PENERAPAN *FINE-TUNING* LLAMA 3.1 8B MENGGUNAKAN UNSLOTH DAN TEKNIK QLORA TERHADAP DETEKSI ULASAN PALSU PRODUK *E-COMMERCE*

(Skripsi)

Oleh

NICHOLAS VITTO ADRIANTO
NPM 2217051024

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

PENERAPAN *FINE-TUNING* LLAMA 3.1 8B MENGGUNAKAN UNSLOTH DAN TEKNIK QLORA TERHADAP DETEKSI ULASAN PALSU PRODUK *E-COMMERCE*

Oleh

NICHOLAS VITTO ADRIANTO

Skripsi

Sebagai Salah Satu Syarat untuk Mendapatkan Gelar
Sarjana Komputer

Pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

ABSTRAK

PENERAPAN *FINE-TUNING* LLAMA 3.1 8B MENGGUNAKAN UNSLOTH DAN TEKNIK QLORA TERHADAP DETEKSI ULASAN PALSU PRODUK *E-COMMERCE*

Oleh

Nicholas Vitto Adrianto

Ulasan palsu pada platform *e-commerce* merusak kepercayaan konsumen dan mendistorsi integritas pasar. Penelitian ini bertujuan mengembangkan sistem deteksi ulasan palsu yang akurat dan efisien secara komputasi menggunakan *Large Language Model* (LLM) Llama 3.1 8B. Metode yang diterapkan adalah *fine-tuning* menggunakan kerangka kerja Unsloth dan teknik *Quantized Low-Rank Adaptation* (QLoRA) dengan kuantisasi 4-bit untuk menghemat memori. Menggunakan *dataset* ulasan Amazon yang terdiri dari 40.000 data seimbang antara ulasan asli dan buatan komputer, model dilatih dengan memperbarui hanya 0,52% parameter model. Hasil evaluasi menunjukkan kinerja model yang sangat tinggi dengan akurasi 99,47%, *precision* 99,27%, *recall* 99,66%, dan *F1-score* 99,47%. Secara teknis, implementasi ini berhasil menekan penggunaan memori GPU hingga puncaknya hanya 8,98 GB. Namun, analisis *loss* menunjukkan terjadinya *overfitting* setelah *epoch* kedua, yang mengindikasikan bahwa pelatihan optimal cukup dilakukan dalam dua siklus. Penelitian ini membuktikan bahwa Llama 3.1 8B dengan teknik QLoRA merupakan solusi efektif untuk mendeteksi ulasan palsu pada infrastruktur terbatas.

Kata kunci: Deteksi ulasan palsu, *E-commerce*, *Large language model*, Llama 3.1, *Fine-tuning*, QLoRA, Unsloth.

ABSTRACT

APPLICATION OF LLAMA 3.1 8B FINE-TUNING USING UNSLOTH AND QLORA TECHNIQUE IN E-COMMERCE PRODUCT FAKE REVIEW DETECTION

By

Nicholas Vitto Adrianto

Fake reviews on e-commerce platforms damage consumer trust and distort market integrity. This study aims to develop an accurate and computationally efficient fake review detection system using the Llama 3.1 8B Large Language Model (LLM). The method employed is fine-tuning using the Unsloth framework and Quantized Low-Rank Adaptation (QLoRA) technique with 4-bit quantization to conserve memory. Using an Amazon review dataset consisting of 40,000 balanced original and computer-generated reviews, the model was trained by updating only 0.52% of the parameters. Evaluation results demonstrated superior performance with 99.47% accuracy, 99.27% precision, 99.66% recall, and 99.47% F1-score. Technically, this implementation successfully reduced peak GPU memory usage to only 8.98 GB. However, loss analysis indicated overfitting after the second epoch, suggesting that optimal training requires only two cycles. This study proves that Llama 3.1 8B with the QLoRA technique is an effective solution for detecting fake reviews on limited infrastructure.

Keywords: Fake review detection, E-commerce, Large language model, Llama 3.1, Fine-tuning, QLoRA, Unsloth.

Judul Skripsi	: **PENERAPAN FINE-TUNING LLAMA 3.1 8B MENGGUNAKAN UNSLOTH DAN TEKNIK QLORA TERHADAP DETEKSI ULASAN PALSU PRODUK E-COMMERCE**
Nama Mahasiswa	: Nicholas Vitto Adrianto
Nomor Pokok Mahasiswa	: 2217051024
Program Studi	: S1 Ilmu Komputer
Jurusan	: Ilmu Komputer
Fakultas	: Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

Dr. rer. nat. Akmal Junaidi, M.Sc.	**Sandi Badiwibowo Atim, M.Kom.**
NIP. 19710129 199702 1 001	NIP. 19900603 2024061 002

2. Mengetahui

Ketua Jurusan Ilmu Komputer	Ketua Program Studi S1 Ilmu Komputer
Dwi Sakethi, S.Si., M.Kom.	**Tristiyanto, S.Kom., M.I.S., Ph.D.**
NIP. 19680611 199802 1 001	NIP. 19810414 200501 1 001

MENGESAHKAN

1. Tim Penguji

Ketua : **Dr. rer. nat. Akmal Junaidi, M.Sc.** .....

Sekretaris : **Sandi Badiwibowo Atim, M.Kom.** .....

Penguji Utama : **Tristiyanto, S.Kom., M.I.S., Ph.D.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 19711001 200501 1 002

Tanggal Lulus Ujian Skripsi: **18 Februari 2026**

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Nicholas Vitto Adrianto

NPM : 2217051024

Menyatakan bahwa skripsi saya yang berjudul **"Penerapan *Fine-Tuning* Llama 3.1 8B Menggunakan Unsloth dan Teknil QLoRA Terhadap Deteksi Ulasan Palsu Produk E-commerce"** merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya tulis ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil jiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang saya terima.

Bandar Lampung, 20 Februari 2026

Penulis,

Nicholas Vitto Adrianto

NPM. 2217051024

RIWAYAT HIDUP

Penulis dilahirkan di Kota Bandar Lampung pada tanggal 31 Desember 2003 sebagai anak kedua dari dua bersaudara dari Bapak Desiderius Dwipujo Priyanto dan Ibu Metty Maria Goretti. Penulis telah menyelesaikan pendidikan formal di SD Xaverius 3 Way Halim di Bandar Lampung pada tahun 2016. Kemudian SMP Xaverius 4 Way Halim di Bandar Lampung pada tahun 2019, dan SMA Xaverius Bandar Lampung pada tahun 2022. Pada tahun yang sama, penulis terdaftar sebagai mahasiswa program studi S1 Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung melalui jalur *undangan* atau SNMPTN (sekarang SNBP).

Selama menjadi mahasiswa di Ilmu Komputer, penulis aktif dalam berbagai aktivitas baik di dalam maupun di luar Universitas Lampung. Aktivitas yang dilakukan adalah sebagai berikut:

1. Menjadi anggota dari Keluarga Mahasiswa Katolik di FMIPA dan Universitas Lampung dan mengikuti kegiatan dan pembinaan rohani.
2. Menjadi asisten dosen untuk mata kuliah Matematika, Struktur Data dan Algoritma, Pemrograman Interpreter, dan Pemrograman Deklaratif di Jurusan Ilmu Komputer.
3. Menjadi anggota dari UKM English Society Universitas Lampung dalam cabang *newscasting* dan mendapatkan Juara 1 di Lomba *Newscasting* di *English Olympic 2024* yang diselenggarakan oleh Politeknik Negeri Ujung Pandang.

4. Menjadi penerima beasiswa Bank Indonesia pada tahun 2024, dan menjadi anggota divisi Lingkungan Hidup (LH) di organisasi Generasi Baru Indonesia (GenBI) Universitas Lampung.
5. Melaksanakan kegiatan Kuliah Kerja Nyata (KKN) di Desa Rukti Harjo, Kecamatan Seputih Raman, Kabupaten Lampung Tengah selama 32 hari dari 6 Januari 2025 sampai 6 Februari 2025 untuk melaksanakan kegiatan pengabdian kepada masyarakat.
6. Melaksanakan kegiatan Kerja Praktik di Dinas Perindustrian dan Perdagangan Provinsi Lampung selama 40 hari dari 16 Juni 2025 sampai 1 Agustus 2025.

MOTTO

Everything has to be perfect!

(Gumball Watterson)

PERSEMBAHAN

Kepada Tuhan Yang Maha Esa,

Yang telah memberikan berkat dan karunia-Nya kepada penulis untuk menyelesaikan skripsi ini.

Kedua Orang Tuaku, dan Kakak yang Tercinta

Yang telah mendukung, mendoakan, dan menjadi motivator penulis dalam setiap langkah kehidupan. Terima kasih telah selalu mempercayai saya untuk menyelesaikan skripsi ini.

Almamater Tercinta, Universitas Lampung.

SANWACANA

Puji syukur kepada Tuhan Yang Maha Esa atas berkat dan anugerah-Nya, penulis dapat menyelesaikan skripsi dengan judul **"Penerapan *Fine-Tuning* Llama 3.1 8B Menggunakan Unsloth dan Teknik QLoRA Terhadap Deteksi Ulasan Palsu Produk *E-commerce*"**. Skripsi ini merupakan salah satu syarat untuk menyelesaikan perkuliahan program studi S1 Ilmu Komputer dan mendapatkan gelar Sarjana Komputer di Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.

Penyelesaian skripsi ini melibatkan berbagai pihak melalui bimbingan, saran, dan dukungan kepada penulis. Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada semua pihak yang terlibat dalam proses penyusunan skripsi ini, antara lain:

1. Kedua orang tua saya, kakak, dan keluarga besar saya yang telah memberikan dukungan, doa, dan membimbing penulis di setiap langkah kehidupan.
2. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., I.P.M., selaku Rektor Universitas Lampung.
3. Bapak Dr. Eng. Heri Satria, S.Si., M.Si., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
4. Bapak Dwi Sakethi, S.Si., M.Kom., selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung, sekaligus Dosen Pembimbing Akademik yang selalu memberikan dukungan dan bimbingan kepada penulis selama berkuliah di Jurusan Ilmu Komputer, Universitas Lampung.
5. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc., selaku Dosen Pembimbing Utama atas keikhlasannya untuk membantu penulis dalam penyusunan skripsi dengan sabar, memberikan saran, kritik, pengarahan, dan ilmu yang bermanfaat.

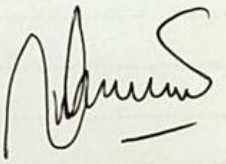
6. Bapak Sandi Badiwibowo Atim, M.Kom., selaku Dosen Pembimbing Pembantu atas kebaikannya dalam memberikan bimbingan, saran, masukan, dan ilmu yang bermanfaat kepada penulis untuk menyelesaikan skripsi dengan lancar.
7. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D., selaku Ketua Program Studi S1 Ilmu Komputer dan Dosen Pembahas Utama yang telah memberikan ide selama penulis menentukan tema skripsi, dan juga memberikan saran dan masukan kepada penulis guna untuk menyempurnakan penulisan skripsi.
8. Ibu Yunda Heningtyas, M.Kom., selaku Sekretaris Jurusan Ilmu Komputer yang telah membantu penulis dalam urusan akademik dan administrasi selama perkuliahan.
9. Ibu Ade Nora Maela, selaku Administrator Jurusan Ilmu Komputer yang membantu penulis dalam urusan administrasi selama di Jurusan Ilmu Komputer.
10. Seluruh dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu yang bermanfaat dan pengalaman yang baru kepada penulis.
11. Kakak-kakak dan teman-teman Keluarga Mahasiswa Katolik Universitas Lampung dan FMIPA yang telah membina iman saya dan mengikuti kegiatan-kegiatan rohani yang seru.
12. Kakak-kakak dan teman-teman di UKM English Society Universitas Lampung yang telah melatih saya selama berada di organisasi untuk mempersiapakan penulis dalam perlombaan di cabang *newscasting*.
13. Kakak-kakak dan teman-teman penerima Beasiswa Bank Indonesia 2024 dan Generasi Baru Indonesia dan yang telah membantu penulis dalam mengembangkan diri penulis untuk bersiap untuk menciptakan seorang pemimpin masa depan.
14. Bapak/Ibu dan segenap pegawai di tempat Kerja Praktik saya, Dinas Perindustrian dan Perdagangan Provinsi Lampung, yang telah membimbing saya selama pelaksanaan kerja praktik untuk pengalaman kerja yang berguna.
15. Rekan-rekan sekelompok kerja praktik: Fahreza, Yusuf, David, Firman, dan Joe yang saling mendukung selama pelaksanaan kerja praktik.

16. Keluarga KKN Desa Rukti Harjo Kelompok 2: Billy, Cynthia, Tangguh, Kak Yulia, Amel, dan Audy, yang membuat pengalaman yang tak terlupakan.
17. Teman-teman Jurusan Ilmu Komputer FMIPA Universitas Lampung angkatan 2022 yang ikut serta dalam perjuangan perkuliahan dari awal hingga akhir.
18. Dan terakhir, untuk semua pihak yang tidak bisa disebutkan satu per satu, atas kontribusi yang bermanfaat dalam menyelesaikan skripsi ini.

Penulis menyadari bahwa masih terdapat banyak ketidaksempurnaan dalam penulisan skripsi ini. Maka dari itu, penulis menerima secara terbuka kritik dan saran yang membangun sebagai pembelajaran untuk kedepannya. Penulis berharap semoga skripsi ini bermanfaat bagi pihak yang terlibat dan dunia IT yang masih berkembang.

Bandar Lampung, 20 Februari 2026

Penulis,



Nicholas Vitto Adrianto
NPM. 2217051024

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR KODE PROGRAM

Halaman

I. PENDAHULUAN

1.1 Latar Belakang & Masalah

Perkembangan teknologi telah berkembang pesat di berbagai belahan dunia. Internet menjadi salah satu teknologi yang sudah ada di berbagai kalangan masyarakat, baik itu di perkotaan maupun perdesaan. Menurut laporan dari We Are Social terhadap dunia digital tahun lalu (2025), sebanyak 6,04 miliar penduduk di dunia merupakan pengguna internet, atau 73,2% dari populasi dunia. Pada tahun ini, angka pengguna internet baru menembus 6 miliar dari 8 miliar penduduk dunia, di mana tahun-tahun sebelumnya masih berjumlah dibawahnya. Sejak berdirinya *world wide web*, muncul berbagai jenis platform internet yang dapat kita gunakan. Baik itu media sosial, platform *messaging,* surat elektronik, dan masih banyak lagi. Tetapi ada satu jenis internet yang sedang menjadi tren pada saat ini, yaitu *e-commerce*. Platform *e-commerce* menyediakan layanan transaksi jual beli barang atau jasa dengan perantaraan jaringan internet (Syaifudin *et al*., 2025). Berdasarkan laporan dunia digital tahun 2025, 74,5% pengguna menggunakan internet untuk melakukan transaksi perbelanjaan di platform *e-commerce* (We Are Social, 2025).

