**RANCANG BANGUN SISTEM ANALISIS SENTIMEN KINERJA KEJAKSAAN AGUNG BERBASIS DATA SCRAPING TIKTOK MENGGUNAKAN NATURAL LANGUAGE PROCESSING**

**SKRIPSI**

Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan

Pendidikan Jenjang Sarjana Terapan

Pada Politeknik Negeri Lhokseumawe



**Oleh:**

**FURQAN**

|  |  |  |
| --- | --- | --- |
| **NIM** | **:** | **2021573010051** |
| **Program Studi** | **:** | **Teknik Informatika** |
| **Jurusan** | **:** | **Teknologi Informasi dan Komputer** |

**JURUSAN TEKNOLOGI INFORMASI & KOMPUTER**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI LHOKSEUMAWE**

**PENGESAHAN PEMBIMBING**

**Skripsi yang berjudul** *Rancang Bangun Sistem Analisis Sentimen Kinerja Kejaksaan Agung Berbasis Data Scraping Tiktok Menggunakan Natural Language Processing*, disusun oleh Furqan, NIM. 2021573010051, Program Studi Teknik Informatika, Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe, telah memenuhi syarat untuk dipertanggungjawabkan dihadapan dewan penguji.

**Nama Mahasiswa** : Furqan

**NIM** : 2021573010051

**Program Studi** : Teknik Informatika

|  |  |
| --- | --- |
|  | Buketrata, 24 Juni 2025 |
| Pembimbing I, | Pembimbing II, |
|  |  |
| **Huzaeni., SST., M. IT** | **Muhammad Reza Zulman, S.S.T., M.Sc** |
| NIP. 19700601 199501 1 001 | NIP. 19920501 202203 1 005 |

Mengetahui,

|  |  |
| --- | --- |
| Ketua Jurusan | Ketua Program Studi |
| Teknologi Informasi dan Komputer, | Teknik Informatika, |
|  |  |
| **Salahuddin., ST., M.C.S** | **M. Khadafi, S.T.,M.T** |
| NIP. 19740424 200212 1 001 | NIP. 19920501 202203 1 005 |

**PENGESAHAN PENGUJI**

**Skripsi yang berjudul** *Rancang Bangun Sistem Analisis Sentimen Kinerja Kejaksaan Agung Berbasis Data Scraping Tiktok Menggunakan Natural Language Processing*, disusun oleh Furqan, NIM. 2021573010051, Program Studi Teknik Informatika, Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe, telah memenuhi syarat untuk dipertanggungjawabkan dihadapan dewan penguji.

Komisi Sidang,

|  |  |
| --- | --- |
| Asasasdasdasd  NIP. asdasdasda | …………………………………….  (Ketua Sidang) |

|  |  |
| --- | --- |
| Asasasdasdasd  NIP. asdasdasda | …………………………………….  (Sekretaris Sidang) |

|  |  |
| --- | --- |
| Asasasdasdasd  NIP. asdasdasda | …………………………………….  (Penguji I) |

|  |  |
| --- | --- |
| Asasasdasdasd  NIP. asdasdasda | …………………………………….  (Penguji II) |

|  |  |
| --- | --- |
| Asasasdasdasd  NIP. asdasdasda | …………………………………….  (Penguji III) |

Mengetahui,

|  |  |
| --- | --- |
| Ketua Jurusan | Ketua Program Studi |
| Teknologi Informasi dan Komputer, | Teknik Informatika, |
|  |  |
| **Salahuddin., ST., M.C.S** | **M. Khadafi, S.T.,M.T** |
| NIP. 19740424 200212 1 001 | NIP. 19920501 202203 1 005 |

**HALAMAN PERNYATAAN**

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk mendapatkan gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis orang lain, kecuali yang secara tertulis diacu dalam naskah dan disebut dalam daftar pustaka.

|  |
| --- |
| Lhokseumawe, …. Juni 2025 |
|  |
| **Furqan** |
| NIM. 2021573010051 |

**KATA PENGANTAR**

**ABSTRAK**

**ABSTRACT**

**DAFTAR ISI**

[PENGESAHAN PEMBIMBING ii](#_Toc202646164)

[PENGESAHAN PENGUJI iii](#_Toc202646165)

[HALAMAN PERNYATAAN iv](#_Toc202646166)

[KATA PENGANTAR v](#_Toc202646167)

[ABSTRAK vi](#_Toc202646168)

[ABSTRACT vii](#_Toc202646169)

[DAFTAR ISI viii](#_Toc202646170)

[DAFTAR GAMBAR xi](#_Toc202646171)

[DAFTAR TABEL xii](#_Toc202646172)

[RINGKASAN xiii](#_Toc202646173)

[BAB I 1](#_Toc202646174)

[PENDAHULUAN 1](#_Toc202646175)

[1.1 Latar Belakang Masalah 1](#_Toc202646176)

[1.2 Rumusan Masalah 4](#_Toc202646177)

[1.3 Tujuan Penelitian 4](#_Toc202646178)

[1.4 Batasan Masalah 5](#_Toc202646179)

[1.5 Manfaat Penelitian 6](#_Toc202646180)

[BAB II 7](#_Toc202646181)

[TINJAUAN PUSTAKA 7](#_Toc202646182)

[2.1 State of the Art 7](#_Toc202646183)

[2.2 Kerangka/Tinjauan Teoritis 10](#_Toc202646184)

[2.2.1 *Scraping* Data Komentar pada Postingan Media Sosial 10](#_Toc202646185)

[2.2.2 Penggunaan *Natural Language Processing* (*NLP*) 11](#_Toc202646186)

[2.2.3 Pendekatan *Lexicon-Based*: VADER 13](#_Toc202646187)

[2.2.4 Algoritma *Machine Learning: Random Forest* 16](#_Toc202646188)

[2.2.5 Evaluasi Model Klasifikasi 18](#_Toc202646189)

[2.2.6 Arsitektur Sistem: Laravel Framework dan Flask API 20](#_Toc202646190)

[BAB III 22](#_Toc202646191)

[METODOLOGI PENELITIAN 22](#_Toc202646192)

[3.1 Data dan Pengumpulan Data 22](#_Toc202646193)

[3.1.1 Sumber Data 22](#_Toc202646194)

[3.1.2 Teknik Pengumpulan Data 23](#_Toc202646195)

[3.2 Analisis Kebutuhan Sistem 26](#_Toc202646196)

[3.2.1 Analisis Kebutuhan Fungsional 27](#_Toc202646197)

[3.2.2 Analisis Kebutuhan Non-Fungsional 28](#_Toc202646198)

[3.3 Rancangan Sistem 30](#_Toc202646199)

[3.3.1 Metodologi perancangan Sistem 31](#_Toc202646200)

[3.3.2 Flowchart Sistem 33](#_Toc202646201)

[3.3.3 Rancangan Business Rules 38](#_Toc202646202)

[3.3.4 Use Case Diagram 40](#_Toc202646203)

[3.3.5 Arsitektur Sistem (Laravel & Flask API) 48](#_Toc202646204)

[3.3.6 Activity Diagram 50](#_Toc202646205)

[3.3.7 Sequence Diagram 65](#_Toc202646206)

[3.3.8 Class Diagram 78](#_Toc202646207)

[3.3.9 *Entity Relationship Diagram* (*ERD*) 81](#_Toc202646208)

[3.3.10 Perancangan Antar Muka Pengguna 84](#_Toc202646209)

[3.4 Rancangan Metode Penelitian 87](#_Toc202646210)

[3.4.1 Alur penelitian 88](#_Toc202646211)

[3.4.2 Tahapan *Preprocessing* data Komentar 90](#_Toc202646212)

[3.4.3 Pelabelan Komentar dengan VADER 94](#_Toc202646213)

[3.4.4 Prediksi Sentimen Menggunakan model *Random Forest* 98](#_Toc202646214)

[3.5 Perancangan Teknik Pengujian 103](#_Toc202646215)

[3.5.1 Pelatihan model *Random forest* 103](#_Toc202646216)

[3.5.2 *K-Fold Cross validation* 107](#_Toc202646217)

[3.5.3 *Confusion Matrix* 109](#_Toc202646218)

[3.5.4 Perancangan Teknik pengujian *Black Box* 112](#_Toc202646219)

[3.5.5 Perancangan Teknik Pengujian White Box 114](#_Toc202646220)

[BAB IV 116](#_Toc202646221)

[HASIL DAN PEMBAHASAN 116](#_Toc202646222)

[4.1 Hasil Tampilan Website 116](#_Toc202646223)

[4.2 Hasil *Preprocessing* Data Komentar TikTok 120](#_Toc202646224)

[4.3 Hasil *Labeling* Data 122](#_Toc202646225)

[4.4 Hasil Pelatihan Model *Random Forest* 127](#_Toc202646226)

[4.4.1 *K-Fold Cross Validation* 128](#_Toc202646227)

[4.4.2 *Confusion Matrix* 132](#_Toc202646228)

[4.5 Prediksi Sentimen Menggunakan *Machine Learning* (*Random Forest*) 136](#_Toc202646229)

[4.6 Hasil Pengujian *BlackBox* 139](#_Toc202646230)

[4.7 Pengujian *White Box* 145](#_Toc202646231)

[BAB V 146](#_Toc202646232)

[KESIMPULAN DAN SARAN 146](#_Toc202646233)

[DAFTAR PUSTAKA 147](#_Toc202646234)

**DAFTAR GAMBAR**

[Gambar 3. 1 Tahapan pengumpulan data 25](#_Toc202648392)

[Gambar 3. 2 Contoh hasil *scraping* 26](#_Toc202648393)

[Gambar 3. 3 Metodelogi perancangan sistem (*waterfall methodology*) 32](#_Toc202648394)

[Gambar 3. 4 Flowchart sistem (login) 34](#_Toc202648395)

[Gambar 3. 5 Flowchart sistem (fitur admin) 36](#_Toc202648396)

[Gambar 3. 6 Flowchart sistem (fitur user) 38](#_Toc202648397)

[Gambar 3. 7 *Use case diagram* 41](#_Toc202648398)

[Gambar 3. 8 Arsitektur sistem 49](#_Toc202648399)

[Gambar 3. 9 *Activity diagram* admin (registrasi) 51](#_Toc202648400)

[Gambar 3. 10 *Activity diagram* admin (login) 52](#_Toc202648401)

[Gambar 3. 11 *Activity diagram* admin (tombol *scraping*) 54](#_Toc202648402)

[Gambar 3. 12 *Activity diagram* admin (tombol analisis ML) 56](#_Toc202648403)

[Gambar 3. 13 *Activity diagram* admin (tombol unduh hasil analisis sentimen) 58](#_Toc202648404)

[Gambar 3. 14 *Activity diagram* user (registrasi) 60](#_Toc202648405)

[Gambar 3. 15 *Activity diagram* user (login) 61](#_Toc202648406)

[Gambar 3. 16 *Activity diagram* user (tampilan visualisasi hasil analisis sentimen) 62](#_Toc202648407)

[Gambar 3. 17 *Activity diagram* user (unduh data) 63](#_Toc202648408)

[Gambar 3. 18 *Sequence diagram* registrasi admin 66](#_Toc202648409)

[Gambar 3. 19 *Sequence diagram* registrasi user 67](#_Toc202648410)

[Gambar 3. 20 *Sequence diagram* login admin 68](#_Toc202648411)

[Gambar 3. 21 *Sequence diagram* login user 69](#_Toc202648412)

[Gambar 3. 22 *Sequence diagram scraping* komentar Tiktok (admin) 71](#_Toc202648413)

[Gambar 3. 23 *Sequence diagram* analisis sentimen ML (admin) 73](#_Toc202648414)

[Gambar 3. 24 *Sequence diagram* unduh hasil analisis sentimen (admin) 75](#_Toc202648415)

[Gambar 3. 25 *Sequence diagram* melihat visualisasi hasil analisis sentimen (user) 77](#_Toc202648416)

[Gambar 3. 26 *Class diagram* 79](#_Toc202648417)

[Gambar 3. 27 *Entity Relationship Diagram* (ERD) 82](#_Toc202648418)

[Gambar 3. 28 *User interface* (register) 84](#_Toc202648419)

[Gambar 3. 29 *User interface* (login) 85](#_Toc202648420)

[Gambar 3. 30 *User interface* (dashboard) 85](#_Toc202648421)

[Gambar 3. 31 *User interface* (result) 86](#_Toc202648422)

[Gambar 3. 32 Alur penelitian 89](#_Toc202648423)

[Gambar 3. 33 Contoh hasil dalam database 98](#_Toc202648424)

[Gambar 3. 34 Struktur teknik coverage pada pengujian white blox 115](#_Toc202648425)

**DAFTAR TABEL**

[Tabel 3. 1 Kebutuhan perangkat lunak (software) 28](#_Toc202648362)

[Tabel 3. 2 Kebutuhan perangkat keras (hardware) 29](#_Toc202648363)

[Tabel 3. 3 Rancangan *business rules* 39](#_Toc202648364)

[Tabel 3. 4 *Use case* registrasi, login dan logout akun (Admin & User) 42](#_Toc202648365)

[Tabel 3. 5 *Use case* jalankan *scraping* komentar TikTok (Aktor Admin) 43](#_Toc202648366)

[Tabel 3. 6 *Use case* jalankan analisis sentimen ML (aktor Admin) 44](#_Toc202648367)

[Tabel 3. 7 *Use case* lihat hasil analisis sentimen (aktor user) 45](#_Toc202648368)

[Tabel 3. 8 *Use case* unduh hasil analisis sentimen (aktor admin & user) 46](#_Toc202648369)

[Tabel 3. 9 Contoh implementasi 96](#_Toc202648370)

[Tabel 3. 10 Contoh implementasi 105](#_Toc202648371)

[Tabel 3. 11 pembagian data latih dan data uji 109](#_Toc202648372)

[Tabel 3. 12 Rancangan confusion matrix 110](#_Toc202648373)

[Tabel 3. 13 Perancangan skenario pengujian black box 112](#_Toc202648374)

**RINGKASAN**

Kejaksaan Agung merupakan lembaga pemerintahan yang memainkan peran vital dalam penegakan hukum di Indonesia. Sebagai institusi yang memiliki kewajiban untuk menjaga keadilan, kinerja Kejaksaan Agung sering kali menjadi sorotan publik. Masyarakat menyuarakan pendapat mereka melalui berbagai saluran, termasuk media sosial, yang kini menjadi salah satu platform utama untuk menyampaikan opini dan kritik terhadap kinerja lembaga pemerintah. Dalam hal ini, TikTok, sebagai platform berbagi video pendek yang sangat populer, menyediakan banyak data komentar yang dapat digunakan untuk memahami persepsi publik terkait kinerja Kejaksaan Agung.

Namun, dengan jumlah data komentar yang sangat besar dan beragam, analisis manual menjadi tidak efektif. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem analisis sentimen otomatis yang dapat menganalisis komentar-komentar pada TikTok dengan menggunakan teknologi Natural Language Processing (NLP). Sistem ini akan mengklasifikasikan sentimen komentar ke dalam kategori positif, negatif, atau netral dengan menggunakan metode VADER (Valence Aware Dictionary and sEntiment Reasoner), yang berbasis leksikon.

Penelitian ini diharapkan dapat memberikan wawasan mengenai opini publik terhadap kinerja Kejaksaan Agung, yang nantinya dapat digunakan untuk mendukung transparansi dan akuntabilitas kinerja lembaga pemerintah. Dengan menggunakan metode NLP dan VADER, sistem ini akan membantu mempercepat proses analisis sentimen serta menghasilkan hasil yang lebih objektif dan akurat, yang berkontribusi pada pengambilan keputusan yang lebih baik oleh instansi terkait.

Penelitian ini memberikan manfaat bagi pengambil kebijakan di Kejaksaan Agung, masyarakat yang ingin memantau kinerja lembaga pemerintah, serta pengembangan lebih lanjut dalam pemanfaatan NLP untuk analisis data media sosial dengan karakteristik bahasa yang dinamis.

# BAB I

# PENDAHULUAN

## Latar Belakang Masalah

Kejaksaan Agung adalah lembaga pemerintahan yang sangat vital dalam sistem hukum Indonesia. Kejaksaan Agung tidak hanya memiliki peran sebagai penegak hukum, tetapi juga sebagai pelaksana putusan pengadilan, serta memiliki kewenangan dalam bidang intelijen penegakan hukum, seperti pencegahan korupsi, kolusi, dan nepotisme, sesuai dengan Undang-Undang Nomor 11 Tahun 2021 tentang Perubahan atas Undang-Undang Nomor 16 Tahun 2004 tentang Kejaksaan Republik Indonesia, pasal 1 ayat (1) dan pasal 30B.

Keberadaan Kejaksaan Agung yang berperan besar dalam menciptakan keadilan sangat diperhatikan oleh publik, terutama dalam hal transparansi dan akuntabilitas kinerjanya. Dalam sistem hukum yang demokratis, independensi Kejaksaan Agung menjadi kunci untuk memastikan keadilan dalam penegakan hukum. Kinerja Kejaksaan Agung menjadi subjek perhatian masyarakat yang diekspresikan melalui berbagai saluran komunikasi, salah satunya adalah media sosial seperti TikTok. Platform ini menjadi tempat bagi masyarakat untuk menyuarakan pandangan mereka tentang lembaga pemerintah. Untuk mengukur dan memahami opini publik terhadap kinerja Kejaksaan Agung secara lebih objektif dan efisien, dibutuhkan teknologi analisis sentimen yang dapat mengolah data komentar secara otomatis [1].

