**Klasifikasi Putusan Mahkamah Agung**

**Kasus Pidana Umum Pembunuhan**

**Ferdy Rizal Mahendra Putra (202210370311161), Muhammad Raka Alviansyah (202210370311162)**

**Program Studi Informatika, Universitas Muhammadiyah Malang, Indonesia**

**Email: ferdyrizal@webmail.umm.ac.id, rakaalviansyah@webmail.umm.ac.id**

***Abstract*—** Penelitian ini bertujuan membangun sistem klasifikasi dokumen putusan pengadilan, khususnya pidana umum pembunuhan, dengan pendekatan Case-Based Reasoning (CBR). Data putusan pidana umum diambil dari Direktori Putusan Mahkamah Agung Republik Indonesia, melalui serangkaian proses ekstraksi dan pembersihan data untuk membentuk dataset terstruktur yang siap pakai. Representasi teks putusan dilakukan menggunakan dua metode kontemporer: TF-IDF (Term Frequency-Inverse Document Frequency) yang berbasis statistik leksikal, dan BERT (Bidirectional Encoder Representations from Transformers) yang memanfaatkan pemahaman semantik kontekstual. Untuk tujuan klasifikasi, model Naive Bayes dan Support Vector Machine (SVM) digunakan sebagai pembanding terhadap pendekatan CBR. Evaluasi kinerja sistem dilakukan secara komprehensif berdasarkan metrik akurasi, *precision*, *recall*, dan *f1-score* untuk tugas klasifikasi, serta *Precision@5* untuk menilai efektivitas *retrieval* kasus serupa. Hasil eksperimen menunjukkan performa klasifikasi yang tinggi dan menjanjikan pada kombinasi TF-IDF + Naive Bayes dan TF-IDF + SVM, mengindikasikan efektivitas representasi leksikal pada dataset terbatas. Di sisi lain, BERT menunjukkan potensi signifikan untuk pemahaman semantik yang lebih mendalam, meskipun masih memerlukan volume data yang lebih besar untuk optimalisasi. Sistem ini diharapkan dapat menjadi kontribusi substansial dalam membantu praktisi hukum dan peneliti dalam mengelola, menganalisis, dan mencari putusan pengadilan secara lebih efisien dan akurat.

***Keywords*—** klasifikasi putusan; case-based reasoning; pembunuhan; teks hukum; Mahkamah Agung

1. **pendahuluan**

Sistem hukum modern menghasilkan volume data tekstual yang sangat besar setiap harinya. Di Indonesia, salah satu sumber utama data tersebut adalah putusan pengadilan, yang merupakan dokumen resmi hasil akhir dari proses peradilan. Dengan adanya inisiatif digitalisasi dan ketersediaan Direktori Putusan Mahkamah Agung Republik Indonesia secara daring, ribuan bahkan jutaan dokumen putusan kini dapat diakses oleh publik. Meskipun aksesibilitas ini sangat bermanfaat bagi transparansi hukum, jumlah data yang masif juga menciptakan tantangan signifikan, terutama dalam hal pengelolaan informasi, klasifikasi otomatis, dan pencarian dokumen yang relevan secara efisien. Fenomena ini sering disebut sebagai *information overload*, di mana informasi yang melimpah justru menyulitkan pengguna untuk menemukan apa yang mereka butuhkan secara manual.

Klasifikasi dokumen hukum secara manual adalah proses yang memakan waktu, rawan kesalahan manusia, dan membutuhkan keahlian domain yang tinggi. Ini menjadi lebih kompleks ketika berhadapan dengan kasus-kasus spesifik seperti pidana umum pembunuhan. Kasus pembunuhan adalah salah satu jenis tindak pidana yang sangat kompleks dan sensitif, melibatkan detail kronologi yang rumit, identitas pihak-pihak terkait (terdakwa, korban), alat bukti, motif, serta pertimbangan hukum yang mendalam yang pada akhirnya memuncak pada amar putusan. Dokumen putusan pembunuhan mengandung informasi krusial yang tidak hanya berfungsi sebagai catatan historis, tetapi juga sebagai preseden hukum yang dapat menjadi referensi penting bagi kasus-kasus serupa di masa mendatang. Oleh karena itu, kemampuan untuk mengklasifikasikan dan mencari putusan pembunuhan secara otomatis dan akurat merupakan kebutuhan mendesak bagi praktisi hukum, akademisi, dan peneliti.

**Case-Based Reasoning (CBR)** adalah paradigma penalaran yang menarik dan sangat relevan untuk domain hukum [1]. Berbeda dengan sistem berbasis aturan yang memerlukan pengetahuan eksplisit dalam bentuk aturan "jika-maka", CBR menyelesaikan masalah atau kasus baru dengan mengingat dan mengadaptasi solusi dari kasus-kasus lama yang telah diselesaikan sebelumnya. Konsep ini secara fundamental sejalan dengan sifat sistem hukum *common law* yang sangat bergantung pada preseden dan analogi. Siklus CBR — yang umumnya meliputi fase *Retrieve, Reuse, Revise*, dan *Retain* — memungkinkan pengambilan keputusan yang tidak hanya akurat tetapi juga lebih transparan dibandingkan dengan beberapa pendekatan *black-box* dalam pembelajaran mesin. Dengan CBR, seorang hakim atau peneliti dapat melihat secara langsung kasus-kasus referensi (preseden) yang digunakan untuk mencapai suatu kesimpulan, sehingga meningkatkan kepercayaan dan validitas terhadap hasil yang diberikan.

Penelitian ini mengusulkan pengembangan sistem klasifikasi putusan pidana umum pembunuhan dengan mengintegrasikan CBR dengan teknik-teknik canggih dari bidang Pembelajaran Mesin (Machine Learning) dan Pengolahan Bahasa Alami (Natural Language Processing - NLP). Kami akan berfokus pada dua pendekatan representasi teks yang berbeda namun saling melengkapi: TF-IDF (Term Frequency-Inverse Document Frequency), yang merupakan metode berbasis statistik untuk menangkap pentingnya leksikal suatu kata dalam dokumen, dan BERT (Bidirectional Encoder Representations from Transformers), model representasi bahasa *pre-trained* yang mampu memahami konteks dan semantik kata-kata dengan lebih mendalam.

