BAB II

TINJAUAN PUSTAKA

2.1 Pidana Umum

Pidana Umum adalah suatu tindak kejahatan yang dapat dilakukan oleh siapa saja tanpa terkecuali, yang meliputi berbagai kejahatan yang terencana maupun tidak terencana dan dapat dilakukan oleh siapapun secara umum (Mangkepriyanto, 2019). Segala aturan yang disahkan tercantum dalam Kitab Undang-Undang Hukum Pidana (KUHP). Dalam konteks analisis data, karakteristik perkara pidana umum bersifat dinamis dan fluktuatif dari waktu ke waktu, sehingga memerlukan pendekatan komputasional untuk mengidentifikasi pola dan kecenderungan kejadian. Data pidana umum yang bersifat historis dan berurutan dapat dimanfaatkan sebagai dasar penerapan metode machine learning untuk melakukan prediksi jumlah perkara serta analisis perkembangan kriminalitas secara kuantitatif.

2.2 *Case Management System* (CMS)

Case Management System merupakan sistem informasi yang digunakan untuk mencatat, menyimpan, dan mengelola data kasus atau perkara secara terstruktur. Dikutip dari Jurnal Suara Hukum, Jaksa Agung Republik Indonesia telah menerbitkan Instruksi Jaksa Agung Nomor 3 Tahun 2020 (Insja No. 3 Tahun 2020). tentang Penggunaan Aplikasi Sistem Manajemen Perkara (*Case Management System*) sehingga adanya upaya yang menekankan para Jaksa yang menangani perkara untuk melakukan input data penanganan perkara dan melengkapi administrasi persuratan pada setiap tahapan penanganan perkara yang ditanganinya, sehingga terdapat peningkatan kinerja dan pelayanan publik yang

dipenuhi oleh institusi Kejaksaan terhadap masyarakat yang mencari keadilan (Suryadi & Supardi, 2021).

CMS menyediakan data historis yang konsisten dan terdokumentasi dengan baik, sehingga sangat sesuai digunakan sebagai sumber data dalam penelitian berbasis machine learning. Data yang tersimpan dalam CMS dapat diolah lebih lanjut melalui tahapan pra-pemrosesan sebelum digunakan untuk pelatihan model prediksi dan klasifikasi. Pemanfaatan CMS sebagai sumber data mendukung penerapan analisis berbasis data dan pengambilan keputusan yang lebih objektif.

2.3 *Machine* Learning

Machine Learning merupakan cabang dari kecerdasan buatan yang berfokus pada pengembangan algoritma yang memungkinkan sistem komputer mempelajari pola dari data dan membuat keputusan atau prediksi tanpa diprogram secara eksplisit. Pendekatan ini memanfaatkan data historis sebagai dasar pembelajaran sehingga model mampu melakukan generalisasi terhadap data baru. *Machine learning* banyak diterapkan pada permasalahan prediksi dan klasifikasi karena kemampuannya dalam mengolah data berukuran besar serta menangani hubungan nonlinier antarvariabel (Permana, 2023). Dalam konteks data kriminalitas dan perkara, *machine learning* digunakan untuk menganalisis tren, memprediksi jumlah kejadian, serta mengelompokkan status atau kategori tertentu berdasarkan atribut data yang tersedia.

2.4 Klasifikasi

Klasifikasi merupakan suatu teknik dalam data mining yang digunakan untuk memetakan data ke dalam berbagai kelompok (kelas) yang sudah ditentukan

sebelumnya (Sahbana et al., 2025). Mengenali pola yang menjadi pembeda dalam setiap data ke dalam berbagai kelas berdasarkan atribut yang relevan merupakan tujuan utama dari klasifikasi. Menurut Sundari (2023) Hal menarik dari proses klasifikasi adalah fungsi yang mampu memprediksi kelas atau label yang cocok untuk data yang tidak memiliki kelas. Implementasi klasifikasi dalam kehidupan sehari-hari bisa dijumpai dalam deteksi spam e-mail, identifikasi penyakit berdasarkan gejala, penentuan profil pelanggan, pengenalan pola, dan sebagainya. Klasifikasi memiliki berbagai jenis metode seperti Algoritma C4.5, *Naive Bayes*, *Convolutional Neural Network,* dan lainnya. Dalam penelitian ini, penekanan metode *1D-Convolutional Neural Network* sangat berpengaruh untuk klasfikasi kategori status perkara.

2.5 Supervised Learning

Supervised learning merupakan metode pembelajaran mesin yang menggunakan data berlabel sebagai dasar pelatihan model. Dalam pendekatan ini, model mempelajari hubungan antara data masukan dan target keluaran yang telah diketahui. Dua tugas utama dalam *supervised learning* adalah prediksi dan klasifikasi. Prediksi bertujuan untuk memperkirakan nilai atau jumlah di masa depan berdasarkan pola data historis, sedangkan klasifikasi digunakan untuk menentukan kelas atau kategori tertentu dari suatu data. Kombinasi *supervised learning* dengan model *deep learning*, seperti LSTM untuk prediksi dan 1D-CNN untuk klasifikasi, terbukti efektif dalam meningkatkan akurasi hasil analisis (Bowo, 2024).

2.6 *Long Short-Term Memory* (LSTM)

Long Short-Term Memory merupakan varian dari *Recurrent Neural Network* yang dirancang khusus untuk menangani data deret waktu (*time series*). LSTM memiliki mekanisme sel memori dan gerbang (*input gate, forget gate, dan output gate*) yang memungkinkan model mempertahankan informasi penting dalam jangka panjang. Dengan struktur tersebut, LSTM mampu mengatasi permasalahan *vanishing gradient* yang umum terjadi pada RNN konvensional (Rosyd et al., 2024). Penerapan LSTM telah banyak digunakan dalam penelitian prediksi data runtun waktu, seperti prediksi harga bahan pokok dan analisis tren kriminalitas, karena kemampuannya dalam mempelajari pola temporal secara akurat (Wiranda & Sadikin, 2019).

2.7 *Deep Learning*

Deep Learning merupakan pengembangan dari *machine learning* yang menggunakan jaringan saraf tiruan dengan banyak lapisan tersembunyi (*deep neural network)* untuk menganalisasi dan memproses data (Karim et al., 2025). Pendekatan ini memungkinkan proses ekstraksi fitur dilakukan secara otomatis dan bertingkat, sehingga sangat efektif dalam menangani data kompleks. *Deep learning* terbukti unggul dalam tugas prediksi dan klasifikasi karena mampu memodelkan pola nonlinier yang sulit ditangkap oleh metode konvensional (Yanto et al., 2021). Berbagai arsitektur *deep learning* telah digunakan dalam penelitian, seperti *Recurrent Neural Network* (RNN) dan *Convolutional Neural Network* (CNN), yang masing-masing memiliki keunggulan sesuai dengan karakteristik data yang dianalisis.

2.8 *1 Dimension - Convolutional Neural Network* (1D-CNN)

Convolutional Neural Network merupakan arsitektur *deep learning* yang menggunakan operasi konvolusi untuk mengekstraksi fitur penting dari data. Dalam implememtasinya, CNN CNN bisa digunakan untuk memproses data dalam bentuk gambar dengan mengenali pola visual meliputi warna, bentuk, dan tekstur (Sari et al., 2025). CNN tidak hanya terbatas pada pengolahan citra dua dimensi, tetapi juga dapat diterapkan pada data satu dimensi melalui pendekatan 1D-CNN. Pada 1D-CNN, proses konvolusi dilakukan pada data berurutan sehingga efektif dalam mengenali pola lokal pada data deret waktu atau data numerik berurutan. Penelitian menunjukkan bahwa 1D-CNN memiliki performa yang baik dalam tugas klasifikasi karena kemampuannya dalam mengekstraksi fitur secara otomatis tanpa memerlukan rekayasa fitur manual (Pramana, 2024).

2.9 Python & Tensorflow

Python merupakan bahasa pemrograman populer dan dikenal oleh berbagai *developer Machine Learning*, *Data Scientist,* maupun *Data Miner*. Python memiliki berbagai kelebihan yakni Mudah dipelajari dan dipahami, dan memiliki banyak *library* dan *Package Manage* (ID, 2021). Selain itu, *Python* meriupakan bahasa pemrograman yang *open source* dan terbuka untuk siapa saja secara gratis.

Tensorflow merupakan *library* perangkat lunak yang dikembangkan oleh Google Bratin dalam organisasi penelitian Mesin Cerdas Google, yang bertujuan untuk melakukan *Machine Learning* dan penelitian *Deep Neural Network*. *Tensorflow* bekerja dengan mempermudah perhitungan banyak ekspresi matematis dimana masalahnya adalah waktu yang dibutuhkan untuk melakukan perhitungan.

2.10 Teknologi Ekstraksi Data (*Web Scraping* dan API)

Web scraping merupakan sebuah teknik otomatisasi yang digunakan untuk mengekstrak data dari suatu situs web secara terprogram. Berbeda dengan metode konvensional di mana pengumpulan data dilakukan secara manual melalui aktivitas penggandaan (*copy-paste*), web scraping memanfaatkan skrip kode atau perangkat lunak khusus untuk mengurai (*parsing*) struktur kode halaman web dan mengambil informasi spesifik dalam skala besar secara efisien. Dalam perkembangannya, teknik ini menjadi pondasi krusial dalam ekosistem ilmu data (*data science*) karena mampu mentransformasikan data tidak terstruktur (*unstructured data*) yang tersebar di internet menjadi data terstruktur (*structured data*) yang siap dianalisis.