Pertumbuhan platform *e-commerce* merupakan hal yang menarik perhatian pengguna internet karena transformasi digital serta strategi pemasaran yang lebih modern membuat segala transaksi bisa dilakukan di internet (Syaifudin *et al*., 2025). Selain itu, pandemi COVID-19 telah mendorong masyarakat untuk melakukan pembelanjaan secara online akibat dari kebijakan *physical distancing* pada saat itu di mana pembelanjaan secara langsung (luring) dibatasi atau tutup (Dewi & Firdaus, 2024). Berbagai platform *e-commerce* telah muncul, baik itu platform global seperti Alibaba*,* Amazon, dan eBay maupun platform di Indonesia

seperti Tokopedia dan Shopee. Amazon merupakan salah satu platform *e-commerce* global modern terbesar yang menjual layanan dan produk-produk seperti *e-book*, barang pokok, sekunder, tersier, dan lain sebagainya. Laporan peringkat dari Similarweb menempatkan Amazon sebagai platform website paling banyak dikunjungi ke-15 dengan 2,71 miliar kunjungan, sedangkan laporan dari Semrush menempatkannya sebagai website paling banyak dikunjungi ke-13 dengan 2,67 miliar kunjungan (We Are Social, 2025).

Sebagai platform global yang terpercaya dan andal, Amazon menjadi pilar utama dalam ritel modern untuk penjualan berbagai produk dan layanan. Untuk mempertahankan kepercayaan konsumen yang tinggi, ulasan produk atau layanan akan menjadi salah satu faktor yang meningkatkan tingkat kepercayaan antara konsumen, dan produk. Berdasarkan sebuah statistik terkait ulasan pada platform *e-commerce,* 93% calon konsumen membaca ulasan sebelum membeli produk atau layanan serta menyatakan bahwa ulasan dapat memengaruhi keputusan pembelian tersebut (Chen *et al*., 2022). Sayangnya, karena ketergantungan ini, ulasan dapat dieksploitasi untuk suatu keuntungan atau kerugian terhadap penjual atau platform *e-commerce* dengan maraknya ulasan palsu. Ulasan palsu, baik yang positif maupun negatif secara langsung memengaruhi daya beli konsumen dan mendistorsi pasar (Clarisa & Areta A, 2022). Menurut penelitian yang dilakukan oleh CapitalOne Research terhadap ulasan palsu (2025), secara global, ulasan palsu diproyeksikan akan merugikan konsumen online sekitar $770.7 miliar pada tahun 2025.

Bagi penjual yang sah di platform *e-commerce*, dampaknya akan sangat pengaruh terhadap mereka dan juga platform tersebut. Seorang penjual anonim melaporkan produknya "diserang secara kejam" oleh berbagai ulasan palsu, mengakibatkan rata-rata peringkat bintangnya jatuh dari 4.6 menjadi 4.1 hanya dalam waktu tiga minggu (Anonim, 2025). Platform *e-commerce* telah melakukan upaya tindakan untuk menghindari ancaman ulasan palsu untuk meluas. Amazon bekerja sama dengan *Better Business Bureau* (BBB) dalam upaya untuk melawan ulasan palsu dengan menggugat para penanggung jawab atas ulasan palsu ini (Amazon, 2025). Dan bahkan Amazon sendiri juga menggunakan teknologi ML untuk mencegah ulasan palsu untuk memasuki platform mereka dengan cara mendeteksi ulasan palsu dan mampu menghapus 275 juta ulasan palsu (Amazon, 2024). Meskipun

demikian, keberadaan ulasan palsu tetap meraja lela karena semakin lama, ulasan palsu akan menjadi lebih sulit dideteksi oleh sistem atau bahkan model LLM yang digunakan. Sebuah penelitian menemukan bahwa rata-rata 30% ulasan online dianggap palsu, dan terdapat 47% produk terlaris di platform *e-commerce* Amazon memiliki ulasan yang tidak dapat diandalkan (CapitalOne Shopping, 2025).

Deteksi ulasan palsu adalah sebuah teknik untuk menghindari ancaman hilangnya kepercayaan konsumen. Hal ini semakin lama semakin berkembang akibat dari munculnya metode-metode. Metode deteksi awal menggunakan teknik machine learning (ML) tradisional seperti SVM*, Naive Bayes,* dan *Random Forest*, lalu berkembang menjadi metode *deep learning* seperti CNN, LSTM, dan transformer *models* Menurut penelitian Mir *et al.* (2023), metode SVM, *Naive Bayes*, k-*Nearest Neighbors* (k-NN) dan *Random Forest* digunakan untuk model klasifikasi, dan menemukan bahwa SVM mempunyai akurasi 87,81%, *Naive Bayes* mempunyai akurasi 78,43%, k-NN mempunyai akurasi 77,18%, dan *Random Forest* mempunyai akurasi 83,43%, sehingga SVM mempunyai tingkat akurasi tertinggi.

Tetapi saat ini, terdapat perkembangan kemajuan teknologi dari *Natural Language Processing* (NLP) yang paling terbaru yaitu *Large Language Model* (LLM). Papageorgiou *et al.* (2024) menemukan bahwa model ini kini dapat mendeteksi dan bahkan menghasilkan berita palsu (*hoax*) yang mendekati bahasa alami, yang sering kali tidak dapat dibedakan oleh manusia atau pendeteksi sekarang. Dan saat ini juga, pendeteksi berita palsu telah mengalami perkembangan yang mengesankan. Maka dari itu, penelitian ini dapat menerapkan sebuah model LLM yang dapat mendeteksi ulasan palsu pada sebuah produk *e-commerce.* Berbagai *pre-trained model* LLM telah muncul, seperti *Generative Pre-trained Transformer* (GPT) yang sering digunakan dalam berbagai aplikasi seperti *chatbot ChatGPT*. GPT-4 mampu memahami dan menjawab pertanyaan dari *input* teks dan gambar serta mendorong pemahaman informasi yang lebih rinci dan komprehensif. Sejak itu, berbagai model LLM baru dapat memahami konteks dan pertanyaan sekaligus efisiensi memori dan kinerja agar dapat digunakan oleh banyak orang untuk kegunaan mereka.

Penelitian ini mengusulkan untuk mengambil langkah berikutnya, yaitu memanfaatkan salah satu LLM *open source* paling canggih dan efisien saat ini,

yaitu Meta Llama 3.1 dengan parameter 8 miliar. Model ini menawarkan keseimbangan kuat antara kemampuan dan efisiensi, dengan konteks 128K yang diperluas dan kemampuan multibahasa (Grattafiori *et al*., 2024). Namun, *fine-tuning* LLM, terutama yang berparameter besar membutuhkan sumber daya komputasi, khususnya memori RAM, GPU, dan penyimpanan yang besar dan mahal (Shafelbilyunazra & Prasetya, 2025).

Untuk mengatasi tantangan tersebut, penelitian ini akan mengimplementasikan pustaka Unsloth. Unsloth adalah perangkat optimasi yang secara drastis mengurangi penggunaan memori hingga 70-90% dan mempercepat pelatihan dua sampai lima kali dibandingkan metode standar, tanpa kehilangan akurasi. Unsloth telah mendukung beberapa model LLM, seperti Llama, Qwen3, Deepseek, *Gemma*, dan beberapa model lainnya (Han *et al*., 2023). Penelitian ini juga menggunakan teknik *Quantized Low-Rank Adaptation* (QLoRA), yang merupakan perkembangan dari proses *Low-Rank Adaptation (*LoRA*)*. QLoRA memanfaatkan *fine-tuning* pada model LLM dengan kuantisasi 4-bit untuk menghemat penggunaan memori GPU (Dettmers *et al*., 2023). Penelitian ini bertujuan untuk mengembangkan sistem deteksi ulasan palsu yang akurat dan efisien secara komputasi, sehingga layak untuk diterapkan dalam skala besar.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, adapun rumusan pertanyaan yang menjadi masalah pokok pada penelitian ini, yaitu:

a. Bagaimana melakukan *fine-tuning* terhadap *Large Language Model* (LLM) Llama 3.1 8B menggunakan Unsloth dan teknik QLoRA untuk mendeteksi ulasan produk palsu pada platform *e-commerce* dengan menghemat sumber daya komputasi?
b. Bagaimana mengevaluasi model *Large Language Model* (LLM) Llama 3.1 yang telah di-*fine-tuned* untuk mendeteksi ulasan produk palsu pada platform *e-commerce*?

1.3 Batasan Masalah

Untuk memastikan bahwa penelitian ini tidak meluas dan tetap fokus terhadap tujuan yang tercapai, adapun beberapa batasan ruang lingkup penelitian harus dibuat, yaitu:

a. Karena bahasa yang didukung oleh model LLM tidak termasuk bahasa Indonesia, *dataset* yang digunakan berasal dari Kaggle merupakan *dataset* ulasan produk pada platform *e-commerce* Amazon berbahasa Inggris dengan jumlah 40 ribu data, dengan rasio 50:50 data ulasan asli dan palsu (*computer generated*).
b. Model LLM yang digunakan adalah Llama versi 3.1 dengan parameter 8 miliar kata yang telah di-*fine-tuned* untuk memberikan *instruction* khusus agar dapat menjawab *prompt* lebih baik.
c. Penelitian ini tidak mencakup pembuatan (*generated*) teks ulasan palsu dengan model LLM, hanya mencakup pendeteksi teks ulasan palsu.
d. Penelitian ini akan menggunakan pustaka *Unsloth* untuk mengoptimasikan pemuatan dan pelatihan model LLM, serta menghemat pemakaian memori GPU dan RAM.
e. *Fine-tuning* model LLM menggunakan *Parameter-Efficient Fine-Tuning* (PEFT) dengan teknik *Quantized Low-Rank Adaptation* (QLoRA) untuk mengoptimasi proses *fine-tuning*.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian yang dilakukan ini, yaitu:

a. Membangun dan melakukan *fine-tuning* model Llama 3.1 dengan parameter 8 miliar dengan optimasi *Unsloth* dan teknik *fine-tuning* QLoRA untuk mendeteksi ulasan palsu terhadap produk di platform *e-commerce*.
b. Mengevaluasi dan mengukur kinerja dari model Llama 3.1 berparameter 8 miliar yang telah di-*fine-tuned* dengan optimasi *Unsloth* dan teknik QLoRA terhadap deteksi ulasan palsu terhadap produk di platform *e-commerce*.

1.5 Manfaat Penelitian

Dengan adanya penelitian ini, adapun manfaat yang bisa diberikan, yaitu:

a. Merintis pengguna Llama 3.1 untuk kepentingan pendeteksi ulasan palsu berbasis teknologi NLP, agar dapat digunakan untuk perkembangan *fine-tuning* model Llama dan LLM selanjutnya, khususnya untuk mendeteksi suatu hal yang mencurigakan seperti ulasan palsu, berita palsu, penipuan, dan sebagainya, serta dapat menerapkan teknik pelatihan yang menghemat memori pada komputasi dan waktu pelatihan.
b. Membantu platform *e-commerce* dan bisnis untuk mengimplementasikan deteksi ulasan palsu berbasis LLM dalam platform mereka tanpa memerlukan infrastruktur GPU yang kompleks, agar dapat melindungi integritas ekosistem *e-commerce*, dan meningkatkan kepercayaan konsumen dengan efektif dan efisien.

II. LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian ini tidak terlepas dari penelitian terkait yang membahas deteksi ulasan palsu terhadap platform *e-commerce* yang menjadi sumber referensi sekaligus perbandingan dengan penelitian yang sedang dilakukan saat ini.

Tabel 1. Penelitian terdahulu yang terkait.

Peneliti	Judul Penelitian	Metode	Hasil
Alamsyah *et al*. (2023)	Deteksi *Fake Review* Menggunakan Metode *Support Vector Machine* dan *Naïve Bayes* Di Tokopedia	*SVM* dan *Naive Bayes*	Hasil pengujian menunjukkan bahwa model *SVM* mencapai akurasi 94,38%, sedangkan model *Naive Bayes* mencapai akurasi 91,57%.
Purifyregalia *et al*. (2025)	*Detecting Fake Reviews in E-Commerce: A Case Study on Shopee Using Support Vector Machine and Random Forest*	*SVM* dan *Random Forest*	Hasil eksperimen menunjukkan bahwa model *SVM* unggul dengan mencapai nilai akurasi 88,84% dibandingkan model *Random Forest* dengan akurasi 80,39%

Peneliti	Judul Penelitian	Metode	Hasil
Awalina *et al*. (2022)	Perbandingan *Pretrained Model Transformer* Pada Deteksi Ulasan Palsu	*CNN* dan Model p*re-trained* BERT (ALBERT, RoBERTa, dan DistilBERT)	RoBERTa unggul dengan nilai akurasi 90,8% dibandingkan model lainnya *seperti* ALBERT *dan* DistilBERT serta CNN.
Hadi *et al*. (2025)	*Fake Review Detection on Digital Platforms Using The RoBERTa Model : A Deep Learning and NLP Approach*	Model *fine-tuned* RoBERTa	Model *fine-tuned* RoBERTa mencapai tingkat akurasi sebesar 86,25%, yang unggul atas model *Logistic Regression*, K-NN, dan *LDA*-BPText CNN.
Mohawesh *et al*. (2024)	*Fake Review Detection Using Transformer-Based Enhanced LSTM and RoBERTa*	RoBERTa dan LSTM	Penelitian ini menggabungkan model RoBERTa dengan lapisan LSTM. Model *dataset* OpSpam menghasilkan akurasi 96,03%, sedangkan *dataset* Deception, model menghasilkan akurasi 93,15%.