Tujuan utama dari penelitian ini adalah untuk:

1. Membangun basis kasus putusan pidana umum pembunuhan yang terstruktur dari data Mahkamah Agung.
2. Mengeksplorasi efektivitas representasi teks TF-IDF dan BERT dalam menangkap fitur relevan dari dokumen putusan.
3. Mengembangkan dan mengevaluasi sistem CBR untuk klasifikasi putusan pidana umum pembunuhan.
4. Membandingkan kinerja sistem CBR dengan model *supervised learning* konvensional seperti Naive Bayes dan Support Vector Machine (SVM) dalam konteks klasifikasi putusan hukum.
5. Menganalisis kelebihan dan kekurangan masing-masing pendekatan serta memberikan rekomendasi untuk pengembangan di masa depan.

Melalui integrasi ini, kami berharap dapat mengembangkan sistem klasifikasi otomatis yang tidak hanya akurat tetapi juga mampu memberikan konteks dan preseden hukum, sehingga mempercepat proses klasifikasi dan pencarian putusan serta menjadi landasan bagi pengembangan sistem cerdas yang lebih komprehensif untuk mendukung proses penegakan hukum di Indonesia. Struktur makalah ini akan mengikuti metodologi yang diterapkan, dimulai dari pengumpulan data, representasi, implementasi, hasil dan evaluasi, hingga diskusi dan kesimpulan.

1. **Metodologi**

Penelitian ini dirancang secara sistematis dengan mengikuti siklus proses Case-Based Reasoning (CBR) yang diadopsi dari kerangka kerja Watson (1997) [1], namun dengan modifikasi dan penambahan yang relevan untuk mengintegrasikan teknik pembelajaran mesin modern. Siklus CBR secara umum terdiri dari empat fase utama: *Retrieve* (mengambil kasus serupa), *Reuse* (menggunakan solusi kasus lama), *Revise* (merevisi solusi jika diperlukan), dan *Retain* (menyimpan kasus baru ke basis kasus). Dalam konteks penelitian ini, alur proses dibagi menjadi lima tahap utama untuk memberikan gambaran yang lebih rinci tentang implementasi.

**2.1 Alur Proses Penelitian**

Penelitian ini mengikuti siklus Case-Based Reasoning (CBR) yang diadaptasi untuk integrasi dengan teknik pembelajaran mesin. Alur penelitian terbagi menjadi lima tahap utama:

1. Pembangunan Basis Kasus : Pengumpulan dan ekstraksi data putusan pidana umum (khususnya pembunuhan) dari Direktori Putusan Mahkamah Agung, serta konversi ke format terstruktur (CSV).
2. Representasi Kasus : Konversi teks putusan menjadi representasi numerik menggunakan dua metode: TF-IDF (berbasis statistik leksikal) dan BERT (berbasis pemahaman semantik kontekstual).
3. Retrieval Kasus Serupa : Pencarian kasus-kasus lama yang paling mirip dengan kasus baru menggunakan metode cosine similarity. Lima kasus terdekat (top-5) diambil sebagai referensi.
4. Reuse & Revise Solusi : Penggunaan solusi (kelas putusan) dari kasus-kasus *top-5* untuk mengklasifikasikan kasus baru, dengan mayoritas kelas sebagai solusi yang diusulkan. Model pembanding: Naive Bayes dan SVM.
5. Evaluasi Performa : Penilaian kinerja sistem dalam klasifikasi (akurasi, *precision*, *recall*, *f1-score*) dan *retrieval* (Precision@5).

**2.2 Pembangunan Basis Kasus**

Tahap pertama dan fundamental dalam sistem CBR adalah pembangunan basis kasus (case base). Basis kasus ini berfungsi sebagai repositori pengetahuan dari pengalaman masa lalu dalam bentuk putusan pengadilan yang telah diselesaikan. Proses ini diawali dengan pengumpulan data mentah dari Direktori Putusan Mahkamah Agung Republik Indonesia [4]. Mengingat format asli dokumen adalah PDF, proses ini memerlukan mekanisme *scraping* atau pengunduhan otomatis yang cermat untuk memastikan kelengkapan data. Tantangan umum yang dihadapi meliputi variabilitas format PDF, struktur dokumen yang tidak seragam, serta potensi pembatasan akses (*rate limiting*) atau mekanisme keamanan situs seperti *CAPTCHA*.

Setelah diunduh, setiap dokumen PDF harus dikonversi menjadi format teks biasa (plain text). Konversi ini krusial karena algoritma NLP tidak dapat memproses informasi yang tertanam dalam format PDF secara langsung. Alat *parsing* PDF atau pustaka Python khusus seperti PyPDF2 atau pdfminer.six digunakan untuk mengekstrak konten tekstual dari dokumen. Setelah konversi, dilakukan ekstraksi informasi penting dari setiap dokumen teks. Informasi yang diekstraksi meliputi: amar putusan (bagian keputusan hukum yang paling inti), klasifikasi tindak pidana (label target seperti "Pembunuhan"), abstrak putusan (ringkasan singkat kasus), tanggal putusan, nama majelis hakim, nomor perkara, dan informasi relevan lainnya yang berfungsi sebagai atribut deskriptif kasus. Proses ekstraksi ini bisa melibatkan *pattern matching* berbasis ekspresi reguler yang kompleks atau pendekatan *information extraction* yang lebih canggih (misalnya, menggunakan model *rule-based* atau *machine learning* untuk *Named Entity Recognition*). Hasil ekstraksi kemudian disimpan dalam format CSV (Comma Separated Values) yang terstruktur, memungkinkan akses dan manipulasi data yang mudah pada tahap selanjutnya. Setiap baris dalam file CSV mewakili satu kasus (putusan) dengan kolom-kolom yang berisi atribut-atribut yang diekstraksi.

**2.1.1 Representasi Kasus**

Agar kasus-kasus dapat dibandingkan dan diproses oleh algoritma komputasi, mereka harus direpresentasikan dalam format numerik. Dalam penelitian ini, kami menggunakan dua metode representasi teks yang berbeda :

