BAB 2
TINJAUAN PUSTAKA

2.1. Penelitian yang Relevan

2.1.1. Matriks Penelitian

Beberapa penelitian yang relevan membahas topik peningkatan kemampuan pemahaman LLM pada domain hukum khususnya terkait dengan teks kontrak dan kecerdasan buatan penalaran hukum, diantaranya sebagai berikut:

Penulis	Judul	Pendekatan dan Hasil
Zheng dkk., (2025)	*Automating construction contract review using knowledge graph-enhanced large language models*	Menggunakan metode integrasi antara LLM dengan *GraphRAG* untuk meningkatkan akurasi dan interpretasi dalam proses automasi identifikasi risiko kontrak. Pendekatan yang dilakukan bersifat *tuning-free* dengan melakukan retrieval dan penalaran klausul kontrak (Zheng dkk., 2025).
Gao dkk., (2025)	*Application of large language models to intelligently analyze long construction contract texts*	Melakukan metode segmentasi semantik menggunakan model embedding dari OpenAI (text-embedding-ada-002) untuk memadatkan/ kompresi teks panjang pada kontrak konstruksi (FIDIC Red Book, kontrak konstruksi standar internasional) untuk memuat teks kontrak panjang (sebesar 69k berdasarkan perhitungan token GPT-2) ke dalam panjang konteks maksimal 8k pada GPT-4 (Gao dkk., 2025). Tujuan akhir dari penerapan metode ini adalah aplikasi QA dalam rangka menyediakan informasi terkait isi kontrak.
Wong dkk., (2024)	*Construction contract risk identification based on knowledge-augmented language models*	Memanfaatkan logika berpikir pakar kontrak konstruksi dan menyimpan aturan-aturannya dalam data vektor sebagai *knowledge base* bagi LLM menganalisa klausul. Satu per satu klausul yang cocok di-retrieve untuk dibandingkan dengan draft klausul, untuk selanjutnya ditentukan analisa terbaik dengan metoda voting (Wong dkk., 2024).

Wang dkk., (2024)	*Prompts and Large Language Models: A New Tool for Drafting, Reviewing and Interpreting Contracts?*	Membahas implikasi hukum penggunaan prompt dan LLM dalam *drafting, review*, dan interpretasi kontrak, dengan menyoroti status *prompt* dalam kaitannya dengan *evidence rule* serta perlunya dokumentasi *prompt* sebagai bagian dari praktik kontraktual modern, serta untuk menjaga kerahasiaan dan privasi data dapat diimplementasikan LLM *on-premise.* (Wang, Brydon T, 2024).
Shu dkk., (2024)	*LawLLM: Law Large Language Model for the US Legal System*	Pelatihan model LawLLM menggunakan pendekatan multi-tugas yang mencakup *Similar Case Retrieval* (SCR), *Precedent Case Recommendation* (PCR), dan *Legal Judgment Prediction* (LJP). Memproses data hukum mentah melalui teknik preprocessing khusus, seperti sumarisasi perkara dan ekstraksi putusan, serta menggunakan metode pembelajaran dalam konteks *(in-context learning)* untuk meningkatkan akurasi penalaran. Menggunakan GPT-4 untuk mengekstrak infromasi perkara (Shu dkk., 2024).
Kasundra dkk., (2024)	*Adapting Open-Source LLMs for Contract Drafting and Analyzing Multi-Role vs. Single-Role Behavior of ChatGPT for Synthetic Data Generation*	Menggunakan ChatGPT dan GPT-4 dengan strategi prompting yang dirancang khusus untuk menggenerasi konten hukum sintetis namun tetap realistik dalam membuat dataset tugas drafting kontrak, dengan keragaman ±107 kontrak sebagai kombinasi dari industri (15 jenis industri) dan jenis kontrak (setiap industri 3-4 jenis kontrak). Strategi *single-role scenario* dinilai lebih efektif untuk menghasilkan data sintetis berkualitas tinggi, daripada *multi-role scenario* (Kasundra & Dhankhar, 2024). Dataset sintetis yang digenerasi tersebut digunakan untuk melakukan *fine-tuning* model *open source* yang lebih kecil (13 miliar parameter) seperti LLaMA2 atau Nous.
Jayesh dkk., (2024)	*LAWTRIX: NLP Powered Legal Revolution*	LawTrix berinteraksi dengan user melalui suatu bot. Kemudian bot berinteraksi dengan mesin inferensi NLP yang melakukan retrieval terhadap *knowledge base*. Jika dibutuhkan, LLM akan melakukan penerjemahan bahasa bila input menggunakan bahasa tertentu (Jayesh Kirtane, 2024).

Dahl dkk., (2024)	*Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models*2/23/26 11:20:00 AM	Membandingkan jawaban LLM yang menggunakan *reference-based* dan *reference-free* untuk mendeteksi dan mengukur tingkat halusinasi. LLM juga diuji dengan pertanyaan yang memiliki premis salah untuk menguji bias kontra-faktual, serta dievaluasi tingkat kepercayaan diri LLM dalam menjawab pertanyaan hukum. Tahapan ini perlu untuk menjaga *integrity* kebenaran jawaban (Dahl dkk., 2024).
Parsa dkk., (2023)	*Legal Tech, the Law Firm and the Imagination of the Right Legal Answer*	Sistem pakar hukum berbasis *rule-based* dianggap dapat mengurangi kreativitas dan interpretasi hukum karena tidak cukup untuk membangun analisis kontekstual. Untuk mendapatkan sistem kepakaran hukum berbasis komputer, diperlukan mekanisme kontrol atas proses otomatisasi yang menjadi *man-in-the-middle* untuk mempertahankan kreativitas dan interpretasi hukum *lawyer* yang dinamis (Parsa dkk., 2023).
Kwon-Yan Lam dkk., (2023)	*Applying Large Language Models for Enhancing Contract Drafting*	Menggunakan LLM untuk menghasilkan klausul kontrak berdasarkan *prompt* pengguna, kemudian dilakukan validasi klausul dengan membandingkannya terhadap klausul serupa dari sumber terpercaya yang tersedia dalam bentuk representasi vektor (Kwok-Yan Lam dkk., 2023). Hasil validasi dicek ulang secara manual oleh manusia.
Janatian dkk., (2023)	*From Text to Structure: Using Large Language Models to Support the Development of Legal Expert Systems*	Menggunakan LLM (GPT-4) untuk memproses otomatis teks peraturan sehingga menghasilkan representasi hukum terstruktur. Untuk pembuatan draft jalur/alur logika pakar menggunakan JusticeCreator Automatic Pathway Generator (JCAPG). Pendekatan otomatis ini mempercepat ekstraksi alur logika dari peraturan yang ada, untuk kemudian dapat digunakan lebih lanjut sebagai *augmented intelligence* bagi sistem pakar untuk menghubungkan fakta dengan kesimpulan (Janatian dkk., 2023).
Schilder (2023)	*Legal Expertise Meets Artificial Intelligence: A Critical Analysis of*	Mengintegrasikan *Intelligent Assistance* dalam *framework* kecerdasan buatan yang menggunakan penalaran LLM serta dilengkapi dengan