Berdasarkan Tabel 1 di atas, terdapat beberapa penelitian yang terkait dengan penelitian tentang deteksi ulasan palsu menggunakan model NLP yang ada. Penelitian oleh Alamsyah *et al*. (2023) yang berjudul "Deteksi *Fake Review*

Menggunakan Metode *Support Vector Machine* dan *Naïve Bayes* Di Tokopedia" meneliti pendeteksian ulasan palsu pada *platform e-commerce* Tokopedia, dengan cara membandingkan antara dua model ML, *Support Vector Machine* (SVM) dan *Naïve Bayes*. Penelitian ini memanfaatkan teknik *Latent Dirichlet Allocation* (LDA) untuk melakukan pelabelan data secara otomatis dan TF-IDF untuk fitur ekstraksi. Berdasarkan hasil pengujian, metode SVM terbukti lebih akurat dengan nilai akurasi mencapai 94,38%, sedikit lebih tinggi dibandingkan *Naïve Bayes* yang menghasilkan akurasi sebesar 91,57%, meskipun kedua model dinilai cukup baik dalam mengidentifikasi ulasan palsu.

Penelitian yang dilakukan oleh Purifyregalia *et al*. (2025) berjudul "*Detecting Fake Reviews in E-Commerce: A Case Study on Shopee Using Support Vector Machine and Random Forest*" berfokus pada deteksi ulasan palsu di *e-commerce* Shopee Indonesia, khususnya pada produk akun premium Netflix, dengan menerapkan algoritma *Support Vector Machine* (SVM) dan *Random Forest*. Penelitian ini juga menggunakan metode *Latent Dirichlet Allocation* (LDA) untuk pelabelan data otomatis menjadi kategori *Original* (OR) dan *Computer-Generated* (CG), dan TF-IDF untuk pembobotan kata. Hasil evaluasi menunjukkan bahwa algoritma SVM memiliki performa yang lebih unggul dalam menangani data teks berdimensi tinggi dengan akurasi sebesar 88,84%, dibandingkan dengan *Random Forest* yang memperoleh akurasi 80,39%.

Penelitian oleh Awalina *et al*. (2022) berjudul "Perbandingan *Pretrained Model Transformer* Pada Deteksi Ulasan Palsu" melakukan perbandingan komprehensif antara berbagai model *pre-trained* Transformer (BERT, XLNet, ALBERT, RoBERTa, DistilBERT) dengan pendekatan *fine-tuning* melawan model *Deep Learning* CNN yang menggunakan berbagai *word embedding* untuk mendeteksi ulasan palsu pada *dataset* OpSpam oleh Ott *et al*. (2011). Hasil eksperimen menunjukkan bahwa model RoBERTa mengungguli seluruh model *transformer* dan CNN lainnya dengan mencapai akurasi tertinggi sebesar 90,8%, *precision* 90%, *recall* 91,8%, dan *f1-score* 90,8%. Meskipun RoBERTa memiliki kinerja klasifikasi terbaik, penelitian ini juga menemukan bahwa DistilBERT merupakan model yang paling efisien dari segi waktu komputasi pelatihan, hanya membutuhkan waktu 200,5 detik.

Penelitian oleh Hadi *et al*. (2025) berjudul "*Fake Review Detection on Digital Platforms Using The RoBERTa Model : A Deep Learning and NLP Approach*" mengeksplorasi penggunaan model *Robustly Optimized* BERT *Approach* (RoBERTa) untuk mendeteksi ulasan palsu pada *dataset* OpSpam oleh Ott *et al*. (2011), dengan pendekatan *Deep Learning* dan NLP. Penelitian ini membandingkan kinerja RoBERTa dengan metode lain seperti *Logistic Regression*, K-NN, dan LDA-BPTextCNN, di mana proses pelatihan menunjukkan konvergensi *loss* yang signifikan meskipun validasi *loss* cenderung tidak stabil. Hasil akhirnya menunjukkan bahwa RoBERTa mencatatkan performa terbaik dengan akurasi 86,25%, menegaskan kemampuannya dalam menangkap nuansa linguistik yang kompleks untuk mendeteksi konten manipulatif dibandingkan metode pembanding lainnya.

Penelitian oleh Mohawesh *et al*. (2024) berjudul "*Fake Review Detection Using Transformer-Based Enhanced LSTM and RoBERTa*" mengusulkan model hibrida baru yang menggabungkan arsitektur *transformer* RoBERTa dengan lapisan *Long Short-Term Memory* (LSTM) untuk meningkatkan deteksi ulasan palsu melalui pemahaman semantik dan linguistik yang lebih dalam. Diuji pada *dataset* OpSpam oleh Ott *et al*. (2011) dan Deception oleh Li *et al*. (2014), model *hybrid* ini berhasil mengungguli metode *state-of-the-art* lainnya dengan akurasi yang sangat tinggi, yaitu 96,03% pada *dataset* Ott *et al*. dan 93,15% pada dataset Deception. Selain kinerja tinggi, penelitian ini juga menerapkan teknik *Shapley Additive Explanations* (SHAP) untuk memberikan transparansi dan penjelasan rasional mengenai mengapa sebuah ulasan diklasifikasikan sebagai palsu atau asli.

Penelitian-penelitian di atas menggunakan metode ML dan *deep learning* dalam klasifikasi ulasan palsu. Penelitian dari Purifyregalia *et al*. (2025) dan Alamsyah *et al*. (2023) menggunakan algoritma klasifikasi ML seperti SVM, *Random Forest*, dan *Naïve Bayes*, sedangkan Awalina *et al*. (2022) dan Hadi *et al*. (2025) mengembangkan metode klasifikasi *deep learning* seperti *transformer* RoBERTa, CNN, maupun *Logistic Regression*. Selain itu, *dataset* yang digunakan pada penelitian di atas juga bervariasi. Penelitian dari Purifyregalia *et al*. dan Alamsyah *et al*. menggunakan teknik *web scrapping* untuk mengambil data dari *platform e-commerce* seperti Tokopedia dan Shopee, sedangkan Awalina *et al*. dan Hadi *et al.*

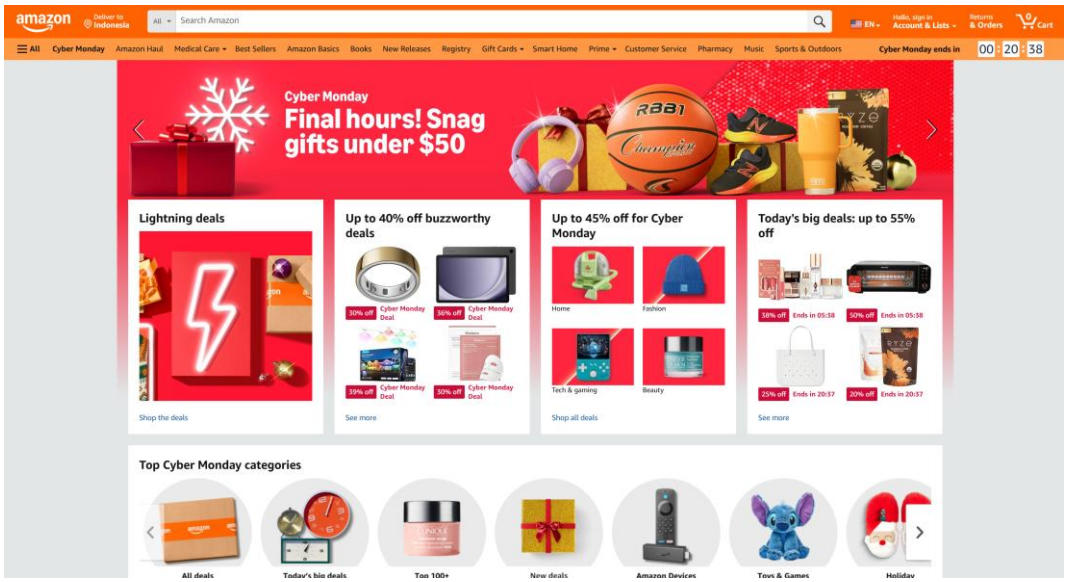
menggunakan *dataset* OpSpam dari Ott *et al*. (2011). Penelitian oleh Mohawesh *et al*. (2024) mengoptimalkan pendekatan ini melalui arsitektur *hybrid* RoBERTa-LSTM untuk menangkap pola semantik yang lebih rumit, serta menggunakan dua *dataset*, yaitu OpSpam oleh Ott *et al*. dan Deception oleh Li *et al*. (2014). Penelitian ini akan menerapkan pembaruan teknologi dengan menerapkan *Large Language Model* (LLM) generatif dari *Meta AI*, *Llama* 3.1 dengan parameter yang sangat besar agar dapat memahami konteks lebih mendalam. Penelitian ini menggunakan *dataset* dari platform *e-commerce* Amazon yang disediakan oleh Salminen *et al*. (2022). Penelitian ini juga menangani permasalahan efisiensi komputasi dan biaya sumber daya komputasi pada saat melatih model yang berat ini, dengan menggunakan pustaka *Unsloth* dan teknik QLoRA untuk menghemat penggunaan memori grafis sebesar 70% sampai 90% dan mempercepat proses *fine-tuning*, menjadikan penerapan teknologi LLM canggih lebih efisien dan efektif untuk deteksi ulasan palsu dibandingkan pendekatan sebelumnya (Han *et al*., 2023).

2.2 *E-commerce*

E-commerce (*electronic commerce*), atau dalam bahasa Indonesia disebut perdagangan elektronik adalah ekosistem transaksi jual beli produk atau jasa yang dilakukan melalui media internet. Perdagangan elektronik bisa dilakukan antara organisasi dan konsumen atau *business to consumer* (B2C), antara kedua organisasi atau *business to business* (B2B), maupun antara kedua konsumen atau *consumer to consumer* (C2C). Platform *e-commerce* telah mendukung transformasi digital dan mengembangkan ekonomi digital terhadap pembelanjaan produk dan jasa di dunia, serta mengubah gaya hidup sosial manusia dari secara fisik menjadi lewat berbagai kanal (Syaifudin *et al*., 2025).

Contoh *platform e-commerce* di dunia ada *Amazon* dan *Alibaba,* sedangkan *platform e-commerce* lokal terdapat Tokopedia, Lazada, dan Shopee. Sayangnya, keberadaan ulasan palsu merusak integritas pasar e-*commerce* dengan mendistorsi persepsi kualitas produk. Bagi mereka, kegagalan mendeteksi ulasan palsu dapat mengakibatkan hilangnya kepercayaan konsumen terhadap penjual dan juga *platform e-commerce* itu sendiri, serta kerugian ekonomi digital pada jangka

panjang. Oleh karena itu, pengembangan sistem deteksi otomatis yang akurat bukan hanya masalah teknis, tetapi imperatif bisnis yang strategis untuk menjaga reputasi ekosistem pasar digital.



Gambar 1. Halaman utama *platform e-commerce* Amazon.

2.3 Ulasan Palsu

Ulasan (*review*) pada *platform e-commerce* merupakan hal yang sangat penting bagi konsumen, apalagi mereka yang ingin membeli produk atau jasa tersebut. Ulasan-ulasan tersebut dapat digunakan oleh calon pembeli sebagai faktor dalam mengambil keputusan terhadap apakah mereka membutuhkan produk atau jasa yang dilihat tersebut. Sayangnya, muncul juga berbagai ulasan palsu bersifat merugikan pembeli dan juga penjual dan dapat menghancurkan reputasi dari platform *e-commerce* (Alamsyah *et al*., 2023).

Ulasan palsu (*fake reviews*) merupakan ulasan yang sengaja ditulis dengan tujuan untuk menipu konsumen *Ulasan palsu* bersifat merugikan karena dapat menurunkan kualitas ulasan secara umum, dan menghilangkan rasa kepercayaan konsumen (Clarisa & Areta A, 2022). Ulasan palsu secara umum dibagi menjadi tiga jenis, ulasan tidak jujur (*untruthful reviews*), ulasan merek (*brand reviews*), dan ulasan yang bukan sama sekali ulasan (*non-reviews*) (Kumar, 2020):

1. ***Untruthful reviews***: Ulasan ini berisi pendapat yang tidak jujur dan disengaja, biasanya bertujuan untuk memuji produk yang buruk atau memperburuk produk-produk pesaing agar produk ini menjadi produk yang satu-satunya terbaik disbanding produk lain.
2. ***Brand reviews***: Ulasan ini hanya berfokus terhadap merek dari produk tersebut secara umum, bukan produk secara khusus, sering kali bersifat promosi berlebihan atau menjelekkan nama merek untuk menguntungkan merek lain.
3. ***Non-reviews***: Ulasan ini tidak berisi ulasan, melainkan iklan atau pendapat yang tidak relevan terhadap produk ataupun merek.

Deteksi ulasan palsu adalah masalah yang menantang karena pembuat ulasan palsu dapat meniru gaya ulasan asli. Untungnya, terdapat beberapa pola yang memungkinkan ulasan-ulasan tersebut bersifat palsu. Penelitian dari Kumar (2020) dan Sun *et al*. (2024) menemukan beberapa karakteristik dari ulasan palsu, yaitu:

1. **Fitur tekstual:** Ulasan palsu sering kali lebih pendek, menggunakan bahasa yang beranggap berlebihan (hiperbola), atau memiliki pola *grammar* yang repetitif. Mereka mungkin kurang mendetail tentang aspek sensorik penggunaan produk dibandingkan ulasan asli.
2. **Fitur perilaku:** Ulasan palsu sering terjadi mengikuti pola waktu pengiriman ulasan. Biasanya terjadi lonjakan ulasan tiba-tiba terhadap suatu produk atau jasa, dan sering kali terjadi perubahan rata-rata peringkat secara signifikan.

Tantangan terkini adalah munculnya ulasan palsu yang dihasilkan oleh AI (*computer-generated fake reviews*). Ulasan ini sering kali memiliki tata bahasa yang sempurna dan narasi yang alami, membuatnya sulit dideteksi oleh metode ML tradisional, sehingga memerlukan kemampuan pemahaman semantik mendalam dari LLM seperti *Llama* 3.1 untuk mendeteksinya.

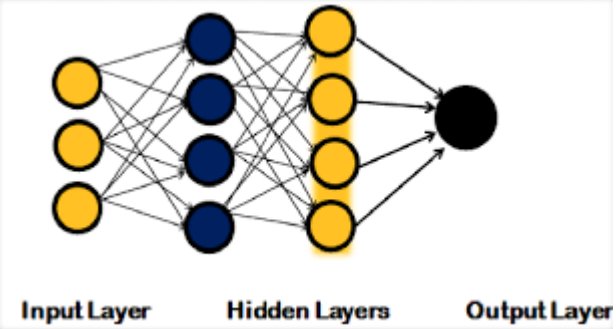
2.4 Klasifikasi

Klasifikasi dalam bidang kecerdasan buatan (AI) merupakan teknik mengelompokkan data dalam *dataset* ke dalam sebuah kategori atau kelas

menggunakan label. Karena label kelas tersebut, klasifikasi termasuk dalam jenis *supervised learning* dalam teknik *machine learning* (ML). *Supervised learning* melatih algoritma untuk mempelajari data dari *dataset* dan target kelas yang sesuai yang berupa label *string.* Salah satu contoh klasifikasi berdasarkan penelitian ini menggunakan teknik klasifikasi untuk membuat komputer membedakan antara ulasan asli dan ulasan palsu. Klasifikasi juga dapat digunakan untuk alat pendeteksi pada *dataset* bermodal gambar atau teks (Santoso, 2023).