Dalam era digital, media sosial telah menjadi salah satu alat komunikasi utama yang digunakan masyarakat untuk menyampaikan opini, termasuk pandangan terhadap kinerja Kejaksaan Agung [2]. Namun, tantangan yang dihadapi dalam penilaian kinerja lembaga ini tidak hanya terbatas pada faktor internal, melainkan juga dipengaruhi oleh persepsi publik yang sering kali terbentuk melalui informasi yang beredar di media. Oleh karena itu, penting untuk memahami bagaimana interaksi antara media sosial dan persepsi publik dapat memengaruhi penilaian terhadap kinerja Kejaksaan Agung dalam menjalankan peran dan fungsinya sebagai lembaga penegak hukum di Indonesia .

media sosial seperti TikTok telah menjadi salah satu platform yang banyak digunakan oleh masyarakat untuk berbagi opini dan mengekspresikan pandangan terkait isu-isu sosial, politik, dan pemerintahan, termasuk kinerja Kejaksaan Agung, memungkinkan pengguna untuk berinteraksi secara langsung melalui fitur komentar pada postingan video-video terkait isu publik. Dengan lebih dari 113 juta pengguna di Indonesia, TikTok memiliki potensi besar untuk digunakan sebagai sumber data dalam memahami opini masyarakat [3].

Namun, analisis terhadap komentar-komentar yang ada di TikTok secara manual bukanlah pekerjaan yang mudah, terutama karena jumlah data yang sangat besar dan beragam. Komentar-komentar tersebut bisa mencakup sentimen positif, negatif, atau netral, yang mencerminkan pandangan masyarakat terhadap berbagai aspek kinerja Kejaksaan Agung. Oleh karena itu, diperlukan suatu sistem yang mampu melakukan pengumpulan data komentar secara otomatis dari TikTok dan menganalisis sentimen dari komentar tersebut secara efektif dan efisien [4].

Kondisi ideal yang diharapkan adalah tersedianya aplikasi yang dapat melakukan scraping data komentar dari TikTok dan menganalisis sentimen dari data tersebut menggunakan teknologi *Natural Language Processing* (NLP). Menurut penelitian oleh Elik Hari Muktafin, Kusrini dan Emha Taufiq Luthfi (2020) *Natural Language Processing* (NLP) merupakan salah satu bidang dari kecerdasan buatan yang berfokus pada pemrosesan bahasa alami manusia untuk memungkinkan komputer memahami, menafsirkan, dan memanipulasi teks atau data secara otomatis. Dalam konteks analisis sentimen, NLP digunakan untuk mengklasifikasikan teks ke dalam kategori sentimen tertentu, seperti positif, negatif, atau netral, berdasarkan pola bahasa yang ditemukan dalam teks [5]. Dengan teknologi ini, data teks dari komentar di TikTok dapat dianalisis untuk mengidentifikasi kecenderungan sentimen masyarakat terhadap kinerja Kejaksaan Agung. Hasil dari analisis ini diharapkan dapat memberikan informasi yang berharga bagi pengambil kebijakan di lembaga pemerintahan, sehingga dapat digunakan untuk meningkatkan transparansi dan akuntabilitas kinerja.

Dengan adanya aplikasi ini, diharapkan masyarakat dan pihak-pihak yang berkepentingan dapat lebih mudah memantau persepsi publik terhadap kinerja Kejaksaan Agung. Informasi yang diperoleh dari hasil analisis sentimen ini juga dapat digunakan sebagai acuan dalam proses evaluasi kinerja, perbaikan kebijakan, dan pengambilan keputusan yang lebih baik di masa mendatang. Sehingga, penelitian ini diharapkan dapat memberikan manfaat nyata bagi masyarakat, khususnya dalam konteks peningkatan kualitas kinerja lembaga pemerintah seperti Kejaksaan Agung.

## Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, penelitian ini akan berfokus pada beberapa permasalahan terkait analisis sentimen terhadap kinerja Kejaksaan Agung melalui data hasil scraping di media sosial TikTok. Adapun rumusan masalah yang diidentifikasi dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun sistem yang mampu melakukan scraping komentar TikTok secara otomatis berdasarkan kata kunci tertentu yang relevan dengan kinerja Kejaksaan Agung?
2. Bagaimana menerapkan proses preprocessing dan analisis sentimen menggunakan pendekatan *hybrid lexicon-based* (VADER) yang diperkuat dengan kamus manual dan hasilnya digunakan sebagai data latih untuk membangun model klasifikasi sentimen berbasis *Machine Learning* (*Random Forest*)?
3. Bagaimana menyajikan hasil akhir klasifikasi sentimen dari model Machine Learning secara informatif melalui sistem berbasis web?

## Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai:

1. Mengembangkan sistem yang mampu melakukan pengambilan data komentar TikTok secara otomatis melalui proses scraping berdasarkan kata kunci yang berkaitan dengan kinerja Kejaksaan Agung.
2. Mengimplementasikan proses preprocessing berbasis *Natural Language Processing* (NLP), serta menerapkan analisis sentimen menggunakan metode VADER yang diperkuat dengan kamus normalisasi dan kamus sentimen manual sebagai dasar pelabelan data untuk pelatihan model Machine Learning.
3. Membangun dan mengintegrasikan model klasifikasi sentimen berbasis *Random Forest* yang dilatih menggunakan data hasil VADER, dan menyajikan hasil prediksinya dalam sistem berbasis web berupa tabel dan visualisasi data sentimen.

## Batasan Masalah

Agar penelitian ini dapat berjalan secara fokus dan terarah, terdapat beberapa batasan yang ditetapkan, yaitu:

1. Penelitian hanya memfokuskan pada komentar TikTok yang relevan dengan isu kinerja Kejaksaan Agung, berdasarkan kata kunci tertentu seperti “kejaksaan agung”, “kejagung”, “kinerja kejaksaan”, dan sejenisnya.
2. Proses *scraping*, *preprocessing*, analisis sentimen dengan VADER, dan pelatihan model *Machine Learning* dilakukan secara offline menggunakan Python.
3. Sistem berbasis web dikembangkan menggunakan Laravel dan hanya digunakan untuk menampilkan hasil analisis akhir dari model Machine Learning serta memfasilitasi proses *scraping* dengan menghubungkan ke Flask API.

Pendekatan analisis sentimen yang digunakan terbatas pada klasifikasi tiga kelas: positif, netral, dan negatif. Analisis sentimen tingkat lanjut seperti emosi, ironi, atau sarcasm tidak dibahas dalam penelitian ini. Hasil analisis sentimen disajikan dalam bentuk tabel dan visualisasi sederhana seperti *pie chart* dan summary jumlah komentar per-kategori sentimen.

## Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Penelitian ini bermanfaat bagi pengembangan sistem berbasis web yang mampu menyajikan hasil analisis sentimen dari data media sosial secara otomatis, meskipun proses *scraping* dan analisis utama dilakukan secara offline.
2. Bagi Institusi Akademik penelitian ini bermanfaat Sebagai referensi bagi penelitian selanjutnya yang berkaitan dengan penggunaan teknik NLP pada analisis data media sosial, serta untuk pengembangan aplikasi berbasis web yang mendukung pengolahan data secara otomatis.
3. Penelitian ini juga bermanfaat Bagi Dunia Lembaga Pemerintahan Sebagai alat untuk monitoring dan evaluasi terhadap persepsi masyarakat terhadap kinerja suatu lembaga, yang dapat digunakan untuk mendukung evaluasi kebijakan dan perbaikan kinerja institusi lembaga di masa mendatang.

# BAB II

# TINJAUAN PUSTAKA

1. State of the Art

Pada penelitian ini, tinjauan pustaka dilakukan untuk memahami berbagai metode dan pendekatan yang telah digunakan dalam analisis sentimen pada data yang diambil dari media sosial. Hal ini penting untuk mengetahui bagaimana penelitian sebelumnya telah dilakukan, serta apa yang menjadi kelebihan dan kekurangannya. Untuk itu, perbandingan metode, hasil yang diperoleh, serta persamaan dan perbedaan dengan penelitian sebelumnya akan dipaparkan seperti berikut:

Tabel 2. *State of the art*

|  |
| --- |
| 1. Ridhwan Ardiyansyah et al. /2024 2. Judul Artikel :Sentimen Komentar YouTube Dengan Sentiment Intensity Analyzer Dari NLTK. 3. Metode Yang Digunakan : Sentiment Intensity Analyzer dari NLTK. 4. Hasil Penelitian : Evaluasi sentimen komentar YouTube dengan akurasi tinggi dan cepat, menggunakan skor intensitas emosi. 5. Persamaan : Sama-sama menggunakan analisis sentimen berbasis NLP untuk mengevaluasi opini publik. 6. Perbedaan : Fokus pada platform YouTube, sedangkan penelitian saat ini berfokus pada TikTok. |
| 1. Amanda Amalia et al./2023 2. Judul Artikel : Rancang Bangun Aplikasi Sentimen Pendapat Masyarakat Indonesia Terhadap Vaksin COVID-19 Dengan Menggunakan Algoritma LSTM Berbasis Web. 3. Metode Yang Digunakan : Long Short-Term Memory (LSTM), Word Embedding. 4. Hasil Penelitian : Akurasi 74,46% dalam analisis sentimen pada tweet tentang vaksin COVID-19. 5. Persamaan : Sama-sama menggunakan teknik NLP untuk analisis sentimen. 6. Perbedaan : Penelitian ini menggunakan LSTM untuk Twitter, sedangkan penelitian saat ini berfokus pada analisis sentimen di TikTok menggunakan metode berbasis leksikon seperti VADER. |
| 1. Maria Mega Mala Olhang/2020 2. Judul Artikel : Analisis Sentimen Pengguna Twitter Terhadap COVID-19 di Indonesia Menggunakan Metode *Naïve Bayes Classifier* (NBC). 3. Metode Yang Digunakan :*Naïve Bayes Classifier* (NBC). 4. Hasil Penelitian : Akurasi 36% dengan klasifikasi sentimen positif dan negatif berdasarkan data Twitter tentang COVID-19. 5. Persamaan : Penggunaan data dari media sosial untuk analisis sentimen. 6. Perbedaan : Platform penelitian menggunakan Twitter, sedangkan penelitian saat ini menggunakan TikTok. |
| 1. Dianati Duei Putri/2022 2. Judul Artikel : Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode *Naïve Bayes Classifier*. 3. Metode Yang Digunakan :*Naïve Bayes Classifier* (NBC). 4. Hasil Penelitian : Hasil menunjukkan 75% sentimen positif, 79% netral, dan 82% negatif dengan tingkat akurasi 80%. 5. Persamaan : Fokus pada analisis sentimen kinerja institusi publik melalui media sosial. 6. Perbedaan : Penelitian ini menggunakan Twitter sebagai platform data, sementara penelitian sekarang menggunakan TikTok. |
| 1. Tri Rivanie/2021 2. Judul Artikel : Analisis Sentimen Terhadap Kinerja Menteri Kesehatan Indonesia Selama Pandemi COVID-19. 3. Metode Yang Digunakan :*Support Vector Machine* (SVM) dan *Naïve Bayes*. 4. Hasil Penelitian : SVM memiliki akurasi lebih tinggi (72.57%) dibandingkan dengan Naïve Bayes (66.45%) dalam mengklasifikasi sentimen terkait kebijakan kesehatan. 5. Persamaan : Sama-sama menggunakan analisis sentimen untuk mengevaluasi kinerja institusi pemerintah berdasarkan data media sosial. 6. Perbedaan : Menggunakan SVM dan Naïve Bayes dengan validasi k-Fold pada Twitter, sedangkan penelitian ini menggunakan TikTok dengan NLP untuk analisis sentimen. |
| 1. Elik Hari Muktafin et al./2020 2. Judul Artikel : Analisis Sentimen pada Ulasan Pembelian Produk di Marketplace Shopee Menggunakan Pendekatan Natural Language Processing. 3. Metode Yang Digunakan : K-Nearest Neighbors (KNN) dengan pendekatan TF-IDF dan NLP. 4. Hasil Penelitian : Akurasi 76,92% dalam analisis sentimen pada ulasan produk Shopee. 5. Persamaan : Sama-sama menggunakan pendekatan *Natural Language Processing* (NLP) untuk analisis sentimen. 6. Perbedaan : Penelitian tersebut menggunakan algoritma KNN dengan pendekatan TF-IDF untuk analisis ulasan produk di Shopee. sedangkan penelitian ini menggunakan pendekatan berbasis leksikon dengan algoritma VADER untuk menganalisis sentimen komentar TikTok. Fokus penelitian tersebut adalah marketplace Shopee, sedangkan penelitian ini berfokus pada platform media sosial TikTok. |
| 1. Nawang A. Hapsari & Aries D. Indriyanti/2023 2. Judul Artikel : Analisis Sentimen pada Aplikasi Dompet Digital Menggunakan Algoritma Random Forest. 3. Metode Yang Digunakan : VADER untuk labeling, Random Forest untuk klasifikasi. 4. Hasil Penelitian : Akurasi klasifikasi sentimen dompet digital (Gopay dan Shopeepay) dengan Random Forest mencapai sekitar 85%. 5. Persamaan : Sama-sama menggunakan VADER sebagai pelabel dan Random Forest untuk klasifikasi sentimen berbasis media sosial. 6. Perbedaan : Menggunakan Twitter dan RapidMiner, sedangkan penelitian ini menggunakan TikTok dan integrasi Laravel–Flask–Python. |

1. Kerangka/Tinjauan Teoritis
2. *Scraping* Data Komentar pada Postingan Media Sosial

Pengambilan data dari media sosial melalui proses *scraping* merupakan tahap awal yang sangat penting dalam analisis sentimen. Menurut Eka Yuniar et al. (2022), *scraping* dilakukan dengan cara mengekstrak elemen-elemen informasi dari platform sosial seperti Twitter menggunakan metode otomatis yang terstruktur. Informasi yang diperoleh dapat berupa komentar, metadata pengguna, dan konten relevan lainnya. Prosedur *scraping* umumnya mencakup penentuan target URL, pemuatan konten HTML dari halaman tersebut, ekstraksi data berdasarkan elemen yang dibutuhkan, dan penyimpanan hasilnya dalam format terstruktur seperti CSV atau JSON [6]​.

*Scraping* memungkinkan peneliti mengakses data dalam jumlah besar secara *real-time* dari platform yang bersifat dinamis seperti media sosial. Data ini menjadi bahan baku utama dalam analisis berbasis teks, karena mampu merefleksikan opini publik secara langsung dan aktual. Penelitian Yuniar et al. juga menunjukkan bahwa setelah proses *scraping*, perlu dilakukan tahapan lanjutan seperti pembersihan data (*cleansing*), normalisasi, dan konversi data ke dalam bentuk yang siap dianalisis [6].

Dalam penelitian ini, proses scraping difokuskan pada komentar pengguna TikTok yang berkaitan dengan Kejaksaan Agung Republik Indonesia. TikTok dipilih karena menjadi platform yang sangat aktif dengan bentuk komunikasi yang khas: singkat, informal, dan banyak menggunakan bahasa gaul, simbol, serta singkatan. Hal ini menjadikan proses pengambilan data lebih menantang dibanding platform lain. Untuk itu, scraping dilakukan dengan bantuan Selenium dan *undetected\_chromedriver*, agar dapat mengakses komentar meskipun terdapat mekanisme anti-bot. Komentar yang berhasil dikumpulkan akan digunakan sebagai dataset utama dan diproses lebih lanjut melalui metode *Natural Language Processing* (NLP), guna mengidentifikasi sentimen yang terkandung di dalamnya secara sistematis dan terstruktur.

1. Penggunaan *Natural Language Processing* (*NLP*)

*Natural Language Processing* (NLP) merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) yang berfokus pada bagaimana sistem komputer dapat memahami, menafsirkan, dan menghasilkan bahasa manusia secara otomatis. Dalam konteks analisis sentimen, NLP berperan dalam mengolah teks mentah dari pengguna menjadi data yang terstruktur, sehingga sentimen yang terkandung di dalamnya dapat diklasifikasikan ke dalam kategori seperti positif, negatif, atau netral. Muktafin et al. menjelaskan bahwa NLP dapat digunakan untuk meningkatkan akurasi dalam analisis sentimen melalui tahapan pra-pemrosesan data. Tahapan ini mencakup proses seperti normalisasi kata, penghilangan karakter tidak relevan, tokenisasi, dan penyederhanaan teks. Langkah-langkah ini penting karena teks pada media sosial sering kali tidak mengikuti kaidah bahasa formal, sehingga diperlukan upaya transformasi untuk menyamakan makna kata yang berbeda dalam penulisan namun memiliki arti yang serupa [5].

Dalam penelitian ini, NLP diterapkan untuk menangani karakteristik unik komentar TikTok yang cenderung informal, menggunakan bahasa gaul, singkatan, serta kombinasi simbol dan emotikon. Oleh karena itu, tahapan pra-pemrosesan disesuaikan secara khusus berdasarkan karakteristik data dan metode yang digunakan. Proses ini dimulai dengan *case folding*, yakni mengubah seluruh huruf menjadi huruf kecil agar seragam. Selanjutnya dilakukan pembersihan karakter tidak relevan menggunakan ekspresi reguler (*regex*), seperti menghapus tautan URL, mention, hashtag, angka, serta tanda baca. Setelah itu, sistem menerapkan normalisasi kata dengan menggunakan kamus normalisasi untuk mengubah kata tidak baku (seperti “gk”, “ga”, atau “bgt”) menjadi kata baku dalam Bahasa Indonesia. Sebagai tambahan, sistem juga menyertakan proses penggantian kata idiomatik atau slang lokal menjadi padanan Bahasa Inggris umum, guna meningkatkan akurasi penerjemahan. Setelah semua proses pra-pemrosesan dilakukan, komentar diterjemahkan ke dalam Bahasa Inggris menggunakan API Google Translate agar dapat dianalisis oleh algoritma VADER yang berbasis leksikon Bahasa Inggris.

Tahapan-tahapan tersebut merupakan bagian dari pendekatan NLP yang disesuaikan untuk memastikan bahwa teks komentar TikTok yang tidak terstruktur dapat diolah secara efektif. Dengan menerapkan strategi ini, sistem dapat mengubah opini publik dalam bentuk komentar menjadi data kuantitatif yang dapat dianalisis secara sistematis, sehingga mendukung proses pengambilan keputusan berbasis data dalam mengevaluasi persepsi masyarakat terhadap kinerja Kejaksaan Agung​ [5].