	Large Language Models as Intelligent Assistance Technology	*knowledge base* untuk mencapai *augmented intelligence* (Frank Schilder, 2023).
Moon dkk., (2022)	*Automated system for construction specification review using natural language processing*	Otomatisasi dilakukan pendekatan bertahap, yaitu: (i) pengembangan *thesaurus* semantik khusus untuk istilah konstruksi dengan Word2Vec untuk memahami kosakata yg berneda; (ii) untuk pemahaman teks yang memiliki struktur berbeda, digunakan model *Named Entity Recognition* (NER). Pendekatan-pendekatan tersebut mengatasi konflik semantik yang ada di antara dokumen yang berbeda, sehingga sistem dapat dengan cepat meninjau spesifikasi teknis pada kontrak konstruksi (Moon dkk., 2022).
Huttner and Merigoux, (2022)	*Catala: Moving towards the future of legal expert systems*	Mengembangkan bahasa pemrograman yang spesifik pada domain hukum, dimana sintaksis dibuat eksplisit serta dirancang meniru struktur logika bahasa hukum. Bahasa ini digunakan untuk sistem pakar hukum berbasis aturan, terutama yang melibatkan perhitungan seperti perpajakan dan ketenagakerjaan. (Huttner & Merigoux, 2022).
Douglas Raevan Faisal, (2022)	*Towards Building a Legal Virtual Assistant Based on Knowledge Graphs*	Membangun *legal virtual assistant* (LVA) dengan 2 sistem terpisah: (i) konversi PDF menjadi *knowledge graph* (KG) menggunakan Lex2KG; (ii) retrieval dengan LVA menggunakan sumber data dari KG yang sudah dibuat. *Use case* yang diajukan khususnya terkait dengan QA, dengan pendekatan pada yurisdiksi hukum Indonesia (Douglas Raevan Faisal, 2022).
Caixia Zou dkk., (2022)	*The Rationale and Approach of the Legal Expert System Construction*	Menggunakan *reasoning* yang memungkinkan adanya pengecualian terhadap aturan atau kesimpulan yang telah ditetapkan. Pendekatan ini digunakan untuk menangani kompleksitas dan ketidakpastian yang sering muncul dalam proses hukum, di mana aturan atau kesimpulan dapat diubah atau dibatalkan jika ada informasi baru atau pengecualian yang relevan (Caixia Zou dkk., 2022).

Atkinson dkk., (2020)	*Explanation in AI and law: Past, present and future*	Menjabarkan pentingnya *contrastive explanation* dalam penalaran hukum dengan menggunakan *counterfactuals* dan *hypotheticals* (Atkinson dkk., 2020).

2.1.2. Kesenjangan Penelitian *(Research Gap)*

Berdasarkan telaah sistematis terhadap penelitian-penelitian terdahulu pada bidang pemanfaatan kecerdasan buatan dalam analisis kontrak dan penalaran hukum, dapat diidentifikasi bahwa meskipun telah terdapat kemajuan signifikan dalam aspek pemahaman semantik, reasoning berbasis LLM, serta integrasi *knowledge-based* systems, masih terdapat sejumlah kesenjangan fundamental yang belum terjawab secara memadai. Kesenjangan tersebut dapat diklasifikasikan sebagai berikut:

1. Kesenjangan Metodologi

Sejumlah penelitian terdahulu telah menunjukkan keberhasilan LLM dan sistem berbasis pengetahuan dalam menganalisis klausul kontrak, baik melalui integrasi knowledge graph (Zhang dkk., 2025; Faisal dkk., 2022), pendekatan *retrieval* berbasis vektor (Wong dkk., 2024; Kwok-Yan Lam dkk., 2023), maupun *prompting* dan *in-context learning* (Shu dkk., 2024; Kasundra & Dhankhar, 2024). Namun demikian, sebagian besar pendekatan tersebut secara implisit mengasumsikan bahwa *template* atau jenis kontrak telah diketahui sejak awal.

Penelitian seperti Gao dkk. (2025) dan Moon dkk. (2022) langsung memfokuskan analisis pada isi klausul kontrak setelah dokumen dianggap berada dalam domain yang benar (misalnya kontrak konstruksi atau spesifikasi teknis). Demikian pula, pendekatan *rule-based* dan *expert system* klasik (Huttner & Merigoux, 2022; Caixia Zou dkk., 2022) beroperasi dalam kerangka aturan yang telah ditentukan berdasarkan jenis dokumen tertentu.

Belum banyak penelitian yang secara eksplisit merancang pipeline deterministik bertahap yang memisahkan secara metodologis antara:

1. tahap pengenalan atau pemilihan template kontrak *(document-level decision)*, dan
2. tahap pemetaan atau pencocokan pasal *(clause-level decision)*.

Akibatnya, mekanisme pengambilan keputusan awal yang menjadi fondasi seluruh proses automasi *review* kontrak sering kali tidak dievaluasi secara terpisah, tidak

transparan, dan sulit direproduksi, terutama ketika pendekatan generatif LLM digunakan secara langsung.

2. Kesenjangan Kontekstual

Sebagian besar penelitian yang dikaji berfokus pada kontrak berbahasa Inggris dan yurisdiksi hukum tertentu, terutama Amerika Serikat atau konteks internasional seperti kontrak FIDIC (standar internasional yang diterbitkan oleh *Fédération Internationale des Ingénieurs-Conseils*, Federasi Internasional Insinyur Konsultan) (Gao dkk., 2025; Wong dkk., 2024; Moon dkk., 2022). Bahkan penelitian yang menekankan aspek penalaran hukum lanjutan, seperti LawLLM (Shu dkk., 2024), masih sangat bergantung pada karakteristik sistem hukum dan dataset Amerika Serikat.

Meskipun terdapat upaya pengembangan sistem berbasis *knowledge graph* untuk konteks Indonesia (Douglas Raevan Faisal, 2022), fokus utama penelitian tersebut masih pada question answering dan retrieval informasi, bukan pada pengenalan template kontrak dan pemetaan pasal secara sistematis.

Dengan demikian, terdapat kesenjangan kontekstual dalam pengembangan metode automasi *review* kontrak yang benar-benar mempertimbangkan karakteristik kontrak hukum berbahasa Indonesia, baik dari sisi terminologi, struktur dokumen, maupun praktik penyusunan kontrak yang berlaku di Indonesia.

3. Kesenjangan Fokus Analisis

Sebagian besar penelitian terdahulu mengevaluasi performa sistem AI hukum berdasarkan akurasi hasil akhir, kualitas jawaban, atau kesesuaian penalaran dengan pendapat pakar (Shu dkk., 2024; Wong dkk., 2024; Jayesh dkk., 2024). Namun, dalam konteks hukum yang menuntut kehati-hatian tinggi, pendekatan evaluasi tersebut belum cukup memadai.

Penelitian mengenai halusinasi dan bias LLM (Dahl dkk., 2024) telah menunjukkan bahwa model dapat memberikan jawaban yang tampak meyakinkan namun keliru. Meskipun demikian, masih jarang penelitian yang mengaitkan fenomena tersebut dengan analisis distribusi skor kemiripan, ketajaman pemisahan antara kandidat relevan dan tidak relevan, serta risiko *false positive*, khususnya pada tahap awal pengambilan keputusan.

Selain itu, evaluasi komparatif antara metode berbasis leksikal (misalnya Jaccard dan TF-IDF), *embedding* semantik, dan LLM umumnya dilakukan secara parsial, tanpa

membedakan karakteristik skor dan implikasi *threshold* keputusan pada level dokumen dan level pasal secara terpisah.

4. Kesenjangan dalam Aspek Determinisme dan Akuntabilitas Sistem

Pendekatan berbasis LLM menawarkan fleksibilitas tinggi dalam pemahaman konteks dan penalaran hukum (Schilder, 2023; Wang dkk., 2024), namun di sisi lain menimbulkan tantangan terkait determinisme, kontrol keputusan, dan akuntabilitas. Fenomena *overconfidence* dan halusinasi yang diidentifikasi oleh Dahl dkk. (2024) menunjukkan potensi risiko operasional yang signifikan apabila LLM digunakan tanpa mekanisme kontrol tambahan.

Sebaliknya, pendekatan *rule-based* atau *expert system* klasik (Parsa dkk., 2023; Huttner & Merigoux, 2022) bersifat deterministik dan transparan, namun sering dikritik karena kurang fleksibel dalam menangkap konteks hukum yang dinamis. Hingga saat ini, masih terbatas penelitian yang merancang mekanisme *hybrid* atau evaluatif yang mengombinasikan fleksibilitas LLM dengan kontrol deterministik berbasis *threshold*, skor kuantitatif, dan justifikasi yang dapat diaudit.

