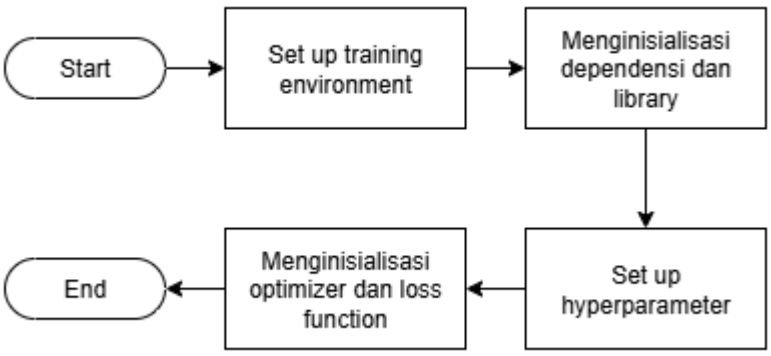
3.4.2 Pemilihan LLM



Gambar 15. Alur Pemilihan LLM

Tahap ini bertujuan untuk memilih LLM yang sesuai dengan masalah dan kebutuhan. Proses pemilihan LLM dapat diamati pada Gambar 15. Tahap pertama adalah menentukan tujuan atau task yang akan dilakukan, yakni question and answer (chatbot). Pada tahap ini juga, dilakukan identifikasi kebutuhan yang akan dijadikan pertimbangan untuk memilih LLM, antara lain memahami spesialisasi LLM, ketersediaan, kompatibilitas, ukuran, dan versi kuantisasinya. Setelah itu, melakukan studi literatur untuk mencari referensi mengenai LLM yang akan digunakan. LLM yang akan dipilih adalah yang mendemonstrasikan hasil menjanjikan pada penelitian sebelumnya, yang mengindikasikan potensi kesuksesan untuk tugas yang akan dilakukan dalam penelitian ini. Sumber data yang digunakan adalah artikel ilmiah, artikel dari website yang kredibel, dan leaderboard yang menampilkan beberapa benchmark populer.

3.4.3 Training Setup



Gambar 16. Training Set Up

Ketika mengatur training setup, ada tiga hal yang perlu diperhatikan, yakni men-set up environment training, menentukan hyperparameters, dan menentukan optimizer serta loss function. Gambar 16 menampilkan tahapan training set up.

Di tahap set up environment training, penting untuk mengonfigurasi hardware yang memiliki performa yang tinggi, seperti GPU atau TPU, serta memastikan instalasi software yang dibutuhkan, misalnya framework seperti Torch atau TensorFlow. Pastikan hardware yang akan digunakan kompatibel dengan kebutuhan dan supaya men-training model dengan computational power dengan efektif. Pastikan pula environment yang dipakai telah terinstall hardware yang dibutuhkan. Pastikan juga software dan/atau framework yang digunakan dalam versi terupdate untuk meningkatkan training dan kemampuan model. Gunakan juga software yang dapat menyederhanakan proses loading pre-trained model dan tokenizer. Pastikan semua komponen software, termasuk library dan dependencynya, kompatibel dengan framework dan setup hardware.

Dalam memilih hardware, pertimbangkan memory yang dibutuhkan. LLM biasanya membutuhkan memory GPU yang tinggi, jadi memilih GPU yang memiliki VRAM besar (16 GB ke atas) akan sangat membantu proses training.

Kemudian, mengatur hyperparameter. Penting untuk mengatur hyperparameter seperti learning rate, batch size, dan epoch untuk mengoptimasi kemampuan model.