Klasifikasi bisa menggunakan metode ML tradisional maupun *deep learning*. Metode ML untuk klasifikasi sudah ada banyak aplikasi dan algoritma yang digunakan. Contoh algoritma ML yang sering digunakan dalam klasifikasi ada *Support Vector Machine* (SVM), *Random Forest*, *Naive Bayes*, dan masih banyak lagi. SVM memanfaatkan *hyperplane* yang memisahkan antar dua kelas sebagai batas keputusan (*decision boundary*), sedangkan *Naive Bayes* menggunakan teorema *Bayes* yang menghitung probabilitas pada data tertentu (Alamsyah *et al*., 2023). *Random Forest* merupakan perkembangan dari algoritma *Decision Tree* dengan cara menggabungkan beberapa *decision tree* yang dilatih pada sebagian *dataset* yang berbeda. *Random Forest* memanfaatkan *majority voting* untuk melakukan klasifikasi pada *dataset* (Purifyregalia *et al*., 2025).

Klasifikasi metode *deep learning* menjadi perkembangan lanjut dari metode ML tradisional. Metode ini memanfaatkan *neural network* untuk melakukan analisis yang lebih kompleks *Neural network* memanfaatkan lapisan-lapisan dan neuron untuk melakukan klasifikasi yang lebih mendalam (Qamar & Zardari, 2023).



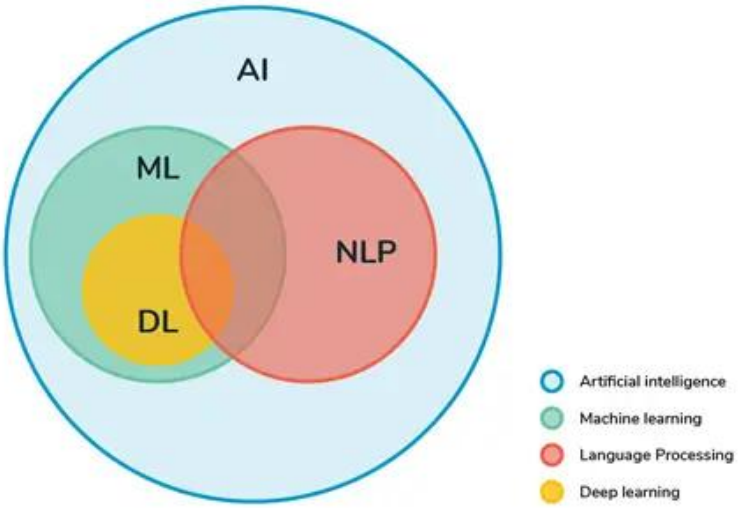
Gambar 2. Lapisan dalam *neural network* (Qamar & Zardari, 2023).

Contoh algoritma *deep learning* yang sering dipakai dalam penelitian bertopik klasifikasi adalah *Recurrent Neural Network* (RNN), *Convolutional Neural Network* (CNN), dan *Long Short-Term Memory* (LSTM). CNN biasanya digunakan dalam tugas-tugas *Computer Vision* (CV), tetapi telah terbukti juga bahwa CNN mampu menyelesaikan permasalahan yang berhubungan dengan pemrosesan bahasa alami (Awalina *et al*., 2022).

2.5 *Natural Language Processing* (NLP)

Menurut Stryker dan Holdsworth (n.d.), *natural language processing* (NLP), atau dalam bahasa Indonesia disebut pemrosesan bahasa alami merupakan bidang AI yang menggunakan algoritma untuk memungkinkan mesin seperti sistem komputer untuk memahami, menafsirkan, dan menghasilkan bahasa manusia dengan cara yang bernilai dan kontekstual. NLP telah menjadi salah satu teknologi AI yang sedang berkembang pesat dalam beberapa tahun terakhir, akibat perkembangan teknologi seperti *deep learning* yang meningkatkan pemahaman mesin terhadap bahasa manusia. NLP dibagi menjadi dua jenis, *natural language understanding* (NLU) yang berfungsi untuk menganalisis makna dari dokumen, dan *natural language generation* (NLG) yang berfungsi untuk membuat teks dari komputer. (Deeplearning.ai, 2023).

NLP telah digunakan untuk beberapa aplikasi dalam kehidupan sehari-hari, seperti analisis sentimen, mesin penerjemah, *voice assistant*, *chatbot*, dan analisis teks untuk mendeteksi *spam* dan *fraud*. Berhubungan dengan penelitian ini, deteksi ulasan palsu pada *platform e-commerce* merupakan salah satu aplikasi NLP yang sangat penting untuk membantu pengembangan NLP, apalagi menggunakan model LLM yang merupakan perkembangan dari NLP (Sutoyo, 2023).

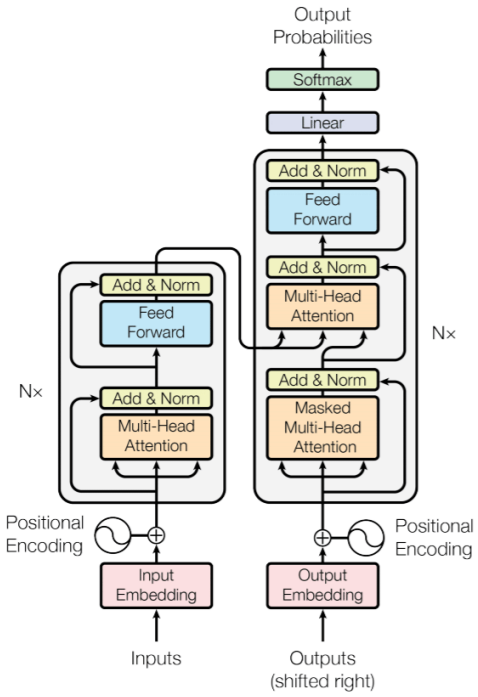


Gambar 3. Posisi bidang NLP pada teknologi AI (Sutoyo, 2023).

2.6 *Transformer*

Transformer merupakan arsitektur model yang menyelesaikan permasalahan keterbatasan *Recurrent Neural Network* (RNN). Penelitian oleh *Google Brain* berjudul "*Attention is All You Need*" memperkenalkan arsitektur *transformer* yang seperti judul penelitiannya, hanya memerlukan mekanisme *attention* untuk mengolah elemen-elemen dalam urutan *input* dan *output* sehingga tidak memerlukan model perulangan (*recurrence*) yang mengakibatkan hambatan paralelisasi pada GPU. Mekanisme *self-attention* merupakan mekanisme yang sangat penting dalam arsitektur *transformer*. Mekanisme tersebut memungkinkan model untuk menimbang relevansi kata-kata lain dalam kalimat terhadap kata yang sedang diproses (Vaswani *et al*., 2017).

Arsitektur *transformer* dibagi menjadi dua, yaitu *encoder* dan *decoder*. Arsitekturnya dapat pada **Gambar 4**.



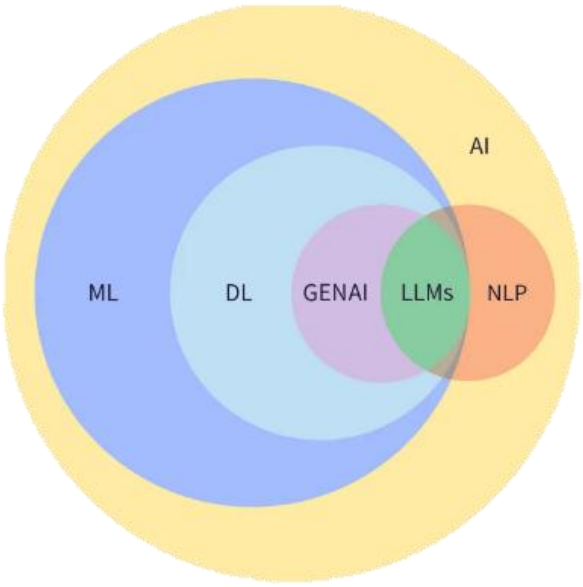
Gambar 4. Arsitektur *transformer* (Vaswani *et al*., 2017).

1. ***Encoder*:** Bagian *encoder* berfungsi untuk mengubah masukan yaitu berupa teks atau token menjadi vektor yang bersifat kontekstual. *Encoder* terdiri dari N (biasanya 6) lapisan, dan setiap lapisan mengandung lapisan *multi-head attention* dan *feed forward*.
2. ***Decoder*:** Bagian *decoder* berfungsi untuk menghasilkan *output* berdasarkan representasi dari encoder dan *output* sebelumnya. *Decoder* mengandung N lapisan di mana setiap lapisan mengandung lapisan *multi-head attention* dan *feed forward*. Selain itu, *decoder* juga mengandung tambahan lapisan *masked multi-head attention* untuk membantu *output* dalam tumpukan encoder.

Model-model LLM umumnya menggunakan arsitektur *transformer* seperti di atas. Contoh *encoder-only* LLM adalah *Bidirectional Encoder Representations from Transformers* (BERT), sedangkan contoh *decoder-only* LLM adalah *Generative Pre-trained Transformer* (GPT) dan *Llama*.

2.7 *Large Language Model* (LLM)

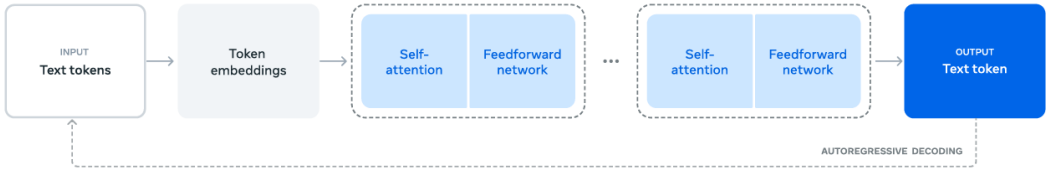
Large language model (LLM) merupakan *pre-trained* model NLP dan *deep learning* yang canggih dan dirancang untuk memahami dan menghasilkan bahasa manusia dengan memerhatikan makna dan *syntax* dari teks dokumen. Model ini biasanya dilatih pada korpus teks berskala sangat besar (biasanya berupa *dataset* korpus dokumen yang terdiri dari triliun *token*) untuk memprediksi *token* berikutnya dalam sebuah urutan teks. Seiring berjalannya waktu, LLM sudah digunakan dalam beberapa aplikasi dalam kehidupan sehari-hari, seperti *chatbot* dalam layanan dukungan sebuah perusahaan, dan sebuah alat untuk mendeteksi penipuan, *spamming*, serta berita palsu (*hoax*) (Papageorgiou *et al*., 2024). Maka dari itu, deteksi ulasan palsu pada *e-commerce* bisa menggunakan LLM yang makin paham konteks dari kalimat dan pola yang terkandung sehingga dapat mendeteksinya sebaik metode-metode pendeteksi lainnya (Z. Wang, 2025).



Gambar 5. Posisi LLM pada teknologi AI (Dataiku, n.d.).

2.8 *Llama* 3.1

Llama 3.1 merupakan seri dari *pre-trained* LLM yang dikembangkan oleh Meta AI yang dirilis pada Juli 2024. Model ini mewakili puncak dari evolusi arsitektur *decoder-only* yang dioptimalkan. Llama 3.1 memperkenalkan beberapa perubahan arsitektur kunci yang memungkinkannya bersaing dengan model LLM lainnya. Model *Llama* dilatih berdasarkan korpus teks *multilingual* yang sangat besar dengan 15.6T *tokens* dan *context window* 128K *tokens*. Karena *token* yang besar tersebut, LLM ini mampu mendukung beberapa bahasa, yang terdiri dari bahasa Inggris, Jerman, Prancis, Italia, Portugis, Hindi, Spanyol, dan Thai. Selain model *base*, Llama 3.1 juga menyediakan model *instruct*, yaitu model Llama 3.1 yang di-*fine-tune* berdasarkan data instruksi atau *prompt* untuk mengikuti perintah yang diberikan oleh pengguna (Grattafiori *et al*., 2024).



Gambar 6. Arsitektur Llama 3.1 (Grattafiori *et al*., 2024).

Gambar 6 menggambarkan arsitektur Llama 3.1. Secara umum, fitur dan arsitektur model Llama 3.1 dapat dijelaskan sebagai berikut:

1. ***Input*:** Llama 3.1 menyediakan *tokenizer* menggunakan algoritma *Byte Pair Encoding* (BPE) berbasis *tiktoken* yang mampu memecahkan korpus teks yang panjang menjadi unit *token*. *Input* teks
2. ***Embeddings*:** Llama 3.1 menggunakan *Rotary Positional Embeddings* (RoPE) untuk menyandikan informasi posisi dalam urutan teks. Nilai frekuensi RoPE disetel menjadi 500.000 untuk mendukung konteks yang lebih panjang.
3. ***Transformer block*:** Blok ini terdiri dari beberapa tumpukan blok yang berbeda untuk varian setiap parameter, seperti varian 8B mempunyai 32 blok, sedangkan varian 405B mempunyai 126 blok. Blok ini dibagi menjadi dua bagian, lapisan *attention* dan *feed-forward*.

   a. ***Self-attention*:** Llama 3.1 menggunakan *Group-Query Attention* (GQA), yaitu membagi *query heads* menjadi delapan kelompok-kelompok, di mana setiap kelompok berbagi satu pasangan *key* dan value. Teknik ini membuat kinerja *attention* menjadi jauh lebih cepat dan hemat memori.
   b. ***Feed-forward network*:** Digunakan untuk meneruskan data dari lapisan *attention* ke blok transformer selanjutnya. Bekerja seperti *neural network* secara umum.
4. ***Output*:** Model menghasilkan probabilitas untuk token berikutnya. Model memilih satu token yang paling mungkin untuk melanjutkan kalimat tersebut.
5. ***Auto-regressive decoding*:** Setelah melalui tahap-tahap sebelumnya, data akan kembali ke tahapan *input* untuk prediksi *token* berikutnya.