5. Kesenjangan Evaluasi Multi-Tahap *(Multi-Stage Decision)*

Mayoritas penelitian mengevaluasi sistem AI hukum sebagai satu kesatuan proses, tanpa memisahkan evaluasi pada setiap tahap pengambilan keputusan (Atkinson dkk., 2020; Wong dkk., 2024). Masih jarang penelitian yang secara sistematis mengevaluasi performa metode AI pada tahap pemilihan template kontrak dan tahap pemilihan atau pencocokan pasal dengan metrik, *threshold*, dan karakteristik skor yang disesuaikan dengan tujuan masing-masing tahap.

Berdasarkan kesenjangan-kesenjangan tersebut, diperlukan penelitian yang:

a. merancang *pipeline* automasi *review* kontrak bertahap;
b. mengevaluasi secara komparatif metode leksikal, *embedding* semantik, dan LLM;
c. menganalisis distribusi skor, *mean gap*, dan stabilitas keputusan; serta
d. menekankan konteks kontrak hukum berbahasa Indonesia.

Penelitian ini secara khusus diarahkan untuk mengisi kesenjangan tersebut dengan menempatkan karakteristik pengambilan keputusan deterministik sebagai prasyarat utama dalam sistem automasi *review* kontrak.

2.2. Landasan Teori

2.2.1. Anatomi kontrak

Untuk keperluan model dalam mendeteksi, mengkategorikan, serta memahami konteks klausul, penting untuk memahami dasar-dasar anatomi kontrak. Pada dasarnya anatomi kontrak terbagi dalam 3 bagian yaitu pendahuluan, isi, dan penutup, dengan penjelasan sebagai berikut (Salim H. S, 2019):

1. Bagian pendahuluan
    a. Pembuka: memuat sebutan atau nama kontrak, tanggal ditandatangani, tempat ditandatangani
    b. Identitas para pihak (komparisi): penyebutan identitas dan kapasitas para pihak, serta pendefinisiannya
    c. Penjelasan (premis): penjelasan mengapa kontrak dilakukan
2. Bagian isi
    a. Klausul definisi: berisi berbagai definisi untuk pengartian khusus seisi kontrak.
    b. Klausul transaksi: berisi tentang transaksi yang dilakukan, mencakup objek, nilai, mekanisme, dsb.
    c. Klausul spesifik: mengatur hal-hal spesifik terkait transaksi dan kewajiban-kewajiban seputar transaksi
    d. Klausul ketentuan umum (boilerplate): domisili, penyelesaian sengketa, pilihan hukum, korespondensi, pengakhiran, keseluruhan perjanjian, dsb.
3. Bagian penutup

    Terdapat 2 hal yang terncantum dalam bagian ini:
    a. Penutup: menerangkan perjanjian dan menyatakan ulang mengenai keterikatan para pihak secara sadar dan sukarela
    b. Ruang tanda tangan: tempat untuk para pihak menandatangani kontrak

Pada prakteknya, substansi dari setiap kontrak bisa berbeda. Namun demikian, pada prinspinya anatomi kontrak tersebut berlaku secara umum.

2.2.2. Kecerdasan Buatan

Kecerdasan buatan (*Artificial Intelligence*/AI) merupakan bidang ilmu yang berfokus pada pengembangan sistem komputasi yang mampu meniru kemampuan kognitif manusia, seperti penalaran, pembelajaran, dan pengambilan keputusan.

Perkembangan AI tidak terlepas dari peningkatan kemampuan komputasi dan ketersediaan data, yang memungkinkan mesin untuk menyelesaikan permasalahan yang semakin kompleks dan sebelumnya sulit dicapai oleh sistem komputasi konvensional. Sejak awal, AI dikembangkan dengan tujuan menjembatani kemampuan algoritmik mesin dengan kecerdasan dan intuisi manusia (Aggarwal, 2021).

Kecerdasan buatan dikembangkan untuk mensimulasikan cara berpikir dan perilaku kecerdasan manusia melalui pemanfaatan program komputer yang mampu memahami dan mempelajari pola perilaku manusia. Sistem AI dirancang agar dapat meniru pengalaman mental dan pola tindakan manusia, seperti kemampuan untuk belajar, menilai, serta bereaksi terhadap situasi yang tidak secara eksplisit diprogram sebelumnya, sebagaimana diterapkan pada berbagai aplikasi cerdas. Kemampuan yang dikembangkan dalam kecerdasan buatan meliputi pembelajaran, pengenalan pola, penalaran, pemecahan masalah, persepsi visual, dan pemahaman bahasa. Secara umum, kecerdasan buatan terbagi menjadi *applied artificial intelligence*, yaitu sistem cerdas yang dirancang untuk menyelesaikan tugas tertentu secara spesifik, dan *generalized artificial intelligence*, yaitu sistem yang memiliki kemampuan umum untuk menangani berbagai jenis tugas (Yessy Asri dkk., 2025).

2.2.2.1. Jaccard Similarity

Jaccard *similarity* mengukur kemiripan antara dua teks berdasarkan rasio jumlah kata yang dimiliki bersama dengan jumlah keseluruhan kata unik dari kedua teks tersebut (Bag dkk., 2019). Pendekatan ini sederhana dan cepat, namun sangat sensitif terhadap perbedaan redaksi. Secara matematis, Jaccard *similarity* dihitung menggunakan persamaan berikut:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

Terhadap teks klausul draft dan teks template, dilakukan konversi ke huruf kecil dan pemisahan kata berdasarkan spasi dengan text.lower().split(), kemudian dibungkus ke dalam set() agar Jaccard bekerja dengan himpunan unik. Misalnya dengan pemisahan seperti berikut ini:

a. *Draft*: `{pihak, pertama, berhak, mengakhiri, perjanjian, ini, secara, sepihak, apabila, terjadi, pelanggaran}`

b. *Template*: `{jika, terjadi, pelanggaran, maka, pihak, pertama, dapat, menghentikan, perjanjian, secara, sepihak}`

Maka irisan kata yang sama terdapat 7 kata yaitu: `{pihak, pertama, terjadi, pelanggaran, perjanjian, secara, sepihak}`. Kemudian, gabungan dari seluruh kata unik adalah 15 kata. Dengan demikian, skor Jaccard *similarity* dapat dihitung dari 7/15 dengan hasil 0,4667.

2.2.2.2. TF-IDF

TF-IDF *(Term Frequency–Inverse Document Frequency)* merupakan metode pembobotan kata yang mengukur tingkat kepentingan suatu kata dalam sebuah dokumen relatif terhadap kumpulan dokumen lainnya. TF-IDF menghitung frekuensi kemunculan kata dalam dokumen *(term frequency)* serta seberapa jarang kata tersebut muncul di seluruh dokumen *(inverse document frequency)* (Dikmen dkk., 2025). Meski tidak memahami konteks semantik, metode ini dapat mengenali pasal-pasal dengan terminologi khas yang berulang. Adapun rumus TF-IDF adalah sebagai berikut:

$$TF(t,d) = f(t,d) \tag{2}$$

Keterangan:

t = term (kata atau token) tertentu

d = dokumen

$f(t,d)$ = jumlah kemunculan term t dalam dokumen d

$$IDF(t) = \log\left(\frac{N}{df(t)}\right) \tag{3}$$

Keterangan:

N = jumlah total dokumen dalam korpus

$df(t)$ = jumlah dokumen yang mengandung *term* t

log = fungsi logaritma untuk menekan pengaruh term yang sangat umum

Bobot TF-IDF untuk suatu term dalam dokumen dihitung dengan mengalikan kedua komponen tersebut:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) \tag{4}$$

Hasil pembobotan TF-IDF dari seluruh *term* dalam suatu dokumen selanjutnya membentuk sebuah vektor representasi dokumen dalam ruang berdimensi k, di mana k merupakan jumlah term unik dalam korpus.