Meskipun memberikan kemudahan dalam akuisisi data, implementasi web scraping wajib memperhatikan etika penambangan data dan aspek legalitas. Setiap penyedia layanan situs web umumnya memiliki berkas penunjuk standar komunikasi berupa dokumen robots.txt yang tersimpan di dalam direktori akar (*root*) peladen (*server*). Berkas tersebut mendefinisikan batasan mengenai bagian halaman mana saja yang diizinkan untuk diakses oleh robot penjelajah (*crawler/scraper*) maupun bagian yang dilarang (*disallow*). Pelanggaran terhadap aturan ini dapat menyebabkan beban berlebih pada peladen (*server overload*) akibat tingginya frekuensi permintaan (*request rate*), sehingga pengembang skrip wajib menerapkan interval waktu penundaan (*delay*) demi menjaga kestabilan operasional sistem target.

2.10.1 Protokol HTTP (*Hypertext Transfer Protocol*)

Protokol HTTP (*Hypertext Transfer Protocol*) merupakan standar dasar yang mengatur mekanisme komunikasi data di dalam ruang lingkup jaringan

internet. Mekanisme ini bekerja berdasarkan arsitektur *Client-Server*, di mana peramban web (*browser*) atau skrip otomatisasi bertindak sebagai pihak peminta (*client*), sedangkan komputer yang menyimpan data situs web bertindak sebagai penyedia layanan (*server*). Proses transmisi data diawali ketika *client* mengirimkan pesan HTTP *Request* menuju URL target, kemudian *server* akan memproses permintaan tersebut dan mengembalikan pesan HTTP *Response* yang berisi kode status beserta muatan data (*payload*).

Di dalam arsitektur HTTP, terdapat beberapa metode permintaan (*request methods*) yang memiliki fungsi spesifik, di antaranya:

1. *GET*: Digunakan untuk meminta atau mengambil data dari suatu sumber (*resource*) tanpa memberikan dampak perubahan pada data di sisi peladen.
2. *POST*: Digunakan untuk mengirimkan entitas data atau parameter tertentu menuju peladen, biasanya dimanfaatkan dalam proses autentikasi login atau pengiriman formulir pencarian data yang dinamis.

Setelah peladen menerima permintaan, sistem akan mengirimkan respons yang disertai dengan *HTTP Response Status Codes* sebagai indikator tingkat keberhasilan transaksi data. Kode berawalan 2xx (seperti 200 OK) menandakan bahwa permintaan berhasil diproses sempurna. Kode berawalan 4xx (seperti 403 *Forbidden* atau 404 *Not Found*) menunjukkan adanya kesalahan di sisi *client*, seperti ketiadaan hak akses atau alamat URL yang tidak valid. Sementara kode berawalan 5xx (seperti 500 *Internal Server*

Error) mengindikasikan bahwa kegagalan penarikan data disebabkan oleh malafungsi pada sistem intrnal peladen.

2,10.2 *Application Programming Interface* (API)

Application Programming Interface (API) merupakan sekumpulan antarmuka, protokol, dan kakas pembangun perangkat lunak yang berfungsi sebagai jembatan komunikasi antar-aplikasi yang berbeda agar dapat saling bertukar data secara instan. Pada arsitektur web modern, API berperan memisahkan lapisan presentasi (*front-end*) dengan lapisan basis data (*back-end*). Komunikasi ini umumnya berjalan secara asinkronus menggunakan teknologi AJAX (*Asynchronous JavaScript and XML*) atau fungsi *fetch*, di mana halaman web dapat memperbarui isi datanya tanpa harus melakukan pemuatan ulang (*reload*) pada keseluruhan dokumen HTML.

Ketika pengguna berinteraksi dengan portal informasi digital, peramban web di balik layar akan menembak sebuah URL *endpoin*t API internal untuk mengambil data berformat murni. Karakteristik portal web yang memanfaatkan sub-sistem API internal ini memberikan keuntungan besar bagi efisiensi aktivitas *web scraping*. Peneliti tidak perlu lagi melakukan ekstraksi teks yang rumit dari elemen penanda visual dokumen HTML (HTML tags seperti atau <table>), melainkan cukup menangkap parameter pertukaran data tersebut melalui analisis lalu lintas jaringan (*network traffic inspection*) dan mengarahkan skrip pengumpul data langsung menuju *endpoint* API peladen untuk mendapatkan data yang homogen dan bersih.

2.11 Struktur Data *JavaScript Object Notation* (JSON)

JavaScript Object Notation (JSON) adalah sebuah format pertukaran data berbasis teks standar yang bersifat ringan (*lightweight*), independen, dan mudah dibaca serta ditulis oleh manusia, sekaligus mudah diparsing dan dibuat oleh mesin komputer. Meskipun mengadopsi konvensi penulisan dari bahasa pemrograman JavaScript, JSON merupakan format data yang sepenuhnya bebas dari ketergantungan bahasa (*language-independent*). Hampir seluruh bahasa pemrograman modern, termasuk Python, telah menyediakan modul bawaan khusus untuk mendukung konversi data teks JSON menjadi struktur data internal yang dapat dimanipulasi secara langsung.

Karakteristik utama JSON terletak pada efisiensi ruang penyimpanannya jika dibandingkan dengan format pendahulunya seperti XML (*Extensible Markup Language*). JSON tidak membutuhkan tag pembuka dan penutup yang repetitif, melainkan mengandalkan karakter tokenisasi yang minimalis. Sifatnya yang ringkas ini meminimalkan ukuran berkas (*file size*) saat proses transmisi data berlangsung di dalam jaringan, sehingga mampu menghemat konsumsi *bandwidth* peladen dan mempercepat waktu respons pengiriman data hasil ekstraksi dari sistem basis data.

Struktur data di dalam JSON dibangun berdasarkan dua jenis elemen universal yang sangat fleksibel:

1. Koleksi Pasangan Kunci dan Nilai (*Key-Value Pairs*): Di dalam bahasa pemrograman, struktur ini diimplementasikan sebagai objek (*object*), kamus (*dictionary*), atau tabel *hash*. Elemen ini diapit oleh tanda kurung

kurawal ({}). Setiap kunci (*key*) wajib bertipe data string dan dipisahkan dari nilainya (*value*) menggunakan tanda titik dua(:).

2. Daftar Nilai Terurut (*Ordered List of Values*): Struktur ini diimplementasikan sebagai larik (*array*), vektor, atau daftar (list). Elemen ini diapit oleh tanda kurung siku ([]) dan setiap anggotanya dipisahkan menggunakan tanda koma(,).

Tipe data yang diizinkan untuk mengisi nilai (*value*) di dalam dokumen JSON sangat bervariasi, meliputi tipe data *string* (wajib diapit tanda petik dua), *number* (*integer* maupun pecahan*), boolean (true* atau *false)*, serta nilai kosong (*null*). Selain itu, JSON mendukung struktur bersarang (*nested structure*), di mana sebuah objek dapat berisi objek lain atau larik di dalamnya, sehingga memungkinkan representasi hubungan data hierarkis yang rumit menjadi satu kesatuan berkas teks yang padat.

2.12 Pra-pemrosesan Data (*Data Preprocessing*)

Di dalam siklus pengembangan sistem kecerdasan buatan berbasis *Machine Learning* maupun *Deep Learning*, tahapan pra-pemrosesan data (*data preprocessing*) memegang peranan yang sangat vital dan menentukan tingkat keberhasilan performa model. Fenomena ini erat kaitannya dengan paradigma baku komputer berbunyi "*Garbage In, Garbage Out*", yang menegaskan bahwa seunggul apa pun arsitektur algoritma yang dibangun, sistem akan menghasilkan keluaran prediksi yang bias dan tidak akurat apabila disuapi oleh data masukan yang berkualitas buruk, kotor, atau cacat.

Data mentah yang diperoleh secara langsung dari lapangan, termasuk data hasil web scraping otomatis, umumnya berada dalam kondisi tidak murni. Dataset

tersebut kerap mengandung komponen derau (noise), ketidakkonsistenan format penulisan, nilai pencilan (outliers), hingga ketiadaan data pada kolom tertentu (missing values). Algoritma jaringan saraf tiruan kontemporer (seperti LSTM dan CNN) beroperasi sepenuhnya menggunakan perhitungan matriks matematika linier tingkat tinggi. Ketiadaan proses pembersihan data akan mengakibatkan *error* pada komputasi internal, mengacaukan proses pembaruan bobot (*weight adjustment*) saat fase pelatihan, serta menurunkan kemampuan generalisasi model terhadap data baru.

Untuk mengukur kelayakan suatu dataset sebelum memasuki fase pemodelan eksperimental, terdapat empat dimensi utama kualitas data yang wajib dipenuhi melalui proses pra-pemrosesan:

1. Kelengkapan (*Completeness*): Merujuk pada pemenuhan aspek ketersediaan seluruh nilai data yang diperlukan. Dataset tidak boleh memiliki sel kosong pada fitur-fitur krusial yang menjadi variabel prediktor sistem.
2. Konsistensi (*Consistency*): Menuntut keseragaman format dan tipe data di seluruh baris dataset. Sebagai contoh, representasi nilai tanggal tidak boleh bercampur antara format numerik dengan teks.
3. Akurasi (*Accuracy*): Memastikan bahwa data yang tercatat di dalam sistem merepresentasikan kebenaran kondisi riil di lapangan secara valid tanpa adanya kesalahan input atau bias instrumen perekam data.
4. Kebersihan (*Cleanliness*): Menunjukkan tingkat sterilitas data dari elemen pengganggu sisa proses ekstraksi, seperti tag sintaksis sistem yang

terbawa, spasi ganda yang tidak sengaja terbentuk, atau karakter asing non-alfanumerik yang tidak bermakna.