1. Pendekatan *Lexicon-Based*: VADER

Pendekatan berbasis lexicon-based adalah metode analisis sentimen yang memanfaatkan daftar kata atau **leksikon** yang telah diberi skor sentimen sebelumnya. Metode ini tidak memerlukan proses pelatihan model. Salah satu algoritma yang banyak digunakan dalam pendekatan ini adalah **VADER (*Valence Aware Dictionary and sEntiment Reasoner*).** Algoritma VADER dirancang khusus untuk menganalisis sentimen pada teks pendek, informal, dan ekspresif, seperti komentar media sosial, ulasan produk, atau cuitan [7].

Algoritma VADER menghasilkan empat skor sentimen utama, yaitu positif, netral, negatif, dan *compound*. *Compound score* merupakan representasi dari keseluruhan intensitas sentimen suatu teks. Perhitungan nilai *compound* didapatkan dari penjumlahan skor valensi setiap kata dalam teks, kemudian dinormalisasi sehingga nilainya berada dalam rentang −1 hingga +1. Rumus yang digunakan untuk perhitungan tersebut adalah sebagai berikut:

Compount =

Skor compound pada VADER dihitung dengan menjumlahkan total skor valensi (x) dari seluruh kata dalam kalimat, kemudian dinormalisasi dengan konstanta (α) bernilai 15. Pendekatan ini menjadikan VADER lebih sensitif terhadap ekspresi emosional dalam teks media sosial, karena mempertimbangkan berbagai aspek seperti kata penguat (*booster words*), kata negasi, kapitalisasi, hingga tanda baca [7].

Penentuan klasifikasi sentimen dalam VADER umumnya didasarkan pada ambang nilai compound. N. Sham et al. (2016) menjelaskan bahwa ambang batas default ±0.05 seringkali kurang sensitif dalam membedakan sentimen secara nyata. Oleh karena itu, dalam penelitian N. Sham et al. (2016), digunakan ambang batas ±0.3, yang dinyatakan mampu meningkatkan akurasi klasifikasi karena memberikan pemisahan yang lebih jelas antara sentimen netral dan non-netral [7].

Studi serupa yang dilakukan oleh Abimanyu et al. (2020) juga menerapkan algoritma VADER untuk menganalisis komentar pengguna Twitter terkait akun gim Apex Legends. Dalam penelitian tersebut, klasifikasi sentimen menggunakan ambang batas yang lebih tinggi, yaitu ±0.5. Penggunaan ambang batas ini menyebabkan hanya komentar dengan intensitas sentimen yang sangat tinggi yang dikategorikan sebagai positif atau negatif, sedangkan komentar lainnya cenderung diklasifikasikan sebagai netral [8].

Hasil klasifikasi sentimen oleh VADER dalam studi Abimanyu et al. (2020) menunjukkan distribusi sebagai berikut: 18% positif, 4,8% negatif, dan 77,2% netral. Apabila dibandingkan dengan pelabelan manual oleh penutur asli (*native speaker*) yang menghasilkan 27% positif, 10,8% negatif, dan 62,2% netral, terlihat adanya perbedaan distribusi. Perbedaan ini diyakini disebabkan oleh penentuan ambang batas klasifikasi yang digunakan [8].

Berdasarkan perbandingan antara penelitian N. Sham et al. (2016) dan Abimanyu et al. (2020), dapat disimpulkan bahwa pemilihan ambang batas (threshold) memiliki pengaruh signifikan terhadap distribusi label sentimen. Oleh karena itu, dalam penelitian ini, digunakan ambang batas ±0.3, mengikuti pendekatan yang digunakan oleh N. Sham et. al. (2016) [7].

Pertimbangan penggunaan ambang batas ini adalah karena dianggap lebih moderat dan sesuai dengan karakteristik komentar pada media sosial TikTok yang cenderung menggunakan bahasa informal, singkatan, serta ekspresi yang tidak selalu ekstrem. Dengan penetapan ambang batas ini, diharapkan dapat mengurangi dominasi kelas netral yang terlalu besar, sekaligus meningkatkan sensitivitas sistem dalam mendeteksi sentimen yang lebih halus [7] [8].

Implementasi VADER untuk Pelabelan Sentimen Komentar TikTok Dalam sistem yang dikembangkan, VADER (*Valence Aware Dictionary and sEntiment Reasoner*) digunakan untuk melakukan pelabelan awal sentimen pada komentar TikTok. Mengingat leksikon VADER yang berbasis Bahasa Inggris, seluruh komentar berbahasa Indonesia melalui proses penerjemahan otomatis menggunakan Google Translate terlebih dahulu. Tahapan ini kemudian dilanjutkan dengan perhitungan skor sentimen dan pemberian label berdasarkan nilai compound menggunakan ambang batas ±0.3. Pemilihan metode ini didasarkan pada kepraktisannya serta kemampuannya yang telah terbukti dalam berbagai studi untuk memberikan hasil klasifikasi yang cukup representatif terhadap persepsi pengguna media sosial [7] [8].

1. Algoritma *Machine Learning: Random Forest*

*Random forest* merupakan salah satu algoritma *supervised learning* yang umum digunakan dalam tugas klasifikasi, termasuk analisis sentimen. Algoritma ini termasuk dalam metode *ensemble learning*, yaitu pendekatan yang mengombinasikan sejumlah model lemah, dalam hal ini pohon keputusan (decision trees), untuk membentuk model prediktif yang kuat dan stabil. Dalam Random Forest, setiap pohon dilatih menggunakan subset data dan fitur yang dipilih secara acak, dan hasil akhir ditentukan melalui sistem pemungutan suara (*majority voting*) antar pohon [9].

Kelebihan utama *Random Forest* terletak pada kemampuannya mengurangi risiko *overfitting* yang sering terjadi pada pohon keputusan tunggal, serta kemampuannya menangani data berdimensi tinggi seperti teks. Selain itu, *Random Forest* juga cukup tahan terhadap noise dan data tidak seimbang, karena struktur ensemble-nya menghasilkan prediksi yang lebih konsisten dibandingkan model klasifikasi tradisional lainnya [9].

Secara umum, proses kerja *Random Forest* melibatkan tiga tahap utama. Pertama, data latih dibagi ke dalam beberapa subset acak melalui teknik *bootstrap sampling*. Kedua, setiap pohon dalam hutan dilatih dengan subset tersebut, dan pada setiap percabangan pohon, hanya dipilih sebagian kecil dari seluruh fitur untuk pemisahan data. Ketiga, untuk melakukan klasifikasi terhadap data baru, setiap pohon dalam hutan memberikan prediksinya, dan prediksi akhir dipilih berdasarkan suara terbanyak [9].

Penelitian oleh M. Tsaqif dan W. Maharani (2025) menunjukkan bahwa algoritma *Random Forest* memberikan performa klasifikasi emosi yang lebih baik daripada *decision tree* tunggal. Dalam penelitian tersebut, data media sosial dianalisis menggunakan teknik TF-IDF (*Term Frequency-Inverse Document Frequency*) sebagai representasi fitur , dan *Random Forest* menghasilkan akurasi sebesar 76,17%, mengungguli decision tree yang hanya memperoleh 72,62%. Hal ini membuktikan bahwa Random Forest sangat efektif untuk tugas klasifikasi berbasis teks seperti analisis sentimen [9].

Dalam konteks penelitian ini, algoritma *Random Forest* digunakan untuk memprediksi sentimen komentar TikTok yang sebelumnya telah diberi label secara otomatis menggunakan metode VADER. Komentar-komentar tersebut terlebih dahulu melalui tahap pra-pemrosesan yang mencakup pembersihan karakter tidak relevan, normalisasi kata menggunakan kamus, dan translasi ke dalam Bahasa Inggris. Hasil skor compound dari VADER digunakan untuk memberikan label "positif", "netral", atau "negatif" terhadap komentar, yang kemudian menjadi target (label) untuk proses pelatihan model. Fitur yang digunakan berasal dari representasi teks hasil pra-pemrosesan, seperti vektor kata atau skor frekuensi. Dengan pendekatan ini, sistem mampu mengotomatisasi klasifikasi komentar baru tanpa perlu pelabelan ulang, serta mempertahankan akurasi dan efisiensi kerja model dalam analisis sentimen publik terhadap kinerja Kejaksaan Agung.

1. Evaluasi Model Klasifikasi

Evaluasi model klasifikasi adalah tahapan krusial untuk mengukur kapabilitas algoritma Random Forest dalam memprediksi sentimen komentar TikTok secara akurat, dengan menggunakan label awal dari VADER sebagai ground truth. Salah satu instrumen evaluasi yang fundamental adalah Confusion Matrix, yaitu suatu tabel yang merepresentasikan jumlah prediksi benar dan salah untuk setiap kelas. Confusion Matrix mencakup beberapa elemen utama, antara lain:

1. *True Positive* (TP): jumlah instansi positif yang diprediksi dengan benar.
2. *True Negative* (TN): jumlah instansi negatif yang diprediksi dengan benar.
3. *False Positive* (FP): jumlah instansi negatif yang salah diprediksi sebagai positif.
4. *False Negative* (FN): jumlah instansi positif yang salah diprediksi sebagai negatif.

Dari *confusion matrix* tersebut, dapat dihitung beberapa metrik kinerja model:

1. Akurasi (*Accuracy*) – proporsi prediksi yang benar dari total data:

Mengukur proporsi total prediksi yang benar dari keseluruhan data. Meskipun sederhana dan sering digunakan, metrik ini dapat menyesatkan apabila distribusi kelas dalam dataset tidak seimbang [10].

1. Presisi (*Precision*) – kualitas prediksi positif yang dibuat:

Menilai kualitas prediksi kelas positif yang dihasilkan. Nilai presisi yang tinggi dapat mengindikasikan bahwa model jarang menghasilkan false positive [10].

1. *Recall*(*Sensivity*) – kemampuan model menangkap seluruh kejadian positif:

Mengukur kemampuan model dalam mengidentifikasi seluruh kejadian positif. Recall yang tinggi menunjukkan bahwa model mampu menemukan hampir semua kasus positif, meskipun terkadang harus mengorbankan akurasi [10].

1. F1-*Score* – merupakan rata-rata harmonik antara *precision* dan *recall*:

Metrik ini sangat relevan ketika kedua aspek (presisi dan recall) memiliki kepentingan yang setara, khususnya pada dataset yang memiliki ketidakseimbangan kelas [10].

Studi oleh Vakili et al. (2020) menunjukkan bahwa metrik-metrik tersebut, ketika diterapkan untuk membandingkan berbagai algoritma model klasifikasi termasuk Random Forest, mengindikasikan bahwa Random Forest mampu memberikan performa terbaik dalam hal metrik-metrik tersebut saat diuji terhadap beberapa dataset industri IoT [10].

Dalam penelitian ini, metrik-metrik evaluasi tersebut dihitung secara terpisah untuk setiap kelas sentimen: positif, netral, dan negatif. Pendekatan one-vs-rest digunakan untuk mengonstruksi *confusion matrix* per kelas. Evaluasi yang komprehensif ini esensial untuk memastikan bahwa model tidak memiliki bias terhadap salah satu kelas, terutama kelas netral yang seringkali dominan dalam data media sosial. Dengan demikian, penggunaan presisi, recall, dan F1-Score dianggap krusial untuk menilai kinerja model secara adil dan menyeluruh [10].

1. Arsitektur Sistem: Laravel Framework dan Flask API

Sistem ini dikembangkan menggunakan kerangka kerja **Laravel**, sebuah framework PHP yang mengimplementasikan arsitektur Model–View–Controller (MVC). Arsitektur MVC pada Laravel memisahkan logika aplikasi (controller), struktur data (model), dan antarmuka pengguna (view), yang mempermudah proses pengembangan dan pemeliharaan kode. Laravel juga dilengkapi dengan fitur-fitur modern seperti routing, middleware, Eloquent ORM, dan sistem templating Blade, yang secara signifikan mempercepat pembangunan aplikasi web berskala menengah hingga besar [11].

Untuk menjalankan skrip Python, seperti proses scraping data TikTok menggunakan Selenium dan analisis sentimen VADER, diperlukan sebuah jembatan karena Laravel tidak mendukung eksekusi Python secara natif. Oleh karena itu, dikembangkanlah API berbasis Flask yang berfungsi untuk menangani permintaan HTTP guna menjalankan fungsi-fungsi tersebut. Flask dipilih karena karakteristiknya yang ringan, fleksibel, dan mudah dikembangkan untuk tujuan spesifik seperti *restful* API [12].

Arsitektur sistem dibangun dengan Laravel berperan sebagai *frontend* dan pengendali utama, yang mengirimkan permintaan ke *endpoint* Flask melalui protokol HTTP. Flask akan menerima permintaan tersebut, mengeksekusi skrip Python yang relevan, dan kemudian mengembalikan hasil dalam format data JSON. Respons dari Flask ini akan diterima oleh Laravel, disimpan ke dalam basis data melalui model, dan ditampilkan pada view sesuai kebutuhan pengguna.

Studi komparasi performa API antara Laravel dan Flask menunjukkan bahwa meskipun Laravel menunjukkan keunggulan pada jumlah permintaan menengah, Flask menunjukkan performa yang lebih stabil pada dataset berukuran besar. Sebaliknya, Laravel cenderung mengalami penurunan performa akibat kompleksitas fitur MVC yang lebih tinggi [12]. Temuan ini mengindikasikan bahwa pemanfaatan Flask hanya pada bagian yang memerlukan eksekusi Python merupakan arsitektur yang optimal. Dengan demikian, kombinasi Laravel (yang kuat pada sisi frontend dan manajemen routing) dan Flask (yang efisien untuk eksekusi Python melalui API) menjadikan sistem ini modular, *scalable*, dan optimal dalam menjalankan tugasnya masing-masing.

# BAB III

# METODOLOGI PENELITIAN

1. Data dan Pengumpulan Data
2. Sumber Data

Dalam penelitian ini, data dikumpulkan dari platform media sosial TikTok, secara spesifik kolom komentar pada video publik yang membahas "kinerja Kejaksaan Agung Republik Indonesia." Proses akuisisi data ini dilakukan secara otomatis menggunakan skrip Python yang dikembangkan dengan *Selenium WebDriver* dan metode simulasi sesi pengguna melalui pemanfaatan file cookies pribadi.

Penting untuk dicatat bahwa *cookies* tersebut diperoleh melalui proses login manual pada browser terlebih dahulu. Setelah login berhasil, cookies disimpan dalam format file JSON yang berisi *cookies* aktif agar dapat digunakan kembali secara otomatis dalam operasi *scraping*. Pendekatan ini memungkinkan skrip Python untuk mensimulasikan sesi login aktif dan mengakses halaman komentar secara autentik, sehingga menjamin stabilitas proses *scraping* serta kepatuhan terhadap kebijakan TikTok.

Pencarian video dilakukan berdasarkan kata kunci spesifik, meliputi "kejaksaan agung," "kinerja kejaksaan agung," dan "kejagung." Sistem kemudian akan menelusuri setiap video yang berhasil ditemukan berdasarkan kata kunci tersebut, Proses ini melibatkan pemuatan halaman video menggunakan *browser* otomatis yang telah tersinkronisasi dengan *cookies* pengguna, diikuti dengan ekstraksi elemen-elemen komentar yang tersedia pada halaman tersebut.

Adapun data komentar yang dikumpulkan meliputi:

1. Teks komentar
2. Nama pengguna (*username*)
3. Tanggal komentar
4. Jumlah suka
5. Jumlah balasan
6. Video ID
7. Kata kunci pencarian (taggar)

Seluruh hasil *scraping* disimpan ke dalam basis data MySQL, tepatnya pada tabel “komentar\_mentah”, yang menjadi sumber utama data mentah untuk proses *preprocessing*, pelabelan sentimen, pelatihan model klasifikasi, serta prediksi komentar baru di sistem berbasis web.

1. Teknik Pengumpulan Data

Dalam penelitian ini, teknik pengumpulan data dilakukan melalui pendekatan *automated web scraping* dengan memanfaatkan skrip Python yang terintegrasi dengan *Selenium WebDriver*. Untuk menjaga autentikasi sesi pengguna dan memastikan akses yang stabil ke halaman TikTok, sistem menggunakan file *cookies* dalam format JSON yang diperoleh dari hasil *login* manual pengguna melalui *browser*.

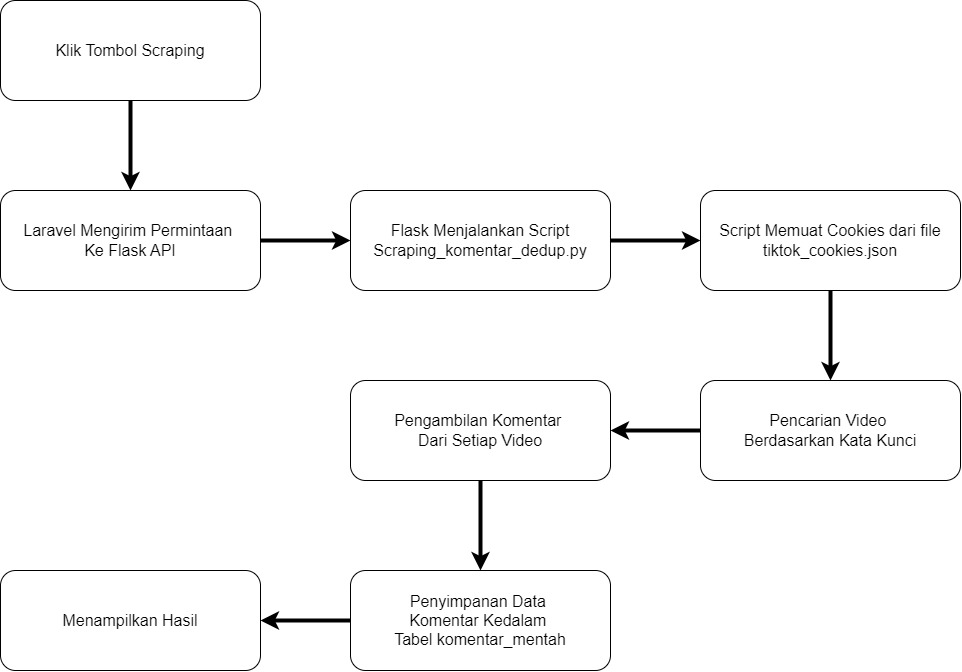
Proses *scraping* ini diotomatisasi melalui antarmuka admin pada sistem berbasis web Laravel. Ketika pengguna menekan tombol "Scraping", sistem akan mengirimkan permintaan HTTP (*Hypertext Transfer Protocol*) ke Flask API, yang kemudian menjalankan skrip *scraping* Python. Skrip tersebut akan memuat halaman TikTok berdasarkan kata kunci yang telah ditentukan, lalu mengekstrak komentar dari setiap video yang ditemukan.