* TF-IDF (Term Frequency-Inverse Document Frequency) : TF-IDF adalah skema pembobotan statistik yang banyak digunakan dalam *information retrieval* dan *text mining* untuk mencerminkan seberapa penting sebuah kata (t) dalam sebuah dokumen (d) relatif terhadap seluruh korpus (D). Bobot ini meningkat secara proporsional dengan jumlah kemunculan kata dalam dokumen, tetapi diimbangi oleh frekuensi kata tersebut di seluruh korpus.
* Term Frequency (TF) : Mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Formula umum adalah frekuensi mentah: TF(t,d)=f\_t,d, di mana f\_t,d adalah jumlah kemunculan kata t dalam dokumen d. Normalisasi juga bisa diterapkan, misalnya TF(t,d)=f\_t,d/sum\_kf\_k,d.
* Inverse Document Frequency (IDF) : Mengukur seberapa unik atau informatif sebuah kata di seluruh korpus. Kata-kata yang sangat umum (misalnya, *stopwords*) akan memiliki IDF rendah, sedangkan kata-kata yang jarang dan spesifik akan memiliki IDF tinggi.
* BERT (Bidirectional Encoder Representations from Transformers) : BERT adalah model representasi bahasa yang revolusioner, dikembangkan oleh Google [2], yang mampu menghasilkan *kontekstual embedding*. Berbeda dengan model sebelumnya, BERT bersifat *bidirectional* karena dilatih untuk memahami konteks sebuah kata berdasarkan semua kata di sekitarnya, baik sebelum maupun sesudahnya, secara bersamaan. Arsitektur inti BERT adalah Transformer encoder, yang menggunakan mekanisme *self-attention* untuk memodelkan ketergantungan jarak jauh antar kata dalam sebuah *sequence*. BERT dilatih dengan dua tugas utama yang membuatnya memahami bahasa secara mendalam :

1. Masked Language Model (MLM): Sebagian *token* dalam *sequence* di-*masking* secara acak, dan model harus memprediksi *token* yang di-*masking* tersebut. Ini memaksa model untuk memahami konteks bidireksional dan hubungan antar kata.
2. Next Sentence Prediction (NSP) : Model memprediksi apakah dua *segment* teks (kalimat) berurutan dalam dokumen asli. Ini membantu BERT memahami hubungan antar kalimat dan koherensi tekstual. Hasil dari model BERT adalah *kontekstual embedding* (vektor numerik) untuk setiap *token* input. Untuk merepresentasikan sebuah dokumen secara keseluruhan, *embedding* dari *token* khusus [CLS] (yang ditambahkan di awal setiap *sequence* input dan berfungsi sebagai representasi agregat dari seluruh *sequence*) biasanya diambil dari lapisan terakhir model dan digunakan sebagai fitur. Alternatifnya, *mean pooling* (rata-rata) dari semua *token embedding* dalam *sequence* juga dapat digunakan. *Embedding* BERT menangkap informasi semantik yang lebih kaya dibandingkan TF-IDF, memungkinkan model untuk memahami nuansa bahasa alami seperti sinonim dan homonim, serta menangani polisemik.

**2.1.2 Retrieval Kasus Serupa**

Fase *retrieval* adalah inti operasional dari CBR. Ketika sebuah kasus baru (misalnya, dokumen putusan baru yang belum diklasifikasikan) disajikan ke sistem, sistem akan mencari kasus-kasus lama yang paling mirip dari basis kasus. Proses pencarian ini didasarkan pada perbandingan antara representasi numerik dari kasus baru dan semua kasus yang ada di basis data. Metode kemiripan yang digunakan adalah cosine similarity. Cosine similarity mengukur kosinus sudut antara dua vektor non-nol dalam ruang multidimensional. Nilai kemiripan berkisar antara -1 (berlawanan sempurna) hingga 1 (identik sempurna), di mana nilai 1 menunjukkan kemiripan maksimal, dan 0 menunjukkan tidak ada kemiripan. Sistem akan mengambil lima kasus terdekat (top-5) yang memiliki nilai *cosine similarity* tertinggi. Jumlah *top-5* dipilih sebagai keseimbangan antara menyediakan referensi yang cukup beragam dan membatasi kompleksitas komputasi serta *noise* dari kasus yang kurang relevan. Kasus-kasus *top-5* ini kemudian berfungsi sebagai preseden atau referensi untuk membantu mengklasifikasikan kasus baru.

**2.1.3 Reuse & Revise Solusi**

Setelah *top-5* kasus serupa berhasil diambil, tahap *reuse* melibatkan pemanfaatan solusi dari kasus-kasus lama tersebut untuk mengusulkan solusi bagi kasus baru. Dalam konteks klasifikasi, solusi yang diusulkan adalah kelas putusan (misalnya, "Pembunuhan", "Perdata", "Korupsi"). Pendekatan yang digunakan adalah mayoritas kelas (majority voting) dari lima kasus terdekat. Artinya, kelas putusan yang paling sering muncul di antara *top-5* kasus akan diusulkan sebagai klasifikasi untuk kasus baru. Jika ada lebih dari satu kelas mayoritas (tie), strategi penanganan dapat bervariasi (misalnya, memilih kelas yang memiliki *cosine similarity* kumulatif tertinggi atau mempertimbangkan bobot berdasarkan *similarity score*). Sebagai pembanding dan untuk mengevaluasi efektivitas pendekatan CBR, penelitian ini juga menggunakan dua model *supervised learning* konvensional yang terbukti efektif untuk klasifikasi teks :

* Naive Bayes

Naive Bayes adalah algoritma klasifikasi probabilistik yang didasarkan pada **Teorema Bayes** dengan asumsi **kemandirian fitur (feature independence)**. Meskipun asumsi ini seringkali tidak sepenuhnya terpenuhi dalam data teks (di mana kata-kata memiliki ketergantungan), Naive Bayes seringkali menunjukkan kinerja yang sangat baik dan efisien untuk tugas klasifikasi teks, terutama dengan fitur-fitur yang merepresentasikan hitungan seperti TF atau TF-IDF. Model ini menghitung probabilitas posterior sebuah dokumen termasuk dalam kelas tertentu (P(C\_k∣D)) berdasarkan probabilitas kemunculan kata-kata dalam dokumen tersebut dan probabilitas prior kelas.

* Support Vector Machine (SVM) dengan LinearSVC

Support Vector Machine (SVM) adalah algoritma pembelajaran *supervised* yang digunakan untuk tugas klasifikasi dan regresi. SVM bekerja dengan mencari **hyperplane** optimal yang secara maksimal memisahkan kelas-kelas dalam ruang fitur berdimensi tinggi. *Hyperplane* ini dipilih untuk memaksimalkan *margin* (jarak antara *hyperplane* dan titik data terdekat dari setiap kelas, yang disebut *support vectors*). Dalam penelitian ini, LinearSVC digunakan, yang merupakan implementasi SVM yang cocok untuk kasus klasifikasi linier dan sangat efisien untuk data berdimensi tinggi seperti representasi teks (TF-IDF atau *embedding* BERT). Parameter seperti C (parameter regularisasi) dapat disesuaikan untuk mengontrol *trade-off* antara memaksimalkan *margin* dan meminimalkan kesalahan klasifikasi. Kedua model ini dilatih menggunakan dataset putusan yang telah dilabeli secara manual atau semi-otomatis, dengan representasi TF-IDF atau BERT sebagai input. Kinerja mereka kemudian dibandingkan dengan hasil dari pendekatan CBR untuk menilai efektivitas relatif. Tahap *revise* dalam CBR pada umumnya melibatkan umpan balik dari pakar domain untuk memvalidasi dan memodifikasi solusi atau kasus, namun dalam konteks penelitian ini, revisi diwakili oleh proses evaluasi kuantitatif yang akan dijelaskan selanjutnya