2.13 Teknik *Data Cleaning* dan Imputasi *Missing Value*

2.13.1 Penanganan Data Duplikat dan Kontradiktif

Data duplikat didefinisikan sebagai kondisi di mana terdapat dua atau lebih baris data yang memiliki nilai atribut yang identik secara keseluruhan di dalam satu dataset. Dalam proses *web scraping* dari portal *Case Management System* (CMS) Kejaksaan, redundansi data ini rentan terjadi akibat interupsi jaringan saat proses penarikan, pengulangan halaman pencarian (*paginating*), atau ketidaksempurnaan sistem basis data sumber dalam memperbarui log perkara. Keberadaan data duplikat wajib dieliminasi karena dapat memicu distorsi serius pada fase pelatihan model kecerdasan buatan. Model akan cenderung memberikan bobot berlebih pada pola data yang mengalami replikasi tersebut, sehingga memicu kondisi *overfitting* di mana model cerdas dalam mengenali data latih namun gagal melakukan generalisasi pada data baru.

Selain redundansi, permasalahan data kontradiktif juga menjadi atensi khusus dalam pembersihan data. Data kontradiktif terjadi ketika beberapa baris data memiliki nomor perkara yang sama tetapi menampilkan status perkembangan, nama jaksa, atau tanggal registrasi yang saling bertolak belakang. Penanganan anomali ini memerlukan pendekatan berbasis validasi logika, seperti menetapkan aturan kronologis berdasarkan stempel waktu (*timestamp*) paling mutakhir, atau melakukan penghapusan secara selektif

(*dropping*) apabila tingkat inkonsistensinya dinilai merusak integritas dataset penanganan perkara.

2.13.2 Klasifikasi *Missing Value*

Ketiadaan data atau *missing value* merupakan suatu kondisi tidak lengkapnya informasi pada sel-sel tertentu di dalam matriks dataset. Secara teoretis, penyebab terjadinya fenomena ini dikelompokkan ke dalam tiga mekanisme utama menurut Little dan Rubin, yaitu:

1. *Missing Completely at Random* (MCAR)**:** Terjadi apabila hilangnya data pada suatu fitur sama sekali tidak memiliki keterkaitan dengan nilai fitur itu sendiri maupun fitur lainnya di dalam dataset. Sifat hilangnya data ini murni bersifat acak, misalnya kegagalan peladen web memuat kolom nama jaksa pada beberapa nomor perkara tertentu akibat gangguan teknis sesaat.
2. *Missing at Random* (MAR): Terjadi jika peluang hilangnya data pada suatu variabel memiliki hubungan sistematis dengan variabel lain, tetapi tidak berkaitan dengan nilai laten dari variabel yang hilang tersebut. Sebagai contoh, kolom "Pasal Tindak Pidana" sering kali kosong khusus pada perkara-perkara yang tercatat berasal dari Polsek tertentu karena adanya keterlambatan administratif penginputan berkas di unit tersebut.
3. *Not Missing at Random* (NMAR): Merupakan kondisi yang paling kompleks, di mana mekanisme hilangnya data berkaitan langsung dengan nilai dari variabel yang hilang itu sendiri. Contohnya, status penyelesaian perkara tidak diisi oleh operator justru karena perkara

tersebut statusnya masih menggantung atau belum memiliki kekuatan hukum tetap.

2.13.3 Metode Imputasi Nilai Hilang

Untuk menjaga keseimbangan struktur data tanpa harus membuang baris yang mengandung informasi berharga, diterapkan teknik imputasi, yaitu prosedur pengisian nilai yang kosong menggunakan nilai pengganti yang logis secara statistik. Pemilihan metode imputasi sangat bergantung pada tipe data dari karakteristik fitur yang ditangani:

1. Imputasi Modus (Nilai Kerapatan Tertinggi): Sangat ideal diterapkan pada fitur bertipe kategorikal atau teks, seperti nama Jaksa Penuntut Umum (JPU) atau asal penyidik kepolisian. Skema ini bekerja dengan mencari nilai yang paling sering muncul di dalam kolom tersebut, kemudian menjadikannya sebagai nilai substitusi pada sel yang kosong.
2. Imputasi Mean atau Median**:** Diterapkan pada data numerik fluktuatif. Imputasi menggunakan nilai rata-rata (*mean*) cocok untuk data yang berdistribusi normal, sedangkan nilai tengah (*median*) lebih disarankan jika dataset mengandung banyak nilai pencilan (*outliers*) guna menghindari bias perhitungan.

2.14 Penyandian Fitur Kategorikal (*Feature Encoding*)

2.14.1 Konsep Representasi Numerik

Arsitektur *Deep Learning* kontemporer seperti *Long Short-Term Memory* (LSTM) dan *Convolutional Neural Network* (CNN) pada dasarnya merupakan sekumpulan operasi aljabar linier yang memproses nilai-nilai numerik berskala kontinu dalam bentuk matriks atau tensor. Algoritma ini

tidak dibekali kemampuan bawaan untuk menginterpretasikan makna semantik dari deretan karakter string atau teks alfabetis secara langsung.

Oleh karena itu, fitur-fitur penting dari CMS Kejaksaan yang mayoritas bertipe data teks (seperti jenis tindak pidana, status perkara, dan nama penegak hukum) wajib melalui tahapan transformasi penyandian fitur (*feature encoding*). Proses ini bertugas memetakan setiap label teks menjadi representasi kode angka yang terstruktur tanpa menghilangkan esensi informasi dan hubungan korelatif antar-variabel yang dikandungnya.

2.14.2 Metode *Label Encoding*

Label Encoding merupakan teknik penyandian yang bekerja dengan mengonversi setiap nilai unik di dalam suatu kolom kategorikal menjadi nilai integer urut, dimulai dari angka 0 hingga $N-1$ (di mana N mewakili jumlah total kategori unik). Sebagai ilustrasi, jika terdapat fitur asal penyidik dengan kategori "Polres Labuhanbatu", "Polsek Rantauprapat", dan "Polsek Bilah Hilir", maka metode ini akan mengubahnya secara berurutan menjadi nilai numerik 0, 1, dan 2.

Meskipun metode ini sangat efisien dari segi penggunaan ruang memori karena tidak menambah dimensi kolom baru pada dataset, *Label Encoding* memiliki kelemahan intrinsik apabila diterapkan pada fitur nominal (kategori tanpa tingkatan). Algoritma matematika pada jaringan saraf dapat salah menginterpretasikan urutan angka tersebut sebagai sebuah hierarki nilai bobot, di mana kategori dengan angka 2 dianggap memiliki kedudukan lebih tinggi atau lebih penting daripada angka 0.

2.14.3 Metode *One-Hot Encoding*

Guna mengatasi bias pengurutan yang muncul pada *Label Encoding*, digunakan teknik *One-Hot Encoding*. Metode ini bekerja dengan memecah fitur kategorikal tunggal menjadi beberapa fitur biner baru (sering disebut sebagai *dummy variables*) sesuai dengan jumlah total kategori unik yang ada. Setiap baris data pada kolom biner baru tersebut hanya akan bernilai angka 1 (*active*) untuk merepresentasikan keberadaan kategori tersebut, dan bernilai 0 (*inactive*) untuk kategori lainnya.

Kelebihan utama dari *One-Hot Encoding* adalah kemampuannya dalam menghilangkan asumsi urutan numerik dari fitur nominal, sehingga seluruh kategori diperlakukan secara setara di hadapan model neural network. Namun, implementasi metode ini harus memperhatikan fenomena *curse of dimensionality*. Jika suatu kolom memiliki ratusan variasi nilai unik (kardinalitas tinggi), maka *One-Hot Encoding* akan melipatgandakan jumlah kolom dataset secara ekstrem, yang berpotensi meningkatkan beban komputasi dan memperlambat proses pelatihan model.

2.15 Skalasi dan Normalisasi Data (*MinMax Scaler*)

2.15.1 Definisi Skalasi Data

Skalasi data merupakan prosedur pra-pemrosesan yang bertujuan untuk menyamakan rentang atau skala nilai dari seluruh fitur numerik yang terdapat di dalam dataset. Dalam kasus riil penanganan perkara di Kejaksaan, terdapat perbedaan skala yang cukup kontras antar-fitur; misalnya fitur kuantitas perkara bulanan berkisar di angka puluhan atau ratusan, sementara fitur hasil penyandian biner hanya bernilai 0 dan 1.

Jika dataset langsung diumpankan ke model tanpa adanya pembatasan skala, maka fitur dengan magnitudo nilai intrinsik yang besar akan mendominasi perhitungan fungsi biaya (*cost function*). Hal ini menyebabkan algoritma mengabaikan fitur bernilai kecil meskipun fitur tersebut memiliki korelasi yang kuat terhadap target prediksi. Skalasi data menjamin setiap variabel memberikan kontribusi yang proporsional selama proses pembaruan parameter model dilakukan.

2.15.2 Formula Matematis *MinMax Scaler*

Salah satu metode transformasi skala yang paling populer dalam implementasi algoritma berbasis *gradien descend* adalah *MinMax Scaler*. Metode ini melakukan transformasi linier terhadap data asli untuk diposisikan ke dalam interval batasan baru yang tetap, umumnya berada pada rentang angka 0 sampai dengan 1. Secara matematis, kalkulasi normalisasi ini dijabarkan melalui persamaan berikut:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Dimana :

X = Jumlah perkara data asli/mentah pada bulan tertentu

X_{min} = Jumlah perkara paling sedikit

X_{max} = Jumlah perkara paling banyak

X_{norm} = Nilai hasil Normalisasi.