Hyperparameter nantinya akan disesuaikan pada proses training untuk mencari konfigurasi terbaik. Beberapa key hyperparameter yang diperhatikan adalah:

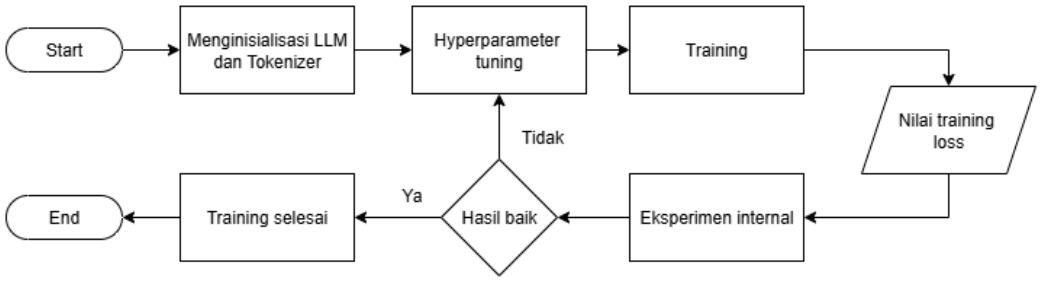
a. Learning rate: seberapa cepat bobot model diupdate per training steps. Semakin besar nilainya, semakin cepat proses training, mengurangi overfitting, dan semakin cepat bobot berubah. Jika nilainya kecil, training akan berjalan stabil, membutuhkan banyak epoch, dan proses training lebih lama karena penyesuain bobot yang kecil per update.
b. Batch size: jumlah data yang diproses oleh model di tiap iterasi. Semakin besar nilainya, maka akan semakin banyak update, convergence yang lebih cepat, performa yang baik, dan membutuhkan banyak VRAM. Jika nilainya kecil, update lebih sedikit dan VRAM lebih sedikit
c. Epoch: full pass melalui keseluruhan training dataset. Semakin besar nilainya, maka model akan belajar lebih baik tetapi mungkin terjadi overfitting. Jika nilainya kecil, model bisa jadi undertrain dan proses learning insufficient.
d. Rank: menentukan rank/dimensi dari matriks LoRA, mempengaruhi kompleksitas dan kapasitas dari model. Semakin besar rank, semakin expressive power, tapi mungkin overfit. Jika nilainya kecil, akan mengurangi overfit.
e. Alpha: scaling factor untuk update bobot. Nilai alpha yang besar dibutuhkan supaya performa model baik. Rule of thumb nya adalah rank * 1 atau rank * 2. Nilai alpha yang besar memberikan penekanan pada informasi baru, sementara nilai yang kecil membuat model lebih bergantung pada parameter model yang asli.
f. Dropout: Mengecualikan beberapa neuron secara acak untuk mencegah model terlalu bergantung pada pola tertentu.
g. Warmup steps: Meningkatkan learning rate bertahap pada saat memulai training. Nilai yang direkomendasikan adalah 5-10% dari total steps.
h. Gradient Accumulation Steps: Mensimulasikan nilai batch size yang lebih besar dengan mengakumulasikan gradient dari beberapa batch kecil sebelum mengupdate weight. Hyperparameter ini dapat menghemat penggunaan memori, namun membutuhkan waktu fine-tuning yang lebih

lama. Biasanya digunakan untuk proses fine-tuning yang menggunakan GPU tunggal.

Tuning hyperparameter dilakukan berulang-ulang untuk menemukan kombinasi konfigurasi yang paling optimal dan menghasilkan output terbaik. Proses ini biasanya terjadi secara trial and error, tracking setiap hyperparameter adjustmen. Pada penelitian ini, akan menggunakan metode random search dengan pertimbangan penyesuaian resource.

Selanjutnya menginisialisasi optimizers dan loss function. Penting untung memilih optimizer yang memperbarui bobot secara efisien dan loss function yang tepat untuk mengukur kemampuan model. Pada umumnya, dalam training LLM, akan digunakan optimizer AdamW, ekstensi dari Adam. Adam sendiri mengombinasikan keuntungan dari AdaGrad dan RMSprop, membuatnya cocok untuk task dengan dataset yang besar dan space berdimensi tinggi. AdamW yang merupakan ekstensi dari Adam yang memasukkan weight decay regularization untuk mengatasi masalah overfitting yang timbul di Adam. AdamW cocok digunakan di kondisi di mana regularization dibutuhkan, seperti mencegah overfitting pada model yang besar dan mem-finetune pre-trained model.

