BAB 2
LANDASAN TEORI

2.1 Penelitian Pendahulu

Tinjauan pustaka ini dilakukan untuk memetakan posisi penelitian saat ini dibandingkan dengan penelitian-penelitian sebelumnya. Tabel 2.1 menyajikan ringkasan penelitian terdahulu yang membahas ekstraksi data, pemrosesan dokumen hukum, dan implementasi *Large Language Models* (LLM).

Tabel 2.1. Ringkasan Perbandingan Penelitian Pendahulu

No	Aspek / Fitur Utama	P1	P2	P3	P4	P5	P6	Penelitian Ini
1	Domain Hukum / Internasional (RTA)	×	✓	✓	✓	×	✓	✓
2	Penggunaan *Large Language Model*	×	×	✓	✓	✓	✓	✓
3	*Advanced Layout Parsing* (Docling)	×	✓	×	×	×	×	✓
4	*Orchestration Framework* (LangChain)	×	×	×	×	×	✓	✓
5	Output Terstruktur (SQL / JSON Schema)	✓	✓	×	×	✓	✓	✓
6	*Sequential Context Handling*	×	×	✓	×	×	×	✓

Keterangan Jurnal:

1. P1: Malashin et al. (2024) - *Image Text Extraction and NLP of Medical Reports* [13].
2. P2: Garofalakis et al. (2020) - *Transformation of Greek Legal Documents into Open Data* [3].
3. P3: Jayapradha et al. (2024) - *CodifiedCant: Enhancing Legal Document Accessibility* [4].
4. P4: Anwansedo et al. (2024) - *AI-Enhanced Online Marketplaces for Trade Relations* [5].
5. P5: Naranbaatar et al. (2025) - *LLM used in traffic accident data extraction* [8].
6. P6: Seabra et al. (2024) - *Contrato360 2.0: LLM and Agents* [9].

Berdasarkan tinjauan pada Tabel 2.1, terlihat pola bahwa penelitian sebelumnya telah mengeksplorasi ekstraksi informasi otomatis dari berbagai domain. Penelitian oleh Malashin et al. (2024) menunjukkan efektivitas

pemrosesan dokumen medis melalui optimasi OCR dan teknik NLP [13]. Di domain hukum, Garofalakis et al. (2020) menekankan perlunya transformasi dokumen tidak terstruktur menjadi format yang dapat diproses oleh mesin untuk mendukung transparansi pemerintah [3].

Penggunaan LLM mulai mendominasi literatur terbaru, seperti pada penelitian Naranbaatar et al. (2025) yang mengevaluasi berbagai model untuk ekstraksi data kecelakaan lalu lintas, namun mencatat adanya tantangan pada pemahaman urutan peristiwa (*event sequencing*) yang menunjukkan bahwa LLM masih kesulitan menjaga konsistensi konteks sekuensial pada dokumen naratif panjang [8]. Dalam manajemen kontrak, Seabra et al. (2024) mengimplementasikan sistem *multi-agent* dan *text-to-SQL* untuk meningkatkan akurasi jawaban atas dokumen kontrak, namun pendekatan QA yang digunakan berfokus pada kontrak berstruktur datar dan tidak mencakup mekanisme pemrosesan dokumen hierarkis berdimensi panjang seperti RTA [9]. Jayapradha et al. (2024) mencoba mengatasi kompleksitas dokumen hukum panjang menggunakan arsitektur *Longformer* untuk summarization, tetapi tidak melakukan ekstraksi terstruktur ke dalam skema basis data relasional [4]. Sementara itu, Garofalakis et al. (2020) mentransformasikan dokumen hukum Yunani ke format *Open Data* berbasis standar Akoma Ntoso, namun belum memanfaatkan kemampuan semantik LLM maupun mekanisme *chunking* untuk menjaga integritas konteks antar segmen dokumen [3].