Melalui penerapan formula tersebut, nilai terkecil pada dataset akan otomatis dikonversi menjadi angka 0, nilai terbesar akan dikonversi menjadi angka 1, dan nilai lainnya akan terdistribusi secara proporsional di dalam ruang interval [0, 1]. Pembatasan rentang nilai ini sangat krusial dalam

mempercepat proses konvergensi gradien sewaktu pelatihan arsitektur *Deep Learning* dilangsungkan serta menjaga kestabilan numerik dari fungsi aktivasi model.

2.16 Karakterisitik Data Deret Waktu (*Time Series Analysis*)

Data deret waktu (*time series data*) didefinisikan sebagai sekumpulan runtunan pengamatan atau observasi terhadap suatu variabel kuantitatif yang dicatat secara kronologis berdasarkan interval waktu yang konstan dan berurutan. Interval waktu tersebut dapat berskala harian, mingguan, bulanan, maupun tahunan. Di dalam ranah komputasi dan statistika, analisis deret waktu tidak hanya bertujuan untuk mendeskripsikan perilaku data masa lalu, melainkan berfokus pada pembangunan model matematis yang mampu mengeksplorasi ketergantungan temporal (*temporal dependencies*) guna memproyeksikan nilai variabel tersebut di masa depan.

Dalam konteks penanganan perkara di Kejaksaan Negeri Labuhanbatu, data jumlah registrasi berkas perkara tindak pidana umum per bulan merupakan manifestasi riil dari data deret waktu. Karakteristik utama yang membedakan data deret waktu dengan data lintas sektoral (cross-sectional) adalah adanya faktor korelasi antar-waktu (autocorrelation). Nilai volume perkara pada bulan berjalan memiliki keterkaitan erat atau dipengaruhi oleh pola volume perkara pada bulan-bulan sebelumnya, sehingga memerlukan pendekatan pemodelan khusus yang sensitif terhadap urutan sekuensial.

Secara teoretis, fluktuasi gerakan data yang terjadi di dalam data deret waktu dapat diuraikan (*decomposed*) menjadi kombinasi dari empat komponen struktural utama, yaitu:

1. Tren (*Trend*): Merupakan kecenderungan arah gerakan data dalam jangka panjang, baik menunjukkan pola kecenderungan menaik (*upward trend*) maupun menurun (*downward trend*). Tren mencerminkan faktor fundamental yang berubah secara gradual, seperti pertumbuhan populasi penduduk atau perubahan kebijakan hukum.
2. Musiman (*Seasonality*): Adalah pola fluktuasi data yang berulang secara teratur dan konsisten dalam kurun waktu kurang dari satu tahun. Pola ini biasanya dipicu oleh siklus kalender, seperti lonjakan kasus tindak pidana tertentu menjelang hari raya keagamaan atau akhir tahun.
3. Siklus (*Cyclical*): Merupakan gelombang fluktuasi naik-turun jangka panjang yang durasinya bersifat dinamis dan tidak berulang dalam periode yang tetap. Siklus umumnya berkaitan dengan perkembangan makro, seperti pergeseran kondisi sosial-ekonomi masyarakat.
4. Irregularitas (*Irregular/Noise*): Adalah komponen pergerakan data yang bersifat acak, tidak teratur, dan tidak dapat diprediksi (*stochastic*). Komponen derau (*noise*) ini biasanya disebabkan oleh peristiwa insidental yang terjadi secara mendadak di lapangan, seperti konflik sosial spontan atau bencana alam.

2.17 Konsep *Sliding Window* dalam Pembelajaran Terawasi

Data deret waktu murni pada dasarnya memiliki struktur univariat tunggal yang bersifat sekuensial linear. Di sisi lain, rumpun algoritma pembelajaran terawasi (*supervised learning*) seperti *Long Short-Term Memory* (LSTM) menuntut format masukan data yang memiliki pasangan fitur prediktor (X) dan label target (y) yang jelas pada setiap barisnya. Oleh karena itu, diperlukan sebuah teknik

transformasi struktural untuk mengonversi deret waktu murni tersebut menjadi matriks data pembelajaran terawasi, yang secara teoretis dikenal dengan konsep *sliding window* atau jendela bergeser.

Proses transformasi ini bekerja dengan merestrukturisasi dataset sekuensial menggunakan pendekatan lag waktu. Nilai data pada titik waktu masa lalu bertindak sebagai variabel independen yang digunakan untuk mengonstruksi pola, sedangkan nilai data pada titik waktu yang akan datang diposisikan sebagai variabel dependen yang wajib ditebak oleh model. Melalui restrukturisasi ini, model *Deep Learning* dapat mempelajari hubungan pemetaan fungsional f antara masa lalu dan masa depan secara matematis.

Lookback atau *window size* merupakan parameter yang menentukan lebar jendela waktu atau jumlah langkah waktu mundur (*time steps*) ke belakang yang digunakan sebagai basis informasi prediktor (X). Penentuan ukuran *lookback* sangat krusial dan bersifat eksperimental. Jika ukuran *lookback* terlalu kecil, model tidak akan mendapatkan informasi konteks histori yang cukup untuk mengenali tren jangka panjang. Sebaliknya, jika *lookback* terlalu besar, beban komputasi model akan meningkat secara drastis dan rentan memasukkan komponen derau yang mengacaukan akurasi.

Sebagai ilustrasi matematis, apabila ditentukan ukuran *lookback* sebesar n, maka untuk memprediksi jumlah perkara pada waktu t (y_t), jendela akan mengambil data histori dari $t-n$ hingga $t-1$ sebagai fitur masukan:

$$X = [x_{t-n}, x_{t-n+1}, \dots, x_{t-1}]$$

Setelah pasangan masukan dan target untuk waktu t terbentuk, jendela akan digeser maju sebanyak satu langkah waktu (*step size = 1*) untuk menyusun

pasangan data baru pada waktu $t + 1$ Langkah pergeseran ini dilakukan secara repetitif menyusuri seluruh panjang data historis hingga membentuk matriks dataset baru yang siap diumpankan ke lapisan LSTM.

2.18 Komponen Arsitektur *1 Dimensions - Convolutional Neural Network* (1D-CNN)

2.18.1 Lapisan Konvolusi 1 Dimensi (*Convolutional Layer 1D*)

Meskipun arsitektur *Convolutional Neural Network* (CNN) pada awalnya jamak diimplementasikan untuk ekstraksi fitur spasial dua dimensi pada data citra atau gambar, variasi Jaringan Saraf Konvolusi 1 Dimensi (1D-CNN) terbukti sangat efektif untuk mengolah data sekuensial dan tekstual yang telah disandikan. Pada lapisan konvolusi 1D, proses ekstraksi fitur dilakukan melalui operasi perkalian matriks antara filter (*kernel*) dengan segmen data masukan secara searah (satu dimensi) menyusuri sumbu waktu atau sumbu fitur sekuensial.

Filter konvolusi bertindak sebagai detektor pola lokal otomatis. Sewaktu filter bergeser menyusuri data penanganan perkara kejaksaan yang sudah melalui proses *encoding*, filter akan mengalkulasi bobot internalnya untuk mendeteksi kombinasi hubungan tersembunyi antar-fitur—seperti keterkaitan spesifik antara jenis pasal tindak pidana tertentu dengan identitas jaksa penuntut yang ditunjuk. Hasil dari operasi konvolusi linear yang dikombinasikan dengan fungsi aktivasi non-linear (seperti *Rectified Linear Unit* / ReLU) akan menghasilkan luaran berupa peta fitur (*feature map*) yang merepresentasikan informasi abstrak tingkat tinggi dari dataset.

2.18.2 Lapisan Pengumpulan (*Pooling Layer / MaxPooling 1D*)

Setelah peta fitur berhasil dikonstruksi oleh lapisan konvolusi, data tersebut diteruskan menuju lapisan pengumpulan (*pooling layer*). Fungsi utama dari lapisan ini adalah untuk melakukan reduksi dimensi spasial (*downsampling*) dari peta fitur secara progresif tanpa kehilangan karakteristik informasi yang paling esensial. Dengan mereduksi dimensi data, lapisan *pooling* mampu menghemat konsumsi daya komputasi peladen, mempercepat durasi pelatihan, sekaligus meminimalkan risiko terjadinya *overfitting*.

Metode *pooling* yang paling umum digunakan adalah *MaxPooling 1D*. Teknik ini bekerja dengan menempatkan sebuah jendela geser berukuran tertentu di atas peta fitur, kemudian hanya mengambil nilai numerik dengan magnitudo terbesar (maksimum) di dalam cakupan jendela tersebut untuk diteruskan ke tahap berikutnya. Pengambilan nilai maksimum ini didasarkan pada asumsi teoretis bahwa nilai numerik tertinggi merepresentasikan keberadaan fitur lokal yang paling dominan dan paling relevan dalam menentukan keputusan klasifikasi.

2.18.3 Lapisan Perataan (*Flatten Layer*) dan *Fully Connected Layer*

Keluaran dari lapisan *pooling* masih berbentuk matriks multidimensi yang terkompresi. Agar data tersebut dapat diolah oleh sub-sistem pengambil keputusan klasifikasi akhir, data wajib dilewatkan melalui lapisan perataan (*flatten layer*). Lapisan *flatten* tidak melakukan modifikasi atau transformasi nilai bobot, melainkan hanya mengubah bentuk (*reshape*) susunan matriks multidimensi tersebut menjadi satu vektor linear satu dimensi tunggal yang panjang.