3.4.4 Fine-tuning



Gambar 17. Alur Fine-tuning

Alur fine-tuning ditunjukkan pada Gambar 17. Pertama, terdapat inisialisasi pre-trained tokenizer dan model. Kemudian, men-setup strategi fine-tuning. Pada penelitian ini akan menggunakan Parameter Efficient Fine-Tuning QLoRA. Metode fine-tuning dipilih sebab metode ini lebih cocok jika ingin memasukkan

pengetahuan tentang ranah spesifik (domain-specific knowledge) [12]. Sebuah penelitian [3] juga menunjukkan bahwa finetuning pada bidang tertentu memberikan hasil yang lebih akurat. Metode PEFT direkomendasikan untuk fine-tuning dengan dataset berukuran kecil [4]. Dikarenakan sumber daya komputasi yang terbatas, metode QLoRA dipakai untuk meraih efisiensi memori ketika fine-tune LLM [11]. Meskipun memakan resource yang lebih sedikit, penerapan 4-bit QLoRA dengan data type NF4 menunjukkan hasil yang sama dengan 16 bit full finetuning dan 16 bit LoRA finetuning pada benchmark akademik [18].

Namun, berdasarkan eksperimen [32], QLoRA memiliki sedikit pengaruh pada performa model dibandingkan LoRA.

	TruthfulQA MC1	TruthfulQA MC2	Arithmetic 2ds	Arithmetic 4ds	BLiMP Causative	MMLU Global Facts
Llama 2 7B base	0.2534	0.3967	0.508	0.637	0.787	0.32
LoRA default 1	0.2876	0.4211	0.3555	0.0035	0.75	0.27
LoRA default 2	0.284	0.4217	0.369	0.004	0.747	0.26
LoRA default 3	0.2815	0.4206	0.372	0.004	0.747	0.27
QLoRA (nf4)	0.2803	0.4139	0.4225	0.006	0.783	0.23
QLoRA (fp4)	0.2742	0.4047	0.5295	0.016	0.744	0.23

Gambar 18. Perbandingan Metode PEFT [32]

Ketika proses fine-tuning, dilakukan training loop dan pengawasan (monitoring) terhadap nilai accuracy dan loss. Nilai accuracy dan loss ini menjadi acuan untuk tuning hyperparameter. Penyesuaian ini akan dilakukan selama nilai training loss belum mencapai 0.5-0.8 dan menghasilkan jawaban dengan halusinasi seminimal mungkin. Hyperparameter tuning akan dilakukan secara berulang selama proses fine-tuning sampai menemukan kombinasi yang optimal [11]

3.4.5 Evaluasi

Evaluasi akan dilakukan menggunakan dataset evaluasi yang merupakan 20% dari dataset training. Data yang ada merupakan hasil parafrase pertanyaan yang ada di training dataset supaya membuat data validation yang berbeda dari yang ada di training set namun tetap dalam lingkup pengetahuan yang sama. Parafrase dilakukan menggunakan GPT-4 di bawah pengawasan human annotator.

Tahap evaluasi akan dilakukan secara kuantitatif dan kualitatif. Evaluasi secara tradisional tidak dilakukan sebab metrik hanya dapat menilai secara semantik, tidak bisa secara makna sintaksis serta tidak linear dengan evaluasi yang dilakukan oleh manusia [2]. Evaluasi secara kuantitatif dilakukan dengan BERTScore. Dalam penelitian sebelumnya [2], BERTScore terbukti menjadi metriks yang dapat diandalkan di antara metriks kuantitatif lainnya.

Seperti yang dilakukan pada penelitian sebelumnya [2], [15], [23], [24], [26], evaluasi secara kualitatif dilakukan menggunakan bantuan LLM GPT-4 yang akan memberikan skor pada jawaban yang dihasilkan oleh model. Metode ini digunakan karena GPT-4 menunjukkan kapabilitas penilaian yang baik pada berbagai penelitian dan dapat selaras dengan evaluasi manusia.