Llama 3.1 diluncurkan dalam tiga parameter besar, yaitu model dengan parameter sebanyak 8 miliar, 70 miliar, dan yang paling besar, 405 miliar. Makin besar parameter model, makin mampu model tersebut untuk menangani pola kompleks. Dalam penelitian ini, kita akan menggunakan model dengan parameter 8 miliar agar tidak memakan waktu yang lama selama pelatihan. Spesifikasi dari semua jenis model *Llama* 3.1 dapat dilihat pada Tabel 2.

Tabel 2. Spesifikasi model Llama 3.1 8B, 70B, dan 405B (Grattafiori *et al*., 2024).

Spesifikasi	**Parameter**		
	8B	**70B**	**405B**
Lapisan	32	80	126
Dimensi Model	4.096	8.192	16.384
Dimensi FFN	14.336	28.672	53.248
Attention Heads	32	64	128
Key/Value Heads		8	
Puncak *Learning Rate*	3×10^{-4}	1.5×10^{-4}	8×10^{-5}
Fungsi Aktivasi		*SwiGLU*	
Ukuran Kosakata		128.000	
Positional Embeddings		RoPE (θ =500,000)	

2.9 *Unsloth*

Unsloth adalah pustaka *framework* optimasi *open-source* yang dikembangkan untuk mempercepat pelatihan dan *fine-tuning* LLM serta mengurangi penggunaan memori GPU secara drastis tanpa mengurangi akurasi dari model LLM. *Unsloth* menyelesaikan masalah terhadap penelitian yang melakukan pelatihan model yang berparameter besar dengan sumber daya komputasi yang terbatas. Tanpa optimasi ini, pelatihan model yang berat, contohnya *Llama* 3.1 8B akan sangat sulit atau bahkan mustahil dilakukan karena keterbatasan memori GPU. Fitur-fitur utama dari Unsloth adalah sebagai berikut (Han *et al*., 2023):

1. **Menggunakan bahasa Triton:** Unsloth menulis ulang *kernel* penting (seperti *forward pass* dan *backward pass* pada lapisan MLP dan *Attention*) menggunakan bahasa pemrograman Triton dari OpenAI. Kernel khusus ini dioptimalkan secara manual untuk arsitektur GPU NVIDIA.
2. **Menggunakan deviasi *backpropagation* manual:** Unsloth menggunakan mesin *backpropagation* manual yang diturunkan secara matematis. Dengan mengetahui turunan eksak dari setiap operasi, Unsloth dapat menghitung gradien tanpa perlu menyimpan semua nilai antara, mengurangi penggunaan memori GPU hingga 60-70% dibandingkan implementasi standar *HuggingFace*.
3. **Mendukung *Parameter Efficient Fine-Tuning* (PEFT):** Unsloth mendukung PEFT sehingga kita tidak perlu melakukan *fine-tuning* semua parameter LLM, hanya sekitar 1% dari parameter saja yang bisa dilatih (*trainable*). Hal ini sangat cocok untuk komputer yang mempunyai spesifikasi yang lemah atau *notebook* gratis dengan GPU yang terbatas. Karena merupakan turunan dari PEFT, LoRA dan QLoRA juga didukung oleh Unsloth untuk menerapkan *adapter* agar menghemat memori GPU selama pelatihan.
4. **Mendukung berbagai platform sistem operasi:** Unsloth mendukung berbagai platform sistem operasi seperti Windows, Linux, WSL, bahkan mendukung Google Colab dan Kaggle Notebook dengan . Pada Google Colab, banyak model LLM bisa dilatih menggunakan GPU T4 gratis dengan *framework* Unsloth untuk mempercepat pelatihan.

5. **Mendukung berbagai model LLM:** Unsloth telah bekerja sama dengan tim-tim pengembang LLM *open source* yang terkenal seperti Llama, Gemma, Qwen, Mistral, dan beberapa LLM lainnya untuk membantu para pengembang LLM dalam *fine-tuning* pada komputer yang kurang kuat. Selain itu, Unsloth juga menyediakan *notebook* gratis yang mengimplementasikan *fine-tuning* LLM

Unsloth telah terbukti menghemat RAM dan memori GPU, serta mempercepat pelatihan *dataset*. Berdasarkan penelitian dari Abruzzo *et al*. (2025) melakukan *benchmark fine-tuning Unsloth* terhadap GPU NVIDIA Blackwell dan mendapatkan pengurangan memori GPU sebesar lebih dari 70% dan percepatan waktu pelatihan sebesar 2 kali dari teknologi konvensional, bahkan mengalahkan *HuggingFace* ditambah dengan *FlashAttention* 2.

2.10 *Parameter-Efficient Fine-Tuning* (PEFT)

Parameter-efficient fine-tuning (PEFT) adalah sebuah teknik dalam *fine-tuning* model *pre-trained* LLM yang hanya mengubah sebagian kecil dari total parameter model tersebut. PEFT telah terbukti mengurangi biaya komputasi, khususnya di memori GPU sebesar 10%. Maka dari itu, teknik ini menyediakan peneliti ilmuwan komputer sebuah pendekatan yang efisien untuk bidang AI dalam mengembangkan model AI yang baru. PEFT dibagi menjadi tiga kategori kategori utama, *Additive*, *Reparameterized*, dan *Selective. Additive* PEFT menambahkan sebagian kecil dari parameter yang bisa dilatih ke dalam model, *Reparameterized* PEFT mengubah parameter, biasanya parameter yang *low-rank* yang akan di-*fine-tuned*. Contoh teknik PEFT yang populer adalah *Adapter* dan *Low-Rank Adaptation* (LoRA). (L. Wang *et al*., 2025).

2.11 *Low-Rank Adaptation* (LoRA)

Menurut Dettmers *et al.* (2023), *Low-Rank Adaptation* (LoRA) merupakan penerapan teknik PEFT yang mengurangi memori dengan cara menggunakan sebagian kecil dari parameter terlatih, dan tidak mengubah keseluruhan parameter

dari model. Misalnya bobot asli W yang berukuran besar ($d \times d$), LoRA melatih pembaruan $\Delta W = B \times A$, di mana B dan A berukuran jauh lebih kecil $d \times r$ dan $r \times d$, dengan $r \ll d$) (Abou Baker *et al*., 2024).

LoRA mempunyai beberapa turunan, salah satunya adalah *Quantized Low-Rank Adaptation* (QLoRA).

2.12 *Quantized Low-Rank Adaptation* (QLoRA)

Quantized Low-Rank Adaptation (QLoRA) merupakan salah satu teknik dari PEFT yang merupakan turunan dari LoRA. Teknik ini memanfaatkan fine-tuning pada model LLM dengan kuantisasi 4-bit untuk menghemat penggunaan memori GPU. QLoRA menggunakan tipe data 4-bit NormalFloat sebagai penyimpanan dan 16-bit BrainFloat untuk komputasi. Beberapa komponen-komponen dari QLoRA terdiri dari (Dettmers *et al*., 2023):

1. **4-bit NormalFloat (NF4)**: QLoRA memperkenalkan tipe data baru, NF4, yang secara informasi-teoretis optimal untuk menyimpan bobot jaringan saraf yang terdistribusi secara normal. Hal ini dapat membuat model yang berparameter besar, seperti *Llama* 3.1 8B mengurangi penggunaan memori GPU sebesar 68%.
2. ***Double quantization***: Teknik ini mengkuantisasi konstanta kuantisasi itu sendiri, menghemat rata-rata 0,37 bit per parameter tambahan.
3. ***Paged optimizers***: QLoRA menggunakan fitur memori terpadu NVIDIA untuk secara otomatis memindahkan status optimizer antara CPU, memori RAM, dan memori GPU saat terjadi lonjakan memori.

2.13 *Hyperparameter*

Hyperparameter merupakan parameter yang bisa diubah yang mengendalikan *fine-tuning* model LLM dengan teknik LoRA atau QLoRA pada *Unsloth*. Nilai-nilai parameter ini ditetapkan sebelum proses pelatihan dimulai dan biasanya tidak dipelajari dari data. Nilai-nilai tersebut akan sangat memengaruhi *output* dari

model, sehingga kita harus mencari nilai yang sesuai untuk mendapatkan hasil yang optimal (Unsloth, 2025). Beberapa *hyperparameter* yang digunakan untuk LLM terdiri dari:

2.13.1 *Learning Rate*

Learning rate menunjukkan seberapa banyak bobot model disesuaikan selama pelatihan setiap *step* pelatihan. Biasanya untuk *learning rate*, nilai yang harus disetel harus sekecil mungkin agar menghasilkan model yang lebih stabil dan tepat, meski memerlukan *epoch* yang lebih banyak, sehingga meningkatkan waktu pelatihan. Terkadang *learning rate* yang rendah bisa menyebabkan *underfitting*, *overfitting*, atau membuat model tidak melatih sama sekali. Untuk LoRA dan QLoRA, biasanya nilai yang paling umum digunakan adalah 2×10^{-4} (Unsloth, 2025).

2.13.2 *Effective Batch Size*

Ukuran *batch* efektif dapat diatur untuk menyeimbangkan stabilitas pelatihan dan banyaknya sumber daya yang tersedia seperti memori GPU. Parameter yang dibutuhkan oleh *effective batch size* adalah *batch size* dan *gradient accumulation steps*. *Batch size* merupakan jumlah data yang akan diolah dalam satu siklus, sedangkan *gradient accumulation* adalah banyaknya mikro-*batch* yang diolah sebelum perhitungan bobot model yang baru. Rumus *effective batch size* ditentukan oleh persamaan (1) (Unsloth, 2025).

$$Effective\ Batch\ Size = N \times G \tag{1}$$

Keterangan:

N : ukuran *batch*

G : langkah *gradient accumulation*

2.13.3 *Epoch*

Epoch merupakan satu iterasi lengkap melalui seluruh *dataset* pelatihan. Jumlah *epoch* yang optimal menyeimbangkan antara pembelajaran yang cukup dan pencegahan penghafalan data latih. Semakin banyak *epoch*, lebih baik pelatihan model, tetapi bisa menyebabkan *overfitting*. Sebaliknya, semakin sedikit, bisa mengurangi waktu pelatihan, tetapi menyebabkan model yang tidak terlatih dengan baik. Untuk *dataset* besar, 1-3 *epoch* sering kali sudah cukup untuk *fine-tuning* LLM (Unsloth, 2025).

2.13.4 LoRA *Rank* & LoRA *Alpha*

LoRA *adapters* terdiri dari LoRA *rank* dan LoRA *alpha*. LoRA *rank* (r) merupakan parameter yang mengendalikan jumlah parameter yang dapat dilatih oleh *adapter* matriks *low-rank*. Nilai r yang lebih tinggi (misal 32, 64, atau 128) memungkinkan model mempelajari fitur yang lebih kompleks tetapi menggunakan lebih banyak memori. LoRA *alpha* (α) adalah faktor *scaling* dari penyesuaian *fine-tuning* untuk bobot LoRA. Biasanya diatur sebanding dengan LoRA *rank* (r). Contohnya adalah $\alpha = r$ atau $2r$ (Unsloth, 2025).

2.13.5 *Weight Decay*

Weight decay merupakan sebuah parameter yang memberikan *penalty* terhadap bobot besar dengan meningkatkan generalisasi model agar dapat mencegah *overfitting*. Biasanya kita tidak boleh menggunakan angka yang terlalu besar karena akan mengurangi nilai bobot model sehingga akan terjadi tidak terlatih dengan optimal. Angka yang sering digunakan adalah 0,01, tetapi bisa sampai 0,1 (Unsloth, 2025).

2.13.6 *Target Modules*

Target modules merupakan lapisan mana dalam transformer yang akan diterapkan LoRA. Penerapan bisa berupa semua lapisan *attention* (q_proj, k_proj, v_proj,

o_proj), *Multilayer Perceptron* (MLP) (gate_proj, up_proj, down_proj), atau kedua-duanya. Biasanya disarankan untuk menerapkannya pada semua lapisan linear untuk mendapatkan hasil yang terbaik (Unsloth, 2025).

2.13.7 LoRA *Dropout*

LoRA *dropout* adalah teknik regularisasi yang diterapkan secara langsung ke dalam bobot *adapter* daripada membekukan model. Teknik ini bekerja hampir sama seperti *dropout* pada *neural network,* yaitu dengan cara menetapkan beberapa nilai aktivasi pada adapter menjadi 0 secara acak untuk mencegah *overfitting*. Nilai parameter ini berupa sebuah probabilitas, biasanya disetel sangat kecil (seperti 0,05 atau 0,1), tetapi bisa juga disetel menjadi 0 untuk tidak mengaktifkan *dropout* sama sekali (PEFT, n.d.). Unsloth merekomendasikan untuk tidak mengaktifkan *dropout* (disetel menjadi 0) karena tidak terlalu berpengaruh dalam pencegahan *overfitting* (Unsloth, 2025).

2.13.8 LoRA *Bias*

Hyperparameter bias dalam LoRA menentukan apakah vektor bias pada model diubah selama pelatihan. *Bias* terdiri dari tiga pilihan nilai, '*none*', '*all*', atau '*lora_only*'. Jika disetel '*none*', maka tidak ada perubahan nilai bias sama sekali dan hanya matriks *low-rank* A dan B yang diperbarui. Jika disetel '*lora_only*', maka hanya nilai bias yang berada di *target modules* LoRA (q_proj, k_proj, v_proj, dan sebagainya) diperbarui. Jika disetel ke '*all*', maka semua modul, termasuk yang di luar *target modules* akan diperbarui nilai bias. Tetapi perlu diketahui bahwa jika parameter ini diaktifkan, maka akan memakan memori lebih banyak sehingga disarankan untuk dinonaktifkan untuk menghemat memori. (PEFT, n.d.).

2.14 Metrik Evaluasi

Metrik evaluasi digunakan untuk mengukur kinerja dari model pendeteksi ulasan palsu di platform *e-commerce*. Hal ini sangat penting agar dapat melihat seberapa

efektif model klasifikasi yang sudah dilatih. Secara umum, metrik evaluasi yang sering digunakan terdiri dari akurasi, *precision*, *recall,* dan F1-score untuk menilai keefektifan model klasifikasi. Selain itu, digunakan juga *confusion matrix* untuk menggambarkan prediksi klasifikasi yang benar dan salah (Hadi *et al*., 2025). Penelitian ini juga menggunakan metrik *loss* selama pelatihan.