Vektor linear hasil perataan tersebut kemudian dihubungkan ke lapisan penuh (*fully connected layer* atau *dense layer*). Pada lapisan ini, setiap neuron masukan terhubung secara menyeluruh dengan seluruh neuron keluaran. Lapisan *dense* bertugas melakukan kombinasi linear tingkat lanjut dari seluruh fitur abstrak yang berhasil diekstraksi oleh kombinasi lapisan konvolusi dan *pooling* sebelumnya, untuk kemudian menerjemahkannya menjadi keputusan penentuan kelas target.

2.18.4 Fungsi Aktivasi *Softmax*

Pada lapisan akhir (*output layer*) dari arsitektur 1D-CNN yang ditujukan untuk kasus klasifikasi multi-kelas—seperti menentukan status akhir berkas perkara hukum yang memiliki lebih dari dua variasi kategori status—wajib digunakan fungsi aktivasi *Softmax*. Fungsi *Softmax* menerima masukan berupa vektor nilai logit mentah berukuran kontinu dari lapisan *dense* sebelumnya, kemudian menormalisasikannya menjadi bentuk distribusi probabilitas. Secara matematis, nilai probabilitas untuk setiap kelas i dihitung dengan membagi nilai eksponensial dari kelas tersebut terhadap total nilai eksponensial seluruh kelas yang tersedia.

Luaran dari fungsi aktivasi *Softmax* akan selalu menghasilkan nilai pecahan berskala antara 0 sampai dengan 1, di mana akumulasi total penjumlahan probabilitas seluruh kelas keluaran dijamin bernilai tepat sama dengan 1. Kelas kategori status berkas perkara yang memiliki nilai probabilitas tertinggi dalam kalkulasi *Softmax* inilah yang akan dipilih oleh sistem sebagai keputusan prediksi akhir model 1D-CNN.

2.19 Metode Pembagian Dataset (*Data Splitting*)

2.19.1 Data Latih (*Training Data*)

Dalam pengembangan model berbasis pembelajaran mendalam (*Deep Learning*), pembagian dataset menjadi beberapa bagian independen merupakan tahapan wajib untuk menjamin objektivitas pengujian. Komponen pertama dan terbesar dari pembagian ini adalah data latih (*training data*). Data latih merupakan sekumpulan sampel data masa lalu yang diumpankan secara langsung ke dalam arsitektur LSTM dan 1D-CNN selama fase pelatihan berlangsung. Fungsi utama dari data latih adalah memberikan ruang bagi algoritma untuk mengeksplorasi, mengenali, dan mengekstrak pola-pola laten, tren temporal, serta hubungan korelatif antar-fitur di dalam dataset penanganan perkara.

Selama proses pelatihan, jaringan saraf tiruan akan mengalkulasi data masukan pada data latih, membandingkan hasil perkiraannya dengan target asli, dan menghitung nilai galat melalui fungsi kerugian (*loss function*). Nilai galat tersebut kemudian digunakan oleh algoritma optimasi (*optimizer*) untuk memperbarui nilai bobot (*weights*) dan bias di setiap lapisan neuron secara iteratif melalui mekanisme *backpropagation*. Oleh karena itu, porsi data latih umumnya mendominasi keseluruhan dataset (misalnya 80%) agar model memiliki basis pengetahuan yang kaya dan komprehensif sebelum diuji pada skenario riil.

2.19.2 Data Uji (*Testing Data*)

Komponen kedua dari pembagian dataset adalah data uji (*testing data*). Berbeda dengan data latih, data uji bersifat sepenuhnya independen dan harus

disembunyikan secara rapat dari model selama proses pelatihan berlangsung. Model sama sekali tidak diizinkan untuk melihat atau mengakses informasi di dalam data uji hingga seluruh proses penyesuaian bobot internal selesai dilakukan. Tujuan utama dari penyediaan data uji adalah untuk mengukur kemampuan generalisasi (*generalization capability*) model kecerdasan buatan terhadap data baru yang belum pernah ditemui sebelumnya.

Pengujian menggunakan data uji bertindak sebagai simulasi riil di lapangan untuk membuktikan apakah model yang dibangun benar-benar cerdas dalam mendeteksi pola umum, atau justru terjebak dalam kondisi *overfitting* (kondisi di mana model menghafal data latih dengan sangat baik tetapi gagal total saat dihadapkan pada data baru). Dalam penerapannya, peneliti harus memastikan tidak terjadi kebocoran data (*data leakage*), di mana informasi dari data uji tidak sengaja merembes ke data latih selama fase pra-pemrosesan, seperti saat kalkulasi nilai minimum dan maksimum pada normalisasi skala data.

2.20 Metrik Evaluasi Kinerja Regresi (RMSE dan MAPE)

2.20.1 *Root Mean Squared Error* (RMSE)

Untuk mengukur tingkat keandalan dan akurasi dari model *time series forecasting* seperti LSTM dalam memprediksi kuantitas perkara bulanan di Kejaksaan, diperlukan metrik evaluasi statistik kuantitatif. Metrik pertama yang umum digunakan adalah *Root Mean Squared Error* (RMSE). Secara teoretis, RMSE mengukur akar kuadrat dari rata-rata selisih kuadrat antara nilai yang diprediksi oleh model dengan nilai aktual yang tercatat di lapangan.

Persamaan matematis untuk mengalkulasi nilai RMSE didefinisikan sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Di mana penjelasan dari komponen simbol pada rumus di atas adalah:

n : Menunjukkan jumlah total sampel data yang dievaluasi.

y_i : Menunjukkan nilai aktual atau volume perkara riil pada baris data ke-i.

$\hat{y}_i$: Menunjukkan nilai hasil prediksi atau estimasi kuantitas perkara dari model pada baris data ke-i.

Karakteristik utama dari metrik RMSE terletak pada operasi pengkuadratan nilai selisih error sebelum dirata-ratakan. Hal ini menyebabkan RMSE sangat sensitif terhadap nilai galat yang besar atau pencilan (*outliers*). Jika terdapat satu saja prediksi model yang meleset terlalu jomplang dari nilai asli, maka nilai RMSE akan meningkat secara signifikan. Oleh karena itu, semakin kecil nilai RMSE yang dihasilkan (mendekati angka 0), maka dapat disimpulkan bahwa performa model prediksi sekuensial tersebut memiliki tingkat presisi yang semakin tinggi.

2.20.2 *Mean Absolute Percentage Error* (MAPE)

Meskipun RMSE memberikan gambaran matematis yang akurat mengenai besarnya nilai galat, metrik tersebut memiliki kelemahan karena satuannya bergantung pada skala data asli, sehingga sulit dinterpretasikan oleh pihak pengambil kebijakan awam di Kejaksaan. Untuk melengkapinya, digunakan metrik *Mean Absolute Percentage Error* (MAPE). MAPE

mengukur rata-rata persentase kesalahan absolut antara nilai prediksi terhadap nilai aktual, yang dihitung melalui formula berikut:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_1-\hat{y}_i}{y_i}\right| \times 100\%$$

Di mana komponen simbol di dalam persamaan tersebut mengacu pada definisi yang sama dengan metrik sebelumnya, dengan tambahan penggunaan nilai absolut |...| untuk mengeliminasi dampak tanda negatif pada selisih pengurangan. Luaran dari kalkulasi MAPE berbentuk nilai persentase secara langsung. Menurut standar kriteria evaluasi Lewis, model prediksi dikategorikan memiliki performa yang sangat baik (*highly accurate forecast*) apabila nilai MAPE berada di bawah 10%, dikategorikan baik jika bernilai antara 10% hingga 20%, dan dikategorikan layak apabila berada di rentang 20% hingga 50%.

2.21 Metrik Evaluasi Klasifikasi (*Confusion Matrix*)

2.21.1 Struktur Tabel *Confusion Matrix*

Berbeda dengan kasus regresi yang mengukur jarak numerik, evaluasi performa pada kasus klasifikasi multi-kelas—seperti model 1D-CNN untuk menentukan status akhir berkas perkara—diukur menggunakan instrumen tabel kontingensi yang disebut *Confusion Matrix* (Matriks Kekacauan). *Confusion Matrix* menyajikan perbandingan silang secara mendetail antara label kelas aktual yang sebenarnya terjadi di lapangan dengan label kelas prediksi yang dihasilkan oleh model. Melalui matriks ini, peneliti dapat melihat tidak hanya tingkat keberhasilan model, tetapi juga jenis kesalahan spesifik yang dilakukan model saat mengelompokkan data.

Pada struktur dasarnya, matriks ini dibangun oleh empat parameter fundamental yang merepresentasikan kombinasi keputusan, yaitu:

1. *True Positive* (TP): Kondisi di mana model berhasil memprediksi suatu data masuk ke dalam kelas target tertentu, dan secara aktual data tersebut memang termasuk dalam kelas target tersebut.
2. *True Negative* (TN): Kondisi di mana model memprediksi bahwa data tidak termasuk dalam kelas target tertentu, dan secara aktual data tersebut memang bukan bagian dari kelas target itu.
3. *False Positive* (FP): Terjadi ketika model salah memprediksi suatu data termasuk dalam kelas target, padahal secara aktual data tersebut milik kelas lain (Kesalahan Tipe I).
4. *False Negative* (FN): Terjadi ketika model salah memprediksi bahwa data tidak termasuk dalam kelas target, padahal secara aktual data tersebut adalah bagian dari kelas target tersebut (Kesalahan Tipe II).