Metode ini dilakukan dengan membandingkan jawaban yang ada pada dataset (golden answer) dan jawaban yang dihasilkan oleh model [33]. Metriks evaluasi diadopsi tanpa modifikasi dari penelitian tentang question-answering system untuk domain programming [28] sebab sesuai dengan konteks penelitian ini. Skala Likert 1—5 digunakan untuk penilaian pada masing-masing metriks. Nilai yang semakin tinggi pada setiap metriks menandakan kualitas yang lebih baik pada metriks tersebut. Penjelasan mengenai matriks kualitatif dipaparkan pada Tabel 4.

Tabel 4. Metriks Kualitatif untuk Evaluasi Jawaban

Metriks	Deskripsi
Keramahan (*friendliness*)	Mengevaluasi gaya dari jawaban yang dihasilkan
Keringkasan (*conciseness*)	Mengevaluasi apakah jawaban yang dihasilkan ringkas.
Kebergunaan (*helpfulness*)	Mengevaluasi apakah model menjawab pertanyaan pengguna sesuai kebutuhan.
Akurasi (*faithfulness*)	Mengevaluasi apakah model menjawab pertanyaan dengan benar.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa:

1. Karyawan United Tractors membutuhkan kemudahan dalam mengakses informasi organizational knowledge. Oleh karena itu, dilakukan proses fine-tuning LLM dengan data perusahaan untuk pengembangan chatbot guna mempermudah akses informasi oleh karyawan. Fine-tuning dilakukan dengan menggunakan dataset bersumber dari annual report perusahaan tahun 2023 yang dikonversi menjadi pasangan pertanyaan dan jawaban. Proses ini menggunakan model Llama 3.1 8B yang dilatih menggunakan QLoRA agar efisien terhadap penggunaan sumber daya. Proses fine-tuning dilakukan menggunakan environment Google Colab dengan GPU T4. Fine-tuning dimulai dari inisialisasi LLM dan tokenizer, hyperparameter tuning, kemudian melakukan proses training. Setelah itu, eksperimen internal dilakukan, dan jika hasilnya belum baik, maka akan dilakukan kembali hyperparameter tuning dan dilanjutkan dengan training. Training akan diakhiri jika hasil eksperimen sudah baik. Proses fine-tuning dilakukan sebanyak delapan belas kali iterasi untuk mencari konfigurasi optimal. Kemudian, model dievaluasi menggunakan BERTScore dan evaluasi menggunakan GPT-4. Fine-tuning telah berhasil dilakukan, ditunjukkan dengan adanya peningkatan skor dari model dasar.
2. Model LLM yang telah di-fine-tune menggunakan dataset annual report perusahaan dalam bahasa Indonesia menunjukkan peningkatan performa

dibandingkan base model. Pada evaluasi dengan GPT-4, skor rata-rata aspek keramahan naik dari 3.84 menjadi 4.09, meskipun nilai modus masih tetap tinggi pada base model (5). Peningkatan signifikan juga terlihat pada keringkasan, dengan rata-rata naik dari 2.71 menjadi 4. Dari sisi kebergunaan, fine-tuned model mencapai skor rata-rata 3, lebih baik dibandingkan base model dengan 2.02. Untuk akurasi, meskipun performa masih rendah, fine-tuned model tetap unggul dengan rata-rata 2.38 dibandingkan 1.62 pada base model. Secara keseluruhan, skor akhir rata-rata meningkat dari 2.09 menjadi 2.95 setelah fine-tuning.

Evaluasi menggunakan BERTScore juga menunjukkan bahwa fine-tuned model memiliki performa yang lebih baik dibandingkan base model. Rata-rata precision meningkat dari 0.645 menjadi 0.825, recall dari 0.679 menjadi 0.813, serta F1 score dari 0.659 menjadi 0.818, yang mengindikasikan bahwa jawaban model memiliki kesamaan semantik yang tinggi dengan jawaban referensi.