Berdasarkan tinjauan tersebut, terdapat celah penelitian (*research gap*) yang signifikan, yaitu belum adanya integrasi antara *advanced layout parsing* yang memahami struktur visual dokumen (seperti Docling) dengan mekanisme *sequential context handling* pada dokumen *Regional Trade Agreement* (RTA) yang berdimensi panjang dan memiliki struktur hierarkis mendalam. Penelitian ini bertujuan untuk mengisi celah tersebut dengan mengimplementasikan *Sequential Chain* pada *framework* LangChain untuk mentransformasikan teks RTA menjadi basis data relasional PostgreSQL yang utuh dan terstruktur.

2.2 Large Language Model (LLM)

Large Language Model (LLM) merupakan model jaringan saraf dalam (*deep neural network*) yang menandai era baru dalam bidang *Natural Language Processing* (NLP) [14]. Berbeda dengan metode NLP tradisional yang mengandalkan aturan manual (*handcrafted rules*), LLM memiliki kemampuan

untuk memahami dan menghasilkan teks manusia dengan kompleksitas tinggi melalui pelatihan pada dataset berskala masif [15]. Kemampuan ini memungkinkan model menangkap nuansa linguistik dan pola semantik yang mendalam secara otomatis langsung dari data teks [14].

Secara teknis, arsitektur LLM didasarkan pada model *Transformer* yang umumnya terdiri dari dua komponen utama, yaitu *encoder* dan *decoder* [15]. *Encoder* bertanggung jawab untuk memproses teks input menjadi representasi vektor yang kaya makna, sementara *decoder* menggunakan representasi tersebut untuk menghasilkan teks output secara sekuensial. Keunggulan utama arsitektur ini terletak pada mekanisme *self attention*, yang memungkinkan model untuk secara bersamaan mempertimbangkan seluruh kata dalam suatu urutan teks dan memberikan bobot perhatian selektif pada informasi yang paling relevan untuk memahami konteks global [15].

Istilah "*large*" pada LLM merujuk pada skala parameter model yang mencapai puluhan hingga ratusan miliar bobot yang dioptimalkan melalui proses *next word prediction* [14]. Hal ini menempatkan LLM sebagai bagian dari *Generative Artificial Intelligence* (GenAI), di mana model tidak hanya melakukan klasifikasi, tetapi mampu menciptakan konten baru yang sesuai konteks. Perkembangan terbaru bahkan telah melahirkan model dengan kemampuan *multi step reasoning* yang memperluas LLM ke dalam penyelesaian masalah logika, matematika terstruktur, serta penalaran simbolik [15].

Dalam konteks ekstraksi informasi dari dokumen tidak terstruktur, keunggulan utama LLM dibandingkan pendekatan berbasis aturan (*rule based*) dan *regular expression* (*regex*) terletak pada kemampuannya melakukan pemahaman semantik struktural yaitu memahami makna dan hubungan antar elemen dokumen dari konteks naratif, bukan sekadar mencocokkan pola teks secara literal. Penelitian oleh Aggarwal et al. (2025) pada ekstraksi dokumen fiskal pemerintah berskala besar (200+ halaman) mendemonstrasikan bahwa LLM mampu merekonstruksi struktur hierarkis dokumen, mulai dari *Major Head*, *Minor Head*, hingga *Detailed Head* dengan akurasi *Tree Edit Distance Similarity* (TEDS) mencapai 73% hingga 96%, dan secara eksplisit menyimpulkan bahwa *"LLMs do much more than regular expression matching"* [16]. Lebih lanjut, studi komparatif oleh Zang et al. (2025) yang membandingkan lima pendekatan *rule based*, *supervised machine learning*, *fine tuned encoder*, *fine tuned decoder LLM*, dan *zero shot decoder LLM* pada tugas klasifikasi dari teks hukum menemukan bahwa pendekatan berbasis aturan hanya unggul pada tugas dengan pola linguistik yang sangat teratur (*formulaic*), sementara

LLM secara konsisten mengungguli metode lainnya pada tugas yang memerlukan inferensi semantik dan pemahaman konteks implisit [17].

