

## BAB II

### LANDASAN TEORI

#### 2.1. Aplikasi Digital Korlantas Polri

Aplikasi digital Korlantas Polri merupakan aplikasi resmi Kepolisian Negara Republik Indonesia untuk memudahkan masyarakat yang membutuhkan jasa Korps Lalu Lintas (Korlantas). Mengutip dari website resmi aplikasi ini, Digital Korlantas Polri memiliki fitur sebagai berikut:

1. SINAR (SIM Nasional Presisi). Hanya fitur Perpanjangan SIM yang sudah bisa digunakan.
2. SIGNAL (Samsat Digital Nasional). Fitur ini belum diimplementasikan.
3. *National Traffic Management Center*. Fitur ini belum diimplementasikan.
4. *Electronic Traffic Law Enforcement*. Fitur ini belum diimplementasikan.

#### 2.2. CRISP-DM

*Cross Industry Standard Process for Data Mining* (CRISP-DM). Sebuah rangkaian proses dalam data mining yang terdiri dari 6 tahap, dari *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment* (Schröer et al., 2021).

Tabel 2. 1 Fase CRISP-DM (Schröer et al., 2021)

Fase	Deskripsi
<i>Business Understanding</i>	<i>Business understanding</i> harus dilakukan untuk memahami sumber daya apa saja yang tersedia dan apa yang dibutuhkan. Menetapkan tujuan penambangan data adalah salah satu aspek

	<p>terpenting dalam fase ini. Pertama, kita perlu menjelaskan jenis <i>data mining</i> (misalnya klasifikasi) dan kriteria keberhasilan <i>data mining</i> (misalnya akurasi). Rencana proyek harus dibuat.</p>
<p><i>Data Understanding</i></p>	<p>Mengumpulkan data dari sumber data, mengeksplorasi dan mendeskripsikan data, serta memeriksa kualitas data merupakan tugas penting dalam fase ini. Untuk lebih spesifik, panduan pengguna menjelaskan tugas mendeskripsikan data menggunakan analisis statistik dan mengidentifikasi atribut dan organisasinya.</p>
<p><i>Data Preparation</i></p>	<p>Pemilihan data sebaiknya dilakukan dengan menetapkan kriteria inklusi dan eksklusi. Kualitas data yang buruk dapat diatasi dengan pembersihan data. Bergantung pada model yang digunakan (ditentukan pada langkah pertama), atribut turunan harus dibangun. Untuk semua langkah ini, metode yang berbeda dapat diterapkan dan bergantung pada modelnya.</p>
<p><i>Modeling</i></p>	<p>Tahap pemodelan data meliputi pemilihan teknik pemodelan dan pembuatan kasus uji dan model. Semua teknik penambangan data dapat digunakan. Secara umum, pilihannya bergantung pada masalah bisnis dan datanya. Hal terpenting adalah bagaimana menjelaskan pilihannya. Untuk membangun suatu model, parameter tertentu harus ditentukan. Untuk mengevaluasi model, seseorang harus mengevaluasi model berdasarkan kriteria evaluasi dan memilih model terbaik.</p>

<i>Evaluation</i>	<p>Pada tahap evaluasi, hasil dibandingkan dengan tujuan bisnis yang telah ditentukan. Oleh karena itu, hasil harus diinterpretasikan dan tindakan selanjutnya harus ditentukan.</p> <p>Poin lainnya adalah bahwa proses ini perlu ditinjau secara keseluruhan.</p>
<i>Deployment</i>	<p>Tahapan <i>deployment</i> secara umum dijelaskan dalam petunjuk penggunaan. Ini bisa berupa laporan akhir atau komponen perangkat lunak. Panduan pengguna menjelaskan bahwa fase penerapan mencakup perencanaan, pemantauan, dan pemeliharaan <i>deployment</i>.</p>

### 2.3. Text Mining

*Text Mining* atau penambangan teks adalah teknik untuk mengekstraksi informasi dari teks dengan mengenali pola dan tren (Hossain et al., 2021). Menurut Hossain et al (2021), istilah penambangan teks atau analisis teks mengacu pada proses pengambilan informasi melalui sumber daya leksikal, penandaan atau anotasi, dan teknik seperti asosiasi, visualisasi, dan prediksi. Menurut Hassani et al. (2020), *data mining* dan *text mining* berbeda tergantung pada jenis data yang diproses. *Data mining* memproses data terstruktur dari sistem seperti *database*, *spreadsheet*, ERP, CRM, dan aplikasi akuntansi. *Text mining*, di sisi lain, memproses data tidak terstruktur yang ditemukan di dokumen, email, media sosial, dan web.