Secara skripting, teks diproses dengan tokenisasi menggunakan *TfidfVectorizer* untuk memecah teks menjadi token kata, mengabaikan tanda baca, melakukan normalisasi huruf, serta pembobotan. Kedua teks tersebut direpresentasikan dalam vektor fitur TF-IDF.

Setelah kedua teks direpresentasikan dalam bentuk vektor TF-IDF, tingkat kemiripan antar teks dihitung menggunakan *cosine similarity* untuk mengukur kedekatan orientasi antara dua vektor dalam ruang vektor berdimensi tinggi, sehingga fokus pada pola distribusi bobot kata tanpa dipengaruhi oleh panjang teks. Nilai *cosine similarity* berada pada rentang 0 hingga 1, dimana nilai yang lebih tinggi menunjukkan kemiripan terminologi yang lebih besar antara klausul draft dan template.

2.2.3. Natural Language Processing

Pemrosesan bahasa alami (*Natural Language Processing*/NLP) merupakan subbidang kecerdasan buatan yang bertujuan memungkinkan sistem komputasi memahami, menginterpretasikan, dan memproses bahasa manusia dalam bentuk teks maupun suara. NLP berperan penting dalam mengolah data bahasa yang tidak terstruktur dan menjadi komponen utama dalam berbagai sistem otomasi modern yang melibatkan komunikasi manusia–mesin. Seiring meningkatnya ketersediaan data dan kemampuan komputasi, NLP berkembang dari pendekatan berbasis aturan menuju pendekatan statistik dan pembelajaran mesin (Cole Stryker & Jim Holdsworth, 2024).

Perkembangan NLP dipengaruhi oleh kemajuan daya komputasi, algoritma, dan ketersediaan data berskala besar, serta berevolusi melalui beberapa pendekatan utama. Tahap awal NLP didominasi oleh pendekatan berbasis aturan (rule-based), yang mengandalkan aturan sintaksis dan semantik yang dirancang secara manual oleh pakar domain. Seiring meningkatnya kompleksitas data dan kebutuhan aplikasi, pendekatan NLP berkembang ke arah *machine learning* dan *deep learning*. Pendekatan *machine learning* memungkinkan model mempelajari pola statistik dari data. Tahap terkini dalam evolusi NLP ditandai dengan hadirnya *Pretrained Language Models* (PLMs) dan *Large Language Models* (LLMs). Model-model ini dilatih pada korpus teks berskala besar untuk memperoleh representasi bahasa umum, kemudian dapat digunakan atau disesuaikan

untuk berbagai tugas lanjutan. Model *encoder-only* seperti BERT unggul dalam pemahaman *text embedding*, sementara model *decoder-only* seperti GPT dirancang untuk generasi teks autoregresif (Tang dkk., 2026).

Beberapa model terkemuka yang sering dipakai dalam berbagai bidang NLP untuk pemahaman semantik teks diantaranya BERT *(Bidirectional Encoder Representations from Transformers)* yang memungkinkan pemrosesan teks yang memahami konteks informasi secara dua arah (Devlin dkk., 2019). Selain itu, model LaBSE *(Language-agnostic BERT)* sebagai pengembangan dari BERT mendukung multibahasa (lebih 109 bahasa) yang menunjukkan performa kuat pada bahasa yang tidak secara eksplisit menjadi data latih (Feng dkk., 2022).

Pendekatan *embedding* semantik digunakan untuk memetakan teks ke dalam ruang vektor yang merepresentasikan makna semantik secara kontekstual. Dalam penelitian ini, model berbasis Transformer seperti BERT dan LaBSE digunakan sebagai encoder untuk menghasilkan embedding teks. proses pembentukan *embedding* dapat dirumuskan sebagai berikut:

$$v = Encoder(x) \tag{5}$$

Keterangan:

x = teks input (dokumen atau pasal kontrak)

$Encoder(x)$ = model bahasa berbasis Transformer (BERT atau LaBSE)

v = vektor embedding hasil pemetaan teks

2.2.4. Cosine Similarity

Dalam penelitian ini, *cosine similarity* digunakan untuk pengukuran kemiripan numerik pada level representasi vektor. *Cosine similarity* diterapkan untuk menghitung kedekatan antara dua vektor teks yang dihasilkan oleh berbagai pendekatan representasi, termasuk TF-IDF dan embedding semantik dari model neural seperti BERT dan LaBSE. Dengan menghitung kosinus sudut antara dua vektor, metode ini menghasilkan skor kemiripan yang tidak dipengaruhi oleh panjang teks, sehingga memungkinkan perbandingan yang konsisten antar dokumen atau klausula kontrak. Secara matematis, *cosine similarity* antara dua vektor A dan B didefinisikan sebagai berikut:

$$cosine\_similarity_{(A,\,B)} = \frac{A \cdot B}{||A|| * ||B||} \tag{2}$$

A·B merupakan hasil perkalian *dot product* antara dua vektor, sedangkan ||A|| dan ||B|| merupakan panjang vektor yang digunakan untuk menormalkan perhitungan, sehingga pengukuran kemiripan ditentukan oleh orientasi vektor dan tidak dipengaruhi oleh perbedaan panjang teks (Christopher D. Manning dkk., 2009).

Dalam implementasi penelitian ini, *cosine similarity* digunakan di seluruh metode berbasis vektor. Pada metode TF-IDF, setiap dokumen direpresentasikan sebagai vektor bobot kata, kemudian *cosine similarity* dihitung untuk mengukur kedekatan distribusi *term* antar teks. Pada metode *embedding* BERT dan LaBSE, teks terlebih dahulu dipetakan ke dalam vektor semantik berdimensi 768, dan *cosine similarity* digunakan untuk mengukur kedekatan posisi vektor dalam ruang representasi semantik. Dengan demikian, *cosine similarity* berperan sebagai mekanisme pengukuran numerik yang seragam untuk membandingkan keluaran berbagai metode, sementara perbedaan karakteristik hasil pengukuran ditentukan oleh model pembentuk representasi vektornya.

2.2.5. Large Language Model

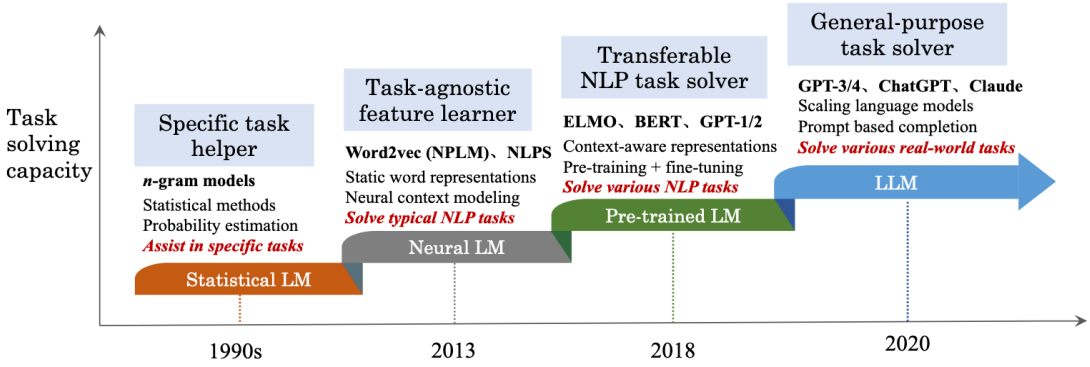
Large language model (LLM) adalah salah satu bentuk kecerdasan buatan yang dilatih dengan dataset dalam jumlah sangat besar. LLM adalah model NLP tingkat lanjut yang menggunakan arsitektur *deep neural network* yang didesain untuk memahami, menggenerasi, dan memproses bahasa manusia pada level tinggi (Reddy dkk., 2024). Dengan data *training* yang sangat besar sebagaimana Tabel 1, LLM menunjukkan kualitas pemahaman kontekstual yang tinggi dibandingkan dengan *Deep Learning* sebagai subset dari *Machine Learning* (Chang dkk., 2024; Hadi dkk., 2024).