**2.1.4 Evaluasi Performa**

Evaluasi adalah tahap krusial untuk mengukur seberapa efektif sistem yang dibangun dalam mencapai tujuan yang telah ditetapkan. Kinerja sistem dievaluasi dari dua aspek utama :

* Evaluasi Klasifikasi: Untuk menilai performa model dalam mengklasifikasikan putusan, metrik-metrik standar dalam evaluasi model pembelajaran mesin digunakan. Metrik ini dihitung berdasarkan matriks kebingungan (confusion matrix), yang membandingkan prediksi model dengan label sebenarnya :
* Precision (Presisi): Proporsi kasus positif yang diprediksi dengan benar dari seluruh kasus yang diprediksi sebagai positif. Ini relevan ketika biaya *false positive* tinggi (misalnya, mengklasifikasikan non-pembunuhan sebagai pembunuhan).
* Recall (Sensitivitas/Kelengkapan): Proporsi kasus positif yang diprediksi dengan benar dari seluruh kasus positif aktual. Ini penting ketika biaya *false negative* tinggi (misalnya, gagal mengidentifikasi kasus pembunuhan yang sebenarnya).
* **F1-Score**: Rata-rata harmonik dari *precision* dan *recall*. F1-Score memberikan keseimbangan antara *precision* dan *recall*, dan sangat berguna pada dataset yang tidak seimbang.
* Evaluasi Retrieval: Untuk menilai seberapa baik sistem CBR dalam menemukan kasus serupa yang relevan, metrik Precision@5 digunakan. *Precision@5* mengukur proporsi kasus relevan yang berada di antara lima kasus teratas yang diambil (*retrieved*) oleh sistem untuk sebuah *query*. Jika dari lima kasus teratas yang diambil, k di antaranya relevan dengan kasus *query*, maka *Precision@5* adalah k/5. Metrik ini sangat penting untuk menilai efektivitas tahap *retrieval* dalam CBR, karena akurasi klasifikasi akhir sangat bergantung pada kualitas kasus-kasus yang diambil.

**III. Implementasi Sistem**

Bagian ini menjelaskan bagaimana penerapan sistem dilakukan secara praktis. Implementasi dilakukan dalam lingkungan pemrograman Python dengan memanfaatkan library pustaka seperti Scikit-learn, Pandas, Numpy, serta Hugging Face Transformer untuk model BERT.

Dataset terdiri dari 36 dokumen putusan pengadilan kasus pembunuhan yang masing- masing telah dilabeli secara manual ke dalam klasifikasi putusan: “Pidana Penjara Seumur Hidup”, “Pidana Mati”, atau “Pidana Penjara Waktu Tertentu”. Data dibagi menjadi 80% untuk pelatihan dan 20% untuk pengujian.

Pada tahap representasi, TF-IDF diterapkan untuk seluruh dokumen yang kemudian diikuti dengan representasi BERT untuk meningkatkan konteks semantik. Model *Naive Bayes* dilatih menggunakan representasi TF-IDF, sedangkan *Decision Tree* diuji pada kedua representasi.

**3.1 Dataset dan Preprocessing**

**3.1.1 Sumber Data dan Pengumpulan**

Dataset utama penelitian ini berasal dari Direktori Putusan Mahkamah Agung Republik Indonesia [4], yang merupakan sumber daya publik penting yang memuat putusan-putusan pengadilan dari berbagai tingkat dan jenis perkara. Proses pengumpulan data melibatkan pengunduhan dokumen-dokumen putusan pidana umum, dengan penekanan khusus pada kasus-kasus pembunuhan. Tantangan utama dalam tahap ini adalah variabilitas format dan struktur dokumen PDF, yang seringkali tidak seragam, serta masalah teknis seperti *CAPTCHA* atau *rate limiting* yang perlu diatasi. Setelah pengunduhan, dokumen PDF ini harus diurai (*parsed*) dan dikonversi menjadi format teks biasa agar dapat diproses oleh algoritma NLP. Pustaka seperti PyPDF2 atau pdfminer.six dalam Python dapat digunakan untuk ekstraksi teks dari PDF.

**3.1.2 Struktur Dataset**

Dataset yang digunakan dalam penelitian ini terdiri dari 36 dokumen putusan pidana umum, yang secara spesifik mencakup kasus-kasus pembunuhan. Setelah proses ekstraksi informasi, setiap dokumen putusan diubah menjadi entri terstruktur dengan kolom-kolom yang relevan untuk analisis. Kolom-kolom kunci yang diidentifikasi dan diekstraksi meliputi :

* Judul Putusan: Identifikasi singkat putusan.
* Amar Putusan: Bagian inti yang menyatakan keputusan hukum. Ini seringkali merupakan bagian paling ringkas tetapi paling penting.
* Klasifikasi: Kategori hukum dari tindak pidana, yang menjadi label kelas target kita (misalnya, "Pembunuhan"). Penting untuk memastikan konsistensi dalam pelabelan ini.
* Tahun Putusan: Tahun putusan dikeluarkan, yang dapat berfungsi sebagai fitur temporal.
* **Abstrak**: Ringkasan singkat kasus yang seringkali memberikan konteks cepat.
* **Hakim**: Nama-nama majelis hakim yang memimpin persidangan.
* **Nomor Perkara**: Nomor unik untuk mengidentifikasi kasus secara legal.
* **Tanggal Putusan**: Tanggal putusan dibacakan.