2.14.1 *Confusion Matrix*

Confusion matrix merupakan metrik evaluasi yang paling umum digunakan untuk menunjukkan visualisasi atau gambaran dari kinerja model dengan membandingkan klasifikasi yang diprediksi oleh model dan klasifikasi sebenarnya. Metrik ini memberikan wawasan terhadap berapa banyak data ulasan yang tepat diprediksi oleh model. Matriks ini terdiri dari empat elemen penting, yaitu *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN), yang masing-masing menghitung banyaknya data yang diprediksi benar atau salah. Elemen-elemen tersebut dapat diturunkan menjadi metrik-metrik untuk menilai kinerja model, yaitu *accuracy*, *precision*, *recall*, dan *F1-score* (Hadi *et al*., 2025). Tabel 3 menunjukkan gambaran dari *confusion matrix*.

Tabel 3. *Confusion matrix* (Maxwell *et al*., 2021).

		Predicted Class	
		Positive	***Negative***
Actual Class	***Positive***	TP	FN
	Negative	FP	TN

Matriks pada tabel di atas menggambarkan pengukuran kinerja klasifikasi, di mana keterangannya adalah sebagai berikut:

1. *True Positive* (TP) : Banyaknya data yang bernilai positif dan diprediksi positif (Benar).
2. *False Positive* (FP) : Banyaknya data yang bernilai positif dan diprediksi negatif (Salah).

3. *False Negative* (FN) : Banyaknya data yang bernilai negatif dan diprediksi positif (Salah).
4. *True Negative* (TN) : Banyaknya data yang bernilai negatif dan diprediksi negatif (Benar).

Beberapa metrik yang diturunkan dari elemen-elemen utama *confusion matrix* terdiri dari:

2.14.2 *Accuracy*

Metrik *accuracy* menunjukkan persentase total dari prediksi yang benar dengan semua prediksi untuk kedua kelas. Nilai akurasi yang tinggi berarti model klasifikasi berjalan dengan baik secara umum (Hadi *et al*., 2025). Rumus dari akurasi ditentukan oleh persamaan (2).

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (2)$$

2.14.3 *Precision*

Metrik *precision* menunjukkan berapa banyak data yang terprediksi positif itu benar-benar positif. Hal ini mencerminkan kemampuan data untuk menghindari *false positives*. Metrik ini menghitung persentase data positif yang terprediksi dengan benar (positif) (Hadi *et al*., 2025). Rumus *precision* dinyatakan pada persamaan (3).

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

2.14.4 *Recall*

Metrik *recall*, atau disebut juga dengan sensitivitas atau *true positive rate* merupakan metrik evaluasi yang mengukur kemampuan model dalam memprediksi semua data yang relevan (positif) dengan benar. Metrik ini menghitung persentase

prediksi positif yang benar atau sesuai dengan kelas aslinya (positif) (Hadi *et al*., 2025). Rumus *recall* dinyatakan pada persamaan (4).

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

2.14.5 *F1-Score*

Metrik *F1-score* merupakan rata-rata harmonik dari *precision* dan *recall*, menyediakan metrik seimbang yang berguna secara khusus untuk *dataset* yang tidak seimbang. Metrik ini menghasilkan nilai yang menyatakan *false postives* dan *false negatives* (Hadi *et al*., 2025). Rumus dari *F1-score* dinyatakan dalam persamaan (5).

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

2.14.6 *Loss*

Metrik *loss* merupakan pengukuran bias atau *error* pada model. Biasanya metrik ini direkam selama pelatihan model. Setiap *epoch* atau iterasi data latih, algoritma optimasi mengubah bobot untuk mengurangi nilai *loss*. Pada saat pelatihan *dataset* terhadap model, *metrik* ini harus berkurang setiap iterasi tertentu atau *epoch*. Ada banyak jenis metrik *loss* yang bisa digunakan. Pada saat divalidasi dengan data validasi, nilai *loss* harus turun. Jika pada saat tertentu nilai *loss* naik, maka data akan mengalami *overfitting*, di mana model "menghafal" *dataset* yang diberikan termasuk data-data yang tidak relevan sehingga akurasi bisa menurun drastis. Untuk klasifikasi bersifat biner seperti penelitian ini, kita akan menggunakan metode *binary cross-entropy* (BCE). BCE, disebut juga dengan *log loss* digunakan dalam model klasifikasi yang memiliki dua kelas, biasanya untuk mengukur seberapa besar *error* atau kesalahan yang dibuat oleh model (Maxwell et al., 2021). Rumus dari BCE diungkapkan pada persamaan (6) (Raju, 2025).

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} \times \log(p(y_i)) + (1 - y_i) \times \log(1 - p(y_i)) \tag{6}$$

Dengan keterangan:

N : banyaknya sampel;

y_i : kelas sebenarnya untuk sampel ke-i; dan

$p(y_i)$: peluang prediksi sampel berada di kelas 1 untuk sampel ke-i.

III. METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini dilakukan pada tempat dan waktu berikut:

3.1.1 Tempat Penelitian

Penelitian ini dilakukan di Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Universitas Lampung, Jalan Prof. Dr. Ir. Sumantri Brojonegoro No.1 Gedong Meneng, Kecamatan Rajabasa, Kota Bandar Lampung.

3.1.2 Waktu Penelitian

Penelitian ini berjalan dari November 2025 sampai Januari 2026 dengan tahapan proses yang lebih jelas di Tabel 4.

Tabel 4. Waktu penelitian.

Tahapan	2025							2026		
	November			Desember				Januari		
	II	III	IV	I	II	III	IV	I	II	III
Pengumpulan *Dataset*										
Pengembangan Model										
Evaluasi Model										
Penulisan Laporan										

3.2 Alat Penelitian

Untuk melaksanakan penelitian ini, diperlukan beberapa peralatan yang membantu penelitian ini terjadi. Peralatan yang digunakan dibagi menjadi perangkat keras (*hardware*) dan perangkat lunak (*software*)

3.2.1 Perangkat Keras (*Hardware*)

Penelitian ini menggunakan perangkat keras *personal computer* (PC) *laptop* dengan spesifikasi dan detail dijelaskan pada Tabel 5.

Tabel 5. Spesifikasi dan detail perangkat keras.

Spesifikasi	Detail
Merek	*Lenovo*
Model	*Ideapad Pro 15* 14AHP
Prosesor	*AMD Ryzen* 7 8845HS (*8 Core / 16 Threads / 3.8 GHz / Turbo Boost up to 5.1 GHz*)
Kartu Grafis	*AMD Radeon* 780M (*Integrated*) *NVIDIA GeForce* RTX 3050 dengan 6 GB VRAM (*Dedicated*)
RAM	16 GB LPDDR5X-6400 MHz
Penyimpanan	1TB M.2 2242 SSD

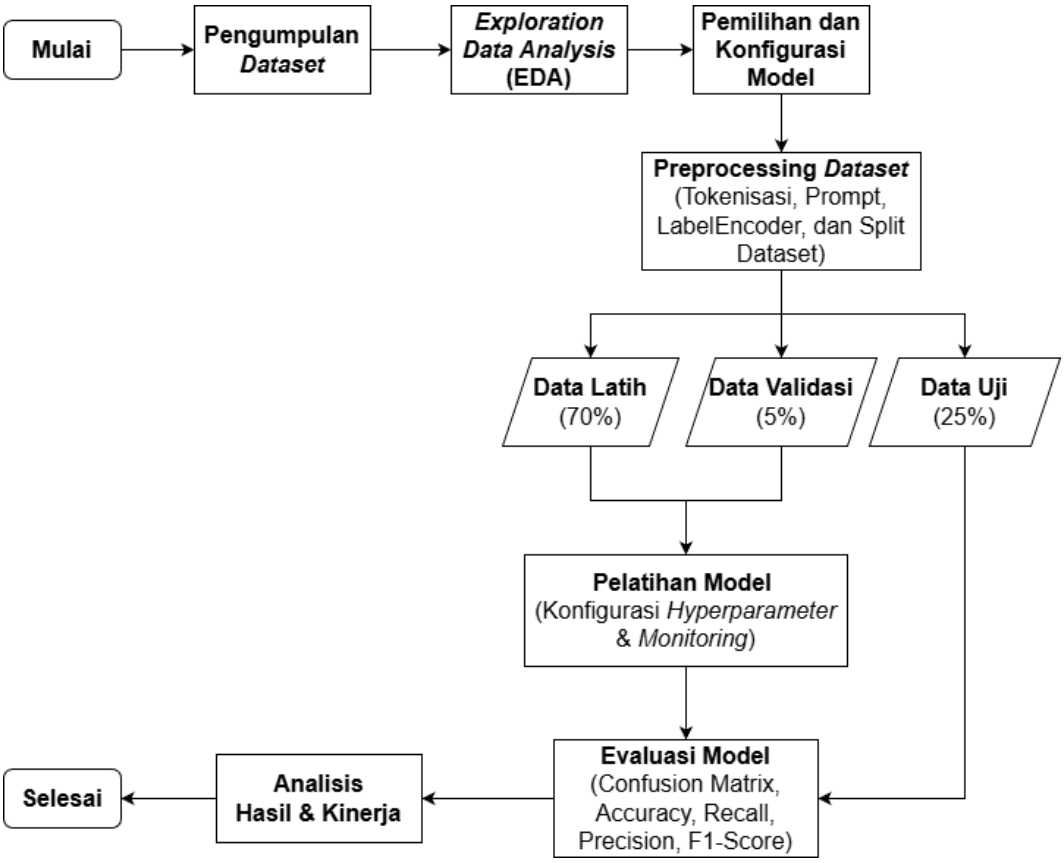
3.2.2 Perangkat Lunak (*Software*)

Selain perangkat keras di atas, penelitian ini juga didukung oleh beberapa perangkat lunak sebagai berikut:

1. Sistem Operasi *Windows* 11 *Home Single Language* 64-bit
2. Python versi 3.12.12
3. Pustaka-pustaka: Unsloth, Transformers, HuggingFace, WandB, dan lain-lain.
4. Google Colab
5. Kaggle Notebook
6. *Web browser* Google Chrome

3.3 Tahapan Penelitian

Penelitian yang dilaksanakan oleh penulis untuk mengembangkan model *fine-tuned Llama 3.1* 8B untuk mendeteksi ulasan palsu dibagi menjadi beberapa tahapan yang digambarkan pada Gambar 7.



Gambar 7. Tahapan penelitian.

Alur penelitian dibagi menjadi lima tahapan utama, terdiri dari memuat model LLM *Llama* 3.1 8B ke dalam program, Lalu, akan dilakukan pengumpulan *dataset* ulasan palsu platform *e-commerce* untuk pelatihan model. Setelah dimuat, *dataset* analisis dan eksplorasi data untuk menentukan strategi Setelah itu, dilakukan tahapan *preprocessing dataset.* Tahap *preprocessing meliputi* tokenisasi, *label encoding*, pemberian instruksi, dan *split dataset* untuk membersihkan *dataset* agar siap dilatih untuk *fine-tuning* model *Llama* 3.1 8B dengan akurasi yang optimal. Setelah itu, model dilatih dengan *Unsloth* dan teknik *QLoRA* berdasarkan *dataset* latih dan validasi untuk menghasilkan model LLM yang *fine-tuned* untuk mendeteksi ulasan

palsu pada *platform e-commerce*. Setelah itu, tahap evaluasi menguji model dengan data uji.

3.3.1 Pengumpulan *Dataset*

Tahapan pengumpulan *dataset* merupakan tahapan untuk mengambil data dan memuatnya ke dalam program. Penelitian ini menggunakan *dataset* yang berasal dari penelitian yang dilakukan oleh Salminen *et al*. (2022) yang berjudul "Creating and detecting fake reviews of online products". *Dataset* tersebut bisa diakses melalui *Kaggle*. *Dataset* terdiri kurang lebih 40.000 data ulasan dari *platform e-commerce* Amazon yang dibagi rata 50:50 antara ulasan asli dan ulasan palsu, yang berupa ulasan yang *computer generated*. *Dataset* terdiri dari empat kolom, yaitu kolom *category* yang menunjukkan kategori produk, kolom *rating* yaitu jumlah bintang yang diberikan dari angka 1 sampai 5, kolom *text_* yang berisi teks ulasan, dan label yang terdiri dari label ulasan asli (OR) dan ulasan *computer generated* (CG). Sampel dari *dataset* ditunjukkan pada Tabel 6 berikut.

Tabel 6. Sampel *dataset* ulasan produk Amazon (Salminen *et al*., 2022).

category	***rating***	***label***	***text_***
Clothing_Shoes_and_Jewelry_5	4.0	OR	I had read some reviews saying that this bra r...
Clothing_Shoes_and_Jewelry_5	5.0	CG	I wasn't sure exactly what it would be. It is ...
Clothing_Shoes_and_Jewelry_5	2.0	OR	You can wear the hood by itself, wear it with ...
Clothing_Shoes_and_Jewelry_5	1.0	CG	I liked nothing about this dress. The only rea...
Clothing_Shoes_and_Jewelry_5	5.0	OR	I work in the wedding industry and have to wor...

3.3.2 *Exploratory Data Analysis* (EDA)

Tahapan *Exploratory Data Analysis* (EDA) adalah sebuah tahapan analisis data awal untuk memahami karakteristik, mencari pola, dan mendeteksi anomali pada *dataset*. Tahapan ini bertujuan untuk membuat strategi *preprocessing dataset* dan pelatihan model serta memahami pengaruh karakteristik *dataset* terhadap model yang akan dibangun pada nantinya. EDA biasanya dilakukan menggunakan teknik-teknik seperti visualisasi *dataset* dan statistik deskriptif lainnya. Penelitian ini menggunakan penggabungan kedua teknik untuk menganalisis *dataset* ulasan yang berlabel ulasan asli (OR) dan ulasan palsu (CG). Analisis data akan dilakukan berdasarkan label kelas untuk melihat perbedaan karakteristik ulasan asli dan palsu yang mungkin muncul. Ada beberapa pengukuran yang digunakan untuk menganalisis karakteristik ulasan dalam *dataset* adalah sebagai berikut:

1. **Visualisasi persentase label ulasan:** Berupa diagram lingkaran yang menunjukkan persentase ulasan asli dan palsu dalam *dataset*. Karena banyaknya ulasan asli dan palsu sama, maka grafik yang diharapkan adalah seimbang antara kedua label tersebut;
2. **Visualisasi distribusi *rating* ulasan:** Berupa diagram batang yang menggambarkan banyaknya ulasan yang memiliki *rating* bintang 1-5 berdasarkan label data. Hal ini menunjukkan keseimbangan ulasan positif dan negatif dalam *dataset*;
3. **Rata-rata jumlah kata dalam kalimat:** Metrik linguistik yang mengukur banyaknya kata dalam satu kalimat ulasan secara rata-rata. Metrik ini menunjukkan perbandingan panjang kalimat dalam ulasan asli dan palsu;
4. **Rata-rata panjang kata:** Metrik linguistik yang mengukur panjang kata, atau banyaknya karakter dalam sebuah kata secara rata-rata. Metrik ini menganalisis gaya penulisan dan keterbacaan dalam ulasan asli dan ulasan palsu;
5. ***Type token ratio* (TTR):** Metrik linguistik untuk mengukur keberagaman kosa kata dalam sebuah teks. TTR dihitung dengan membagi jumlah kata unik (*type*) dengan total jumlah kata (*token*). Semakin tinggi nilai TTR, semakin variatif kosakata yang digunakan. Rumus TTR dinyatakan dalam persamaan (7) di bawah ini:

$$TTR = \frac{Jumlah\ Tipe\ (Kata\ Unik)}{Jumlah\ Token\ (Total\ Kata)} \times 100\% \quad (7)$$

6. **Rata-rata jumlah sebuah kata yang muncul:** Menghitung keberadaan sebuah kata atau kelompok kata dari *dataset* yang muncul. Penelitian ini menghitung keberadaan kata '*recommend*' dan kata ganti orang pertama untuk melihat seberapa banyak kata tersebut muncul dalam ulasan palsu dan ulasan asli.