Tabel 1 - Perbandingan ML, DL, dan LLM

Aspect	Traditional ML	Deep Learning	Large Language Models
Training Data Size	Medium	Large	Very Large
Feature Engineering	Required	Partial	Not Required
Model Complexity	Low	High	Very High
Interpretability	High	Low	Low
Performance	Moderate	High	Very High
Hardware Requirements	Standard	High	Very High
Scalability	Limited	Moderate	High
Fine-tuning	Not Applicable	Possible	Possible
Transfer Learning	Limited	Effective	Effective
Contextual Understanding	Limited	Moderate	High

Perkembangan LLM pada Gambar 1 di bawah ini menunjukkan evolusi kemampuan model bahasa. Dimulai dari *statistical language models* yang hanya mendukung tugas-tugas spesifik, berlanjut ke *neural language models* yang mempelajari

representasi fitur bahasa, kemudian ke *pre-trained language models* yang mampu ditransfer ke berbagai tugas NLP melalui *pre-training* dan *fine-tuning*, hingga mencapai *large language model* (LLM) yang berfungsi sebagai *problem solver* umum berbasis *prompt*.



Gambar 1 - Evolusi Generasi Model Bahasa (Zhao dkk., 2025)

LLM memiliki kemampuan transformatif untuk memprediksi kemiripan dari urutan kata atau menghasilkan teks baru berdasarkan input yang diberikan. LLM merupakan suatu model bahasa lanjutan dengan ukuran parameter masif dan kemampuan belajar sangat besar. Salah satu fitur dari LLM yaitu *in-context learning*, dimana model dilatih untuk menghasilkan teks berdasarkan konteks yang diberikan atau *prompt* (Chang dkk., 2024).

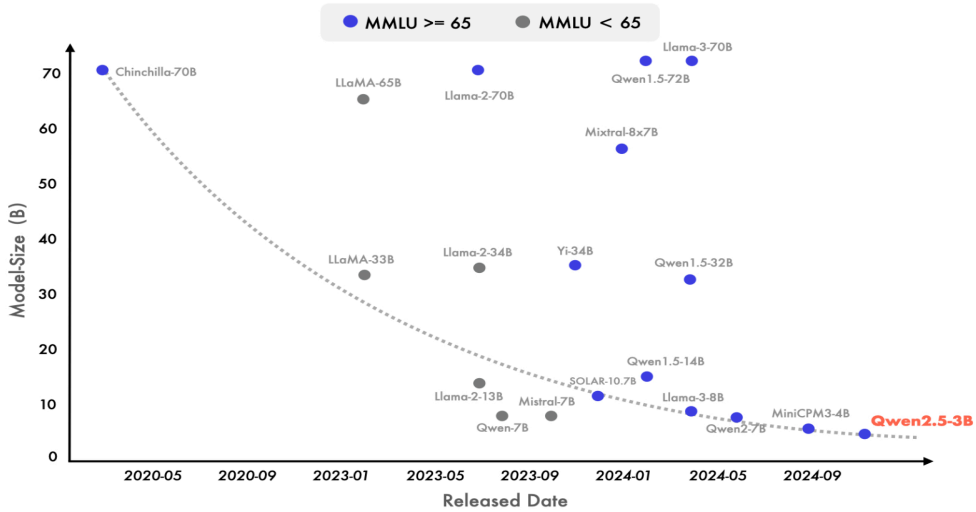
Kemampuan LLM dalam menghasilkan teks berdasarkan konteks yang diberikan dalam prompt memberikan peluang yang besar untuk berbagai *use-case* terutama yang berkaitan dengan pengolahan bahasa. Kekuatan dan kelemahan LLM dapat dilihat sebagai berikut (Chang dkk., 2024):

1. Kekuatan
    a. mampu mengeluarkan teks lingustik secara presisi dan fasih;
    b. performa baik dalam tugas pemahaman bahasa;
    c. kemampuan penalaran aritmatika;
    d. mampu melakukan tugas pengolahan data terstruktur; dan
    e. pemahaman kontekstual yang kuat sehingga dapat merespon sesuai dengan input yang diberikan;
2. Kelemahan
    a. rentan mengalami kebingungan dalam konteks yang kompleks;
    b. berpotensi memunculkan informasi yang keliru dalam dialog (halusinasi);

c. keterbatasan akan pengetahuan terbaru yang *real-time*; dan

d. sensitif terhadap prompt, berisiko pada serangan adversarial.

Kelemahan-kelemahan LLM tersebut dapat teratasi dengan berkembangnya metode pelatihan, peningkatan kualitas dataset, serta perbaikan-perbaikan dalam teknik prompting dan *guardrail* dapat mengurangi berbagai kekurangan yang ada. Gambar 2 menunjukkan perkembangan LLM dari waktu ke waktu menunjukan kinerja model semakin baik bahkan untuk model kecil (Qwen Team, 2024).



Gambar 2 - Perkembangan Capaian MMLU Model (Qwen Team, 2024)

2.2.5.1. Qwen2.5-7B-Instruct

Qwen2.5-7B-Instruct merupakan *instruction-tuned* LLM berbasis arsitektur Transformer *decoder* dengan ukuran sekitar 7 miliar parameter. Model ini dikembangkan melalui proses *pre-training* berskala besar menggunakan korpus hingga puluhan triliun token yang terkurasi. Model tersebut memiliki kemampuan pemahaman bahasa alami yang kuat, kepatuhan terhadap instruksi, serta stabilitas dalam menghasilkan keluaran yang koheren dan terstruktur (Qwen dkk., 2024).

Gambar 3 menyajikan ringkasan hasil evaluasi Qwen2.5-7B-Instruct pada berbagai benchmark standar yang mencakup tugas pemahaman umum, penalaran matematika dan sains, pemrograman, serta alignment. MMLU-Pro dan MMLU-redux digunakan untuk merepresentasikan pemahaman bahasa dan pengetahuan lintas domain, sedangkan LiveBench 0831 digunakan untuk mengamati generalisasi model pada soal yang bersifat dinamis. GPQA, MATH, dan GSM8K merepresentasikan tugas penalaran matematika dan sains dengan tingkat kompleksitas yang berbeda. HumanEval, MBPP, MultiPL-E, dan LiveCodeBench digunakan untuk menggambarkan kemampuan

pemrograman, mulai dari generasi fungsi sederhana hingga penyelesaian masalah lintas bahasa. Sementara itu, IFEval, Arena-Hard, dan MTBench merepresentasikan aspek kepatuhan terhadap instruksi serta kualitas respons dalam konteks alignment.

Datasets	Gemma2-9B	Llama3.1-8B	Qwen2-7B	Qwen2.5-7B
General Tasks				
MMLU-Pro	52.1	48.3	44.1	**56.3**
MMLU-redux	72.8	67.2	67.3	**75.4**
LiveBench 0831	30.6	26.7	29.2	**35.9**
Mathematics & Science Tasks				
GPQA	32.8	32.8	34.3	**36.4**
MATH	44.3	51.9	52.9	**75.5**
GSM8K	76.7	84.5	85.7	**91.6**
Coding Tasks				
HumanEval	68.9	72.6	79.9	**84.8**
MBPP	74.9	69.6	67.2	**79.2**
MultiPL-E	53.4	50.7	59.1	**70.4**
LiveCodeBench	18.9	8.3	23.9	**28.7**
Alignment Tasks				
IFEval	70.1	**75.9**	54.7	71.2
Arena-Hard	41.6	27.8	25.0	**52.0**
MTbench	8.49	8.23	8.26	**8.75**

Gambar 3 - Kinerja Model Instruct di Kelas ±7B (Qwen dkk., 2024)

Dalam penelitian ini, Qwen2.5-7B-Instruct diposisikan sebagai representasi operasional dari pendekatan *Large Language Model* dalam pengenalan template. Model ini mampu melakukan pemrosesan semantik dan penalaran kontekstual berbasis instruksi, sehingga relevan digunakan sebagai *semantic evaluator* dalam sistem pengenalan template pada automasi *review* kontrak. Melalui mekanisme *in-context learning*, evaluasi dan penilaian dapat dilakukan tanpa pelatihan ulang model, menjadikan pendekatan ini fleksibel dan adaptif terhadap variasi dokumen.