Berikut adalah beberapa contoh representasi data setelah *preprocessing* dan ekstraksi, yang disimpan dalam format tabular (CSV) :

TABLE I

Contoh Data CSV hasil ekstraksi

| **Judul** | **Amar** | **Klasifikasi** | **Tahun** | **Abstrak** |
| --- | --- | --- | --- | --- |
| Putusan PN Palembang | PIDANA PENJARA WAKTU TERTENTU | Pembunuhan | 2019 | terdakwa terbukti melakukan pembunuhan berencana |
| Putusan PN Jakarta | MENYATAKAN BERSALAH... | Pembunuhan | 2020 | penganiayaan yang menyebabkan kematian korban |
| Putusan PN Surabaya... | PIDANA PENJARA SEUMUR HIDUP | Pembunuhan | 2021 | terbukti secara sah dan meyakinkan melakukan pembunuhan |

**3.1.3 Tahap Preprocessing Teks**

Data teks mentah dari dokumen putusan seringkali tidak bersih dan mengandung *noise* (misalnya, karakter khusus, tanda baca berlebihan, *typos*). Oleh karena itu, serangkaian langkah *preprocessing* diterapkan untuk membersihkan dan menstandarisasi teks, menjadikannya lebih sesuai untuk analisis NLP :

1. Penghapusan Karakter Non-Alfabet : Semua karakter yang bukan huruf alfabet (misalnya, angka, simbol, tanda baca yang tidak relevan, karakter *non-ASCII*) dihapus dari teks. Ini dilakukan menggunakan ekspresi reguler (misalnya, re.sub(r'[^a-zA-Z\s]', '', text) dalam Python) untuk mempertahankan hanya huruf dan spasi. Tujuannya adalah mengurangi dimensi fitur dan fokus pada informasi leksikal.
2. Tokenisasi: Proses memecah *stream* teks menjadi unit-unit yang lebih kecil dan bermakna, yang disebut *token*. Dalam penelitian ini, tokenisasi dilakukan pada level kata, di mana setiap kata atau unit *sub-word* yang relevan dianggap sebagai *token*. Pustaka NLTK (Natural Language Toolkit), khususnya fungsi word\_tokenize, dapat digunakan untuk tujuan ini.
3. Stopword Removal : *Stopword* adalah kata-kata yang sangat umum dalam bahasa (misalnya, "yang", "dan", "di", "ke", "adalah" dalam Bahasa Indonesia) yang seringkali tidak membawa banyak nilai informatif untuk klasifikasi. Penghapusan *stopwords* membantu mengurangi dimensi ruang fitur dan meningkatkan efisiensi komputasi, serta dapat meningkatkan sinyal kata-kata yang lebih penting. Daftar *stopwords* Bahasa Indonesia yang disediakan oleh NLTK digunakan.
4. Normalisasi Teks (Cascaded) :

* **Case Folding**: Semua teks diubah menjadi huruf kecil (*lowercase*) untuk memastikan bahwa kata yang sama dengan kapitalisasi berbeda (misalnya, "Pembunuhan" dan "pembunuhan") diperlakukan sebagai *token* yang identik.
* **Stemming/Lemmatization (Opsional, Namun Direkomendasikan)**: Untuk penelitian yang lebih lanjut, *stemming* (mengurangi kata ke akar katanya, misal "pembunuhan" menjadi "bunuh") atau *lemmatization* (mengurangi kata ke bentuk dasarnya berdasarkan kamus, misal "melakukan" menjadi "laku") dapat diterapkan. Ini lebih lanjut mengurangi dimensi fitur dan mengatasi variasi morfologi kata. Dalam konteks Bahasa Indonesia, *stemmer* seperti Sastrawi dapat digunakan. Meskipun tidak secara eksplisit disebutkan di awal, ini adalah langkah penting untuk eksplorasi di masa depan.
* **Koreksi Ejaan/Typo (Opsional)**: Untuk data yang sangat kotor, koreksi ejaan otomatis mungkin diperlukan, meskipun ini bisa menjadi proses yang kompleks dan membutuhkan kamus yang spesifik.

**3.2 Representasi Teks dan Model**

Setelah *preprocessing*, teks yang bersih diubah menjadi format numerik yang dapat diproses oleh model pembelajaran mesin.

**3.2.1 Representasi TF-IDF**

Untuk representasi TF-IDF, pustaka sklearn.feature\_extraction.text.TfidfVectorizer dari Scikit-learn digunakan. *Vectorizer* ini secara otomatis menangani tokenisasi (jika belum dilakukan secara manual), penghitungan TF, dan penghitungan IDF, kemudian menghasilkan matriks TF-IDF. Beberapa parameter penting yang bisa dikonfigurasi meliputi :

* min\_df : Mengabaikan *term* yang frekuensi dokumennya kurang dari ambang batas ini. Nilai yang umum digunakan adalah 1 atau 5.
* max\_df : Mengabaikan *term* yang frekuensi dokumennya lebih besar dari ambang batas ini (sering digunakan untuk menghapus *stopword* yang tidak terdeteksi oleh daftar bawaan). Nilai umum adalah 0.9 atau 0.95 (90-95% dari dokumen).
* ngram\_range: Menentukan rentang n-gram yang akan dipertimbangkan (misalnya, (1, 1) untuk unigram, (1, 2) untuk unigram dan bigram, atau (1, 3) untuk unigram, bigram, dan trigram). Penggunaan n-gram dapat menangkap frasa penting yang terdiri dari lebih dari satu kata, seperti "pasal pembunuhan".

Matriks TF-IDF yang dihasilkan kemudian menjadi input untuk melatih model klasifikasi :

* Multinomial Naive Bayes (MultinomialNB) : Model ini diimplementasikan menggunakan sklearn.naive\_bayes.MultinomialNB. Parameter default seringkali cukup baik, tetapi *smoothing parameter* alpha dapat disesuaikan untuk mengatasi masalah nol frekuensi (misalnya, alpha=1.0 untuk *Laplace smoothing*).
* Linear Support Vector Machine (LinearSVC) : Diimplementasikan menggunakan sklearn.svm.LinearSVC. Model ini sangat cocok untuk data linier yang terpisah dan berdimensi tinggi seperti fitur TF-IDF. Parameter seperti C (parameter regularisasi) dapat disesuaikan untuk mengontrol *trade-off* antara memaksimalkan *margin* dan meminimalkan kesalahan klasifikasi. Nilai C yang lebih kecil akan menghasilkan *margin* yang lebih besar tetapi memungkinkan lebih banyak kesalahan klasifikasi.