Seluruh komentar yang berhasil diambil akan disimpan secara otomatis ke dalam tabel “komentar\_mentah” pada basis data MySQL. Pencarian dilakukan menggunakan sejumlah kata kunci spesifik, antara lain: “kejaksaan agung”, “kinerja kejaksaan agung”, dan “kejagung”. Setiap video hasil pencarian akan diakses secara individual, dan seluruh komentar publik yang tersedia akan diambil untuk dianalisis lebih lanjut.

Teknik ini memungkinkan sistem untuk mengumpulkan data komentar dalam jumlah besar secara cepat, konsisten, dan berulang tanpa memerlukan login ulang, selama file *cookies* yang digunakan masih aktif.

Selain itu, untuk menjaga efisiensi dan mencegah beban berlebih dalam proses *scraping*, sistem menerapkan batas maksimum (limit) terhadap jumlah komentar yang diambil per video. Pengaturan limit ini dapat disesuaikan pada konfigurasi dalam skrip *scraping*, sehingga proses pengambilan data tetap optimal dan sesuai kapasitas penyimpanan yang tersedia.

Adapaun alur teknis dari proses pengumpulan data komentar TikTok dilakukan secara otomatis melalui integrasi antara framework Laravel sebagai frontend dan Flask API sebagai jembatan pemrosesan backend. Proses ini diawali dari interaksi pengguna di antarmuka admin, hingga sistem menjalankan skrip Python yang bertugas mengambil komentar dari setiap video TikTok berdasarkan kata kunci tertentu. Seluruh komentar kemudian disimpan ke dalam database untuk dianalisis lebih lanjut. Tahapan lengkap proses tersebut dapat dilihat pada Gambar 3.1 berikut.



Gambar 3. Tahapan pengumpulan data

Berdasarkan Gambar 3.1, proses diawali ketika pengguna menekan tombol "Scraping" pada antarmuka Laravel. Sistem kemudian mengirimkan permintaan ke Flask API, yang akan menjalankan skrip “scraping\_komentar\_dedup.py”. Skrip ini memuat file *cookies* untuk autentikasi, melakukan pencarian video berdasarkan kata kunci, dan mengekstrak komentar dari setiap video yang ditemukan. Seluruh data komentar yang berhasil dikumpulkan disimpan ke dalam tabel “komentar\_mentah”. Setelah itu, data komentar ditampilkan kembali pada halaman web Laravel sebagai hasil *scraping*, serta digunakan untuk proses analisis sentimen pada tahap berikutnya.



Gambar 3. Contoh hasil *scraping*

Seperti yang terlihat pada Gambar 3.2, data komentar TikTok yang diperoleh masih mengandung berbagai elemen informal, seperti emoji, tanda baca tidak standar, singkatan, mention, hingga struktur kalimat yang tidak baku. Oleh karena itu, sebelum dilakukan analisis sentimen, data tersebut harus melalui serangkaian proses pembersihan dan penyesuaian terlebih dahulu. Langkah-langkah ini dikenal sebagai tahapan *preprocessing*, yang akan dijelaskan secara rinci pada bagian selanjutnya sebagai fondasi utama dalam proses analisis berbasis leksikon maupun *Machine Learning*.

1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan tahap awal dalam proses perancangan sistem, yang bertujuan untuk mengidentifikasi kebutuhan pengguna serta batasan teknis sistem yang harus dipenuhi. Kebutuhan sistem dalam penelitian ini dibagi menjadi dua, yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

1. Analisis Kebutuhan Fungsional

Kebutuhan fungsional menggambarkan layanan atau fungsi yang harus dapat dijalankan oleh sistem. Sistem ini memiliki dua jenis pengguna utama, yaitu Admin dan User, dengan hak akses dan fitur yang berbeda.

1. Peran Admin:
2. Login ke dalam sistem melalui halaman web Laravel.
3. Menekan tombol “Scraping” untuk memulai proses pengambilan data komentar dari TikTok.
4. Melihat hasil komentar yang telah berhasil di-scrape pada halaman hasil scraping (halaman “result”).
5. Menekan tombol “Analisis ML” untuk menjalankan prediksi sentimen menggunakan model *Random Forest* melalui integrasi Flask API.
6. Melihat hasil prediksi dan label sentimen pada halaman dashboard analisis.
7. Melihat visualisasi statistik (jumlah komentar per sentimen, grafik batang, dll).
8. Mengunduh hasil analisis sentimen kedalam format “.xlsx”
9. Logout dari sistem.
10. Peran User:
11. Login kedalam sistem.
12. Melihat hasil visualisasi analisis sentimen.
13. Mengunduh hasil analisis sentimen kedalam format “.xlsx”
14. Logout dari sistem.
15. Analisis Kebutuhan Non-Fungsional

Kebutuhan non-fungsional mendefinisikan spesifikasi teknis minimum yang esensial agar sebuah sistem dapat beroperasi secara stabil, efisien, dan terintegrasi dengan baik antara komponen backend dan frontend. Kebutuhan ini tidak menjelaskan apa yang dilakukan sistem, melainkan bagaimana sistem melakukannya, memastikan performa dan kualitas yang optimal.

1. Kebutuhan Perangkat Lunak (Software)

Tabel 3. Kebutuhan perangkat lunak (software)

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Komponen | Spesifikasi | Keterangan |
| 1. | Sistem operasi | Windows 8.1 | Digunakan sebagai platform pengembangan dan eksekusi aplikasi Python serta menjalankan XAMPP atau Laravel. |
| 2. | backend | Python 3.12 | |  | | --- | |  |   Digunakan untuk proses *scraping*, *preprocessing*, dan analisis sentimen. |
| 3. | Web Framework | Laravel 11 | Membangun tampilan antarmuka web dan fitur interaksi admin/user. |
| 4. | Web Server | Xampp (Apache) | Bertindak sebagai server lokal untuk menjalankan laravel dan mengelola permintaan pengguna. |
| 5. | Database | MySQL | Menyimpan data hasil analisis, pengguna, dan informasi komentar yang ditampilkan di web. |
| 6. | API Framework | Flask | Menghubungkan Laravel dengan Python (*Scraping* dan prediksi). |
| 7. | Library Python | requests, pandas, json, re, nltk, sqlalchemy,  undetected\_chrome\_driver, transformers, googletrans, scikit-learn, deep\_translator | Digunakan untuk *scraping* TikTok, *preprocessing*, translasi, dan klasifikasi sentimen. |
| 8. | Browser | Google Chrome (dengan Chromedriver) | Digunakan untuk *scraping* menggunakan Selenium. |

1. Kebutuhan Perangkat Keras (Hardware)

Tabel 3. Kebutuhan perangkat keras (hardware)

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Komponen | Spesifikasi/Minimum | Keterangan |
| 1. | Processor | Intel core i5/setara | Digunakan untuk menjalankan Python dan Laravel dengan performa yang optimal. |
| 2. | RAM | 4 GB atau lebih | Dibutuhkan untuk mendukung multitasking saat scraping, analisis, dan menjalankan server lokal. |
| 3. | Hard Disk | 512 GB HDD/SSD | Menyediakan ruang penyimpanan untuk file scraping, output analisis, file Excel, dan database lokal. |
| 4. | Koneksi Internet | Stabil (≥ 2 Mbps) | Dibutuhkan untuk scraping komentar TikTok dan mengakses sistem web secara real-time. |

1. Rancangan Sistem

Perancangan sistem merupakan tahap yang menjelaskan bagaimana sistem dibangun dari sisi alur proses, interaksi antar pengguna dan sistem, hingga struktur antarmuka dan basis data. Perancangan ini menjadi dasar dalam implementasi sistem agar sesuai dengan kebutuhan pengguna dan tujuan dari penelitian ini. Sistem yang dibangun dalam penelitian ini adalah aplikasi berbasis web yang mengintegrasikan framework Laravel untuk manajemen antarmuka dan data, serta Flask API sebagai jembatan untuk menjalankan proses *scraping* dan analisis sentimen menggunakan bahasa pemrograman Python. Untuk menjaga kejelasan dan efisiensi dalam penyampaian, proses-proses utama dalam sistem ini akan dijelaskan secara terpisah dalam beberapa flowchart. Hal ini dilakukan untuk menghindari visualisasi yang terlalu kompleks dalam satu diagram tunggal, serta untuk mempermudah pemahaman terhadap fungsi masing-masing komponen sistem. Pemisahan flowchart dilakukan berdasarkan titik interaksi pengguna, seperti proses login, fitur admin, dan fitur user biasa.

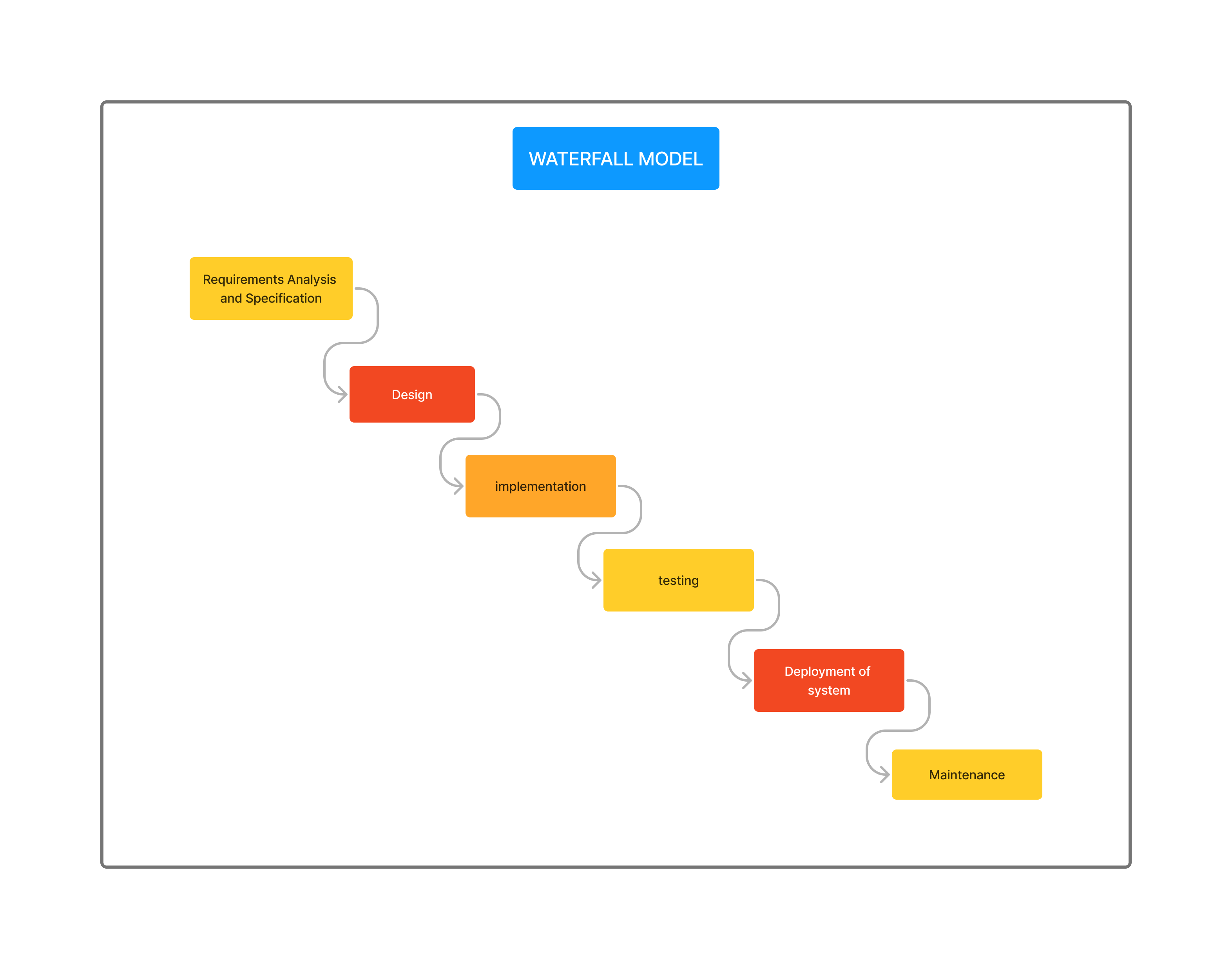
1. Metodologi perancangan Sistem

Perancangan sistem dalam penelitian ini menggunakan metode Waterfall sebagai pendekatan pengembangan sistem. Model Waterfall dipilih karena memiliki alur kerja yang sistematis dan terstruktur dengan tahapan-tahapan yang jelas. Setiap tahap dalam metode ini harus diselesaikan terlebih dahulu sebelum dapat melanjutkan ke tahap berikutnya. Metode ini dinilai cocok karena mampu memastikan bahwa setiap proses seperti analisis kebutuhan, desain, implementasi, hingga pengujian dapat dilakukan secara teratur dan terdokumentasi dengan baik.

Adapun tahapan-tahapan dalam model Waterfall yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Requirements Analysis and Specification, Tahap pengumpulan dan analisis kebutuhan baik dari segi fungsional maupun non-fungsional untuk sistem yang akan dikembangkan.
2. Design, Tahapan untuk merancang arsitektur sistem, database, serta alur proses kerja sistem.
3. Implementation, Tahap penerjemahan hasil desain ke dalam kode program menggunakan framework Laravel, Python Flask API, dan MySQL.
4. Testing, Pengujian sistem untuk memastikan bahwa setiap fungsi berjalan sesuai dengan kebutuhan. Pengujian dilakukan dengan metode black box dan white box.
5. Deployment, Tahap pemasangan atau implementasi sistem ke lingkungan produksi sehingga dapat digunakan oleh pengguna.
6. Maintenance, Tahapan pemeliharaan sistem untuk memperbaiki bug, menambah fitur baru, atau melakukan pembaruan sesuai kebutuhan di masa depan.

Berikut adalah ilustrasi model Waterfall yang digunakan dalam perancangan sistem pada penelitian ini:



Gambar 3. Metodelogi perancangan sistem (*waterfall methodology*)

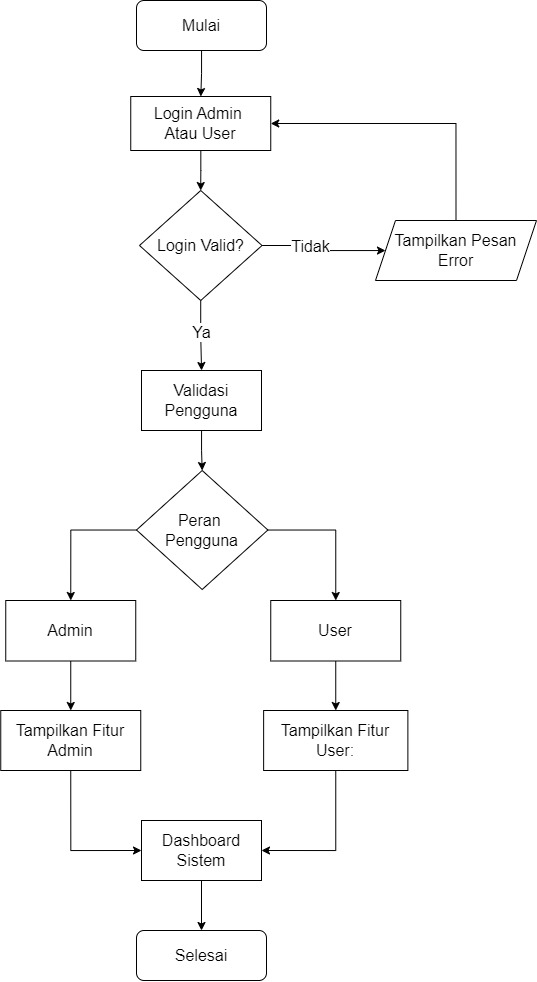
Berdasarkan gambar 3.3 di atas, dapat disimpulkan bahwa proses perancangan sistem dilakukan secara bertahap dan sistematis. Setiap tahapan memiliki peran penting untuk memastikan bahwa sistem yang dikembangkan berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Model Waterfall membantu dalam menjaga kualitas pengembangan sistem karena setiap proses terdokumentasi dengan baik dan dievaluasi sebelum melangkah ke tahap berikutnya. Dengan penerapan metodologi ini, sistem yang dikembangkan diharapkan dapat berjalan optimal dan mudah dalam proses perawatan maupun pengembangan di masa depan.

1. Flowchart Sistem

Flowchart ini menjelaskan alur awal sistem, dimulai dari proses login oleh pengguna hingga pengalihan ke fitur yang sesuai dengan peran pengguna. Setiap pengguna yang berhasil login akan diverifikasi perannya, apakah sebagai admin atau sebagai user biasa. Berdasarkan hasil verifikasi tersebut, sistem akan menampilkan fitur yang dapat diakses sesuai dengan hak akses masing-masing peran. Jika pengguna merupakan admin, maka ia akan memiliki akses untuk melakukan scraping komentar TikTok, menjalankan analisis sentimen menggunakan model *Machine Learning*, serta melihat dan mengelola hasil analisis. Sebaliknya, jika pengguna adalah user biasa, maka ia hanya dapat melihat hasil analisis sentimen yang telah tersedia.