3. Rata-rata skor evaluasi menggunakan GPT-4 yang mencapai 2.95 serta skor F1 BERTScore sebesar 0.818 menunjukkan bahwa meskipun model menghasilkan jawaban yang relevan dan sesuai konteks secara semantik, model belum mampu menghasilkan angka atau fakta kuantitatif yang akurat sesuai data referensi pada dataset annual report perusahaan. Hal ini terlihat dari perbedaan yang signifikan antara nilai BERTScore yang menekankan kesamaan semantik, sementara GPT-4 mengevaluasi dimensi keramahan, keringkasan, kebergunaan, dan akurasi yang lebih ketat terhadap kebenaran faktual.

5.2 Saran

Berdasarkan hasil penelitian dan tinjauan literatur terkait, berikut adalah beberapa saran yang dapat dipertimbangkan untuk penelitian selanjutnya:

1. Fine-tuning knowledge tertentu ke dalam LLM memerlukan strategi khusus. Penggunaan model dengan jumlah pre-training token yang lebih besar tidak selalu efektif dalam mengurangi halusinasi [41]. Oleh karena itu, disarankan

untuk mengombinasikan fine-tuning dengan metode lain, seperti Retrieval-Augmented Generation (RAG). RAG dapat membantu mengurangi kesalahan konten dengan meningkatkan kemampuan model melalui data eksternal, namun implementasi RAG memerlukan sistem retrieval yang efektif serta evaluasi yang robust [39], [41], [42]. Selain RAG, penggunaan chain-of-thought prompting juga dapat dipertimbangkan untuk menurunkan tingkat halusinasi pada model [40]. Kombinasi continuous pre-training, supervised fine-tuning (SFT), dan reinforcement learning seperti Direct Preference Optimization (DPO) dapat dipertimbangkan untuk meningkatkan performa model [49].

2. Metode lain yang dapat diterapkan untuk mengurangi halusinasi adalah dengan melakukan optimasi dan regularisasi model selama proses training [41].
3. Peningkatan kualitas instruksi (instruction improvement) dalam proses fine-tuning juga dapat membantu mengurangi halusinasi pada output model. [41].
4. Mengingat jumlah sampel dalam dataset berpengaruh pada performa LLM, maka disarankan untuk melakukan data augmentation, misalnya dengan teknik parafrase atau back translation, guna menambah variasi data dan meningkatkan generalisasi model.
5. Domain similarity memiliki pengaruh besar terhadap performa LLM. Oleh karena itu, disarankan untuk menggunakan pre-trained model yang telah di-fine-tune pada domain serupa dengan dataset penelitian. Dalam konteks annual report yang banyak memuat informasi finansial, dapat dipertimbangkan untuk menggunakan model LLM dengan spesialisasi pada domain keuangan.
6. Berdasarkan temuan penelitian ini, konfigurasi training steps berpengaruh pada performa model, di mana pada konfigurasi terbaik, performa model telah mencapai plateau pada step keseratus. Oleh karena itu, diperlukan eksplorasi lebih lanjut terkait konfigurasi hyperparameter untuk memperoleh hasil yang lebih optimal.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

[1] M. H. Jarrahi, D. Askay, A. Eshraghi, and P. Smith, "Artificial intelligence and knowledge management: A partnership between human and AI," *Bus Horiz*, vol. 66, no. 1, pp. 87–99, Jan. 2023, doi: 10.1016/j.bushor.2022.03.002.

[2] S. Meyer, S. Singh, B. Tam, C. Ton, and A. Ren, "A Comparison of LLM Finetuning Methods & Evaluation Metrics with Travel Chatbot Use Case," *arXiv preprint*, Aug. 2024, [Online]. Available: http://arxiv.org/abs/2408.03562

[3] A. Balaguer *et al.*, "RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture," *arXiv preprint*, Jan. 2024, [Online]. Available: http://arxiv.org/abs/2401.08406

[4] J. Mathav Raj, V. Kushala, H. Warrier, and Y. Gupta, "Fine Tuning LLM for Enterprise: Practical Guidelines and Recommendations," *arXiv preprint*, Mar. 2024, [Online]. Available: http://arxiv.org/abs/2404.10779

[5] IBM, "What is knowledge management?" [Online]. Available: https://www.ibm.com/think/topics/knowledge-management

[6] IBM, "What is a chatbot?" [Online]. Available: https://www.ibm.com/topics/chatbots

[7] T. Amaratunga, *Understanding Large Language Models*. Apress, 2023. doi: 10.1007/979-8-8688-0017-7.