2.2.5.2. Qwen2.5-VL-72B-Instruct

Model ini merupakan model berbasis Qwen2.5-72B-Instruct yang dilengkapi dengan kemampuan vision encoder. Kemampuannya tidak hanya mengenali objek gambar dan video, juga kemampuan OCR multilingual dan multiorientasi dengan kinerja terbaik di kelasnya(*Qwen/Qwen2.5-VL-72B-Instruct · Hugging Face*, 2025; Team, 2025).

Dalam penelitian ini, Qwen2.5-VL-72B-Instruct bertindak sebagai OCR yang digunakan untuk melakukan ekstraksi teks kontrak dari sumbernya yang berformat PDF gambar.

2.2.6. Format File YAML

Format YAML digunakan sebagai format teks template pasal secara terstruktur. Struktur data dalam YAML tidak menggunakan notasi simbolik, namun memanfaatkan indentasi sebagai hierarki data. Karakteristik ini menjadikan YAML bersifat intuitif, sehingga proses pembuatan, pembacaan, dan modifikasi berkas dapat dilakukan menggunakan editor teks biasa (Oren Ben-Kiki dkk., 2021).

2.2.7. Evaluasi Model

Sistem automasi *review* kontrak dimulai dari tahapan penting yang menjadi objek dari penelitian ini, yaitu pengenalan 2 tahap. Tahap pertama dilakukan untuk mendapatkan *template* yang cocok dengan draft yang diinputkan, kemudian tahap kedua dilakukan untuk mendapatkan pasal yang cocok di dalam folder *template* kontrak.

2.2.7.1. Tahap Pertama (Pemilihan Template)

Pada tahap pertama, pengukuran model ditujukan untuk menilai kemampuan metode kemiripan dalam membedakan template kontrak yang relevan dan tidak relevan terhadap draft kontrak yang diuji. Evaluasi dilakukan terhadap beberapa pendekatan komputasi *text similarity*, baik berbasis Jaccard, TF-IDF, BERT, LaBSE, maupun LLM. Adapun kriteria evaluasi pada tahap ini mencakup hal-hal sebagai berikut:

a. Pencapaian *acceptance threshold* (≥ 0,9)

Model harus mampu menghasilkan nilai kemiripan minimal sebesar 0,9 untuk template yang benar-benar relevan. Nilai *threshold* ini ditetapkan sebagai *acceptance threshold*, yang merepresentasikan tingkat kemiripan semantik yang secara praktis dipersepsikan manusia sebagai "sangat mirip". *Threshold* ini berfungsi secara konservatif, dimana apabila tidak ada *template* yang mencapai nilai tersebut, maka sistem tidak merekomendasikan *template* apa pun dan proses automasi dihentikan.

b. Separabilitas deterministik antara *template* dan *non-template*

Model harus mampu membedakan *template* dan *non-template* secara tegas tanpa menghasilkan gradasi nilai yang terlalu tipis. Hal ini penting untuk menghindari *false positive*, yaitu kondisi dimana template yang tidak relevan tetap memiliki skor tinggi dan berpotensi direkomendasikan dari peringkatnya. Pada sistem rekomendasi berbasis *ranking*, gradasi yang rapat dapat menyebabkan rekomendasi yang keliru.

Evaluasi model difokuskan pada distribusi skor kemiripan kelompok *template* dan *non-template*, serta selisih nilai rata-rata *(mean gap)* di antara keduanya. Dengan pendekatan ini, evaluasi tidak hanya berfokus pada akurasi numerik, tetapi juga pada kemampuan metode dalam mencerminkan persepsi kemiripan manusia serta menjaga kehati-hatian dalam konteks dokumen hukum yang bersifat sensitif.

Pengujian dengan pendekatan tersebut menghasilkan metrik evaluasi sebagai berikut:

a. *Mean Template:* Rata-rata nilai kemiripan dari yang sebenarnya merupakan template, untuk merepresentasikan tingkat kebocoran skor kemiripan terhadap template yang tidak relevan, dengan rumus:

$$MeanTemplate = \frac{\Sigma s_t}{n_t} \quad (3)$$

Keterangan:

s_t : skor kemiripan untuk *template* yang relevan

n_t : jumlah *template* yang relevan

b. *Mean Non-Template:*

Rata-rata nilai kemiripan dari non-template untuk menunjukkan tingkat kemiripan rata-rata yang dihasilkan model terhadap template yang benar-benar relevan, dengan rumus:

$$MeanNonTemplate = \frac{\Sigma s_{nt}}{n_{nt}} \quad (4)$$

Keterangan:

s_t : skor kemiripan untuk *non-template*

n_t : jumlah *non-template*

c. *Count Non-Template* dan *Count Template:* Jumlah data yang diuji (gabungan *template* dan *non-template)*, untuk memastikan perhitungan sesuai dengan jumlah total matrikulasi draft kontrak terhadap template kontrak yang ada.

d. *Mean Gap:* Selisih antara rata-rata template yang benar dengan yang salah, untuk melihat sifat deterministik dari metode evaluasi yang diuji, dengan rumus:

$$MeanGap = MeanTemplate - MeanNonTemplate \quad (5)$$

2.2.7.2. Tahap Kedua (Pemilihan Pasal)

Tahap kedua dalam sistem automasi *review* kontrak berbasis template adalah pemilihan dan pemetaan pasal *(clause mapping)*, yaitu proses menentukan pasal dalam *template* kontrak yang paling sesuai dengan setiap pasal dalam draft kontrak. Tahap ini dilakukan setelah template kontrak yang relevan berhasil ditentukan pada tahap pertama.

Berbeda dengan pemilihan template yang bersifat selektif dan deterministik, pemetaan pasal menghadapi tingkat ambiguitas yang lebih tinggi. Dalam konteks dokumen hukum, dua pasal dapat memiliki substansi yang sama meskipun berbeda secara redaksional, urutan kalimat, maupun penamaan berkas. Oleh karena itu, pendekatan evaluasi yang semata-mata berbasis klasifikasi biner dan *confusion matrix* klasik (seperti *accuracy, precision,* dan *recall*) berpotensi tidak sepenuhnya merepresentasikan kualitas sistem dalam praktik.

2.2.8. Penentuan *Threshold*

Penerapan *threshold* pada tahap ini diperlukan sebagai batas keputusan dalam menentukan apakah suatu *template* atau pasal dianggap cukup relevan untuk direkomendasikan. Pada pengukuran *similarity* berbasis *embedding*, skor kemiripan dengan *cosine similarity* merepresentasikan kedekatan geometris dalam ruang vektor yang berada pada rentang tanpa lompatan dan tidak ada batasan tegas pada interval yang ditentukan *(continuum)* yang dalam penelitian ini adalah antara 0 dan 1.

Berdasarkan penelitian yang dilakukan oleh Klabunde dkk., (2025) dinyatakan bahwa pengukuran *similarity* pada model *neural* menegaskan bahwa *similarity* dan jarak kedekatan geometris tersebut pada dasarnya merupakan konsep yang setara, sehingga nilai yang dihasilkan akan berada dalam spektrum *continuum* dan tidak memiliki batas ontologis universal yang secara tegas memisahkan kategori “mirip” dan “tidak mirip”. Lebih lanjut, penelitian tersebut juga menjelaskan bahwa kemiripan hasil prediksi belum tentu mencerminkan kesamaan representasi internal karena model dapat menghasilkan keluaran yang tampak serupa namun menggunakan representasi internal yang berbeda. Artinya bahwa kemiripan yang tinggi pada satu jenis pengukuran belum tentu mencerminkan kesamaan yang konsisten dalam distribusi yang berbeda sehingga skor yang dihasilkan tetap memerlukan interpretasi kontekstual (Klabunde dkk., 2025).