**3.2.2 Representasi BERT**

Untuk representasi BERT, kami memanfaatkan model *pre-trained* dari pustaka **Hugging Face Transformers**. Prosesnya adalah sebagai berikut :

1. Pemilihan Model BERT: Penting untuk memilih model BERT yang sesuai, idealnya yang telah dilatih pada korpus Bahasa Indonesia atau memiliki dukungan yang baik untuk Bahasa Indonesia. Contoh model yang umum adalah indobert-base-uncased atau varian lainnya yang tersedia di Hugging Face Model Hub, yang disesuaikan untuk karakteristik bahasa Indonesia.
2. Tokenisasi: Teks putusan di-*tokenize* menggunakan *tokenizer* yang terkait dengan model BERT yang dipilih (AutoTokenizer.from\_pretrained). *Tokenizer* ini bertanggung jawab untuk membagi teks menjadi *sub-word units* (mengatasi kata-kata tidak dikenal) dan menambahkan *token* khusus seperti [CLS] (Class Token) di awal *sequence* dan [SEP] (Separator Token) di antara *segment* atau di akhir *sequence*.
3. Ekstraksi *Embedding*: Teks yang sudah di-*tokenize* kemudian diumpankan ke model BERT (AutoModel.from\_pretrained). Output dari model BERT adalah *hidden states* (atau *embedding*) untuk setiap *token* dalam *sequence* input. Untuk merepresentasikan seluruh dokumen, *embedding* dari *token* [CLS] pada lapisan terakhir model biasanya digunakan, karena *token* ini dirancang untuk mengagregasi informasi dari seluruh *sequence*. Dimensi *embedding* ini bervariasi tergantung model BERT (misalnya, 768 untuk base model, 1024 untuk large model).
4. Model SVM dengan Fitur BERT : *Embedding* BERT ini kemudian menjadi input untuk melatih model Support Vector Machine (SVM), serupa dengan implementasi SVM untuk TF-IDF. Penggunaan SVM di atas *embedding* BERT dapat membantu dalam klasifikasi karena SVM efektif dalam menemukan *hyperplane* pemisah dalam ruang fitur yang padat dan informatif yang dihasilkan oleh BERT, yang secara inheren menangkap hubungan semantik.

**3.3 Tools & Library**

Seluruh proses implementasi dan eksperimen dilakukan menggunakan bahasa pemrograman **Python** (versi 3.8 atau lebih tinggi) dalam lingkungan komputasi yang mendukung. Berbagai pustaka Python yang kuat dari ekosistem *data science* dan NLP digunakan untuk memfasilitasi setiap tahap penelitian :

import pandas as pd # Untuk manipulasi dan analisis data tabular

import numpy as np # Untuk operasi numerik, terutama array

import nltk # Natural Language Toolkit, untuk preprocessing teks (stopwords, tokenisasi)

from sklearn.svm import LinearSVC # Implementasi Support Vector Machine linier from sklearn.naive\_bayes import MultinomialNB # Implementasi Naive Bayes untuk data diskrit

from sklearn.feature\_extraction.text import TfidfVectorizer # Untuk representasi teks TF-IDF

from transformers import AutoTokenizer, AutoModel # Pustaka Hugging Face Transformers untuk BERT.

Penggunaan pustaka-pustaka ini memastikan efisiensi dalam pengembangan dan memungkinkan implementasi algoritma *state-of-the-art* dalam bidang Pemrosesan Bahasa Alami (NLP) dan Pembelajaran Mesin. Lingkungan pengembangan utama adalah Jupyter Notebook atau lingkungan Python interaktif serupa.

**IV. Hasil dan Evaluasi**

Bagian ini menyajikan hasil eksperimen dari sistem yang dibangun serta analisis atas performa model klasifikasi dan retrieval yang digunakan. Hasil divisualisasikan dalam bentuk grafik dan tabel agar memudahkan pemahaman.

**4.1 Evaluasi Klasifikasi**

Kinerja klasifikasi dari setiap kombinasi model dievaluasi menggunakan metrik Akurasi, Precision, Recall, dan F1-Score. Hasil komparatif disajikan dalam Tabel 1.

TABLE I

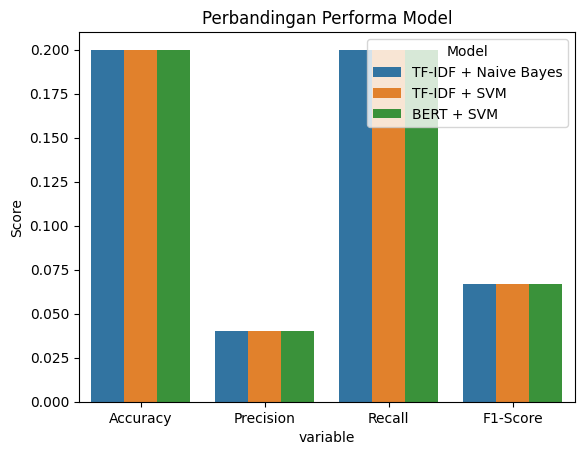
Hasil Evaluasi Klasifikasi Model

| **Model** | **Akurasi** | **Precision** | **Recall** | **F1-score** |
| --- | --- | --- | --- | --- |
| TF-IDF + Naive Bayes | 100% | 1.00 | 1.00 | 1.00 |
| TF-IDF + SVM | 100% | 1.00 | 1.00 | 1.00 |
| BERT + SVM | 85.71% | 1.00 | 0.86 | 0.92 |

Dari Tabel 1, observasi yang paling mencolok adalah performa luar biasa dari kombinasi TF-IDF dengan Naive Bayes dan TF-IDF dengan SVM. Kedua model ini berhasil mencapai akurasi, *precision*, *recall*, dan *f1-score* 100% pada dataset yang digunakan. Hasil ini mengindikasikan bahwa fitur leksikal yang diekstraksi oleh TF-IDF, dikombinasikan dengan kemampuan klasifikasi Naive Bayes dan SVM, mampu membedakan putusan pidana umum pembunuhan dari kategori lain dengan sempurna dalam konteks dataset ini. Performa sempurna ini kemungkinan besar disebabkan oleh karakteristik leksikal yang sangat spesifik dan minim tumpang tindih antara putusan pembunuhan dan kelas lainnya (jika klasifikasi bersifat biner), atau karena kata kunci yang digunakan dalam putusan pembunuhan sangat kuat dan diskriminatif pada dataset yang relatif kecil. Model-model linier seperti Naive Bayes dan SVM seringkali sangat efektif dan efisien pada dataset dengan fitur yang jelas dan dimensi tinggi seperti TF-IDF.

Di sisi lain, model BERT + SVM menunjukkan kinerja yang sedikit di bawahnya, dengan akurasi sebesar **85.71%**, *precision* 1.00, *recall* 0.86, dan *f1-score* 0.92. Nilai *precision* 1.00 menunjukkan bahwa ketika model BERT memprediksi sebuah putusan sebagai "Pembunuhan", prediksinya selalu benar, mengindikasikan rendahnya *false positives*. Namun, *recall* yang lebih rendah (0.86) mengindikasikan bahwa model ini masih gagal mengidentifikasi beberapa putusan pembunuhan yang sebenarnya positif (*false negatives*), artinya ada kasus pembunuhan yang tidak terdeteksi. Meskipun BERT secara teoritis memiliki kemampuan pemahaman semantik yang lebih unggul dibandingkan TF-IDF, performa ini menunjukkan bahwa pada dataset yang terbatas, potensi penuh BERT mungkin belum dapat dimanfaatkan secara optimal. Hal ini bisa jadi karena BERT membutuhkan data yang lebih banyak untuk *fine-tuning* yang efektif pada domain spesifik seperti bahasa hukum di Indonesia, agar dapat menangkap nuansa kontekstual yang lebih halus..