1. Flowchart Login

Flowchart pada Gambar 3.3 menjelaskan alur proses login dan pengelolaan akses pengguna di dalam sistem. Proses dimulai ketika pengguna (baik admin maupun user biasa) mengakses halaman login dan memasukkan kredensial berupa username dan password. Sistem kemudian memverifikasi keabsahan data login yang dimasukkan. Jika validasi gagal, sistem akan menampilkan pesan kesalahan dan meminta pengguna untuk mengulangi proses login. Sebaliknya, jika login berhasil, sistem akan memeriksa peran pengguna dan menampilkan fitur yang sesuai dengan hak akses yang dimiliki. Alur ini merupakan dasar dari pengelolaan kontrol akses sistem, yang membedakan fungsi antara admin dan user biasa.

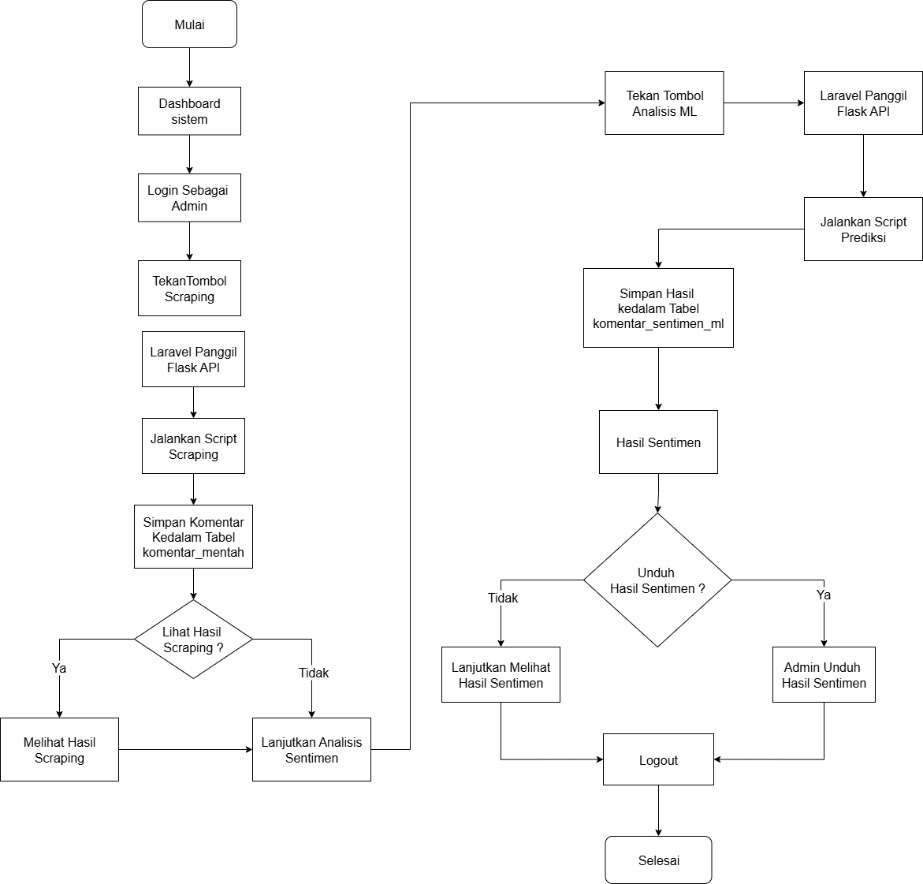


Gambar 3. Flowchart sistem (login)

Gambar 3.4 memperlihatkan bahwa setelah proses validasi login berhasil, sistem melakukan percabangan berdasarkan peran pengguna. Jika pengguna adalah admin, maka sistem menampilkan fitur seperti scraping komentar TikTok, tampilan hasil scraping, analisis Sentimen, tampilan hasil analisis sentimen, dan unduh hasil analisis sentimen. Jika pengguna adalah user biasa, maka sistem hanya menampilkan hasil analisis sentimen dan unduh hasil analisis sentimen yang telah tersedia. Meskipun seluruh pengguna diarahkan ke halaman Dashboard Sistem yang sama, fitur yang tersedia akan disesuaikan secara dinamis berdasarkan peran. Pengaturan ini diterapkan menggunakan kontrol tampilan berbasis role (*role-based access control*) di Laravel, sehingga sistem tetap efisien tanpa perlu memisahkan halaman secara fisik. Alur ini mencerminkan manajemen hak akses yang jelas dan terstruktur dalam sistem.

1. Flowchart Fitur Admin

Gambar 3.5 menggambarkan alur aktivitas yang dapat dilakukan oleh pengguna dengan peran admin setelah berhasil masuk ke sistem. Proses dimulai dari tampilan dashboard, di mana admin dapat memilih untuk menekan tombol *Scraping* guna memulai pengambilan komentar dari TikTok. Proses *scraping* dilakukan melalui integrasi Laravel dan Flask API, dan hasilnya disimpan dalam tabel komentar\_mentah. Setelah proses *scraping* selesai, admin dapat memilih untuk melihat hasil komentar terlebih dahulu atau langsung melanjutkan ke tahap analisis sentimen. Analisis dilakukan dengan menekan tombol Analisis ML, yang akan memicu proses prediksi sentimen menggunakan model *Machine Learning* berbasis *Random Forest*. Hasil analisis disimpan dalam tabel komentar\_sentimen\_ml, dan dapat dilihat atau diunduh oleh admin untuk keperluan dokumentasi lebih lanjut.

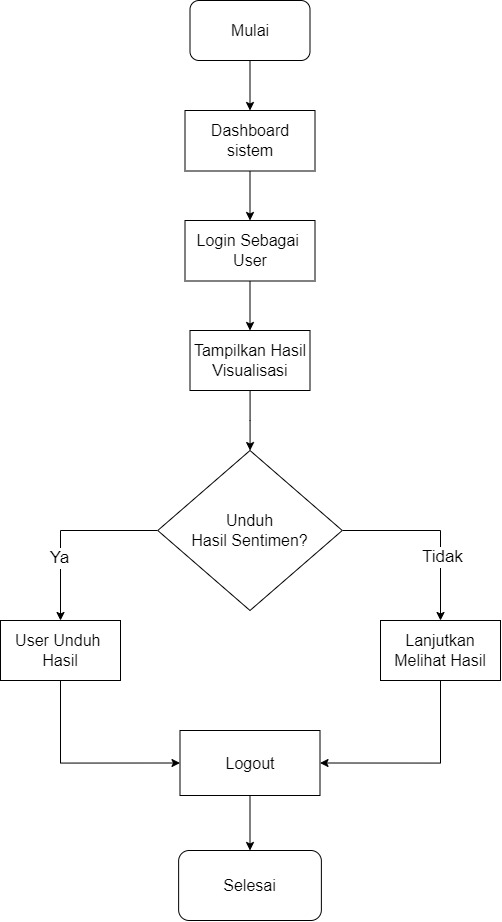


Gambar 3. Flowchart sistem (fitur admin)

Flowchart pada Gambar 3.5 menampilkan seluruh proses kerja admin dalam sistem, mulai dari pengambilan data komentar TikTok, pemrosesan analisis sentimen, hingga akses hasil analisis. Setiap proses dihubungkan secara linier dan logis, dengan percabangan pada titik-titik interaktif seperti pilihan untuk melihat hasil *scraping* terlebih dahulu atau mengunduh hasil sentimen. Alur ini menunjukkan bahwa sistem memberikan fleksibilitas kepada admin dalam mengelola alur kerja analisis sentimen, dengan semua proses berjalan secara terotomatisasi melalui integrasi Laravel dan Flask API. Selain itu, kontrol akses yang hanya diberikan kepada admin memastikan bahwa fitur-fitur ini tidak dapat dijalankan oleh user biasa, sehingga menjaga struktur dan keamanan sistem secara keseluruhan.

1. Flowchart Fitur User

Gambar 3.6 menjelaskan alur aktivitas pengguna dengan peran user biasa setelah berhasil login ke dalam sistem. Setelah diarahkan ke dashboard, sistem secara otomatis menampilkan hasil visualisasi dari analisis sentimen yang telah diproses sebelumnya oleh admin. Informasi ini disajikan dalam bentuk grafik dan tabel, sehingga memudahkan user dalam memahami sentimen publik terhadap topik yang dianalisis. Selanjutnya, user dapat memilih untuk mengunduh hasil tersebut dalam format file, atau melanjutkan melihat hasil analisis langsung dari dashboard.



Gambar 3. Flowchart sistem (fitur user)

Flowchart pada Gambar 3.6 menegaskan bahwa pengguna dengan peran user hanya memiliki akses untuk melihat dan mengunduh hasil analisis yang telah tersedia. Tidak terdapat fitur scraping maupun analisis yang dapat dijalankan oleh user, sesuai dengan batasan hak akses sistem. Jalur alur yang lebih sederhana ini juga menandakan perbedaan peran yang jelas antara admin dan user. Dengan demikian, sistem dapat menjamin bahwa proses manipulasi data dan analisis hanya dapat dilakukan oleh pihak yang berwenang, sementara user tetap dapat memperoleh informasi hasil analisis secara aman dan efisien.

1. Rancangan Business Rules

*Business rules* atau aturan bisnis adalah seperangkat ketentuan yang mengatur bagaimana suatu sistem bekerja dalam konteks domain permasalahan tertentu. Dalam penelitian ini, business rules digunakan untuk mengatur alur proses, hak akses pengguna, serta hubungan antar entitas yang terlibat dalam sistem analisis sentimen berbasis data komentar TikTok. Aturan bisnis ini menjadi dasar bagi pengembangan fungsionalitas sistem dan penerapan kontrol akses sesuai dengan peran pengguna.

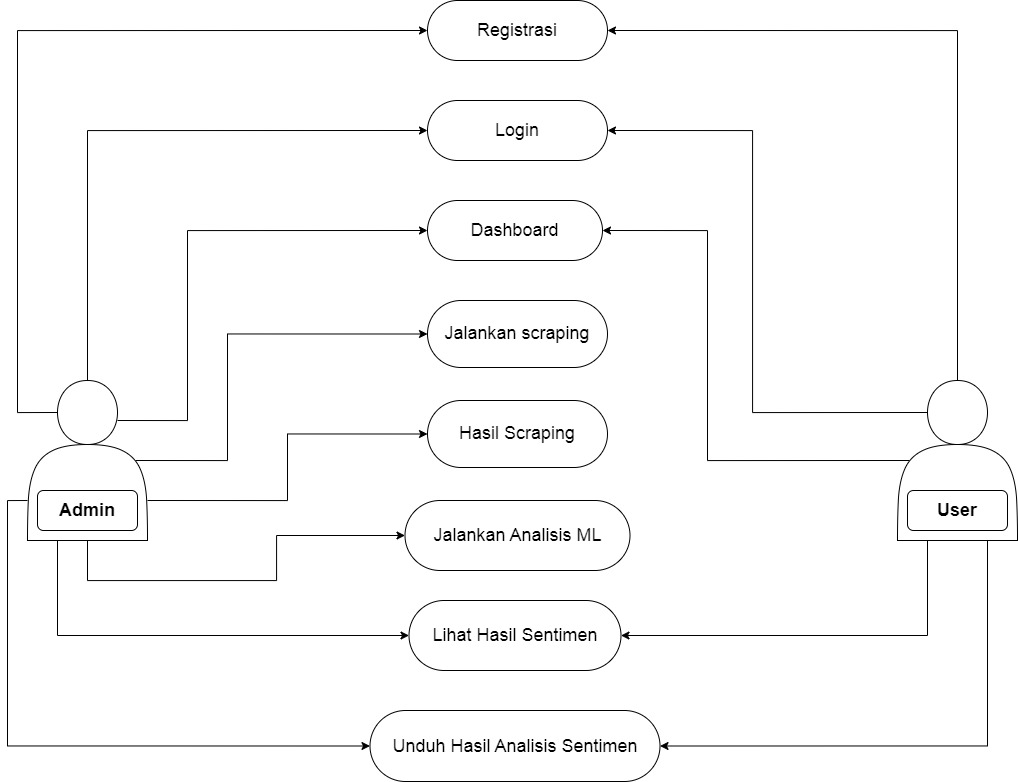
Tabel 3. Rancangan *business rules*

|  |  |
| --- | --- |
| No. | *Business Rules* |
| R1 | Pengguna harus melakukan Registrasi dan login terlebih dahulu untuk dapat mengakses sistem. |
| R2 | Terdapat dua jenis peran pengguna: Admin dan User. |
| R3 | Sistem menampilkan fitur dashboard sesuai peran pengguna yang terdeteksi saat login (*Role-Based Access Control*). |
| R4 | Admin memiliki hak akses untuk menjalankan proses scraping komentar TikTok melalui tombol Scraping pada dashboard. |
| R5 | Setelah proses scraping berhasil, data komentar disimpan ke dalam tabel komentar\_mentah. |
| R6 | Hasil komentar yang telah di-scrape ditampilkan pada halaman Result, dan hanya dapat diakses oleh Admin. |
| R7 | Admin dapat menjalankan analisis sentimen menggunakan tombol Analisis ML, yang akan memicu pemanggilan Flask API. |
| R8 | Hasil analisis sentimen disimpan ke dalam tabel komentar\_sentimen\_ml dan ditampilkan dalam bentuk grafik dan tabel pada dashboard. |
| R9 | Username komentar tidak ditampilkan kepada pengguna dengan peran User, baik dalam tampilan dashboard maupun dalam file hasil unduhan. |
| R10 | Pengguna dengan peran User hanya dapat melihat visualisasi hasil analisis tanpa hak untuk mengedit atau memanipulasi data. |
| R11 | Hanya Admin yang dapat mengunduh seluruh hasil analisis secara lengkap (termasuk username) dalam format file .xlsx. |
| R12 | Proses scraping dan analisis dilakukan melalui integrasi antara Laravel dan Flask API menggunakan permintaan HTTP secara otomatis. |
| R13 | Data hasil scraping maupun hasil analisis tidak dapat diubah atau dimodifikasi oleh pengguna dengan peran User, untuk menjaga integritas data. |
| R14 | Setiap pengguna dapat melakukan proses logout kapan saja untuk mengakhiri sesi akses dengan aman. |

Aturan-aturan bisnis di atas menjadi fondasi utama dalam merancang alur kerja sistem secara terstruktur dan terkontrol. Dengan menerapkan business rules ini, sistem mampu menjalankan setiap proses secara konsisten dan sesuai dengan kebutuhan pengguna. Selain itu, *business rules* juga membantu menjaga keamanan, validitas data, serta efisiensi proses pengumpulan dan analisis data dalam sistem yang dibangun.

1. Use Case Diagram

Use case diagram merupakan bagian dari *Unified Modeling Language* (UML) yang digunakan untuk memodelkan interaksi antara aktor dan sistem. Diagram ini menggambarkan hubungan antara pengguna dan fungsi-fungsi utama sistem yang dapat dijalankan. Dalam konteks penelitian ini, sistem dibangun untuk melakukan proses scraping komentar TikTok, analisis sentimen dengan metode *Machine Learning*, serta penyajian hasil analisis kepada pengguna. Terdapat dua aktor utama dalam sistem, yaitu Admin dan User. Setiap aktor memiliki hak akses dan fungsionalitas yang berbeda, sesuai dengan peran masing-masing. Admin memiliki kontrol penuh terhadap proses pengambilan data, pemrosesan analisis, serta pengelolaan hasil. Sementara itu, user hanya memiliki hak akses terbatas untuk melihat dan mengunduh hasil analisis yang telah tersedia.



Gambar 3. *Use case diagram*

Berdasarkan Gambar 3.7, terlihat bahwa proses interaksi sistem dimulai dari login atau registrasi akun. Setelah login berhasil, pengguna akan diarahkan ke halaman dashboard. Pada dashboard, Admin memiliki kemampuan untuk menjalankan proses scraping komentar dari TikTok, melihat hasil komentar, menjalankan analisis *Machine Learning*, serta mengakses dan mengunduh hasil analisis sentimen. Di sisi lain, User hanya memiliki akses untuk melihat dan mengunduh hasil analisis yang telah tersedia. *Use case diagram* ini menjelaskan bahwa meskipun sistem memiliki tampilan dashboard yang sama, fitur yang dapat diakses sangat tergantung pada peran pengguna yang sedang login. Pemisahan tanggung jawab ini merupakan implementasi dari konsep *role-based access control* (RBAC), yang memastikan bahwa hanya pengguna dengan hak tertentu yang dapat menjalankan proses-proses penting di dalam sistem.