Selain pendekatan geometris, *similarity* juga dapat dipahami melalui perbandingan fitur (karakteristik) yang dimiliki bersama dan fitur yang membedakan dua

objek (Rahnama & Hüllermeier, 2020). Kemiripan dinilai dari keseimbangan antara unsur kesamaan dan perbedaan, sehingga nilainya bersifat kontekstual dan tidak memiliki batas universal yang berlaku di semua situasi. Dalam kerangka tersebut, kemiripan dan ketidakmiripan bukan merupakan dua kategori yang terpisah secara absolut, melainkan dua kutub dalam satu spektrum evaluasi. Setiap pasangan dokumen/pasal dapat memiliki unsur kesamaan dan perbedaan bersamaan dengan proporsinya masing-masing. Oleh karena itu, *threshold* tidak memisahkan dua keadaan ontologi yang berbeda, namun menentukan titik praktis pada spektrum kemiripan untuk tujuan pengambilan keputusan.

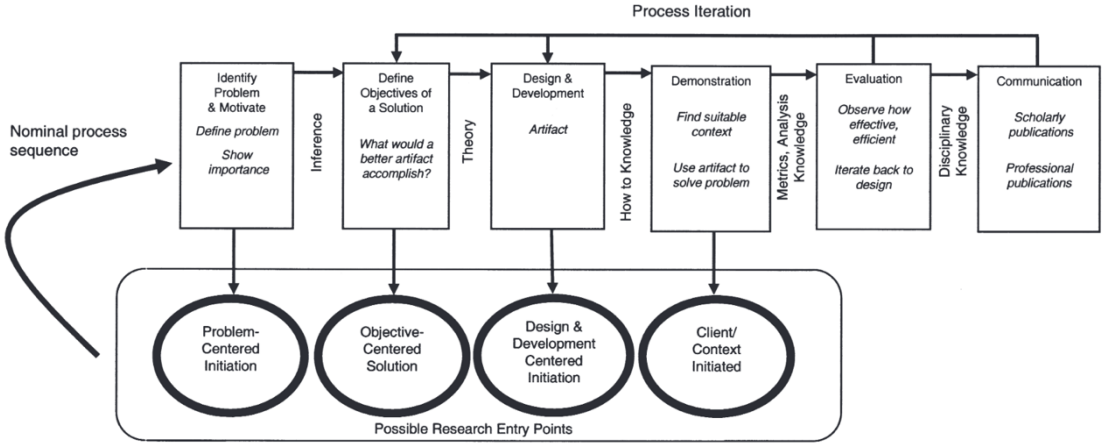
Berdasarkan hal tersebut, nilai *threshold* dalam penelitian ini ditetapkan sesuai kebutuhan tahap pemrosesan dan tingkat kehati-hatian dalam domain hukum. Pada tahap pengenalan *template* digunakan nilai 0,9 agar hanya *template* dengan tingkat kemiripan yang sangat tinggi yang diproses lebih lanjut. Secara teoritis, hanya nilai maksimum (1,0) yang merepresentasikan kesamaan identik dalam *cosine similarity*, sedangkan seluruh nilai di bawahnya berada dalam spektrum gradasi. Dengan demikian, angka 0,9 merepresentasikan tingkat kemiripan yang sangat tinggi untuk menjaga kehati-hatian sistem, bukan sebagai batas teoritis universal antara "mirip" dan "tidak mirip".

Pada tahap pengenalan pasal digunakan nilai 0,7 karena kandidat perbandingan telah lebih terfokus pada *template* yang sudah terpilih. Perbandingan antar pasal dalam satu *template* umumnya memiliki variasi redaksional yang lebih besar, sehingga diperlukan ambang yang lebih fleksibel tanpa mengurangi kualitas hasil. Selain itu, nilai ini juga dipilih untuk menghindari ambang yang terlalu rendah (seperti 0,5), yang pada metode leksikal-statistik (misalnya Jaccard atau TF-IDF) dapat terpengaruh secara signifikan apabila *draft* atau *template* bersifat *bilingual*. Dalam kondisi *bilingual*, sebagian frasa atau kalimat dapat muncul dalam dua bahasa sehingga secara teknis menurunkan skor kemiripan meskipun substansi pasal tetap sejalan. Oleh karena itu, nilai 0,7 dipandang lebih proporsional dan median antara 0,5 dan 0,9 dalam menjaga keseimbangan antara sensitivitas sistem dan konsistensi hasil seleksi.

Meskipun terdapat pendekatan lain yang menetapkan threshold berdasarkan nilai rata-rata *similarity* untuk tujuan distribusi label tertentu (Asri dkk., 2025), pendekatan tersebut bersifat kontekstual dan tidak menjadi standar yang bersifat universal. Dalam penelitian ini, *threshold* digunakan sebagai alat penyaring awal agar sistem tetap berhati-hati sebelum hasilnya dianalisis lebih lanjut terhadap substasi isi pasal.

2.2.9. Design Science Research Methodology (DSRM)

Di antara berbagai pendekatan metodologis yang digunakan dalam penelitian sistem informasi dan rekayasa perangkat lunak, *Design Science Research Methodology* (DSRM) merupakan pendekatan yang berfokus pada perancangan dan evaluasi artefak sebagai solusi terhadap permasalahan nyata yang relevan secara praktis dan ilmiah. Pendekatan ini menempatkan artefak penelitian sebagai pusat kontribusi penelitian, dengan penekanan pada proses evaluasi untuk membuktikan kegunaan dan efektivitas solusi yang dihasilkan, dengan tahapan sebagaimana ditunjukkan Gambar 4.



Gambar 4 - Tahapan DSRM (Peffers dkk., 2007)

Berdasarkan kerangka tersebut, *Design Science Research Methodology* (DSRM) digunakan untuk memandu seluruh tahapan penelitian ini secara sistematis, mulai dari identifikasi masalah, penetapan tujuan solusi, artefak, demostrasi, evaluasi, dan komunikasi.

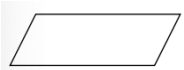
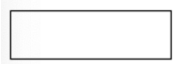
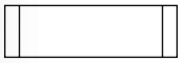
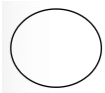
2.2.10. Diagram Alir *(Flowchart)*

Diagram alir *(flowchart)* merupakan representasi visual sebagaimana tergambar pada Tabel 2, yang digunakan untuk menggambarkan urutan langkah penyelesaian suatu masalah atau algoritma secara sistematis. Dalam penelitian ini, flowchart digunakan sebagai alat pemodelan konseptual untuk merepresentasikan alur kerja sistem yang dikembangkan, sehingga hubungan antar tahapan proses dapat dianalisis secara terstruktur sebelum dilakukan implementasi teknis.

Flowchart menyajikan langkah-langkah proses melalui simbol grafis yang dihubungkan oleh garis alur (flow line) untuk menunjukkan arah dan keterurutan proses. Dengan pendekatan ini, flowchart membantu memperjelas struktur logika sistem,

mendukung analisis alur proses, serta berfungsi sebagai dokumentasi yang mendukung evaluasi dan pengembangan sistem yang diteliti (A. B. Chaudhuri, 2020).

Tabel 2 - Simbol Standar Flowchart (A. B. Chaudhuri, 2020)

Simbol	Nama Simbol	Maksud
	Terminal	Menunjukkan awal dan akhir proses.
	Input/Output	Merepresentasikan proses masukan atau keluaran data.
	Process	Menunjukkan aktivitas pemrosesan oleh sistem.
	Predefined Process	Menunjukkan proses yang telah didefinisikan sebelumnya.
	Decision	Menunjukkan titik pengambilan keputusan yang menghasilkan percabangan alur.
	Flow Line	Menunjukkan arah dan urutan proses.
	On-page Connector	Menghubungkan bagian flowchart yang terpisah di halaman yang sama.
	Off-page Connector	Menghubungkan bagian flowchart yang terpisah ke halaman yang berbeda.
	Document Input/Output	Merepresentasikan input/output berbentuk dokumen.