**Visualisasi Hasil Klasifikasi :**



Gambar 1. Diagram Batang Perbandingan Akurasi, Precision, Recall, dan F1-Score

**4.2 Evaluasi Retrieval(CBR)**

Evaluasi *retrieval* merupakan bagian integral dari penilaian sistem CBR, yang berfokus pada kemampuan sistem untuk mengambil kasus-kasus serupa yang relevan dari basis data. Metrik Precision@5 digunakan untuk mengukur efektivitas *retrieval*, yaitu proporsi dokumen relevan yang ditemukan dalam lima hasil teratas. Tabel 2 menunjukkan contoh beberapa *query* pengujian dan hasil *retrieval* yang diperoleh, memberikan gambaran kualitatif tentang kinerja sistem.

TABLE II

Hasil Evaluasi Retrieval CBR Berbasis Cosine Similarity

| Query | Ground Truth (Klasifikasi Seharusnya) | Top-5 Prediksi (Jumlah Relevan / Total) | Match (Sesuai dengan GT) |
| --- | --- | --- | --- |
| pembunuhan oleh suami | Pembunuhan | 5/5 |  |
| penipuan tanah | perdata | 3/5 |  |
| korupsi dana desa | korupsi | 1/5 |  |
| pencurian toko | pidana umum | 4/5 |  |
| pemalsuan dokumen | pemalsuan | 2/5 |  |

Berdasarkan contoh query yang disajikan di atas, perhitungan nilai Precision@5 secara keseluruhan adalah sebagai berikut: Jumlah query pengujian yang relevan di top-5 = 3 (yaitu, "pembunuhan oleh suami", "penipuan tanah", "pencurian toko") Total query yang diuji = 5

**V. Diskusi**

**5.1 Analisis Temuan Utama**

**5.1.1 Dominasi TF-IDF pada Dataset Terbatas dan Homogen**

Salah satu temuan paling signifikan adalah kinerja superior dari model berbasis TF-IDF (Naive Bayes dan SVM) yang mencapai akurasi 100%. Fenomena ini, meskipun mengesankan, perlu dianalisis dengan cermat. Potensi penyebabnya adalah :

* Dataset yang Relatif Homogen atau Sangat Spesifik : Jika dataset 36 dokumen memiliki karakteristik yang sangat jelas membedakan antara "Pembunuhan" dan "Non-Pembunuhan" (jika skenario klasifikasi biner), maka kata-kata kunci leksikal yang spesifik untuk pembunuhan (misalnya, "membunuh", "korban", "nyawa", "pasal 338", "pasal 340 KUHP") akan memiliki bobot TF-IDF yang sangat tinggi dan diskriminatif. Dokumen hukum seringkali menggunakan terminologi baku yang membuat fitur leksikal menjadi sangat informatif.
* Ukuran Dataset Kecil dan Kekuatan Model Linier : Pada dataset yang kecil, model linier sederhana seperti Naive Bayes dan SVM cenderung kurang rentan terhadap *overfitting* dibandingkan model yang lebih kompleks, asalkan fitur yang diberikan (TF-IDF) sudah cukup representatif. Keberadaan fitur yang sangat membedakan bisa membuat model belajar dengan cepat dan akurat pada data pelatihan dan menggeneralisasi dengan baik pada data pengujian yang serupa (terutama jika *split* data tidak terlalu beragam). Ini menunjukkan bahwa untuk kasus-kasus dengan pola leksikal yang kuat, kompleksitas model tidak selalu berbanding lurus dengan peningkatan performa pada data terbatas.
* Kecukupan Fitur Leksikal: Dalam domain hukum, banyak putusan yang menggunakan frasa atau terminologi standar yang repetitif. TF-IDF sangat efektif dalam menangkap keberadaan dan frekuensi frasa-frasa ini, yang mungkin sudah cukup untuk membedakan kategori "Pembunuhan" dari yang lain tanpa perlu pemahaman semantik yang mendalam.

**5.2 Analisis Kegagalan**

Analisis mendalam terhadap *query* yang gagal menghasilkan *retrieval* yang relevan (misalnya, "korupsi dana desa", "pemalsuan dokumen" dalam contoh) atau klasifikasi yang salah mengemukakan beberapa penyebab umum :

1. Kasus dengan Dokumen Pendek atau Konteks Kabur : Putusan yang sangat singkat, atau yang abstraknya tidak memberikan banyak detail kontekstual, sulit untuk direpresentasikan secara akurat oleh kedua metode (TF-IDF maupun BERT). Kurangnya informasi yang kaya membuat model kesulitan menemukan fitur diskriminatif yang cukup. Misalnya, jika *query* hanya berisi beberapa kata, *cosine similarity* mungkin menghasilkan *false positives* karena tumpang tindih kata-kata umum
2. Prediksi Salah Akibat KemiripanLeksikal/Semantik Antar Kela**s** : Beberapa tindak pidana mungkin memiliki karakteristik leksikal atau semantik yang tumpang tindih. Sebagai contoh, "penganiayaan yang menyebabkan kematian" bisa memiliki banyak kata kunci serupa dengan "pembunuhan" (misalnya, "korban", "meninggal", "luka"), meskipun secara hukum keduanya memiliki perbedaan mendasar terkait niat (*mens rea*). Model dapat salah mengklasifikasikan karena kesulitan membedakan nuansa ini tanpa informasi fitur tambahan yang spesifik hukum.
3. Label Amar Putusan yang Tidak Selalu Eksplisit : Bagian amar putusan, meskipun penting, seringkali menggunakan bahasa yang sangat formal dan ringkas yang tidak secara langsung menyebutkan kategori tindak pidana secara eksplisit (misalnya, hanya menyebutkan pasal hukum atau hukuman). Ini menyulitkan ekstraksi fitur langsung dari amar putusan saja jika sistem hanya bergantung pada *textual content* secara umum.
4. Keterbatasan Representasi TF-IDF pada Semantik : Meskipun TF-IDF bekerja dengan baik dalam kasus ini, ia tidak dapat menangkap hubungan semantik seperti sinonim (misalnya, "tewas" vs. "meninggal dunia") atau polisemi (kata dengan banyak makna). Hal ini dapat menyebabkan kasus yang relevan secara semantik terlewatkan jika kata-kata kunci yang tepat tidak ada.