Kedua temuan ini memiliki implikasi langsung terhadap permasalahan ekstraksi dokumen RTA. Regex dapat mendeteksi keberadaan string seperti `Article 5.2` atau `Chapter I`, namun tidak mampu menentukan bahwa `Article 5.2` merupakan anak dari `Chapter I` dan induk dari `Paragraph (a)` informasi yang hanya dapat diinferensi dari posisi naratif dan konteks di sekitarnya. Demikian pula, *parser* berbasis aturan harus ditulis ulang untuk setiap variasi format penulisan RTA, menjadikannya tidak *scalable* untuk korpus lebih dari 400 dokumen dengan konvensi yang berbeda-beda [18]. Kemampuan LLM dalam memahami struktur dokumen dari isyarat spasial seperti indentasi dan pemisah baris (*line breaks*), sebagaimana ditunjukkan oleh Li et al. (2024), semakin memperkuat argumen bahwa LLM dapat menangkap hierarki dokumen tanpa aturan eksplisit yang di-*hardcode* [19]. Bahkan pada sisi *layout analysis* visual, penelitian terbaru oleh Shehzadi et al. (2024) menunjukkan bahwa model *state of the art* untuk deteksi elemen dokumen hanya mencapai akurasi rata-rata 81,6% pada *benchmark* DocLayNet [20], mengindikasikan adanya celah akurasi yang tidak dapat ditutup hanya dengan analisis tata letak visual dan memerlukan pemahaman semantik dari LLM. Oleh karena itu, dalam penelitian ini LLM tidak digunakan untuk interpretasi konten hukum (*legal reasoning*) melainkan sebagai mesin pemahaman semantik struktural yang mampu mengenali dan memetakan hierarki dokumen dari teks naratif, sebuah kapabilitas yang menjadi fondasi bagi transformasi dokumen RTA tidak terstruktur menjadi basis data relasional yang *machine processable*.

2.3 Docling

Docling merupakan perangkat lunak sumber terbuka (*open source toolkit*) yang dikembangkan oleh IBM Research untuk melakukan konversi dokumen tidak terstruktur ke dalam format yang terstruktur dan siap digunakan dalam aplikasi AI [21]. Berbeda dengan alat ekstraksi teks konvensional, *Docling* menggunakan model kecerdasan buatan khusus untuk memahami struktur visual dokumen secara mendalam.

Dua komponen utama yang mendasari kemampuan *Docling* adalah:

1. DocLayNet: Model yang digunakan untuk *layout analysis*, yang mampu mengidentifikasi elemen-elemen dokumen seperti judul, sub judul, paragraf, dan keterangan gambar secara akurat.

2. TableFormer: Model khusus untuk *table structure recognition* yang mampu merekonstruksi tabel kompleks dari format PDF kembali ke dalam struktur data yang sesuai dengan konteks.

Dalam alur kerja ekstraksi informasi, *Docling* berfungsi sebagai lapisan transformasi yang mengubah dokumen PDF asli menjadi representasi terpadu dalam format Markdown. Format Markdown dipilih karena mampu mempertahankan informasi semantik dan struktural dokumen tanpa beban metadata yang berlebihan, sehingga memudahkan *Large Language Model* (LLM) dalam memahami hierarki dan konteks teks [21]. Integrasi *Docling* dengan *framework* orkestrasi seperti LangChain memungkinkan proses ekstraksi dokumen legal berukuran besar dilakukan secara efisien dengan konsumsi sumber daya komputasi yang minimal.

2.4 Framework LangChain

LangChain merupakan *framework* orkestrasi yang dirancang untuk memfasilitasi pengembangan aplikasi berbasis *Large Language Model* (LLM). Munculnya aplikasi berbasis LLM menandai paradigma perangkat lunak baru yang menggabungkan kekuatan penalaran LLM dengan logika program konvensional untuk menjalankan tugas-tugas kompleks [22]. Dalam arsitektur sistem modern, *LangChain* berfungsi sebagai pengatur alur kerja (*orchestration framework*) yang menghubungkan LLM dengan komponen eksternal seperti basis data dan modul pemrosesan dokumen [23].