3.3.3 Pemilihan dan Konfigurasi Model

Tahapan pemilihan model dan konfigurasi merupakan tahapan awal pada penelitian ini. Fokus utama dari tahap ini adalah untuk mempersiapkan infrastruktur model agar dapat berjalan pada sumber daya komputasi yang terbatas, dalam hal ini penulis menggunakan Google Colab dan Kaggle Notebook dengan NVIDIA T4 GPU. Penelitian ini menggunakan model Llama 3.1 8B varian *Instruct* yang telah dikuantisasi menjadi 4-bit dengan cara memuat unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit. Pemilihan model ini didasarkan pada keseimbangan antara kemampuan pemahaman konteks yang mendalam dan efisiensi parameter. Kita dapat memuat model dengan pustaka FastLanguageModel dari Unsloth agar dapat menggunakan kernel-kernel untuk mempercepat pelatihan.

Metode yang dipilih adalah PEFT (*Parameter-Efficient Fine-Tuning*) menggunakan teknik QLoRA (*Quantized Low-Rank Adaptation*). Teknik ini memungkinkan pembaruan bobot model tanpa harus melatih seluruh parameter, melainkan hanya matriks adaptor yang lebih kecil (Rank $r = 16$, Alpha $\alpha = 16$). Pustaka Unsloth digunakan untuk mempercepat proses pelatihan hingga 2-5 kali lipat dan mengurangi penggunaan memori VRAM GPU hingga 70-90%.

3.3.4 *Preprocessing Dataset*

Sebelum *dataset* dilatih ke dalam *pre-trained* model Llama 3.1 8B untuk *fine-tuning*, kita harus melakukan tahapan *preprocessing* untuk memastikan bahwa

dataset siap untuk dilatih sesuai dengan kebutuhan *instruction tuning* pada Llama 3.1. Terdapat beberapa tahapan *preprocessing*, yaitu:

1. ***Label encoding*:** Mengubah label teks menjadi format numerik agar dapat diproses oleh model. Penelitian ini menggunakan fitur *mapping* dari Python, seperti terlihat pada Kode Program 1.

```
label_map = {'CG': 1, 'OR': 0}
df['label_id'] = df['label'].map(label_map)
```

Kode Program 1. *Label encoding.*

2. **Format *prompt* (*instruction tuning*):** Data teks ulasan diubah menjadi format instruksi khusus mengikuti struktur *prompt* Llama 3.1.

   Contoh format yang diterapkan dalam kode untuk data *train* dan *validation*:

   <s>[INST] Classify this product review as fake (1) or genuine (0). Review: {teks_ulasan} [/INST] {label} </s>.

3. **Tokenisasi:** Mengubah kalimat, baik dari *dataset* maupun *prompt* di atas menjadi token menggunakan *tokenizer* bawaan model Llama 3.1 dengan *max sequence length* diatur sebesar 2048 token.

4. **Pembagian Dataset (Splitting):** Dataset dibagi menjadi tiga bagian untuk menjaga validitas evaluasi:

   a. **Data Latih (*Train*):** 70% dari total data, digunakan untuk melatih bobot model.
   b. **Data Validasi (*Validation*):** 5% dari total data, digunakan untuk evaluasi selama proses pelatihan agar tidak terjadi *overfitting*.
   c. **Data Uji (*Test*):** 25% dari total data, digunakan untuk evaluasi akhir kinerja model.

3.3.5 Pelatihan Model

Tahapan ini merupakan tahap inti di mana setelah melalui tahapan *preprocessing*, model Llama 3.1 8B siap dilatih untuk menghasilkan model *fine-tuned* untuk

dataset ulasan palsu. *Hyperparameter* yang digunakan dalam pelatihan model ini dijabarkan pada Tabel 7.

Tabel 7. *Hyperparameter* yang digunakan dalam pelatihan model.

Hyperparameter	**Nilai**
Batch Size	4
Gradient Accumulation Steps	8
Effective Batch Size (BS × GAS*)*	4 × 8 = 32
Learning Rate	2×10^{-4}
Epoch	5
LoRA *Rank*	16
LoRA *Alpha*	16
Target Modules	["q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj", "down_proj”]
Weight Decay	0,01

Pelatihan model menggunakan pustaka *Supervised Fine-Tuning* (SFT) *Trainer* yang disediakan oleh TRL untuk pelatihan *fine-tuning* LLM karena model Llama 3.1 yang digunakan merupakan model yang di-*fine-tuned* untuk memahami instruksi atau *prompt* yang diberikan pada *input* (*instruct model*). Hal tersebut agar model dapat memahami petunjuk selama pelatihan berlangsung. Selain itu, SFT Trainer dapat diintegrasikan ke Unsloth dan QLoRA, yang mampu mengurangi memori yang digunakan pada GPU T4 pada Google Colab dan Kaggle Notebook, serta mempercepat waktu pelatihan secara drastis.

Pustaka *Unsloth* akan melakukan *patch* terhadap sistem komputasi untuk mempercepat pelatihan, sedangkan QLoRA menggunakan kuantisasi 4-bit terhadap LoRA *adapters* dengan menyisipkan matriks-matriks kecil tambahan. Maka dari itu, model hanya melatih sebagian kecil dari total parameter saja untuk menghemat memori GPU secara signifikan. Penjelasan fungsi teknis dari masing-masing *hyperparameter* adalah sebagai berikut:

1. ***Effective Batch Size*:** Ukuran *batch* selama pelatihan. Makin tinggi nilainya, makin baik penggunaan GPU tetapi akan memakan waktu yang lebih lama. Direkomendasikan untuk menetapkan *effective batch size* 32, sehingga penelitian ini menetapkan nilai *batch size* 4 dan *gradient accumulation steps* 8.
2. ***Learning Rate*:** Untuk model klasifikasi dan *instruction-tuned,* nilai ditetapkan serendah mungkin untuk menjaga stabilitas pelatihan. Penelitian ini menggunakan nilai *learning rate* 2×10^{-4}.
3. ***Epoch*:** *Epoch* memastikan model akan dilatih sebanyak satu siklus data latih. Lebih banyak nilai *epoch*, lebih terlatih modelnya, tetapi risiko *overfitting* akan terjadi. Penelitian ini akan menggunakan lima *epoch* agar model dapat menangkap pola dari ulasan asli dan palsu dengan baik.
4. **LoRA *rank* dan LoRA *alpha*:** LoRA *rank* menentukan jumlah parameter yang dilatih, sedangkan LoRA *alpha* menunjukkan faktor scaling untuk bobot LoRA. Makin tinggi nilai LoRA *rank*, makin kompleks pembelajaran fitur yang lebih kompleks. Nilai LoRA *alpha* biasanya diatur sebanding atau bahkan sama dengan nilai LoRA *rank*.
5. ***Target modules*:** Menerapkan *adapter* LoRA pada lapisan transformer tertentu, baik itu semua lapisan *attention* (q_proj, k_proj, v_proj, o_proj), *Multilayer Perceptron* (MLP) (gate_proj, up_proj, down_proj), atau kedua-duanya. Untuk menghasilkan model yang terbaik, penelitian ini akan menggunakan kedua-duanya.
6. ***Weight decay*:** Nilai ini tidak boleh terlalu besar karena akan membuat model terlalu berpikir umum sehingga terjadi *underfitting*, sehingga penelitian ini menggunakan nilai *weight decay* 0.01 untuk mencegah *overfitting* dengan memberikan penalti pada bobot yang terlalu besar.

3.3.6 Evaluasi Model

Setelah pelatihan selesai, kinerja model diuji menggunakan data uji menggunakan metrik-metrik evaluasi seperti *confusion matrix*, *accuracy*, *precision*, *recall*, dan *F1-score*. *Confusion matrix* digunakan untuk memvisualisasikan jumlah prediksi *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative*

(FN), sedangkan *accuracy* berfungsi mengukur persentase prediksi yang benar secara keseluruhan, *precision* mengukur ketepatan prediksi ulasan palsu (seberapa banyak yang diprediksi palsu memang benar-benar palsu), *recall* mengukur kemampuan model menemukan ulasan palsu (seberapa banyak ulasan palsu yang berhasil dideteksi), dan terakhir *F1-score* menghitung rata-rata harmonik antara *precision* dan *recall*, memberikan gambaran performa yang seimbang.

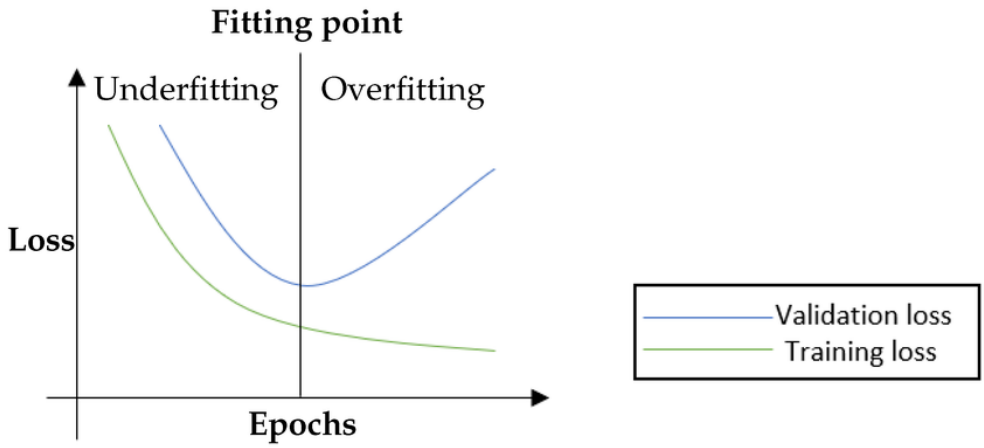
Selain itu, kita juga dapat memantau grafik *training loss* dan *validation loss* untuk memastikan model belajar dengan baik. Dalam kata lain, nilai *loss* menurun dan model tidak mengalami *overfitting*. Terakhir, kita akan menganalisis evaluasi ulasan yang terjadi *missed classification*. *Missed classification* artinya model yang memprediksi sebuah data yang berbeda dengan label aslinya. Biasanya terjadi *False Positive*, yaitu model memprediksi ulasan tersebut palsu padahal sebenarnya itu asli atau *False Negative*, yang berarti model mengklasifikasikan ulasan tersebut asli, padahal sebenarnya palsu.

3.3.7 Analisis Hasil dan Kinerja

Tahapan ini bertujuan untuk menganalisis kinerja dari model yang telah kita latih dan uji (evaluasi) untuk wawasan yang bermakna mengenai kemampuan model dalam mendeteksi ulasan palsu serta efisiensi sumber daya yang digunakan. Tahapan ini dibagi menjadi dua, analisis model dan analisis kinerja pelatihan.

Analisis model dilakukan untuk mencari tahu seberapa akurat model dalam membedakan ulasan asli dan palsu, serta apakah model ini mampu digunakan untuk *platform e-commerce* untuk mendeteksi ulasan palsu tersebut. Analisis dilakukan menggunakan pustaka WandB untuk logging berdasarkan metrik evaluasi *confusion matrix*, *accuracy*, *precision*, *recall*, *F1-score*, dan *loss* untuk *dataset train* dan *validation*. Khusus untuk metrik *loss*, kita dapat melihat grafik *training loss* dan *validation loss* untuk mendeteksi indikasi *overfitting* atau *underfitting*. *Overfitting* biasanya terjadi ketika *validation loss* naik secara drastis, sedangkan *underfitting* terjadi jika tidak ada penurunan *loss* pada model. Pada saat keadaan optimal, model mengalami *validation loss* paling rendah selama *training loss* juga turun. Peristiwa

tersebut disebut juga dengan istilah *fitting point*. Grafik *loss* yang mengalami *underfitting, fitting point,* dan *overfitting* dapat dilihat pada Gambar 8 berikut.



Gambar 8. Grafik kurva *loss* untuk *underftting*, *fitting point*, dan *overfitting* (Alomar *et al*., 2023).

Analisis kinerja komputasi dilakukan untuk mendapatkan wawasan tentang *fine-tuning* LLM menggunakan Unsloth dan QLoRA dan pengaruhnya bagi sumber daya komputasi di GPU T4 pada Google Colab dan Kaggle Notebook. Pustaka Unsloth secara teori dapat mempercepat pelatihan sebesar dua kali melalui *patching kernel* selama pelatihan. Maka dari itu, kita dapat menganalisis komputasi kinerja dengan menggunakan pustaka WandB untuk melihat puncak penggunaan memori *(peak memory usage*), baik itu memori GPU atau RAM, dan mencatat waktu *runtime* pada saat pelatihan *dataset*. Biasanya untuk *dataset* besar, *fine-tuning* LLM biasanya akan memakan waktu dan komputasi yang sangat besar. Karena keterbatasan sumber daya dan *runtime* dari GPU T4, jika *fine-tuning* ini dijalankan tanpa pustaka Unsloth dan QLoRA, maka Google Colab dan Kaggle Notebook akan mengalami *error out-of-memory* atau akan kehabisan *runtime* gratis untuk GPU T4.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan mengenai penerapan *fine-tuning* pada model Llama 3.1 8B menggunakan Unsloth dan teknik QLoRA untuk deteksi ulasan palsu pada produk *e-commerce*, kita dapat simpulkan bahwa:

1. Penelitian ini berhasil mengimplementasikan teknik *Parameter-Efficient Fine-Tuning* (PEFT) menggunakan QLoRA dan optimasi Unsloth pada model *Large Language Model* (LLM) Llama 3.1 8B. Penggunaan teknik ini terbukti sangat efisien dalam penggunaan sumber daya komputasi. Implementasi Unsloth dan QLoRA mampu mengurangi penggunaan memori GPU sehingga dapat dijalankan di lingkungan di mana sumber daya sangat terbatas. Dengan hanya melatih 0,52% parameter (sekitar 41,9 juta parameter) dan menggunakan kuantisasi 4-bit, alokasi memori GPU tertinggi hanya sekitar 8,98 GB dan waktu pelatihan 32 jam 8 menit.