#### 2.3.1. Text Preprocessing

Menurut (Afdhal et al., 2022), *Text Preprocessing* adalah proses pembersihan teks sebelum teks tersebut diolah. Kowsari et al., (2019) mengatakan

bahwa *noise* dan juga fitur yang tidak dibutuhkan dapat mempengaruhi performa dari *Text Mining*. Ada beberapa teknik dalam melakukan *Text Preprocessing*, beberapa contohnya adalah sebagai berikut:

- a. *Tokenization*: proses ini memecah kalimat menjadi kata, beberapa kalimat kecil, atau elemen penting lainnya yang disebut token.
- b. *Stop Words*: proses ini menghapus kata-kata yang tidak signifikan atau penting dalam sebuah kalimat.
- c. Kapitalisasi: proses ini mengubah kalimat menjadi bentuk yang seragam, umumnya huruf kecil.
- d. Bahasa gaul dan singkatan: proses ini mengubah kata gaul dan singkatan menjadi bentuk dasarnya
- e. *Noise removal*: proses ini menghapus karakter yang tidak dibutuhkan seperti tanda baca dan karakter spesial.
- f. Lematisasi: proses mengganti atau menghapus kata imbuhan untuk menjadi kata dasar.

### 2.3.2. *Term Frequency-Inverse Document Frequency (TF-IDF)*

*Term frequency* (TF) adalah Frekuensi istilah dalam dokumen, dimana istilah-istilah tersebut dapat berupa kata atau frasa (Afdhal et al., 2022). Sedangkan *inverse document frequency* (IDF) memberikan bobot yang lebih tinggi pada kata-kata dengan frekuensi tinggi atau rendah dalam dokumen (Kowsari et al., 2019). Kombinasi dari proses ini disebut dengan TF-IDF. Berikut adalah representasi TF-IDF dalam persamaan matematika:

$$W(d, t) = TF(d, t) \times \log\left(\frac{N}{df(t)}\right)$$

N = Jumlah dokumen

$df(t)$  = jumlah dokumen yang mengandung kata  $t$

TF = kemunculan kata dalam dokumen

#### **2.4. Natural Language Processing (NLP)**

Menurut Sangeetha & Kumaran (2022), *Natural Language Processing* (NLP) menggabungkan berbagai disiplin ilmu seperti *Artificial Intelligence* (AI), linguistik, dan ilmu komputer. Tujuan dari NLP adalah untuk komputer dapat mengartikan atau mengerti bahasa natural seperti layaknya manusia untuk melakukan tugas yang mirip dengan yang dilakukan oleh manusia seperti translasi bahasa dan menjawab pertanyaan.

##### **2.4.1. Analisis Sentimen**

Menurut Cristescu et al(2022), analisis sentimen adalah analisis opini dan emosi manusia yang diungkapkan dalam teks tertulis. Analisis sentimen dapat digunakan dalam berbagai bidang, antara lain pemasaran dan penjualan korporat, perawatan kesehatan, dan analisis pasar keuangan. Ada beberapa topik yang dibahas dalam analisis sentimen, namun salah satu topik yang paling banyak diteliti adalah klasifikasi sentimen. Topik ini berfokus pada kegiatan pengelompokan sentimen berdasarkan teks opini untuk diskusi tentang topik yang diminati.

#### **2.5. Recurrent Neural Network (RNN)**

*Recurrent Neural Network* (RNN) adalah *Artificial Neural Network* yang didesain untuk mengenali data berurutan (Wahyudi & Sibaroni, 2022), seperti *time series* dan teks (Manchikanti & Madhurika, 2020). RNN berbeda dengan jaringan syaraf *feedforward* standar karena memiliki koneksi siklus terarah, yang memungkinkannya untuk mengingat input sebelumnya dan menampilkan perilaku

dinamis temporal. Hal ini memungkinkan RNN untuk memproses input secara berurutan sambil mempertahankan keadaan internal atau memori yang menyimpan informasi tentang input sebelumnya, sehingga memungkinkan mereka untuk menangani data berurutan dengan panjang yang berbeda, seperti deret waktu, teks, dan ucapan. RNN berguna untuk pekerjaan yang melibatkan informasi berurutan di mana urutan dan konteks data sangat penting untuk analisis atau prediksi karena kemampuannya untuk menangkap ketergantungan dan pola dalam urutan.