1. *Use case*registrasi, login dan logout (admin & user)

Tabel 3. *Use case* registrasi, login dan logout akun (Admin & User)

|  |  |
| --- | --- |
| Elemen | Deskripsi |
| Use Case | Registrasi, Login, dan Logout Pengguna |
| Deskripsi | Pengguna melakukan pendaftaran akun jika belum memiliki akun, login untuk masuk ke sistem, dan logout untuk mengakhiri sesi. |
| Aktor | Admin & User |
| Precondition | Pengguna mengakses halaman awal sistem (belum login) |
| Postcondition | Sistem mengenali peran dan menampilkan fitur sesuai peran, atau sesi diakhiri saat logout |
| Business Rules | R1, R2, R3, R14 |
| ***Main Flow of Event*** | |
| Actor Action | System Response |
| Pengguna membuka halaman awal sistem | Sistem menampilkan opsi registrasi dan login |
| (Jika belum punya akun) Mengisi form registrasi | Sistem memvalidasi dan menyimpan data ke database |
| (Jika sudah punya akun) Mengisi form login | Sistem memverifikasi kredensial |
| Sistem mengenali peran pengguna | Sistem mengarahkan ke dashboard dengan fitur sesuai peran. |
| Pengguna logout | Sistem mengakhiri sesi dan kembali ke halaman awal |
| **Exceptiona Flow of Event** | |
| Section 1 | Email sudah digunakan → tampilkan pesan error |
| Section 2 | Login gagal → tampilkan “Username/Password salah” |

1. *Use case* jalankan *scraping* komentar TikTok (Aktor Admin)

Tabel 3. *Use case* jalankan *scraping* komentar TikTok (Aktor Admin)

|  |  |
| --- | --- |
| Elemen | Deskripsi |
| Use Case | Jalankan Scraping Komentar TikTok dan Tampilkan Hasil |
| Deskripsi | Admin menekan tombol "Scraping" untuk mengambil komentar dari TikTok dan langsung melihat hasilnya di halaman Result. |
| Aktor | Admin |
| Precondition | Admin telah login dan berada di halaman dashboard |
| Postcondition | Komentar disimpan ke tabel komentar\_mentah, lalu admin diarahkan ke halaman Result |
| Business Rules | R3, R4, R5, R12 |
| ***Main Flow of Event*** | |
| Actor Action | System Response |
| Admin klik tombol Scraping | Laravel kirim permintaan ke Flask API |
| Flask menjalankan proses scraping | Komentar diambil dari TikTok dan disimpan ke tabel komentar\_mentah |
| Sistem arahkan ke halaman Result | Komentar yang berhasil di-scrape ditampilkan dalam tabel |
| **Exceptiona Flow of Event** | |
| Section 1 | Jika scraping gagal → sistem tampilkan pesan error |
| Section 2 | Tidak ditemukan komentar → tampilkan pesan “Data tidak ditemukan” |

1. *Use case* jalankan dan lihat hasil analisis sentimen ML (aktor Admin)

Tabel 3. *Use case* jalankan analisis sentimen ML (aktor Admin)

|  |  |
| --- | --- |
| Elemen | Deskripsi |
| Use Case | Jalankan Analisis Sentimen dan Lihat Hasil |
| Deskripsi | Admin menekan tombol Analisis ML untuk memproses komentar menggunakan model *Machine Learning* (*Random Forest*). kemudian sistem langsung menyimpan dan menampilkan hasil prediksi di dashboard yang sama. |
| Aktor | Admin |
| Precondition | Data komentar tersedia dan belum diprediksi |
| Postcondition | Hasil disimpan di tabel komentar\_sentimen\_ml dan ditampilkan di dashboard |
| Business Rules | R6, R7, R12 |
| ***Main Flow of Event*** | |
| Actor Action | System Response |
| Admin klik tombol “Analisis ML” | Laravel panggil Flask API untuk menjalankan model ML |
| Model ML diproses | Prediksi disimpan ke tabel komentar\_sentimen\_ml |
| Sistem tampilkan notifikasi berhasil | Hasil visualisasi langsung ditampilkan di halaman dashboard |
| **Exceptiona Flow of Event** | |
| Section 1 | Tidak ada data baru → tampilkan pesan “Semua data sudah dianalisis” |
| Section 2 | Proses prediksi gagal → tampilkan pesan error |

1. *Use case* lihat hasil analisis sentimen (aktor user)

Tabel 3. *Use case* lihat hasil analisis sentimen (aktor user)

|  |  |
| --- | --- |
| Elemen | Deskripsi |
| Use Case | Lihat Hasil Visualisasi Sentimen |
| Deskripsi | User melihat hasil analisis sentimen berupa grafik dan tabel tanpa akses ke username |
| Aktor | User |
| Precondition | User telah login dan Data hasil analisis tersedia |
| Postcondition | Hasil ditampilkan tanpa informasi sensitif |
| Business Rules | R1, R7, R8, R9, R10 |
| ***Main Flow of Event*** | |
| Actor Action | System Response |
| User buka halaman hasil analisis (halaman dashboard) | Sistem tampilkan grafik dan tabel tanpa username |
| **Exceptiona Flow of Event** | |
| Section 1 | Jika tidak ada data → tampilkan pesan: “Data analisis belum tersedia” |

1. *Use case* unduh hasil analisis sentimen (aktor admin & user)

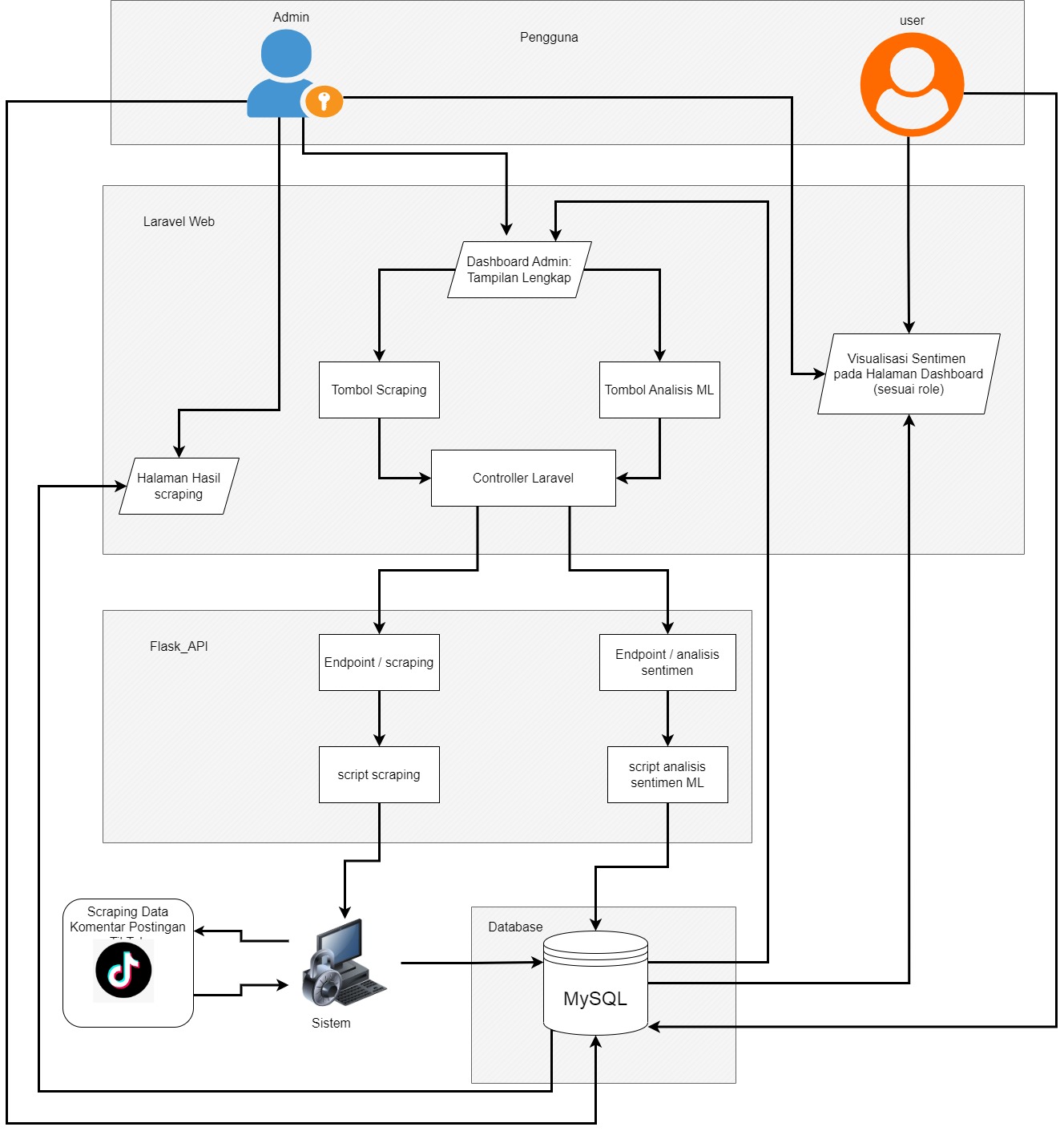
Tabel 3. *Use case* unduh hasil analisis sentimen (aktor admin & user)

|  |  |
| --- | --- |
| Elemen | Deskripsi |
| Use Case | Unduh Hasil Analisis Sentimen |
| Deskripsi | Pengguna mengunduh hasil analisis, dengan data lengkap untuk Admin dan versi terbatas untuk User |
| Aktor | Admin, User |
| Precondition | Hasil analisis tersedia di sistem dan User sudah login |
| Postcondition | Data hasil analisis tersedia di database |
| Business Rules | R9, R10, R11 |
| ***Main Flow of Event*** | |
| Actor Action | System Response |
| Admin/User klik tombol “Unduh Hasil” | Sistem menyiapkan file hasil analisis sesuai peran pengguna |
| File diunduh | Admin menerima file lengkap (dengan username), User menerima versi tanpa username |
| **Exceptiona Flow of Event** | |
| Section 1 | Tidak ada data → tampilkan “Hasil analisis belum tersedia” |
| Section 2 | Gagal ekspor → tampilkan pesan “Terjadi kesalahan saat mengunduh” |

Sebagai rangkuman, seluruh kebutuhan sistem telah dimodelkan dalam bentuk use case berdasarkan peran pengguna, yaitu Admin dan User. Setiap skenario interaksi antara aktor dan sistem telah dijabarkan secara terstruktur, mencakup alur utama, kondisi awal dan akhir, hingga penanganan pengecualian. Use case diagram dan tabel-tabel tersebut menjadi dasar konseptual dalam merancang serta mengimplementasikan fungsionalitas sistem secara menyeluruh. Dengan demikian, proses perancangan sistem dapat dilakukan secara terarah dan sesuai dengan tujuan penelitian.

1. Arsitektur Sistem (Laravel & Flask API)

Subbab ini menjelaskan arsitektur sistem yang dibangun dalam penelitian, yang mengintegrasikan proses scraping komentar TikTok dan analisis sentimen secara otomatis ke dalam satu platform berbasis web. Arsitektur ini terdiri dari beberapa komponen utama, yaitu Laravel Web sebagai antarmuka pengguna, Flask API sebagai penghubung ke proses backend berbasis Python, serta MySQL sebagai basis data pusat penyimpanan komentar dan hasil analisis. Perbedaan peran antara Admin dan User juga memengaruhi fitur yang ditampilkan, dengan sistem yang mengatur tampilan berdasarkan role pengguna saat login. Arsitektur lengkap sistem ditampilkan pada Gambar 3.8.



Gambar 3. Arsitektur sistem

Gambar 3.8 di atas menunjukkan alur integrasi sistem dari sisi pengguna hingga backend. Proses dimulai dari pengguna yang login ke dalam sistem. Setelah berhasil masuk, pengguna akan diarahkan ke dashboard yang sesuai dengan perannya. Admin memiliki akses penuh terhadap fitur-fitur sistem seperti menjalankan scraping komentar TikTok dan melakukan analisis sentimen menggunakan model *Machine Learning* (ML). Tombol “Scraping” pada dashboard Admin akan mengirim permintaan HTTP ke Flask API, yang kemudian menjalankan script scraping berbasis Selenium untuk mengambil komentar dari TikTok. Hasil scraping disimpan secara otomatis ke dalam tabel komentar\_mentah.

Selanjutnya, Admin dapat menekan tombol “Analisis ML” yang akan memicu endpoint Flask lain untuk menjalankan proses prediksi sentimen. Script Python yang digunakan untuk analisis ini telah menggabungkan tahap *preprocessing*, seperti pembersihan teks, normalisasi, dan translasi, ke dalam satu pipeline bersama proses prediksi. Hasil analisis disimpan ke dalam tabel komentar\_sentimen\_ml dan ditampilkan langsung di dashboard.

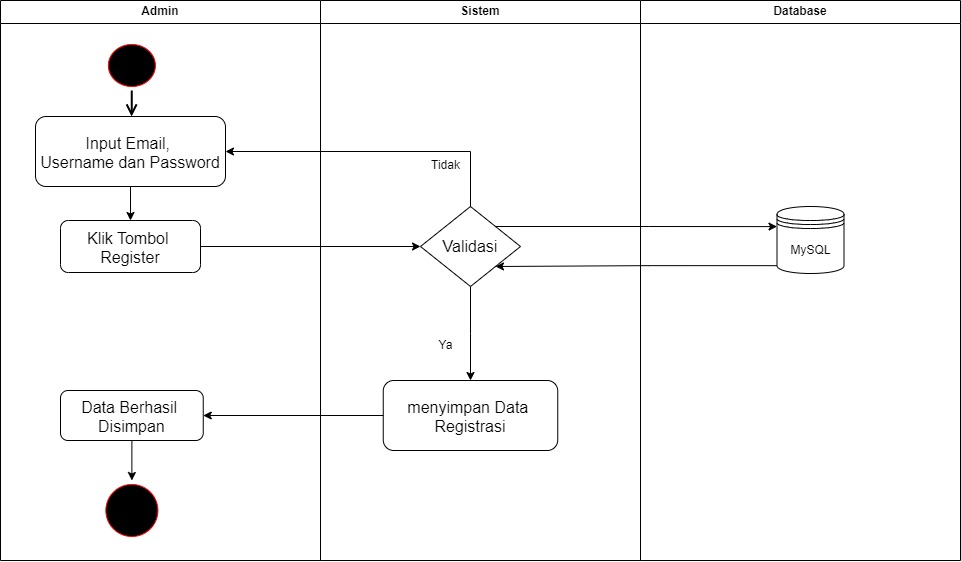
Sementara itu, pengguna dengan peran User hanya dapat mengakses hasil visualisasi sentimen tanpa informasi sensitif seperti username komentar. Hal ini dilakukan untuk menjaga privasi dan integritas data. Sistem ini secara otomatis menyesuaikan tampilan dashboard berdasarkan peran pengguna saat login (*role-based access*). Integrasi Laravel dan Flask API memungkinkan sistem untuk bekerja secara dinamis dan otomatis mulai dari pengumpulan data, pengolahan, hingga penyajian hasil kepada pengguna akhir.

1. Activity Diagram

*Activity diagram* menggambarkan alur aktivitas dalam sistem berdasarkan interaksi antara pengguna dan sistem itu sendiri. Dalam penelitian ini, pengguna dibedakan menjadi dua jenis peran, yaitu Admin dan User. Masing-masing peran memiliki akses dan alur aktivitas yang berbeda sesuai dengan hak akses yang diberikan. Untuk mempermudah pemahaman, diagram aktivitas akan dipisahkan berdasarkan setiap proses utama yang dijalankan oleh pengguna. Dengan demikian, alur proses dapat digambarkan secara lebih rinci, modular, dan mudah dipahami.

1. *Activity Diagram* Admin
2. *Activity diagram* registrasi

Proses registrasi merupakan tahap awal sebelum Admin dapat melakukan login ke dalam sistem. Dalam proses ini, Admin perlu mengisi formulir registrasi yang terdiri atas data penting seperti email, username, dan password. Setelah data diinput dan tombol "Register" ditekan, sistem akan menyimpan informasi tersebut ke dalam basis data menggunakan MySQL. Jika penyimpanan berhasil, sistem akan memberikan notifikasi bahwa proses registrasi berhasil. Tahapan ini penting untuk memastikan hanya pengguna resmi yang dapat memperoleh akses ke sistem dengan hak sebagai Admin.

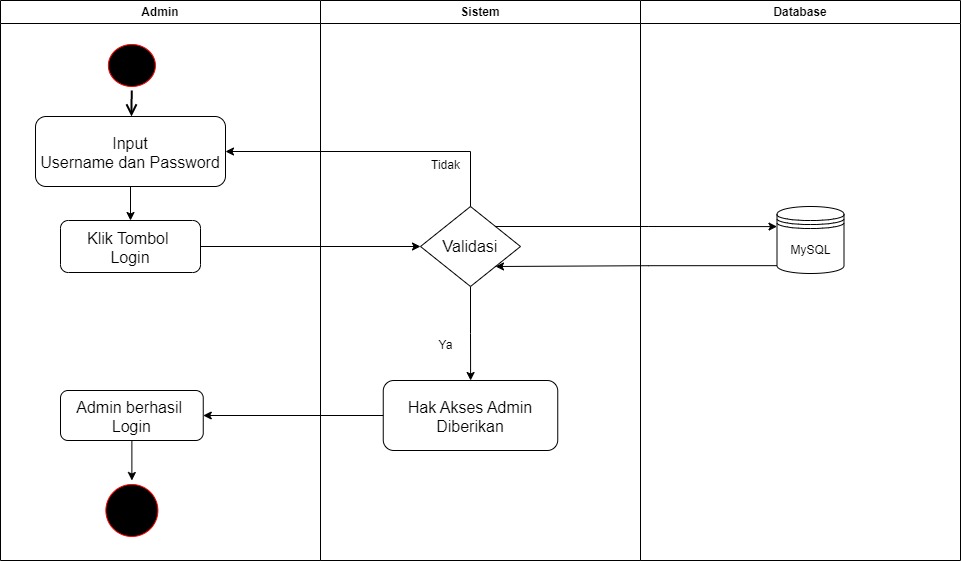


Gambar 3. *Activity diagram* admin (registrasi)

Diagram Gambar 3.9 di atas menggambarkan tahapan registrasi Admin yang dimulai dari pengisian formulir dan diakhiri dengan konfirmasi penyimpanan ke basis data. Seluruh data pendaftaran disimpan secara permanen ke dalam sistem menggunakan MySQL, yang kemudian digunakan untuk proses autentikasi login di tahap selanjutnya. Dengan adanya proses registrasi ini, sistem dapat menjaga integritas pengguna dan memastikan bahwa setiap akses yang masuk berasal dari pihak yang berwenang.