Salah satu fitur utama yang digunakan dalam penelitian ini adalah *Recursive Character Text Splitter*. Komponen ini bekerja dengan memecah teks secara hierarkis menggunakan sekumpulan karakter pemisah yang memiliki urutan prioritas, yaitu `\n##` (header level 2 Markdown), `\n\n` (pemisah antar paragraf), `\n` (pemisah antar baris), dan spasi. LangChain akan mencoba memecah teks pada separator dengan prioritas tertinggi terlebih dahulu, sehingga batas antar *Article* atau *Chapter* yang ditandai oleh header Markdown cenderung dipertahankan utuh. Hanya ketika suatu segmen masih melebihi batas *chunk_size* meskipun sudah dipecah di level header tertinggi, sistem akan turun ke separator berikutnya secara bertahap. Mekanisme ini memastikan bahwa setiap fragmen (*chunk*) tetap mempertahankan keutuhan semantik dan struktur pasal dalam dokumen hukum tidak terpisah di tengah kalimat, sehingga LLM dapat memahami konteks secara utuh dalam batasan jendela konteks yang tersedia.

Selain itu, penelitian ini memanfaatkan mekanisme *Chaining* dan *Prompt Templates*. *Prompt Templates* memungkinkan standarisasi instruksi yang dikirimkan ke LLM, sementara *Output Parsers* yang terintegrasi dengan pustaka *Pydantic* menjamin bahwa hasil ekstraksi memiliki skema data yang konsisten (seperti format JSON). Melalui *Sequential Chain*, sistem dapat memproses setiap fragmen dokumen secara berurutan dengan menyertakan informasi dari fragmen sebelumnya (*previous context*) sebagai referensi tambahan. Pendekatan sekuensial ini menjamin integritas urutan informasi pada dokumen *Regional Trade Agreements* (RTA), di mana interpretasi suatu klausul seringkali bergantung pada ketentuan yang telah didefinisikan di bagian sebelumnya [23]. Dengan demikian, dalam *pipeline* penelitian ini LangChain menjalankan tiga fungsi utama, yaitu sebagai *orchestrator* yang merangkai *prompt template*, model LLM, dan *output parser* melalui mekanisme *LCEL chain*; sebagai *memory manager* yang meneruskan *previous_context* antar iterasi *chunk* untuk menjaga kontinuitas struktur hierarkis; serta sebagai *prompt chaining engine* yang mengelola eksekusi sekuensial setiap *chunk* secara berurutan dari awal hingga akhir dokumen.

2.5 Regional Trade Agreements

Regional Trade Agreements (RTA) didefinisikan oleh *World Trade Organization* (WTO) sebagai perjanjian perdagangan timbal balik antara dua atau lebih mitra dagang yang tidak terbatas pada wilayah geografis yang sama [18]. Berdasarkan ketentuan WTO, setiap RTA yang berlaku harus melalui proses notifikasi dan transparansi sesuai dengan Pasal XXIV *General Agreement on Tariffs and Trade* (GATT) 1994, yang menuntut standarisasi dalam pelaporan teks hukum perjanjian tersebut.