Hochreiter & Schmidhuber (1997) menjelaskan mengenai permasalahan yang dihadapi oleh RNN. Mereka menyoroti dua masalah besar dengan RNN standar yaitu masalah gradien yang meledak atau menjadi terlalu tinggi menyebabkan ketidakstabilan dalam pelatihan, dan masalah gradien yang menghilang menyebabkan gradien menurun secara eksponensial, sehingga mengganggu kemampuan jaringan untuk belajar dari data historis yang jauh. Permasalahan tersebutlah yang mencetuskan pengembangan LSTM.

#### 2.5.1. LSTM

*Long Short-Term Memory* (LSTM) adalah salah satu variasi dari RNN yang dicetuskan oleh Hochreiter pada tahun 1997 (Hochreiter & Schmidhuber, 1997). LSTM dikembangkan untuk mengatasi permasalahan yang ada pada RNN standar. Ko & Chang (2021) mengatakan bahwa LSTM dapat mengingat nilai jangka pendek dan jangka panjang, dan cukup fleksibel untuk mengizinkan jaringan saraf mempertahankan hanya informasi yang diperlukan.

Untuk mengatasi masalah yang ada pada RNN, Hochreiter & Schmidhuber (1997) mengimplementasikan LSTM dengan beberapa komponen utama, yaitu sel memori, *input gate*, *forget gate*, dan *output gate*. Sel memori

bertindak sebagai unit pusat yang menyimpan informasi dan mengontrol aliran informasi. *Input gate* mengontrol masuknya informasi baru ke dalam sel memori, *forget gate* mengontrol retensi atau penghapusan informasi dari sel, dan *output gate* mengontrol aliran informasi dari sel ke output jaringan.

Hochreiter & Schmidhuber (1997) menjelaskan kelebihan dari atribut yang ada pada LSTM dalam hal mengelola data berurutan dan informasi sementara. Atribut-atribut tersebut menunjukkan kemampuan mereka dalam tugas-tugas yang mencakup pengenalan suara, pengenalan tulisan tangan, pemodelan bahasa, prediksi *time series*, dan domain-domain yang serupa. Hochreiter & Schmidhuber (1997) juga mengatakan LSTM dapat digunakan di berbagai aplikasi dunia nyata yang bergantung pada analisis dan pemrosesan data berurutan.

## 2.6. Confusion Matrix

Menurut Afdhal et al. (2022), *confusion matrix* adalah matriks yang terdiri dari informasi tentang nilai prediksi dan hasil aktual, biasanya digunakan untuk menghitung akurasi, *recall*, presisi, dan *F1-Score*. *Confusion matrix* digambarkan dengan tabel yang menunjukkan jumlah data uji yang diklasifikasikan dengan benar dan jumlah data uji yang diklasifikasikan salah.

Tabel 2. 2 *Confusion Matrix*(Afdhal et al., 2022)

<i>Actual Label</i>	<i>Predicted Label</i>	
	<i>Positive(+)</i>	<i>Negative(-)</i>
<i>Positive(+)</i>	<i>True Positive (TP)</i>	<i>False Negative(FN)</i>
<i>Negative(-)</i>	<i>False Positive (FP)</i>	<i>True Negative(TN)</i>

Berikut adalah keterangan dari tabel *confusion matrix*:

1. *True Positives*(TP) adalah jumlah data positif yang diklasifikasikan sebagai nilai positif
2. *False Positives*(FP) adalah jumlah data negatif yang diklasifikasikan sebagai nilai positif
3. *False Negatives*(FN) adalah jumlah data negatif yang diklasifikasikan sebagai nilai positif
4. *True Negatives*(TN) adalah jumlah data negatif yang diklasifikasikan sebagai nilai negatif.