2.21.2 Nilai Akurasi, Presisi, *Recall*, dan *F1-Score*

Dari keempat parameter dasar di dalam tabel *Confusion Matrix* tersebut, dapat diturunkan beberapa nilai metrik evaluasi krusial untuk mengukur efektivitas model klasifikasi secara komprehensif:

Akurasi (*Accuracy*): Rasio total prediksi benar (positif maupun negatif) terhadap keseluruhan jumlah sampel data. Metrik ini memberikan gambaran umum performa global model, dengan rumus:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

Presisi (*Precision*): Mengukur tingkat ketepatan antara data yang diprediksi positif dengan data yang secara aktual memang bernilai positif. Presisi sangat penting untuk meminimalkan kesalahan *False Positive*, dengan rumus:

$$Presisi = \frac{TP}{TP + FP}$$

***Recall* (Sensitivitas):** Mengukur kemampuan model dalam menjaring kembali seluruh informasi dari kelas positif yang ada pada dataset aktual. *Recall* fokus meminimalkan kesalahan *False Negative*, dengan rumus:

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

***F1-Score*:** Nilai rata-rata harmonis (*harmonic mean*) yang mengombinasikan metrik Presisi dan *Recall* ke dalam satu nilai tunggal. Metrik ini menjadi indikator utama yang sangat objektif apabila dataset yang digunakan memiliki kondisi kelas yang tidak seimbang (*imbalanced dataset*), dengan rumus:

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}$$

Di mana TP (*True Positive*), TN (*True Negative*), FP (*False Positive*), dan FN (*False Negative*) dihitung berdasarkan kecocokan prediksi label status perkara oleh model 1D-CNN terhadap label aktual yang tercatat pada dataset pengujian.

2.22 Ekosistem *Library* Pemrograman Python

2.22.1 *Library* Pandas dan NumPy

Python telah menjadi standar de facto dalam pengembangan kecerdasan buatan karena didukung oleh ekosistem pustaka (*library*) sumber terbuka yang sangat masif dan matang. Di dalam tahapan pra-pemrosesan data penanganan perkara kejaksaan, pustaka Pandas memegang peranan sebagai mesin pengelola data utama. Pandas menyediakan struktur data performa tinggi berupa *Dataframe* yang memungkinkan peneliti melakukan manipulasi data tabular, pemfilteran baris, eliminasi data duplikat, pengisian *missing value*, hingga transformasi penggabungan string secara efisien menggunakan baris kode yang minimalis.

Sementara itu, pustaka Numpy (*Numerical Python*) bertindak sebagai fondasi komputasi numerik di tingkat yang lebih rendah (*low-level mathematical engine*). NumPy memperkenalkan struktur data *N-dimensional array* (ndarray) yang dioptimasikan untuk memproses operasi aljabar linier, transformasi matriks, dan kalkulasi vektor matematika berkecepatan tinggi. Seluruh manipulasi data tabular yang dilakukan pada Pandas pada dasarnya berjalan di atas array NumPy, yang nantinya akan dikonversi menjadi format tensor sebelum disuapkan ke dalam algoritma *Deep Learning*.

2.22.2 *Library* TensorFlow dan Keras

Untuk merancang, melatih, dan menguji arsitektur jaringan saraf tingkat tinggi, penelitian ini memanfaatkan pustaka TensorFlow yang diintegrasikan dengan antarmuka Keras. TensorFlow, yang dikembangkan oleh Google Brain Team, merupakan platform *open-source* berskala

komersial yang menyediakan ekosistem komprehensif untuk perhitungan tensor otomatis dan manajemen grafik komputasi. TensorFlow menangani seluruh proses matematis rumit di balik layar, termasuk otomatisasi diferensiasi fungsi kerugian selama fase *gradient descent*.

Keras bertindak sebagai *high-level API* yang berjalan di atas TensorFlow untuk mempermudah peneliti dalam menyusun lapisan-lapisan (*layers*) arsitektur model secara modular. Melalui Keras, pembentukan model dilakukan dengan pendekatan sekuensial (*Sequential API*), di mana peneliti cukup menumpuk lapisan konvolusi 1D, *MaxPooling*, lapisan perataan, lapisan sel LSTM, hingga lapisan *dense* penentu keluaran secara terstruktur. Hal ini mempercepat siklus eksperimen dalam pencarian hiperparameter (*hyperparameter tuning*) terbaik tanpa harus menulis kode operasi matematika dari nol.

2.22.3 *Library* Scikit-Learn dan Requests

Pustaka pendukung terakhir yang memiliki kontribusi penting dalam penelitian ini adalah Scikit-Learn dan Requests. Scikit-Learn merupakan pustaka *machine learning* paling populer di ekosistem Python yang menyediakan modul-modul siap pakai untuk mengeksekusi algoritma data sains konvensional. Dalam skripsi ini, Scikit-Learn tidak digunakan sebagai model inti, melainkan dimanfaatkan secara optimal untuk memanggil modul pra-pemrosesan data standar, seperti kelas `LabelEncoder` untuk penyandian teks, kelas `OneHotEncoder` untuk pembentukan variabel biner, serta kelas `MinMaxScaler` untuk mengeksekusi normalisasi rentang data matematika.

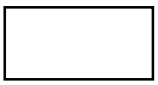
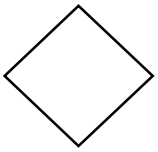
Selain itu, fungsi pembagian dataset (train_test_split) dan modul penghitung metrik seperti *Confusion Matrix* juga disuplai dari pustaka ini.

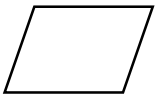
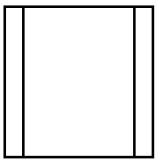
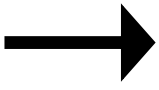
Di sisi hulu pengumpulan data, pustaka Requests bertindak sebagai mesin transmisi protokol internet. Requests memungkinkankan skrip Python yang dibangun oleh peneliti untuk bertindak sebagai *HTTP client* yang mengirimkan sinyal permintaan data (*HTTP requests*) menuju peladen target. Dengan kemampuannya menangani sesi, parameter *query*, dan header HTTP secara otomatis, pustaka Requests menjadi komponen vital dalam mengekstraksi dokumen respons mentah berformat JSON dari *endpoint* API internal website Kejaksaan Negeri Labuhanbatu secara aman dan stabil.

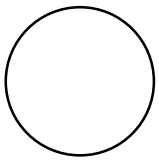
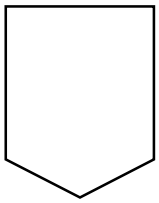
2.23 Deskripsi Bagan Alir (*Flowchart*)

Bagan alir (*flowchart*) merupakan sebuah representasi grafis yang mengilustrasikan urutan langkah-langkah logis, prosedur kerja, atau alur algoritma di dalam suatu sistem komputer maupun aktivitas penelitian. Di dalam dunia rekayasa perangkat lunak dan data sains, bagan alir berfungsi sebagai alat komunikasi visual yang memetakan bagaimana data mengalir dari tahap masukan (*input*), diproses melalui serangkaian instruksi kalkulasi matematika atau logika, hingga menghasilkan keluaran (*output*) tertentu. Penggunaan bagan alir mempermudah pengembang sistem dalam menganalisis efisiensi alur kerja kode program serta meminimalisir kesalahan logika (*logical error*) sebelum algoritma diimplementasikan ke dalam sintaksis bahasa pemrograman. Berikut ini adalah beberapa simbol yang biasa digunakan dalam *flowchart*.

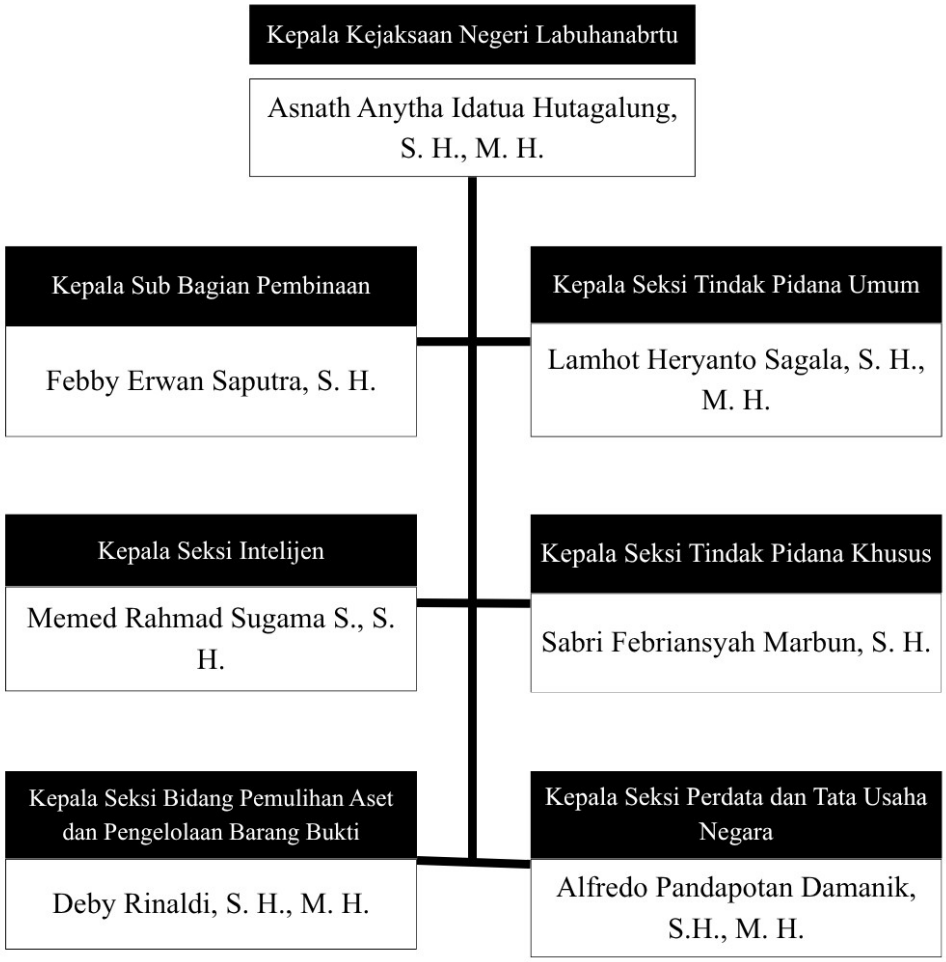
Tabel 2.1 Penjelasan terkait Simbol dalam *Flowchart*