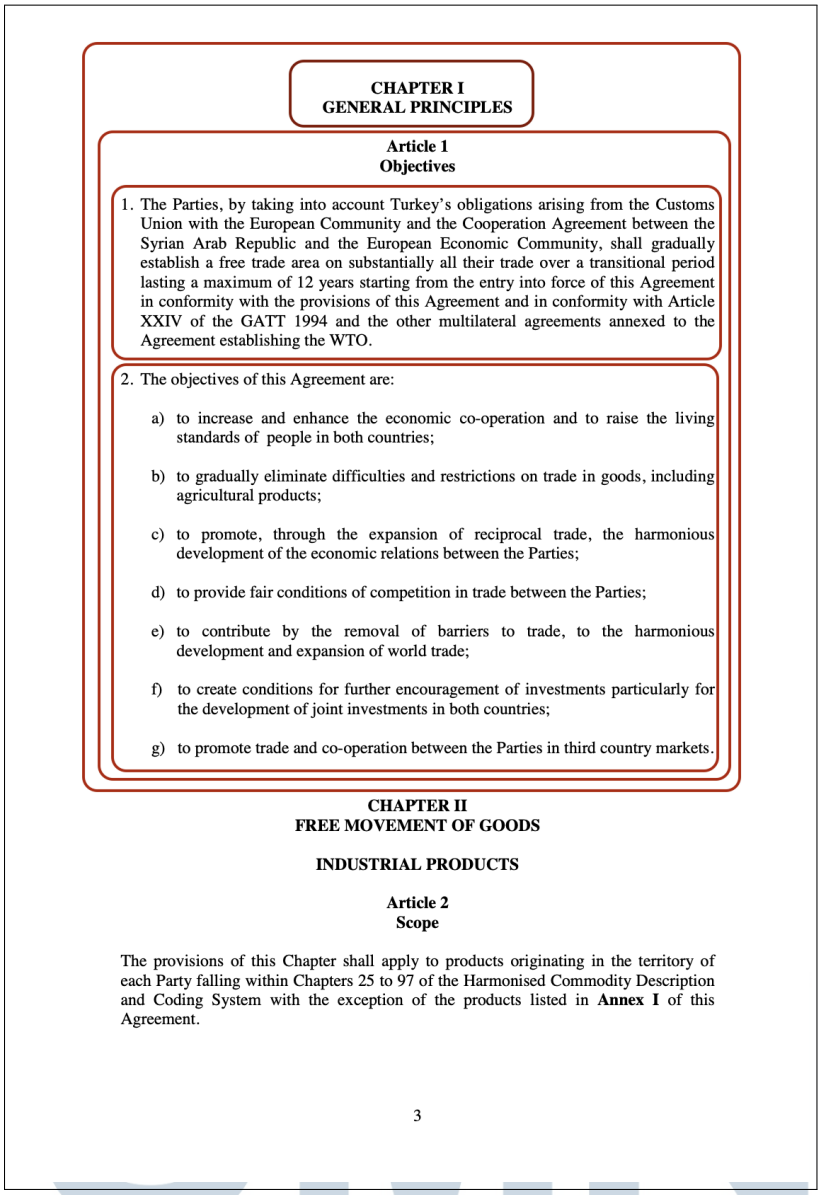
Dalam perspektif informatika hukum (*legal informatics*), dokumen RTA memiliki karakteristik unik berupa struktur data yang terstandarisasi [24]. Penggunaan bahasa yang bersifat *boilerplate* atau berulang menunjukkan bahwa RTA dapat didekonstruksi menjadi modul-modul teks yang dapat direpresentasikan sebagai data terstruktur melalui analisis komputasional [24].

Selain itu, kebutuhan akan representasi pengetahuan hukum (*legal knowledge representation*) menuntut adanya pemetaan informasi yang hierarkis untuk menjaga hubungan logis antar komponen dokumen, mulai dari tingkat makro hingga tingkat mikro [24]. Tantangan dalam manajemen informasi hukum muncul akibat fenomena *spaghetti bowl*, di mana terdapat tumpang tindih aturan

yang kompleks antar berbagai perjanjian [25]. Hal ini memberikan urgensi pada pengembangan sistem ekstraksi otomatis menggunakan *Large Language Model* (LLM) untuk mentransformasikan teks naratif RTA yang masif menjadi skema basis data relasional. Pendekatan komputasional ini memungkinkan analisis kebijakan dilakukan pada skala besar (*scalable*) dengan tingkat presisi yang dapat dibandingkan dengan metode input manual [24].