Berikut ini adalah rumus persamaan *confusion matrix* untuk menghitung akurasi, presisi, *recall* dan *F1-Score* adalah sebagai berikut:

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

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

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

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

## 2.7.Django

Berdasarkan web resmi dari Django (Django Software Foundation, n.d.), Django adalah kerangka web Python tingkat tinggi yang bersifat *open-source* yang mendorong pengembangan cepat dan desain pragmatis yang bersih. Django menangani keamanan dengan serius dan membantu pengembang menghindari banyak kesalahan keamanan umum, seperti *SQL injection*, *cross-site scripting*, *cross-site request forgery* dan *clickjacking*. Sistem otentikasi penggunanya menyediakan cara aman untuk mengelola akun pengguna dan kata sandi.

### 2.7.1. Keamanan Kata Sandi pada Django

Django (2023) secara aman meng-*hash* kata sandi pengguna dan menyimpannya dalam *database*. Sistem otentikasi Django menggunakan algoritma *hashing* untuk mengkonversi kata sandi pengguna ke dalam format yang tidak dapat dengan mudah diubah, sehingga meningkatkan keamanan. Algoritma yang digunakan adalah  $\langle algorithm \rangle \$ \langle iterations \rangle \$ \langle salt \rangle \$ \langle hash \rangle$ .

- *Hashing*: Ketika pengguna membuat akun atau memperbarui kata sandi, Django tidak menyimpan kata sandi itu sendiri. Sebaliknya, Django menggunakan algoritma *hashing* satu arah (PBKDF2, yang menggunakan *hash* SHA256 secara *default*) untuk mengubah kata sandi menjadi nilai *hash*. *Hashing* secara permanen mengubah kata sandi menjadi *string* berukuran tetap yang tidak dapat dengan mudah diubah kembali ke kata sandi asli.
- *Salting*: Django menggunakan konsep *salting*, dimana setiap kata sandi diberikan *string* acak tambahan yang disebut *salt* sebelum di-*hash*. *Salting* mencegah penyerang menggunakan tabel *hash* yang telah dihitung sebelumnya (*rainbow table*) untuk memecahkan kata sandi dengan lebih efisien.
- Penyimpanan: *Hash* yang dihasilkan disimpan dalam *database* bersama dengan algoritma dan *salt* (jika digunakan). Django secara transparan mengelola rincian ini dalam model pengguna bawaan (`Django.contrib.auth.models.User`).
- Validasi: Ketika pengguna *log in*, Django mengambil *hash* kata sandi yang disimpan dari *database*, menerapkan algoritma *hashing* yang sama menggunakan kata sandi yang ditentukan saat *login*, dan membandingkan

*hash* yang dihasilkan dengan *hash* yang disimpan. Jika cocok, autentikasi berhasil.

## 2.8.HTTPS

Menurut Raharjo & Bajuadji (2017), HTTPS merupakan protokol HTTP yang diterapkan protokol SSL/TLS, sehingga pengguna HTTPS juga dapat menggunakan semua fitur keamanan dari protokol SSL/TLS. Protokol HTTPS berjalan pada port 443 secara *default* dan dapat dikonfigurasi untuk menggunakan protokol SSL atau TLS. Protokol HTTPS memberikan perlindungan data yang lebih baik dibandingkan protokol HTTP biasa karena menggunakan kriptografi, khususnya mekanisme enkripsi dan otentikasi, untuk menjamin kerahasiaan dan keaslian paket yang dikirim ke dan dari server.

Aplikasi pada penelitian ini menggunakan sertifikat SSL dari Let's Encrypt. Dari dokumentasi Let's Encrypt (2019), ada dua langkah dalam proses pembuatan sertifikat ini. Pertama, agen membuktikan kepada CA bahwa server web mengontrol domain. Kemudian, agen dapat meminta, memperbarui, dan mencabut sertifikat untuk domain tersebut. Proses ini melibatkan tantangan kriptografi untuk mengonfirmasi kontrol domain, memastikan transmisi data yang aman. Selain itu,

Let's Encrypt mengotomatiskan pembaruan sertifikat, menyederhanakan proses bagi pemilik situs web dan mempromosikan keamanan yang berkelanjutan. Dengan menggunakan protokol terbuka seperti ACME (*Automated Certificate Management Environment*), Let's Encrypt mendorong transparansi, aksesibilitas, dan manajemen sertifikat gratis, yang pada akhirnya berkontribusi pada infrastruktur internet yang lebih tangguh dan aman.