No.	Bentuk / Simbol	Makna	Penjelasan Teoritis
1		*Terminator*	Simbol ini merepresentasikan titik awal (start) atau titik akhir (end) dari suatu runtunan aktivitas di dalam sistem. Setiap dokumen bagan alir wajib diawali dan diakhiri oleh simbol ini untuk menegaskan batasan ruang lingkup proses yang sedang dijalankan.
2		Proses (*Process*)	Simbol persegi panjang mendefinisikan adanya aktivitas pemrosesan data, operasi aritmatika, eksekusi manipulasi data, atau fungsi komputasi linear. Dalam kasus penulisan kode, simbol ini merepresentasikan tahapan pembersihan data seperti pengisian nilai kosong (*imputasi missing value*) atau kalkulasi transformasi rumusan *MinMax Scaler*.
3		Keputusan (*Decision*)	Simbol belah ketupat digunakan untuk menunjukkan kondisi logika atau percabangan yang memerlukan pengambilan keputusan. Simbol ini mengevaluasi suatu kondisi (operasi

			perbandingan) yang menghasilkan dua kemungkinan luaran berupa jawaban biner, yaitu "Ya" (*True*) atau "Tidak" (*False*). Alur data selanjutnya akan bergerak mengikuti jalur yang sesuai dengan hasil evaluasi tersebut.
4		*Data Input/Output*	Simbol jajaran genjang merepresentasikan proses penerimaan data masukan (*input*) dari perangkat luar atau penyajian data keluaran (*output*) hasil pemrosesan sistem tanpa melibatkan proses manipulasi internal.
5		Subproses (*Predefined Process*)	Simbol ini menunjukkan keberadaan sekelompok langkah prosedur atau sub-rutin program yang telah didefinisikan secara terpisah di tempat lain. Dalam arsitektur *Deep Learning*, simbol ini ideal digunakan untuk menggambarkan modul pelatihan internal model LSTM atau CNN-1D yang memiliki arsitektur berlapis-lapis di dalam sub-sistemnya.
6		Garis Alir (*Flow Line*)	Garis alir berfungsi sebagai penghubung antar-simbol yang menegaskan arah pergerakan atau urutan sekuensial dari

			jalannya proses dari satu tahapan ke tahapan berikutnya. Arah mata panah menunjukkan secara mutlak bagaimana kendali program berpindah dari satu instruksi ke instruksi selanjutnya.
7		*On-Page Connector*	Simbol lingkaran kecil digunakan untuk menyambung atau meneruskan alur bagan alir yang terputus, di mana sambungan tersebut masih berada pada halaman dokumen yang sama. Penggunaan konektor ini bertujuan untuk menghindari tumpang tindihnya garis alir yang terlalu panjang sehingga estetika dan keterbacaan diagram tetap terjaga.
8		*Off-Page Connector*	Memiliki fungsi yang serupa dengan *on-page connector*, namun simbol ini digunakan secara khusus apabila alur bagan alir terputus karena keterbatasan ruang halaman dan harus disambung atau diteruskan pada halaman dokumen yang berbeda (halaman berikutnya).

2.24 Struktur Organisasi



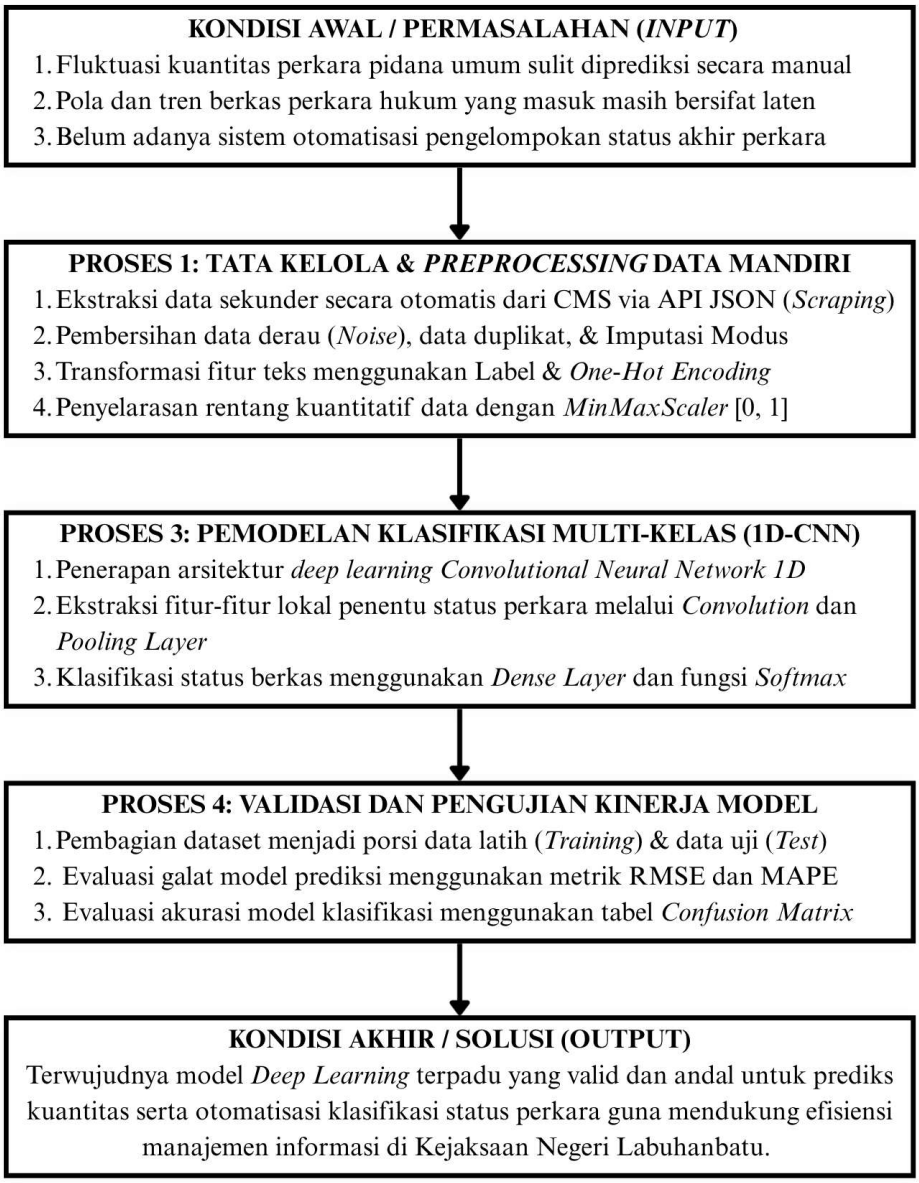
Gambar 2.1 Struktur Organisasi Pejabat Struktural Kejaksaan Negeri Labuhanbatu.(Sumber: Website Resmi (Kejaksaan Negeri Labuhanbatu, 2024))

Kejaksaan Negeri Labuhanbatu merupakan institusi penegak hukum yang dipimpin oleh Kepala Kejaksaan Negeri (Kajari) Asnath Anytha Idatua Hutagalung, S.H., M.H., yang bertanggung jawab atas penyelenggaraan tugas dan fungsi kejaksaan di wilayah hukumnya. Dalam menjalankan tugas tersebut, Kajari

didukung oleh beberapa bidang struktural, yaitu Sub Bagian Pembinaan yang berfungsi menyelenggarakan urusan administrasi umum, kepegawaian, keuangan, dan perlengkapan; Seksi Intelijen yang dipimpin oleh Memed Rahmad Sugama, S.H., yang bertugas melaksanakan kegiatan intelijen penegakan hukum, pengamanan kebijakan penegakan hukum, serta pencegahan tindak pidana; Seksi Tindak Pidana Umum (Pidum) yang berwenang melaksanakan penuntutan dan penyelesaian perkara pidana umum; Seksi Tindak Pidana Khusus (Pidsus) yang menangani perkara tindak pidana khusus, termasuk tindak pidana korupsi; Seksi Perdata dan Tata Usaha Negara (Datun) yang menjalankan fungsi penegakan hukum, bantuan hukum, serta pertimbangan hukum di bidang perdata dan tata usaha negara; serta Seksi Pemulihan Aset dan Pengelolaan Barang Bukti yang bertanggung jawab atas pengelolaan barang bukti dan pemulihan aset negara yang berasal dari tindak pidana.