Meskipun dokumen RTA mengikuti konvensi penulisan perjanjian internasional, terdapat sejumlah ciri format visual dan logika hukum yang membedakannya dari dokumen legal umum dan menjadi penyebab kegagalan alat ekstraksi generik. Dari segi format visual, dokumen RTA memiliki struktur penomoran bersarang (*nested numbering*) yang dapat mencapai tiga hingga empat level kedalaman, seperti *Chapter* $\rightarrow$ *Article* $\rightarrow$ *Paragraph (a)* $\rightarrow$ *Sub paragraph (i)*. Setiap level memiliki konvensi penomoran yang berbeda (angka Romawi, Arab, alfabet, dan Romawi kecil), sehingga hubungan *parent child* antar elemen menjadi kompleks untuk direkonstruksi oleh *parser* berbasis aturan. Selain itu, dokumen RTA seringkali menggabungkan berbagai jenis konten dalam satu halaman, mulai dari narasi hukum yang padat, daftar berpoin (*enumerated list*), hingga tabel tarif dan daftar konsesi yang terintegrasi dalam teks pasal. Gambar 2.1 mengilustrasikan kompleksitas format visual ini pada dokumen Turkey Syria, di mana elemen *Section*, *Sub section*, *Chapter*, dan *Article* muncul dalam urutan yang tidak konvensional pada satu halaman yang sama.

Lebih lanjut, struktur hierarki dokumen RTA tidak seragam antar perjanjian. Dokumen dengan volume pendek seperti Armenia Ukraine (4 halaman) [26] dan Armenia Moldova (6 halaman) [27] hanya memiliki *Article* tanpa *Chapter*, *Section*, maupun *Part*. Dokumen berskala menengah seperti SPARTECA (13 halaman) [28] dan PATCRA (10 halaman) [29] umumnya mengikuti pola *Chapter* $\rightarrow$ *Article*. Sementara itu, dokumen berskala besar seperti Turkey Syria (22 halaman) [30] memiliki urutan *Section* $\rightarrow$ *Sub section* $\rightarrow$ *Chapter* $\rightarrow$ *Article* yang tidak selalu *top down*, di mana *Section* muncul sebelum *Chapter*. Dokumen dengan volume sangat besar seperti China Serbia (46 halaman) [31] mencakup seluruh level hierarki dari *Preamble*, *Part*, *Chapter*, *Section*, *Article*, hingga *Annex*. Ketidakseragaman ini menjadi tantangan signifikan bagi *parser* berbasis aturan yang mengasumsikan struktur tetap pada setiap dokumen.



CHAPTER I
GENERAL PRINCIPLES

Article 1
Objectives

1. The Parties, by taking into account Turkey's obligations arising from the Customs Union with the European Community and the Cooperation Agreement between the Syrian Arab Republic and the European Economic Community, shall gradually establish a free trade area on substantially all their trade over a transitional period lasting a maximum of 12 years starting from the entry into force of this Agreement in conformity with the provisions of this Agreement and in conformity with Article XXIV of the GATT 1994 and the other multilateral agreements annexed to the Agreement establishing the WTO.

2. The objectives of this Agreement are:

a) to increase and enhance the economic co-operation and to raise the living standards of people in both countries;

b) to gradually eliminate difficulties and restrictions on trade in goods, including agricultural products;

c) to promote, through the expansion of reciprocal trade, the harmonious development of the economic relations between the Parties;

d) to provide fair conditions of competition in trade between the Parties;

e) to contribute by the removal of barriers to trade, to the harmonious development and expansion of world trade;

f) to create conditions for further encouragement of investments particularly for the development of joint investments in both countries;

g) to promote trade and co-operation between the Parties in third country markets.

CHAPTER II
FREE MOVEMENT OF GOODS

INDUSTRIAL PRODUCTS

Article 2
Scope

The provisions of this Chapter shall apply to products originating in the territory of each Party falling within Chapters 25 to 97 of the Harmonised Commodity Description and Coding System with the exception of the products listed in **Annex I** of this Agreement.