2. Model Llama 3.1 8B berhasil dievaluasi dan mampu mencapai akurasi sebesar 99,47%, *precision* 99,27%, *recall* 99,66%, dan *F1-score* 99,47% pada data uji dengan waktu evaluasi 56 menit 23 detik dan satu ulasan diproses dalam waktu ~0,333 detik, serta memori GPU selama inferensi terpakai sebanyak ~6,7 GB. Namun, model mengalami *overfitting* setelah epoch ke-2. Hal ini ditandai dengan penurunan konsisten pada *training loss* (mencapai 0,9538 pada *epoch* ke-5) namun terjadi kenaikan signifikan pada *validation loss* mulai dari *epoch* ke-3 (1,5839) hingga puncaknya di *epoch* ke-4 (1,7445). Hal ini menunjukkan bahwa pelatihan model Llama 3.1 untuk kasus ini paling optimal dilakukan maksimal selama 2 *epoch* untuk menjaga kemampuan generalisasi model.

5.2 Saran

Meskipun penelitian ini telah mencapai hasil yang memuaskan, terdapat beberapa keterbatasan yang dapat diperbaiki atau dikembangkan dalam penelitian selanjutnya. Adapun saran yang dapat diberikan adalah sebagai berikut:

1. **Pengembangan Dataset *Multilingual*, Termasuk Bahasa Indonesia:** Penelitian ini terbatas pada *dataset* berbahasa Inggris karena keterbatasan dukungan bahasa model dan ketersediaan data. Untuk penelitian selanjutnya, disarankan untuk mengembangkan atau menggunakan *dataset* ulasan palsu dalam Bahasa Indonesia atau *multilingual*. Hal ini penting agar sistem deteksi dapat diterapkan pada *platform e-commerce* lokal di Indonesia seperti Tokopedia atau Shopee.

2. **Optimalisasi Durasi Pelatihan (*Early Stopping*):** Mengingat model mengalami *overfitting* setelah epoch ke-2, penelitian selanjutnya disarankan untuk menerapkan teknik *Early Stopping*. Pelatihan sebaiknya dihentikan secara otomatis ketika *validation loss* mulai meningkat untuk mencegah model "menghafal" data latih secara berlebihan dan untuk menghemat waktu komputasi.

3. **Pengujian terhadap Variasi Ulasan Palsu yang Lebih Kompleks:** Analisis eksplorasi data menunjukkan bahwa ulasan palsu pada *dataset* ini cenderung lebih pendek dan repetitif, yang mungkin mempermudah model untuk membedakannya berdasarkan panjang kalimat. Penelitian selanjutnya disarankan untuk menguji ketangguhan model menggunakan *dataset* ulasan palsu yang dihasilkan oleh LLM generasi terbaru yang lebih canggih, yang mampu meniru gaya penulisan manusia dengan lebih natural dan panjang kalimat yang bervariasi, guna memastikan model tidak bias terhadap fitur panjang teks semata.

4. **Eksperimen *Hyperparameter* Lanjutan:** Penelitian ini menggunakan konfigurasi *hyperparameter* sesuai rekomendasi dari Unsloth (Rank 16, Alpha 16) dan pelatihan selama 5 *epoch*. Penelitian ini bisa berjalan lebih baik jika

kita menerapkan *hyperparameter tuning* agar dapat melakukan eksperimen lebih lanjut dengan variasi nilai *rank* dan *alpha* yang lebih tinggi, serta jumlah *epoch* yang berbeda, untuk melihat apakah kinerja model masih dapat ditingkatkan atau apakah efisiensi memori dapat lebih dioptimalkan.

5. **Implementasi Sistem Secara *Real-Time*:** Hasil penelitian ini masih berupa model yang berjalan di lingkungan *notebook*. Kedepannya, model ini dapat dikembangkan menjadi sebuah API atau aplikasi web yang terintegrasi langsung dengan sistem *e-commerce*. Hal ini akan memberikan dampak praktis yang nyata dalam melindungi konsumen dan menjaga integritas ekosistem *e-commerce* dari serangan ulasan palsu secara *real-time*.

DAFTAR PUSTAKA

Abou Baker, N., Rohrschneider, D., & Handmann, U. (2024). Parameter-Efficient Fine-Tuning of Large Pretrained Models for Instance Segmentation Tasks. *Machine Learning and Knowledge Extraction*, *6*(4), 2783–2807. https://doi.org/10.3390/make6040133

Abruzzo, P., Perlow, J., & Carpenter, B. (2025). *Train an LLM on NVIDIA Blackwell with Unsloth—and Scale for Production | NVIDIA Technical Blog*. NVIDIA Developer. https://developer.nvidia.com/blog/train-an-llm-on-an-nvidia-blackwell-desktop-with-unsloth-and-scale-it/

Alamsyah, H., Cahyana, Y., & Pratama, A. R. (2023). Deteksi Fake Review Menggunakan Metode Support Vector Machine dan Naïve Bayes Di Tokopedia. *Jutisi : Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, *12*(2), 585. https://doi.org/10.35889/jutisi.v12i2.1222

Alomar, K., Aysel, H. I., & Cai, X. (2023). Data Augmentation in Classification and Segmentation: A Survey and New Strategies. *Journal of Imaging*, *9*(2). https://doi.org/10.3390/jimaging9020046

Amazon. (2024). *How Amazon is using AI to spot fake reviews*. About Amazon. https://www.aboutamazon.com/news/policy-news-views/how-ai-spots-fake-reviews-amazon

Amazon. (2025). *Amazon's latest actions against fake review brokers*. About Amazon. https://www.aboutamazon.com/news/policy-news-views/amazons-latest-actions-against-fake-review-brokers

Anonim. (2025). *Ignored by Amazon: Fake Reviews Are Ruining Our Business – Amazon hs done nothing for months.* Amazon Seller Forums. https://sellercentral.amazon.com/seller-forums/discussions/t/6c4fa06a-a6aa-49ca-92ea-c5278a23e1a8

Awalina, A., Bachtiar, F. A., & Utaminingrum, F. (2022). Perbandingan Pretrained Model Transformer Pada Deteksi Ulasan Palsu Comparison Of Pretrained Transformer Models On Spam Review Detection. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *9*(3), 597–604. https://doi.org/10.25126/jtiik.202295696

CapitalOne Shopping. (2025). *Fake Review Statistics 2025: on Amazon & Other Websites*. https://capitaloneshopping.com/research/fake-review-statistics/

Chatterjee, R., Pandey, M., Thakur, H. K., & Gupta, A. (2024). Checking Counterfeit Critiques on Commodities using Ensemble Classifiers Enhancing Information Credibility. *Procedia Computer Science*, *233*(2024), 570–579. https://doi.org/10.1016/j.procs.2024.03.246

Chen, T., Samaranayake, P., Cen, X., Qi, M., & Lan, Y.-C. (2022). The Impact of Online Reviews on Consumers' Purchasing Decisions: Evidence From an Eye-Tracking Study. *Frontiers in Psychology*, *13*. https://doi.org/10.3389/fpsyg.2022.865702

Clarisa, H., & Areta A, H. (2022). Fake Review and Liabilities Defect Goods in E-Commerce. *The Lawpreneurship Journal*, *2*(1), 19–42. https://doi.org/10.21632/tlj.2.1.19-42

Dataiku. (n.d.). *What Is a Large Language Model?* Dataiku. Diambil 2 Desember 2025, dari https://www.dataiku.com/stories/detail/what-is-a-large-language-model/

Deeplearning.ai. (2023). *Natural Language Processing (NLP)*. Deeplearning.ai. https://www.deeplearning.ai/resources/natural-language-processing/

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). *QLoRA: Efficient Finetuning of Quantized LLMs*. http://arxiv.org/abs/2305.14314

Dewi, S., & Firdaus, R. (2024). Pengaruh Pandemi Terhadap Perilaku Konsumen Dalam E-commerce. *MENAWAN : Jurnal Riset dan Publikasi Ilmu Ekonomi*, *2*(4), 89–92. https://doi.org/10.61132/menawan.v2i4.700

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., … Ma, Z. (2024). *The Llama 3 Herd of Models*. http://arxiv.org/abs/2407.21783

Hadi, Z., Nurkholis, L. M., Imran, B., Riadi, S., & Suryadi, E. (2025). Fake Review Detection on Digital Platforms Using The RoBERTa Model : A Deep Learning and NLP Approach. *Journal Computer and Technology*, *3*(1), 20–25. https://doi.org/https://doi.org/10.69916/comtechno.v3i1.355

Han, D., Han, M., & Unsloth Team. (2023). *Unsloth*. http://github.com/unslothai/unsloth

Kumar, J. (2020). *Fake Review Detection Using Behavioral and Contextual Features*. http://arxiv.org/abs/2003.00807

Li, J., Ott, M., Cardie, C., & Hovy, E. (2014). *Towards a General Rule for Identifying Deceptive Opinion Spam*. 1566–1576.

Maxwell, A. E., Warner, T. A., & Guillén, L. A. (2021). Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—part 2: Recommendations and best practices. *Remote Sensing*, *13*(13). https://doi.org/10.3390/rs13132591

Mir, A. Q., Khan, F. Y., & Chishti, M. A. (2023). *Online Fake Review Detection Using Supervised Machine Learning And BERT Model*. http://arxiv.org/abs/2301.03225

Mohawesh, R., Salameh, H. B., Jararweh, Y., Alkhalaileh, M., & Maqsood, S. (2024). Fake review detection using transformer-based enhanced LSTM and RoBERTa. *International Journal of Cognitive Computing in Engineering*, *5*, 250–258. https://doi.org/10.1016/j.ijcce.2024.06.001

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, *1*, 309–319.

Papageorgiou, E., Chronis, C., Varlamis, I., & Himeur, Y. (2024). A Survey on the Use of Large Language Models (LLMs) in Fake News. *Future Internet*, *16*(8), 298. https://doi.org/10.3390/fi16080298

PEFT. (n.d.). *LoRA Documentation*. Diambil 5 Februari 2026, dari https://huggingface.co/docs/peft/en/package_reference/lora

Purifyregalia, K., Umam, K., Wibowo, N. C. H., & Handayani, M. R. (2025). Detecting Fake Reviews in E-Commerce: A Case Study on Shopee Using Support Vector Machine and Random Forest. *Journal of Applied Informatics and Computing*, *9*(3), 955–965. https://doi.org/10.30871/jaic.v9i3.9514

Qamar, R., & Zardari, B. A. (2023). Artificial Neural Networks: An Overview. *Mesopotamian Journal of Computer Science*, *2023*, 124–133. https://doi.org/10.58496/MJCSC/2023/015

Raju, M. S. U. (2025). *Binary Cross Entropy/Log Loss for Binary Classification*. GeeksforGeeks. https://www.geeksforgeeks.org/deep-learning/binary-cross-entropy-log-loss-for-binary-classification/

Salminen, J., Kandpal, C., Kamel, A. M., Jung, S., & Jansen, B. J. (2022). Creating and detecting fake reviews of online products. *Journal of Retailing and Consumer Services*, *64*, 102771. https://doi.org/10.1016/j.jretconser.2021.102771

Santoso, J. T. (2023). Kecerdasan Buatan. In *Yayasan Prima Agus Teknik*.

Shafelbilyunazra, A., & Prasetya, D. D. (2025). Pengaruh Few-shot Learning pada Kinerja LLM untuk Ekstraksi Entitas Iklan Lowongan Kerja. *Jurnal Pusat Akses Kajian Teknologi Artificial Intelligence*, *5*(2), 418–427.

Sripathi, S., & Shanthi, P. M. (2025). AI Based Knowledge Representation on Spam Review Detection by Bottom -Up Learning Approach. *Communications on Applied Nonlinear Analysis*, *32*(9), 1829–1843.

Stryker, C., & Holdsworth, J. (n.d.). *What Is NLP (Natural Language Processing)?* IBM. Diambil 28 November 2025, dari https://www.ibm.com/think/topics/natural-language-processing

Sun, P., Bi, W., Zhang, Y., Wang, Q., Kou, F., Lu, T., & Chen, J. (2024). Fake Review Detection Model Based on Comment Content and Review Behavior. *Electronics (Switzerland)*, *13*(21). https://doi.org/10.3390/electronics13214322

Sutoyo, E. (2023). *Apa Itu Natural Language Processing?* Edinesia. https://edinesia.com/apa-itu-natural-language-processing/

Syaifudin, A., Yahya, M., Putri, N., Angelica, R., Khalil, S., & Ardina, T. (2025). E-commerce dalam transformasi digital: Menciptakan peluang baru berkelanjutan untuk bisnis modern Info Artikel ABSTRAK. *Jurnal Pembelajaran Pemberdayaan Masyarakat (JP2M)*, *6*(4), 1074–1084. https://doi.org/10.33474/jp2m.v6i4.23426

Unsloth. (2025). *LoRA Hyperparameters Guide*. Unsloth. https://docs.unsloth.ai/get-started/fine-tuning-llms-guide/lora-hyperparameters-guide

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010. http://arxiv.org/abs/1706.03762

Wang, L., Chen, S., Jiang, L., Pan, S., Cai, R., Yang, S., & Yang, F. (2025). Parameter-efficient fine-tuning in large language models: a survey of methodologies. *Artificial Intelligence Review*, *58*(8), 1–64. https://doi.org/10.1007/s10462-025-11236-4

Wang, Z. (2025). Sentiment Analysis of Mobile Phone Reviews Using XGBoost and Word Vectors. *ITM Web of Conferences*, *70*, 03018. https://doi.org/10.1051/itmconf/20257003018

We Are Social. (2025). *Digital 2026 Global Overview Report*. https://datareportal.com/reports/digital-2026-global-overview-report