[8] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd ed. 2025. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/

[9] A. Vaswani *et al.*, "Attention Is All You Need," *Adv Neural Inf Process Syst*, vol. 30, 2017.

[10] S. Ozdemir, *Quick Start Guide to Large Language Models: Strategies and Best Practices for using ChatGPT and Other LLMs*. Addison-Wesley Professional, 2023.

[11] C. Jeong, “Fine-tuning and Utilization Methods of Domain-specific LLMs,” *arXiv preprint arXiv:2401.02981*, 2024.

[12] V. B. Parthasarathy, A. Zafar, A. Khan, and A. Shahid, “The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities,” *arXiv preprint arXiv:2408.13296*, Aug. 2024, [Online]. Available: http://arxiv.org/abs/2408.13296

[13] “meta-llama/Llama-3.1-8B.” Accessed: Jul. 27, 2025. [Online]. Available: https://huggingface.co/meta-llama/Llama-3.1-8B

[14] Turing.com, “Fine-Tuning LLMs : Overview, Methods, and Best Practices.” [Online]. Available: https://www.turing.com/resources/finetuning-large-language-models

[15] Z. Zhang *et al.*, “Quantized Side Tuning: Fast and Memory-Efficient Tuning of Quantized Large Language Models,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 1–17. [Online]. Available: https://github.com/YouAreSpecialToMe/QST

[16] S. A. Ahsan, “Developing a Cybersecurity Domain Chatbot based on an Open Source Large Language Model,” Master Thesis, University of South Bohemia and Deggendorf Institute of Technology, 2024.

[17] E. J. Hu *et al.*, “LoRA: Low-Rank Adaptation of Large Language Models,” *ICLR*, vol. 1, no. 2, Jun. 2022, [Online]. Available: http://arxiv.org/abs/2106.09685

[18] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “QLORA: Efficient Finetuning of Quantized LLMs,” *Adv Neural Inf Process Syst*, vol. 36, 2023, [Online]. Available: https://github.com/TimDettmers/bitsandbytes

[19] Daniel Han, “Introducing Unsloth: 30x faster LLM training,” https://unsloth.ai/introducing.

[20] R. Collobert, K. Kavukcuoglu, and C. Farabet, “Torch7: A matlab-like environment for machine learning,” in *BigLearn, NIPS workshop*, 2011.

[21] Hugging Face, “TRL - Transformer Reinforcement Learning,” https://huggingface.co/docs/trl/en/index.

[22] Hugging Face, “Transformers,” https://huggingface.co/docs/transformers/en/index.

[23] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating Text Generation with BERT,” *arXiv preprint arXiv:1904.09675*, Apr. 2019, [Online]. Available: http://arxiv.org/abs/1904.09675

[24] V.-T. Doan, Q.-T. Truong, D.-V. Nguyen, V.-T. Nguyen, and T.-N. N. Luu, "Efficient Finetuning Large Language Models For Vietnamese Chatbot," in *2023 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, IEEE, Sep. 2023, pp. 1–6.

[25] T. F. Tan *et al.*, "Fine-tuning Large Language Model (LLM) Artificial Intelligence Chatbots in Ophthalmology and LLM-based evaluation using GPT-4," *arXiv preprint arXiv:2402.10083.*, 2024.

[26] R. Ragab and A. Altahhan, "Fine-tuning of Small/medium LLMs for Business Qa on Structured Data," *SSRN 4850031*, 2024, [Online]. Available: https://ssrn.com/abstract=4850031

[27] S. R. R. Pushparaj, "Empowering Emotional Support Chatbots with Large Language Models," Doctoral dissertation, The State University of New York, 2024.