2.25 Kerangka Pemikiran

Kerangka pemikiran dalam penelitian ini disusun sebagai landasan logis untuk menjembatani kesenjangan antara permasalahan riil yang dihadapi di lapangan dengan solusi berbasis kecerdasan buatan (*Machine Learning*). Alur deduktif dan induktif dari pola pikir penelitian ini digambarkan secara sistematis melalui diagram alir konseptual di bawah ini:



Gambar 2.2 Kerangka Pemikiran

Untuk menguraikan logika berpikir yang terkandung dalam bagan konseptual di atas, berikut adalah penjelasan komprehensif dari setiap tahapan yang dilalui dalam penelitian:

1. Kondisi Awal dan Identifikasi Masalah (Input).

Titik tolak penelitian ini didasarkan pada melimpahnya data historis penanganan perkara pidana umum pada Kejaksaan Negeri Labuhanbatu, namun belum dimanfaatkan secara optimal untuk mendukung pengambilan keputusan strategis. Kuantitas perkara yang masuk setiap bulannya memiliki sifat yang fluktuatif dan stokastik, sehingga sulit diprediksi secara konvensional. Di sisi lain, proses pelacakan dan pengelompokan status berkas perkara masih memerlukan pemeriksaan dokumen secara manual yang memakan waktu. Kondisi ini melahirkan kebutuhan mendesak akan sebuah sistem komputasi cerdas yang mampu mengenali pola masa lalu untuk mengestimasi masa depan.

2. Proses 1: Tata Kelola dan Pra-pemrosesan Data Mandiri.

Guna mengatasi kendala birokrasi dan keterbatasan akses database internal, proses pertama diawali dengan pengumpulan data secara mandiri melalui teknik *web scraping* tingkat lanjut pada portal publik CMS Kejaksaan menggunakan skrip Python yang diarahkan langsung ke *endpoint* API JSON. Data mentah berskala historis dari tahun 2022 hingga 2025 yang berhasil ditarik kemudian melewati fase pra-pemrosesan data yang ketat. Tahap ini krusial untuk membersihkan data dari duplikasi, mengimputasi sel yang kosong menggunakan nilai modus, mengubah fitur teks penegak hukum ke dalam matriks angka biner via *Encoding*, serta menormalisasi skala data kuantitatif menggunakan formula *MinMax Scaler* agar proses pembelajaran model menjadi stabil.

3. Proses 2: Pemodelan Prediksi Tren Sekuensial (LSTM).

Proses kedua berfokus pada pembangunan arsitektur jaringan saraf untuk menyelesaikan masalah regresi deret waktu (*time series forecasting*). Data jumlah perkara bulanan yang telah dinormalisasi ditransformasikan ke dalam format supervised learning menggunakan metode *sliding window* dengan ukuran *lookback* tertentu. Data sekuensial ini kemudian diumpankan ke dalam model *Long Short-Term Memory* (LSTM). Melalui gerbang-gerbang internalnya (*input, forget, dan output gate*), LSTM dilatih untuk mengingat informasi penting dan melupakan informasi derau dari masa lalu, sehingga mampu menghasilkan proyeksi jumlah perkara masa depan secara akurat.

4. Proses 3: Pemodelan Klasifikasi Multi-Kelas (1D-CNN).

Proses ketiga berjalan beriringan dengan proses kedua, namun ditujukan untuk menyelesaikan masalah klasifikasi multi-kelas terkait status akhir berkas perkara hukum. Karakteristik fitur-fitur penentu status perkara diekstraksi secara spasial menggunakan arsitektur *Deep Learning* berupa *1-Dimensional Convolutional Neural Network* (1D-CNN). Lapisan konvolusi dan *pooling layer* bekerja memeras informasi lokal yang paling dominan dari kombinasi atribut pasal, jaksa, dan penyidik. Vektor fitur tersebut kemudian diratakan (*flatten*) dan diteruskan ke *fully connected layer* yang menggunakan fungsi aktivasi *Softmax* untuk memetakan probabilitas kelas status berkas perkara secara otomatis.

5. Proses 4: Validasi dan Pengujian Kinerja Model.

Proses keempat merupakan fase pembuktian ilmiah untuk mengukur sejauh mana model yang telah dilatih mampu melakukan generalisasi

terhadap data baru yang belum pernah dilihat sebelumnya. Dataset dibagi secara independen menjadi data latih (*training data*) dan data uji (*testing data*). Keandalan performa model prediksi LSTM dievaluasi berdasarkan nilai minimal dari metrik *Root Mean Squared Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE). Sementara itu, ketepatan performa model klasifikasi 1D-CNN diukur secara komprehensif menggunakan matriks kekacauan (*Confusion Matrix*) untuk mendapatkan parameter Akurasi, Presisi, dan Recall.

6. Kondisi Akhir dan Solusi yang Dihasilkan (Output).

Hasil akhir yang diperoleh dari integrasi kelima proses di atas adalah sebuah model kecerdasan buatan terpadu yang memiliki tingkat validitas dan akurasi tinggi. Model ini menjadi solusi teknologi mutakhir bagi Kejaksaan Negeri Labuhanbatu dalam memprediksi lonjakan atau penurunan volume perkara pidana umum di masa mendatang, sekaligus mengotomatisasi pengelompokan status berkas penanganan perkara. Implementasi sistem ini diharapkan dapat memodernisasi tata kelola administrasi hukum menjadi lebih prediktif dan efisien.

2.26 Penelitian Terdahulu

Penelitian terdahulu merupakan kajian terhadap penelitian-penelitian sebelumnya yang memiliki keterkaitan dengan topik penelitian yang dilakukan. Kajian ini bertujuan untuk mengetahui metode, objek, serta hasil penelitian yang telah dilakukan, sehingga dapat digunakan sebagai dasar pembanding, acuan, dan referensi dalam pengembangan penelitian selanjutnya.

Tabel 2.1 Tabel Penelitian Terdahulu

Peneliti & tahun	Judul penelitian	Metode	Objek/Data Penelitian	Hasil Penelitian	Relevansi
(Cahyani et al., 2023) (Cahyani et al., 2023)	Implementasi *Long Short-Term Memory* untuk Prediksi Harga Bahan Pokok Kedepannya	*Long Short-Term Memory* (LSTM)	Data harga bahan pokok	Model LSTM menghasilkan nilai kesalahan prediksi (MAPE) kurang dari 5%, menunjukkan tingkat akurasi prediksi yang tinggi	Menguatkan penggunaan LSTM untuk prediksi jumlah perkara yang bersifat data deret waktu
(Rosyd et al., 2024) (Rosyd et al., 2024)	Penerapan Metode LSTM dalam Memprediksi Harga Saham PT Bank Central Asia	*Long Short-Term Memory* (LSTM)	Data harga saham PT BCA	Hasil penelitian menunjukkan LSTM mampu memberikan prediksi dengan tingkat error yang lebih rendah dibandingkan metode konvensional	Menjadi referensi penerapan LSTM dalam memprediksi jumlah perkara pidana umum
(Pramana, 2024)	Analisis Perbandingan Evaluasi Metode *Deep Learning* pada	*Deep Learning* (CNN)	Data citra kendaraan	Metode CNN menghasilkan tingkat akurasi klasifikasi di atas 95%	Menjadi dasar penggunaan *Deep Learning* untuk klasifikasi

	Klasifikasi Jenis Kendaraan			dan lebih unggul dibandingkan metode pembanding	status perkara
(Diponegoro et al., 2021) (Diponegoro et al., 2021)	Tinjauan Pustaka Sistematis Implementasi Metode *Deep Learning* pada Prediksi Kinerja Murid	*Deep Learning* (beragam arsitektur)	Data pendidikan	Mayoritas model deep learning menunjukkan peningkatan kinerja prediksi yang signifikan dengan akurasi rata-rata di atas 90%	Memperkuat penggunaan deep learning untuk prediksi dan klasifikasi pada berbagai domain, termasuk perkara pidana umum
(Wiranda et al. , 2019) (Wiranda & Sadikin, 2019)	Penerapan *Long Short Term Memory* pada *Data Time Series* untuk Memprediksi Penjualan Produk PT. Metiska Farma	*Long Short-Term Memory* (LSTM)	Data deret waktu	Model LSTM menunjukkan kemampuan prediksi yang baik dengan tingkat akurasi tinggi (di atas 90%) dalam menangkap pola temporal data	Menguatkan pemilihan LSTM sebagai metode prediksi perkara yang memiliki keterkaitan waktu

Berdasarkan hasil analisis penelitian terdahulu, terdapat beberapa celah penelitian yang menjadi dasar dilakukannya penelitian ini, yaitu:

1. Sebagian besar penelitian terdahulu hanya berfokus pada satu jenis permasalahan, yaitu prediksi atau klasifikasi, tanpa mengintegrasikan kedua pendekatan tersebut dalam satu sistem analisis yang komprehensif.
2. Penelitian dengan metode LSTM umumnya diterapkan pada data ekonomi dan pendidikan, sementara penerapannya pada data perkara pidana umum yang bersumber dari *Case Management System* (CMS) kejaksaan masih sangat terbatas.
3. Penelitian klasifikasi menggunakan CNN mayoritas diterapkan pada data citra dua dimensi, sedangkan pemanfaatan CNN dalam bentuk 1D-CNN untuk klasifikasi status perkara berbasis data numerik dan deret waktu belum banyak dikaji.
4. Penelitian terdahulu belum secara spesifik mengombinasikan pendekatan prediksi jumlah perkara dan klasifikasi status perkara dalam satu kerangka berbasis *machine learning* dan *deep learn*