1. *Activity diagram* login

Login merupakan proses autentikasi yang wajib dilakukan oleh Admin sebelum dapat mengakses sistem. Setelah berhasil melakukan registrasi, Admin akan menggunakan username dan password yang telah terdaftar untuk masuk ke dalam sistem. Proses login ini bertujuan untuk memverifikasi identitas pengguna agar sistem hanya dapat diakses oleh pihak yang sah. Sistem akan mencocokkan kredensial yang dimasukkan dengan data yang tersimpan di basis data. Jika sesuai, maka Admin diarahkan ke dashboard utama. Namun jika data tidak valid, sistem akan menampilkan pesan kesalahan.

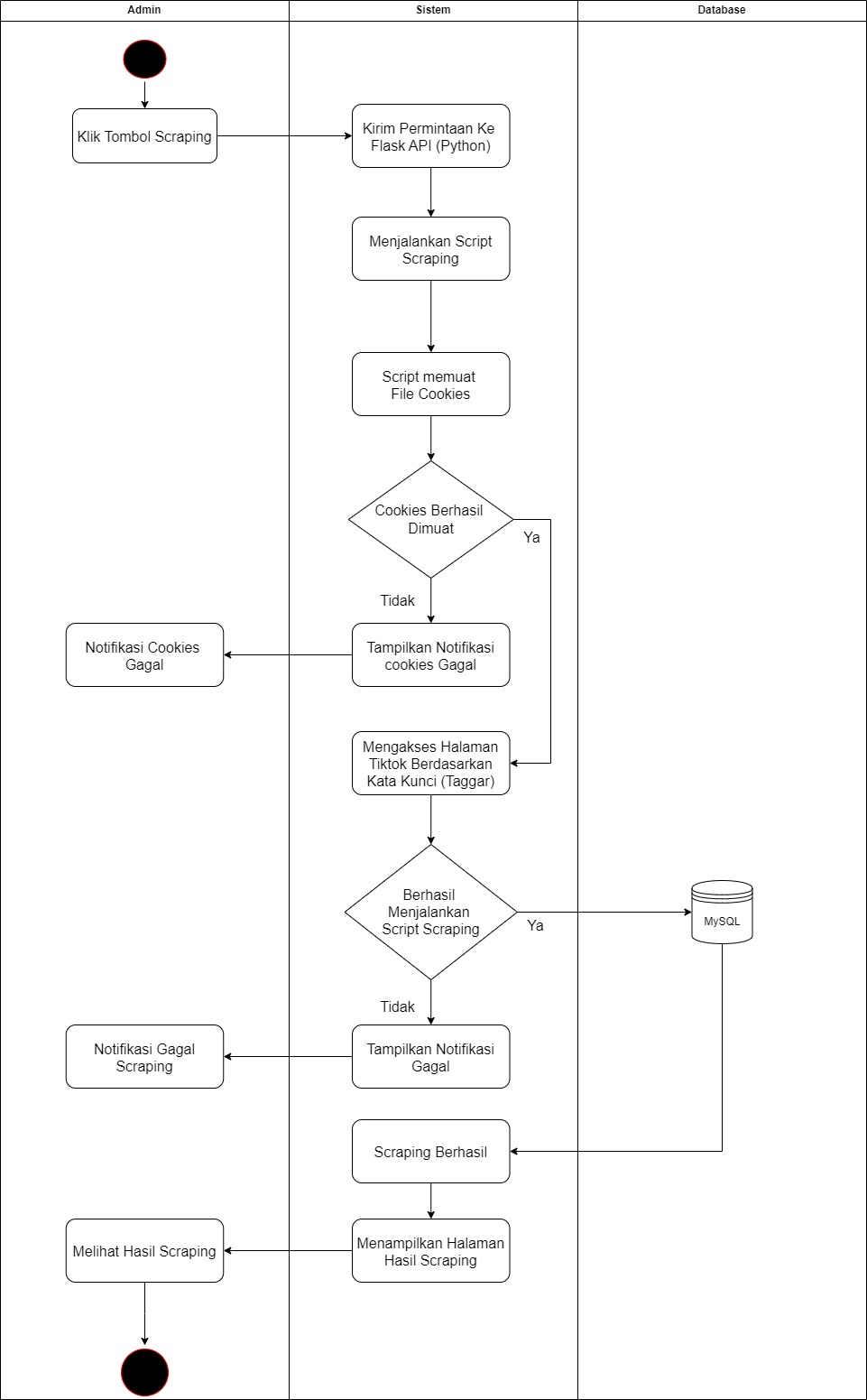


Gambar 3. *Activity diagram* admin (login)

Diagram Gambar 3.10 di atas menjelaskan bahwa login dilakukan dengan memasukkan username dan password, yang kemudian diverifikasi oleh sistem melalui basis data MySQL. Jika data cocok, Admin diberikan akses ke sistem. Jika tidak cocok, sistem memberikan notifikasi kesalahan dan meminta Admin untuk mengulangi proses login. Tahap login ini berperan penting dalam menjaga keamanan sistem dan menghindari akses yang tidak sah ke fitur-fitur administratif.

1. *Activity diagram scraping* komentar TikTok

Setelah berhasil login, Admin memiliki hak akses untuk melakukan pengambilan data komentar dari TikTok melalui fitur scraping yang tersedia di dashboard sistem. Scraping komentar merupakan proses otomatis yang dilakukan oleh Admin untuk mengumpulkan data komentar dari TikTok. Proses ini dimulai setelah Admin menekan tombol “Scraping” pada dashboard sistem. Sistem akan mengirim permintaan ke Flask API yang bertanggung jawab menjalankan script Python scraping\_komentar\_dedup.py. Untuk dapat mengakses konten TikTok dengan sukses, Flask API terlebih dahulu memuat file cookies JSON yang telah disiapkan sebelumnya melalui proses login manual. Cookies ini digunakan agar TikTok mengenali sesi pengguna dan mengizinkan akses ke halaman video dan kolom komentar.

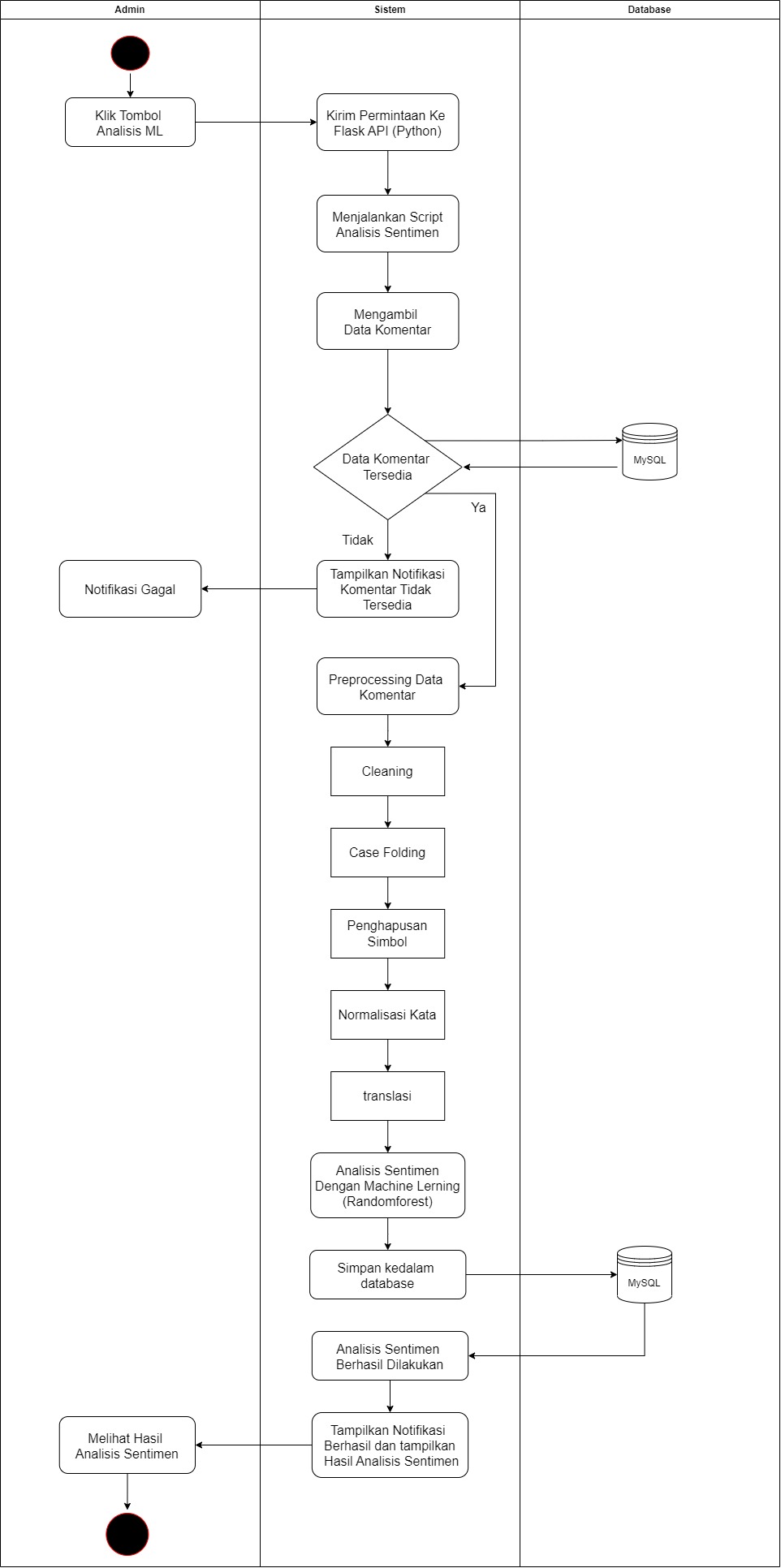


Gambar 3. *Activity diagram* admin (tombol *scraping*)

Diagram Gambar 3.11 di atas menggambarkan bagaimana sistem melakukan proses scraping secara otomatis. Tahapan pertama adalah memuat file *cookies* sebagai bentuk autentikasi agar TikTok mengizinkan akses ke halaman video dan komentar. Jika cookies valid, maka *scraping* dapat dilanjutkan. Proses *scraping* dilakukan berdasarkan tagar tertentu, seperti “kejaksaan agung” atau “kinerja kejaksaan agung”. Komentar yang berhasil diambil akan diteruskan kembali ke sistem dan ditampilkan ke Admin melalui halaman Result. Jika proses gagal, baik karena cookies tidak valid atau gagal mengakses halaman TikTok, sistem akan memberikan notifikasi kesalahan yang sesuai. Dengan adanya validasi *cookies* dan kondisi penanganan error yang jelas, sistem menjadi lebih tangguh dan siap untuk penggunaan berulang secara mandiri oleh Admin.

1. *Activity diagram* analisis sentimen *machine learning*

Setelah proses *scraping* komentar TikTok selesai dilakukan, Admin dapat melanjutkan ke tahap analisis sentimen menggunakan pendekatan *Machine Learning* (*randomforest*). Tahapan ini dimulai saat Admin menekan tombol “Analisis ML” pada dashboard. Sistem akan mengirimkan permintaan ke Flask API, yang selanjutnya mengeksekusi script analisis sentimen untuk memproses komentar dan memprediksi sentimennya secara otomatis. Proses ini dilakukan sepenuhnya tanpa intervensi manual, mencakup pengambilan data, *preprocessing*, translasi hingga klasifikasi dan penyimpanan hasil.



Gambar 3. *Activity diagram* admin (tombol analisis ML)

Diagram Gambar 3.12 di atas menggambarkan alur lengkap proses analisis sentimen yang dilakukan secara otomatis oleh sistem. Tahapan diawali dengan pengambilan data dari tabel komentar\_mentah pada database, dengan kriteria komentar yang belum diproses oleh model *Machine Learning*, yaitu “is\_processed\_ml = 0”. Data yang ditemukan kemudian melewati proses preprocessing, yang meliputi:

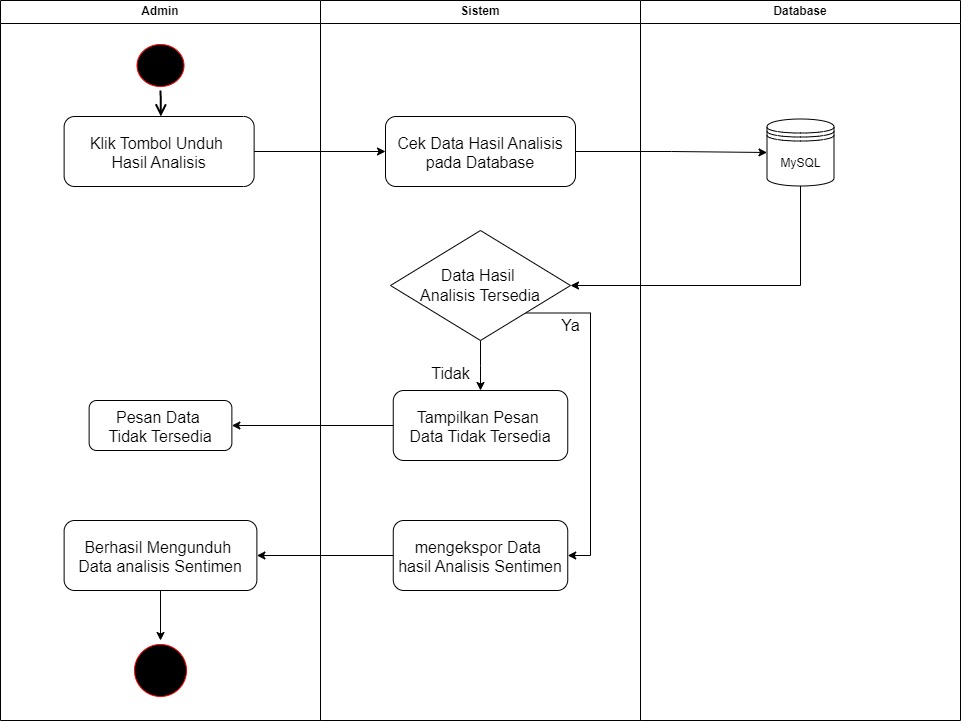
1. *Cleaning*, yaitu penghapusan URL, mention, angka, dan simbol-sombol tidak relevan.
2. *Case folding*, yaitu mengubah seluruh huruf menjadi huruf kecil.
3. Penghapusan simbol tambahan dan whitespace tidak penting.
4. Normalisasi kata menggunakan kamus normalisasi agar kata tidak baku untuk diseragamkan.
5. Translasi kedalam bahasa inggris, yang dilakukan dengan idiom replacement dan Google Translate, agar sesuai dengan model yang dilatih menggunakan teks berbahasa Inggris.

Setelah *preprocessing*, komentar dianalisis menggunakan model *Random Forest* untuk mengklasifikasikan komentar menjadi positif, netral, atau negatif. Hasil prediksi kemudian disimpan ke dalam tabel komentar\_sentimen\_ml, dan ditampilkan ke Admin dalam bentuk tabel hasil analisis dan visualisasi grafik pie di dashboard.

Jika tidak ada data yang tersedia untuk diproses, sistem akan memberikan notifikasi "Komentar Tidak Tersedia" sebagai umpan balik. Dengan demikian, sistem dapat menangani proses analisis dengan cerdas dan responsif, tanpa intervensi manual dari pengguna.

1. *Activity diagram* unduh data hasil analisis

Setelah proses analisis sentimen selesai dilakukan, Admin memiliki opsi untuk mengunduh hasil analisis dalam bentuk file Excel (.xlsx) secara lengkap. Fitur ini disediakan untuk memudahkan Admin dalam menyimpan, membagikan, atau melakukan analisis lanjutan di luar sistem. Proses ini hanya dapat dilakukan oleh Admin sebagai pengguna yang memiliki akses penuh, dan data yang diunduh mencakup seluruh informasi penting seperti komentar asli, hasil *preprocessing*, hasil prediksi, dan nama pengguna (username) dari komentar TikTok.



Gambar 3. *Activity diagram* admin (tombol unduh hasil analisis sentimen)

Diagram Gambar 3.13 di atas menunjukkan alur proses ketika Admin melakukan pengunduhan data hasil analisis. Sistem terlebih dahulu melakukan pengecekan terhadap tabel komentar\_sentimen\_ml untuk memastikan bahwa data hasil analisis tersedia. Jika data ditemukan, sistem akan mengekspor seluruh informasi—termasuk komentar asli, hasil preprocessing, skor sentimen, label sentimen, dan username—ke dalam file Excel atau CSV, dan mengirimkannya kepada Admin untuk diunduh.

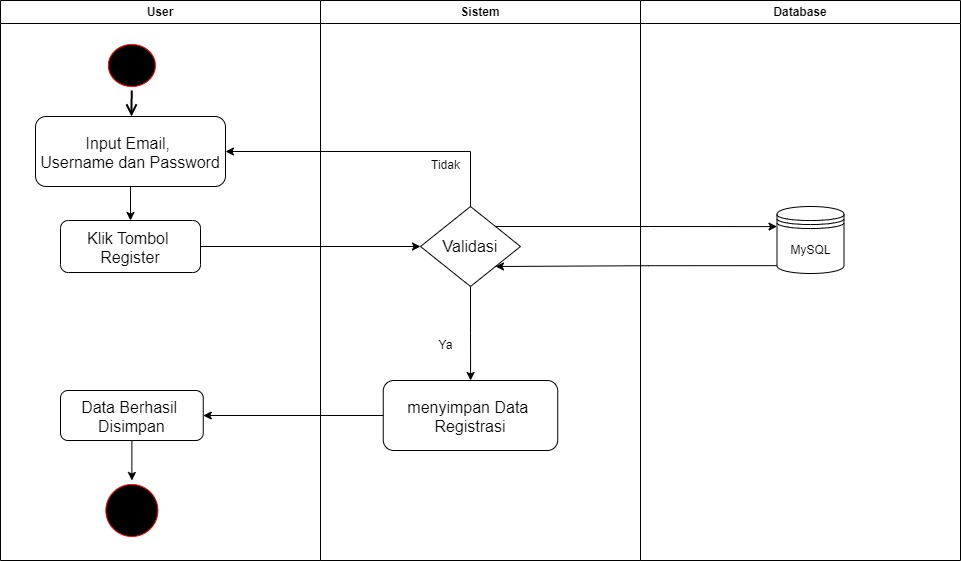
Jika tidak terdapat data di dalam database (misalnya belum dilakukan proses analisis atau scraping), sistem akan menampilkan pesan bahwa data tidak tersedia. Dengan adanya validasi kondisi ini, sistem mampu memberikan pengalaman yang lebih baik kepada pengguna, serta menghindari kesalahan unduhan file kosong. Fitur ini juga memperjelas bahwa hak akses penuh untuk mengunduh seluruh data hanya dimiliki oleh Admin.