[28] O. C. Wijaya and A. Purwarianti, "An Interactive Question-Answering System using Large Language Model and Retrieval-Augmented Generation in An Intelligent Tutoring System on the Programming Domain," in *2024 11th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA)*, IEEE, 2024. doi: https://doi.org/10.1109/ICAICTA63815.2024.10763263.

[29] A. T. U. BR. Lubis, N. S. Harahap, S. Agustian, M. Irsyad, and I. Afrianty, "Question Answering System pada Chatbot Telegram Menggunakan Large Language Models (LLM) dan Langchain (Studi Kasus UU Kesehatan)," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 3, pp. 955–964, May 2024, doi: 10.57152/malcom.v4i3.1378.

[30] J. Hakkarainen, "Fine-tuning an open source chatbot to translate code from Python to Java using Qlora: translating for more energy efficient code," Master's thesis, Lappeenranta–Lahti University of Technology LUT, 2024.

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

[32] S. Raschka, "Finetuning LLMs with LoRA and QLoRA: Insights from Hundreds of Experiments," https://lightning.ai/pages/community/lora-insights/.

[33] D. Banerjee, P. Singh, A. Avadhanam, and S. Srivastava, "Benchmarking LLM powered Chatbots: Methods and Metrics," *arXiv preprint arXiv:2308.04624*, Aug. 2023, [Online]. Available: http://arxiv.org/abs/2308.04624

[34] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1715–1725.

[35] T. Kudo, “Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 66–75.

[36] Y. Susanto *et al.*, “SEA-HELM: Southeast asian holistic evaluation of language models,” *arXiv preprint arXiv:2502.14301*, 2025.

[37] Sahabat-AI, “Sahabat-AI Leaderboard,” May 2025.

[38] A. Grattafiori *et al.*, “The llama 3 herd of models,” *arXiv preprint arXiv:2407.21783*, 2024.

[39] J. Yuan *et al.*, “Towards a holistic evaluation of llms on factual knowledge recall,” *arXiv preprint arXiv:2404.16164*, 2024.

[40] T. Cao, N. Raman, D. Dervovic, and C. Tan, “Characterizing multimodal long-form summarization: A case study on financial reports,” in *COLM 2024*, 2024.

[41] J. Li *et al.*, “The Dawn After the Dark: An Empirical Study on Factuality Hallucination in Large Language Models,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 10879–10899.

[42] L. Huang *et al.*, “A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions,” *ACM Trans Inf Syst*, vol. 43, no. 2, pp. 1–55, 2025.

[43] I. Augenstein *et al.*, “Factuality challenges in the era of large language models,” *Nat Mach Intell*, vol. 6, pp. 852–863, 2024.

[44] C. Kang and J. Choi, “Impact of co-occurrence on factual knowledge of large language models,” in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 7721–7735.

[45] W. Orr and K. Crawford, “Building better datasets: Seven recommendations for responsible design from dataset creators,” *Journal of Data-centric Machine Learning Research*, 2024.

[46] H. Mehrafarin, S. Rajaee, and M. T. Pilehvar, “On the Importance of Data Size in Probing Fine-tuned Models,” in *Findings of the Association for Computational Linguistics: ACL 2022*, 2022, pp. 228–238.

[47] E. Khiu *et al.*, “Predicting Machine Translation Performance on Low-Resource Languages: The Role of Domain Similarity,” in *Findings of the*

Association for Computational Linguistics: EACL 2024, 2024, pp. 1474–1486.

[48] Z. Liu, N. Venkateswaran, É. Le Ferrand, and E. Prud'hommeaux, "How important is a language model for low-resource ASR?," in *Findings of the Association for Computational Linguistics: ACL 2024*, 2024, pp. 206–213.

[49] G. Balaskas, H. Papadopoulos, D. Pappa, Q. Loisel, and S. Chastin, "A Framework for Domain-Specific Dataset Creation and Adaptation of Large Language Models," *Computers*, vol. 14, no. 5, p. 172, 2025.