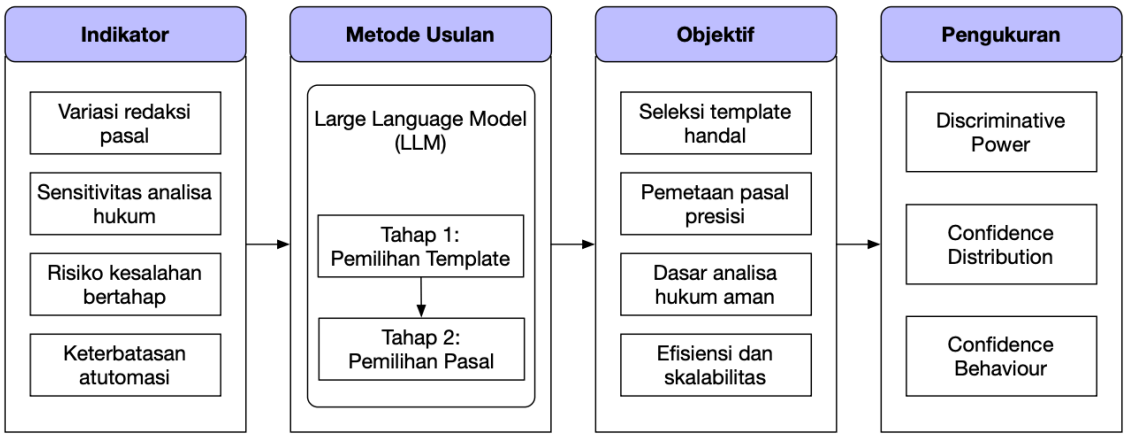
2.3. Kerangka Pemikiran

Penelitian ini berangkat dari permasalahan dalam sistem rekomendasi template kontrak yang menjadi penentu keberhasilan proses automasi *review* draft kontrak berbasis template. Tahap utama evaluasi substansi kontrak akan sangat ditentukan oleh ketepatan pemilihan template yang relevan berdasarkan rekomendasi dari tahap pencarian template dan pemetaan pasal demi pasal, yang akan terintegrasi ke dalam sistem automasi *review* kontrak.

Untuk menjawab permasalahan tersebut, penelitian ini mengusulkan metode rekomendasi template kontrak berbasis kemiripan substansi dengan memanfaatkan kemampuan penalaran terarah LLM dalam menilai template dan pasal yang tepat. Pemilihan LLM sebagai *decision gate* adalah karena alasan praktis dimana LLM dapat diarahkan secara objektif melalui *prompt engineering* untuk mendapatkan daya diskriminasi yang tinggi.

Penelitian ini juga membandingkan beberapa pendekatan komputasi selain LLM, yaitu Jaccard *similarity*, TF-IDF, BERT, dan LaBSE untuk mengukur bagaimana LLM dapat menjadi alternatif dalam pemrosesan *text similarity* berbasis substansi kontekstual. Pendekatan-pendekatan ini diintegrasikan dalam sebuah *pipeline* yang melakukan perbandingan antara teks kontrak asli sebagai *draft* dan teks *template* hasil modifikasi, baik pada tahap seleksi template maupun pada tahap analisis pasal demi pasal.

Objek penelitian meliputi dokumen kontrak dan template kontrak yang direpresentasikan dalam bentuk teks pada tingkat dokumen dan pasal. Evaluasi dilakukan dengan mengukur relevansi hasil pada setiap tahap pengambilan keputusan masing-masing metode, sehingga performa metode yang diusulkan dapat dianalisis secara sistematis dan hasil penelitian dapat digunakan sebagai pondasi bagi sistem automasi review kontrak maupaun penelitian lanjutan.



Gambar 5 - Kerangka Pemikiran

Gambar 5 merupakan alur kerangka pemikiran bagaimana penelitian ini dilakukan, yang menunjukkan keterkaitan antara indikator, metode usulan, objektif, dan pengukurannya sebagai berikut:

1. Indikator *(Indicators)*

   Komponen ini menunjukkan faktor-faktor penting yang menjadi latar belakang penelitian ini dilakukan, yaitu:

   a. Tingginya varisi redaksi pasal kontrak, terutama dalam ragam bahasa hukum Indonesia. Model-model bahasa, baik *encoder* maupun *decoder*, tidak banyak yang dilatih dengan dataset spesifik dengan bahasa hukum Indonesia.
   b. Sensitivitas analisa hukum, dimana setiap kalimat, tanda baca, pengulangan, penempatan kata, akan berpengaruh terhadap pemahaman substansi teks hukum.
   c. Potensi risiko kesalahan bertahap pada sistem automasi review kontrak berbasis *template*, dimana setiap keputusan yang diambil pada satu tahapan akan menentukan tahapan selanjutnya serta mempengaruhi hasil akhir.
   d. Adanya keterbatasan sistem automasi dalam menstandarkan input teks, dimana untuk mencapai automasi penuh yang cerdas memerlukan pola pengolahan teks yang minim interaksi.

2. Metode Usulan *(Propose Method)*

   Peneliti mengusulkan metode perbandingan substansi teks melalui *Large Language Model* (LLM). Sebagai pembanding, metode *text similarity* melalui Jaccard *similarity*, TFIDF, BERT, dan LaBSE juga dievaluasi untuk memahami karakteristik dan keterbatasan masing-masing pendekatan.

   Metode automasi *review* kontrak yang diteliti adalah automasi 2 tahap utama, yaitu tahap pemilihan template kontrak dan tahap pemetaan pasal-pasal terkait. Hasil dari tahap kedua ini dapat diposisikan sebagai “bahan baku” bagi proses analisa hukum substantif, yang berada di luar ruang lingkup pembahasan penelitian ini.

3. Objektif *(Objective)*

   Tujuan dari penelitian ini yaitu:

   a. memperoleh mekanisme seleksi template kontrak yang handal dan mampu meminimalkan risiko kesalahan pada tahap awal automasi;
   b. menghasilkan mendapatkan pemetaan pasal yang presisi untuk analisa lanjutan;
   c. memastikan hasil automasi layak dan aman untuk dilanjutkan ke tahap analisa hukum substantif; dan
   d. meningkatkan relevansi hasil dengan tingkat komputasi yang ringan serta dapat diskalakan pada proses yang lebih besar.

4. Pengukuran *(Measurement)*

   Hasil dari metode yang diusulkan diukur dengan metrik untuk dapat melihat skor sebagai kelayakan aksi dan evaluasi kepercayaan sistem, yaitu:

   a. sejauh mana metode yang diuji mampu membedakan *(discriminative power)* antara teks input dengan teks template yang relevan maupun tidak relevan;
   b. bagaimana distribusi nilai kemiripan *(confidence distribution)* yang dihasilkan metode mencerminkan tingkat keyakinan sistem dalam menilai kesesuaian dan ketidaksesuaian antar teks; dan
   c. sejauh mana nilai kemiripan yang dihasilkan memiliki perilaku yang merepresentasikan tingkas kepercayaan *(confidence behaviour)* sebagai dasar untuk melanjutkan proses automasi ke tahap analisa hukum selanjutnya, serta menghindari risiko *overconfidence* yang dapat menyebabkan propagasi kesalahan antar tahap.

Untuk melengkapi kerangka pemikiran berbasis IPOM tersebut, penelitian ini juga menggunakan pendekatan analisis sebab–akibat *(fishbone)* guna mengidentifikasi sumber risiko, *overconfidence*, dan potensi kesalahan bertahap pada setiap tahap pengambilan keputusan dalam sistem automasi review kontrak. Analisis ini digunakan sebagai alat interpretatif terhadap hasil evaluasi kuantitatif pada Bab IV.