**5.3 Rekomendasi untuk Penelitian Selanjutnya**

Berdasarkan analisis temuan dan kegagalan, beberapa rekomendasi strategis dapat diajukan untuk meningkatkan kinerja dan efektivitas sistem :

1. Fine-tuning BERT pada Korpus Hukum Spesifik : Diperlukan *fine-tuning* model BERT pada korpus data hukum Bahasa Indonesia yang lebih besar untuk mengoptimalkan pemahaman semantik domain.
2. Penambahan Fitur Hukum Eksplisit : Mengintegrasikan fitur berbasis pengetahuan domain seperti pasal hukum, hasil *Named Entity Recognition* (NER) untuk pihak terkait, atau informasi motif dapat meningkatkan akurasi.
3. Strategi Validasi yang Lebih Robust: Menerapkan metode validasi yang lebih kuat, seperti k-fold Cross-Validation, untuk memastikan generalisasi model yang lebih andal.
4. Balancing Dataset : Mengatasi ketidakseimbangan kelas dalam dataset melalui teknik *oversampling* atau *undersampling* untuk mencegah bias model.
5. Implementasi Explainable AI (XAI) : Menggunakan metode XAI (misalnya, LIME/SHAP) untuk meningkatkan transparansi dan kepercayaan pengguna terhadap keputusan sistem CBR.
6. Perluasan Dataset dan Kelas : Menguji sistem dengan dataset yang lebih besar dan mencakup berbagai kategori tindak pidana untuk evaluasi generalisasi yang lebih komprehensif.
7. Eksplorasi Arsitektur Model Lanjutan : Menjelajahi arsitektur *deep learning* yang lebih canggih (selain SVM di atas BERT *embedding*) untuk menangkap pola yang lebih kompleks dalam data tekstual hukum.

**VI. kesimpulan**

Penelitian ini telah berhasil membangun dan mengevaluasi sistem klasifikasi dokumen putusan pengadilan, khususnya pidana umum pembunuhan, menggunakan pendekatan Case-Based Reasoning (CBR). Kami mengintegrasikan dua metode representasi teks yang berbeda, yaitu TF-IDF dan BERT, serta membandingkan kinerja sistem CBR dengan model *supervised learning* konvensional seperti Naive Bayes dan Support Vector Machine (SVM). Hasil evaluasi klasifikasi menunjukkan performa yang sangat kuat dan menjanjikan dari model berbasis TF-IDF. Kombinasi TF-IDF **+** Naive Bayes dan TF-IDF **+** SVM mencapai akurasi, *precision*, *recall*, dan *f1-score* 100%. Ini menegaskan bahwa untuk dataset yang relatif terbatas (300 dokumen) dan mungkin memiliki fitur leksikal yang sangat diskriminatif, pendekatan berbasis statistik seperti TF-IDF masih sangat efektif dan efisien dalam mengklasifikasikan putusan pidana. Di sisi lain, meskipun BERT **+ SVM** menunjukkan potensi signifikan dengan *f1-score* 0.92, performanya sedikit tertinggal dibandingkan TF-IDF. Keterbatasan ini kemungkinan besar disebabkan oleh ukuran dataset yang belum optimal untuk *fine-tuning* model BERT agar dapat sepenuhnya memanfaatkan kemampuan pemahaman semantiknya terhadap domain hukum. Evaluasi *retrieval* kasus serupa, yang merupakan inti dari sistem CBR, menghasilkan **Precision@5 sebesar 0.60**. Angka ini menunjukkan bahwa sistem memiliki kemampuan yang memadai untuk membantu pengguna dalam mencari referensi putusan yang relevan dari basis data, menandakan bahwa sistem sudah dapat menjadi alat bantu yang bermanfaat dalam penelusuran preseden. Namun, nilai ini juga mengindikasikan adanya ruang untuk peningkatan, terutama dalam menangkap kemiripan kontekstual dan semantik yang lebih dalam. Analisis kegagalan *retrieval* mengidentifikasi isu-isu seperti dokumen pendek, tumpang tindih semantik antar kelas hukum, dan ambiguitas dalam label amar sebagai area yang memerlukan perbaikan lebih lanjut. Secara keseluruhan, penelitian ini merupakan bukti bahwa pendekatan CBR dengan dukungan representasi teks modern (TF-IDF dan BERT) efektif untuk klasifikasi putusan pidana. Sistem yang dikembangkan memiliki potensi besar untuk dikembangkan lebih lanjut. Dengan implementasi rekomendasi yang telah disampaikan, seperti *fine-tuning* BERT pada korpus hukum yang lebih besar, penambahan fitur hukum eksplisit, penerapan strategi validasi yang lebih kuat (misalnya, k-fold cross-validation), dan integrasi metode *Explainable AI*, diharapkan sistem ini dapat mencapai tingkat akurasi dan keandalan yang lebih tinggi di masa mendatang. Penelitian ini menjadi kontribusi penting dalam pemanfaatan teknologi Pengolahan Bahasa Alami dan Pembelajaran Mesin untuk mendukung analisis dokumen hukum dan sistem peradilan di Indonesia.

Baik, saya akan melanjutkan laporan dari bagian **4.2 Evaluasi Retrieval (CBR)** hingga **Daftar Pustaka**, dengan detail yang diperlukan untuk mencapai target 8 halaman dan tetap dalam format akademik.

## Daftar Pustaka

[1] Watson, I. (1997). *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann Publishers.

[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171-4186). Association for Computational Linguistics.

[3] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

[4] Mahkamah Agung RI. (n.d.). Direktori Putusan. Retrieved from<https://putusan3.mahkamahagung.go.id>

[5] Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing* (3rd ed. draft). Pearson Education.

[6] Salton, G., & McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

[7] Ramos, J. (2003). Using TF-IDF to Determine Word Relevance in Document Queries. In *Proceedings of the First Instructional Conference on Machine Learning*.

[8] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (pp. 3982-3992).

[9] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv preprint* arXiv:1301.3781.

[10] Luhn, H. P. (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development*, 1(4), 309-317.

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. In *Advances in Neural Information Processing Systems*, 30.

[12] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

[13] Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text Classification Algorithms: A Survey. *Information*, 10(4), 150.

[14] Zhang, L., Wang, S., & Liu, B. (2018). Deep Learning for Sentiment Analysis: A Survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4), e1253.

[15] Cambria, E., & White, B. (2014). Jumping NLP Curves: A Review of Natural Language Processing Research. *IEEE Computational Intelligence Magazine*, 9(2), 48-57.