## 2.9.UML

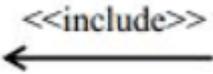
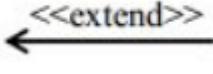
Menurut Kurniawan et al (2021), UML atau *Unified Modeling Language* adalah standar bahasa yang digunakan dalam dokumentasi, spesifikasi, dan konstruksi dalam pengembangan perangkat lunak, dengan menggunakan bahasa pemrograman berorientasi objek.

### 2.9.1. *Use Case Diagram*

*Use case diagram* adalah diagram atau representasi grafis dari beberapa aktor dan interaksi antar aktor yang mewakili suatu sistem dan menjelaskan aktor mana yang dapat melakukan prosedur dalam sistem dan proses mana yang terkait dengan sistem. *Use case* banyak digunakan oleh pengembang aplikasi dalam tahap pemodelan karena secara umum dapat dianggap sebagai hubungan antara aktor dan sistem (Kurniawan et al., 2021).

Tabel 2. 3 *Use case diagram* (Hendini, 2016)

Simbol	Nama	Keterangan
	Aktor	Aktor adalah abstraksi dari orang atau sistem lain yang mengaktifkan fungsionalitas sistem.
	<i>Use case</i>	<i>Use case</i> menggambarkan fungsionalitas yang disediakan oleh sistem dalam bentuk unit yang bertukar pesan antara unit dan aktor, yang dinyatakan dengan kata kerja.
	<i>Association</i>	Asosiasi antara aktor dan <i>use case</i> menggunakan panah terbuka untuk

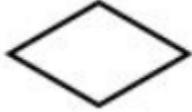
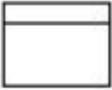
		menunjukkan aktor berinteraksi dengan sistem
	Relasi <i>include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Relasi <i>extend</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan tambahan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

### 2.9.2. *Activity Diagram*

*Activity diagram* diagram yang menggambarkan proses-proses yang terjadi dalam suatu sistem.. *activity diagram* merupakan rancangan alur aktivitas atau alur kerja pada sistem yang akan diimplementasikan. Diagram aktivitas juga digunakan untuk mengidentifikasi atau mengelompokkan aliran tampilan sistem. Diagram aktivitas memiliki komponen-komponen dengan bentuk tertentu yang dihubungkan dengan tanda panah. Tanda panah menunjuk pada urutan kegiatan yang berlangsung dari awal sampai akhir.

Tabel 2. 4 *Activity Diagram* (Hendini, 2016)

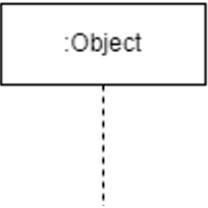
Simbol	Nama	Keterangan
	Status awal	Simbol yang mengawali diagram aktivitas dimulai
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali kata kerja

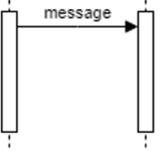
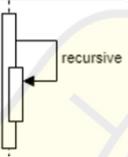
	Percabangan/ <i>decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu
	Penggabungan/ <i>join</i>	Penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
	Status akhir	Simbol yang menandakan bahwa rangkaian aktivitas telah berakhir
	<i>Swimlane</i>	Pembagian diagram aktivitas untuk menunjukkan siapa yang melakukan apa

### 2.9.3. *Sequence Diagram*

Menurut Hendini (2016), *sequence diagram* menggambarkan perilaku objek dalam *use case* dengan menggambarkan masa hidup objek dan pesan yang dikirim dan diterima antar objek.

Tabel 2. 5 *Sequence Diagram* (Hendini, 2016)

Simbol	Nama	Keterangan
	Aktor	Komponen ini menggambarkan seorang pengguna ( <i>user</i> ) yang berada di luar sistem dan sedang berinteraksi dengan sistem.
	Objek	objek berfungsi untuk mendokumentasikan perilaku sebuah objek pada sebuah sistem.

	<i>Lifeline</i>	Menggambarkan daur hidup objek
	<i>Activation box</i>	sebuah eksekusi operasi dari objek panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi
	<i>message</i>	Komponen ini untuk menggambarkan komunikasi antar objek.
	<i>Recursive</i>	Menggambarkan sebuah pengiriman pesan atau fungsi untuk dirinya sendiri





### **BAB III**

## **TEKNOLOGI INFORMASI**

# **UNIVERSITAS DARMA PERSADA**