3

Gambar 2.1. Contoh Kompleksitas Format Visual Dokumen RTA (Turkey Syria Agreement) dengan Struktur Penomoran Bersarang

Dari segi logika hukum, dokumen RTA menggunakan *defined terms* di bagian awal perjanjian (umumnya Article 1) yang berlaku dan direferensi di seluruh dokumen, sehingga kesalahan interpretasi pada satu istilah kunci akan terpropagasi ke seluruh hasil ekstraksi. Variasi format hierarki yang tidak seragam antar RTA, sebagaimana dijelaskan sebelumnya, semakin memperberat tantangan ekstraksi karena tidak ada satu aturan *parsing* yang dapat berlaku universal untuk seluruh korpus dokumen. Karakteristik inilah yang menyebabkan *tool* ekstraksi generik

gagal menangani dokumen RTA secara memadai. Alat berbasis *Optical Character Recognition* (OCR) konvensional hanya mampu mengekstraksi teks sebagai aliran datar (*flat text*) tanpa informasi struktur hierarkis, sehingga informasi *parent child* lenyap sepenuhnya. *Parser* berbasis *regular expression* (regex) tidak mampu menangkap variasi pola penomoran bersarang yang berbeda antar perjanjian, dan setiap kali format penulisan berubah, aturan regex harus ditulis ulang dari awal sehingga pendekatan ini tidak *scalable* untuk korpus RTA yang berjumlah lebih dari 400 dokumen. Demikian pula, pustaka ekstraksi teks PDF standar seperti PyPDF2 atau pdfplumber dapat mengambil konten tekstual, namun kehilangan konteks tata letak (*layout context*) yang krusial untuk menentukan hubungan hierarkis antar elemen dokumen. Oleh karena itu, diperlukan pendekatan yang menggabungkan *advanced layout parsing* berbasis model AI (seperti Docling dengan DocLayNet) untuk mengenali struktur visual dokumen, serta *Large Language Model* untuk memahami hierarki dokumen dari konteks naratif yang tidak dapat ditangkap hanya dari informasi posisional semata.

2.6 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) adalah salah satu metode yang digunakan untuk mengukur performa model. Secara matematis, formulasi standar untuk menghitung nilai agregasi MAE didefinisikan pada Persamaan 2.1:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.1}$$

Dimana:

1. n = jumlah total elemen baris atau variabel data yang dievaluasi dalam basis data.
2. y_i = nilai representasi ideal dari *ground truth* pada observasi ke-i.
3. $\hat{y}_i$ = nilai status ketepatan hasil ekstraksi sistem pada observasi ke-i.

Secara konseptual, MAE mengukur seberapa jauh estimasi atau nilai prediksi yang dihasilkan oleh sistem meleset dari kondisi sebetulnya [32]. Setiap kali model melakukan ekstraksi atau memprediksi suatu nilai ($\hat{y}_i$), nilai tersebut akan dibandingkan langsung dengan nilai *ground truth* asli (y_i) yang bertindak sebagai acuan ideal. Selisih dari kedua variabel tersebut dihitung dalam bentuk

fungsi nilai absolut $|y_i - \hat{y}_i|$, sehingga evaluasi hanya berfokus pada besaran atau jarak kesalahan tanpa memedulikan arah deviasi baik bernilai positif maupun negatif. Seluruh akumulasi kesalahan absolut dari total n baris data kemudian dijumlahkan melalui operasi sigma ($\sum$) dan dibagi dengan ukuran sampel untuk menghasilkan satu nilai rata-rata kesalahan tunggal. Karakteristik utama dari metrik ini adalah pemberian bobot kesalahan yang setara untuk setiap baris data, menjadikannya sangat intuitif dan objektif dalam merepresentasikan rata-rata tingkat ketidaktepatan sistem secara keseluruhan [32]. Disini, semakin kecil nilai MAE yang diperoleh (mendekati nilai nol), maka tingkat akurasi dan ketepatan sistem dalam mengekstrak informasi dinilai semakin tinggi.