1. *Activity Diagram* User

User merupakan pengguna publik yang hanya memiliki akses terbatas dalam sistem. Mereka dapat melihat hasil analisis sentimen dan mengunduh data dalam versi terbatas, tanpa informasi sensitif seperti username. Untuk mengakses sistem, User tetap perlu melalui proses autentikasi terlebih dahulu, sebagaimana pengguna Admin.

1. *Activity diaram* registrasi user

Proses registrasi merupakan tahap awal yang harus dilakukan oleh pengguna publik (User) sebelum dapat mengakses sistem. Melalui halaman registrasi, User diminta untuk mengisi informasi berupa email, username, dan password. Sistem kemudian akan memvalidasi data yang dimasukkan untuk memastikan bahwa seluruh informasi telah diisi dengan benar dan tidak duplikat. Jika validasi berhasil, data disimpan ke dalam basis data MySQL, dan pengguna akan mendapatkan notifikasi bahwa registrasi berhasil. Jika terjadi kesalahan atau data tidak valid, sistem akan menampilkan pesan bahwa registrasi gagal.

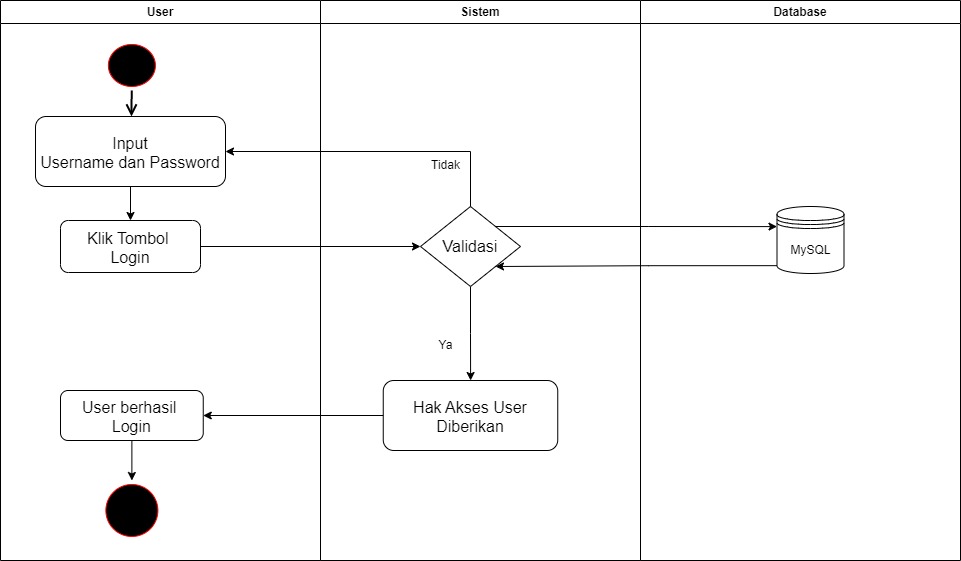


Gambar 3. *Activity diagram* user (registrasi)

Diagram Gambar 3.14 di atas memperlihatkan alur registrasi yang dilakukan oleh User. Data registrasi yang dimasukkan akan divalidasi terlebih dahulu oleh sistem, termasuk pemeriksaan kelengkapan data dan validitas email/username. Jika data valid, sistem akan menyimpan informasi tersebut ke dalam tabel users di basis data MySQL, dan memberikan notifikasi keberhasilan kepada pengguna. Sebaliknya, jika data tidak valid, sistem akan langsung memberikan notifikasi bahwa registrasi gagal. Proses ini merupakan bentuk mekanisme kontrol awal terhadap integritas pengguna yang akan mengakses sistem.

1. *Activity diagram* login user

Setelah berhasil melakukan registrasi, pengguna publik (User) dapat mengakses sistem dengan melakukan login menggunakan username dan password yang telah terdaftar. Proses login ini bertujuan untuk memastikan bahwa hanya pengguna yang sah yang dapat mengakses fitur sistem, sekaligus sebagai implementasi dari mekanisme kontrol akses berbasis peran (role-based access control). Sistem akan memverifikasi kredensial yang dimasukkan dengan mencocokkannya terhadap data yang tersimpan di dalam basis data. Jika kredensial sesuai, maka User diarahkan ke dashboard. Namun jika tidak valid, sistem akan menampilkan notifikasi kesalahan.

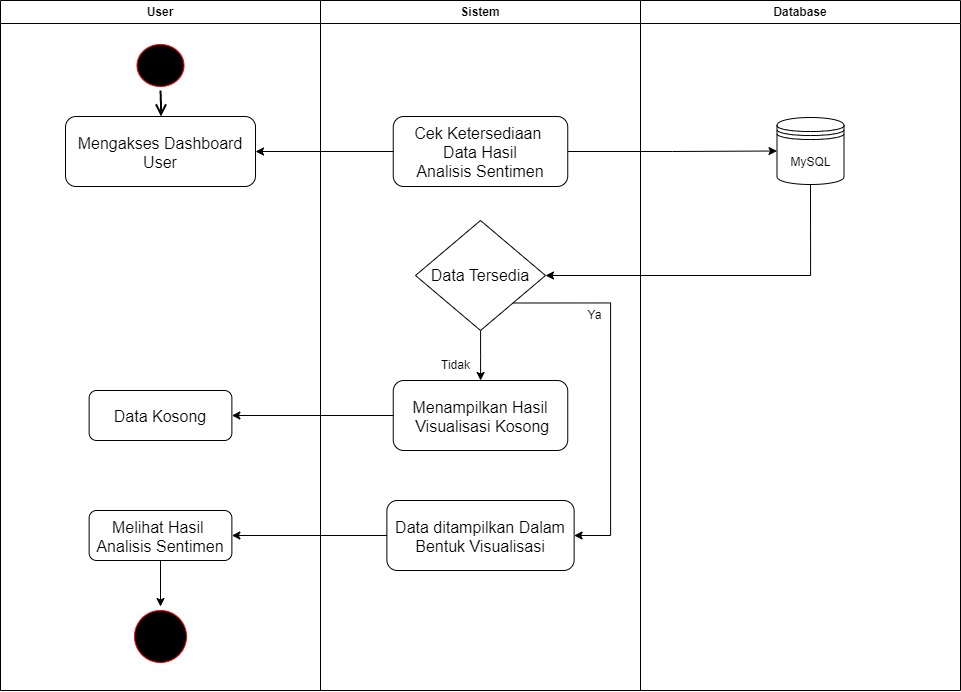


Gambar 3. *Activity diagram* user (login)

Diagram Gambar 3.15 di atas menggambarkan proses autentikasi yang dilakukan oleh User untuk dapat mengakses sistem. Sistem akan mengambil data login dari input pengguna dan memverifikasi keberadaan data tersebut dalam tabel users pada database MySQL. Jika data ditemukan dan kredensial sesuai, sistem akan mengidentifikasi peran sebagai “User”, memberikan hak akses yang sesuai, dan menampilkan pesan bahwa login berhasil. Namun, jika kredensial tidak cocok atau akun belum terdaftar, sistem akan menampilkan notifikasi bahwa login gagal. Login merupakan bagian penting dari sistem untuk menjaga keamanan dan integritas data. Selain itu, hasil login juga menentukan tampilan dan fitur yang dapat diakses, sesuai dengan peran pengguna yang terdeteksi.

1. *Activity diagram* user melihat visualisasi hasil analisis sentimen

Setelah berhasil login, User akan diarahkan ke dashboard publik yang berisi tampilan visualisasi hasil analisis sentimen dari komentar TikTok yang telah diproses oleh Admin. Visualisasi ini ditampilkan dalam bentuk grafik (misalnya pie chart) yang memudahkan pengguna memahami sebaran sentimen seperti positif, netral, atau negatif. User tidak dapat melihat komentar satu per satu maupun username, karena fitur ini dibatasi hanya untuk Admin. Akses User terbatas pada informasi agregat untuk menjaga privasi dan integritas data.



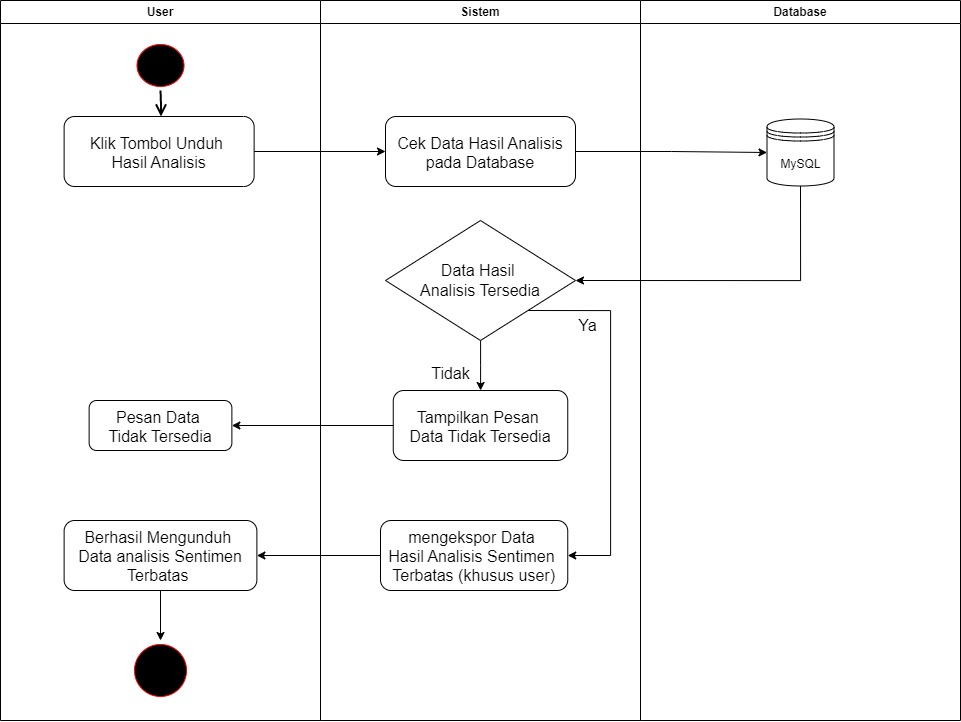
Gambar 3. *Activity diagram* user (tampilan visualisasi hasil analisis sentimen)

Diagram 3.16 di atas menunjukkan bahwa setelah login, User dapat langsung mengakses dashboard dan melihat visualisasi hasil analisis sentimen yang telah diproses sebelumnya oleh sistem. Sistem akan mengambil data ringkasan sentimen dari tabel komentar\_sentimen\_ml, seperti jumlah komentar positif, netral, dan negatif. Data ini kemudian divisualisasikan dalam bentuk grafik, biasanya pie chart atau bar chart.

Pengguna publik tidak dapat melihat username maupun skor individu dari sentimen karena keterbatasan hak akses. Hal ini diterapkan untuk menjaga keamanan dan privasi data, serta sebagai bentuk pembatasan peran antara Admin dan User.

1. *Activity diagram* unduh data hasil analisis (user)

Sistem menyediakan fitur pengunduhan hasil analisis sentimen dalam format file Excel atau CSV bagi pengguna publik (User). Namun, untuk menjaga privasi dan keamanan data, file yang diunduh oleh User tidak memuat informasi sensitif seperti username, melainkan hanya mencakup sentimen, komentar anonim, dan label klasifikasinya. Fitur ini memungkinkan pengguna untuk melakukan analisis mandiri berdasarkan data agregat tanpa melanggar prinsip keamanan data.



Gambar 3. *Activity diagram* user (unduh data)

Diagram Gambar 3.17 di atas menggambarkan alur unduh data oleh pengguna publik. Sistem akan terlebih dahulu melakukan pengecekan terhadap data yang tersedia dalam tabel komentar\_sentimen\_ml. Jika data tersedia, maka sistem menyiapkan file dalam format Excel (.xlsx) yang hanya berisi informasi umum seperti komentar anonim dan label sentimen (positif, netral, negatif). File ini kemudian diberikan kepada User untuk diunduh. Jika data tidak tersedia, sistem akan memberikan notifikasi kepada pengguna bahwa data belum dapat diunduh. Mekanisme ini tidak hanya menjaga pengalaman pengguna, tetapi juga memastikan bahwa informasi pribadi seperti nama pengguna tidak dibagikan kepada pihak yang tidak memiliki hak akses sebagai Admin.

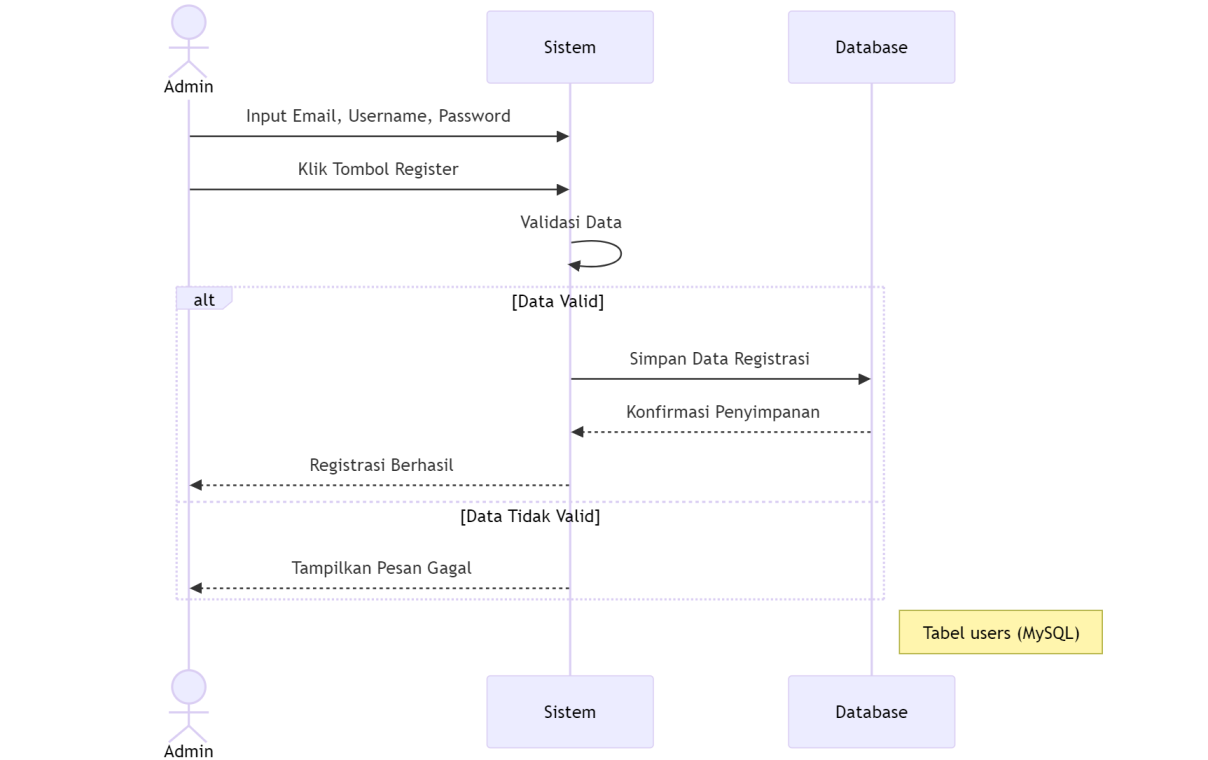
Dengan adanya *activity diagram*, setiap alur proses yang terjadi dalam sistem dapat tergambarkan secara visual dan terstruktur. Diagram ini membantu menjelaskan interaksi antara pengguna (baik Admin maupun User) dengan sistem dalam berbagai skenario, mulai dari proses registrasi, login, *scraping* komentar, analisis sentimen, hingga pengunduhan hasil. Pemisahan antara peran Admin dan User juga ditunjukkan secara jelas, di mana Admin memiliki hak akses penuh terhadap fitur-fitur sistem, sementara User hanya memiliki akses terbatas untuk melihat dan mengunduh hasil visualisasi tanpa informasi sensitif. Dengan menggambarkan seluruh aktivitas ini, perancangan sistem menjadi lebih terarah dan memudahkan dalam proses implementasi serta pengujian di tahap selanjutnya.

1. Sequence Diagram

*Sequence diagram* merupakan jenis diagram UML yang digunakan untuk menggambarkan interaksi antar objek atau komponen dalam sistem berdasarkan urutan waktu. Diagram ini berfokus pada bagaimana pesan dikirim antar entitas dalam proses menjalankan suatu fungsionalitas, termasuk aktor (pengguna) dan komponen sistem. Sequence diagram memudahkan pengembang dalam memahami bagaimana alur komunikasi berjalan ketika sistem merespons input pengguna. Pada sistem ini, sequence diagram dibagi menjadi dua bagian berdasarkan peran pengguna, yaitu Admin dan User.

1. *Sequence Diagram* Registrasi Admin

Proses registrasi Admin dilakukan melalui antarmuka backend sistem. Admin dapat mendaftarkan akun baru dengan mengisi informasi berupa email, username, dan password. Sistem akan memvalidasi data yang dimasukkan, lalu menyimpan informasi ke dalam basis data jika data valid. Proses ini bertujuan untuk memastikan bahwa hanya pengguna internal yang sah dan terverifikasi yang dapat mengakses fitur-fitur sensitif seperti scraping dan analisis sentimen.

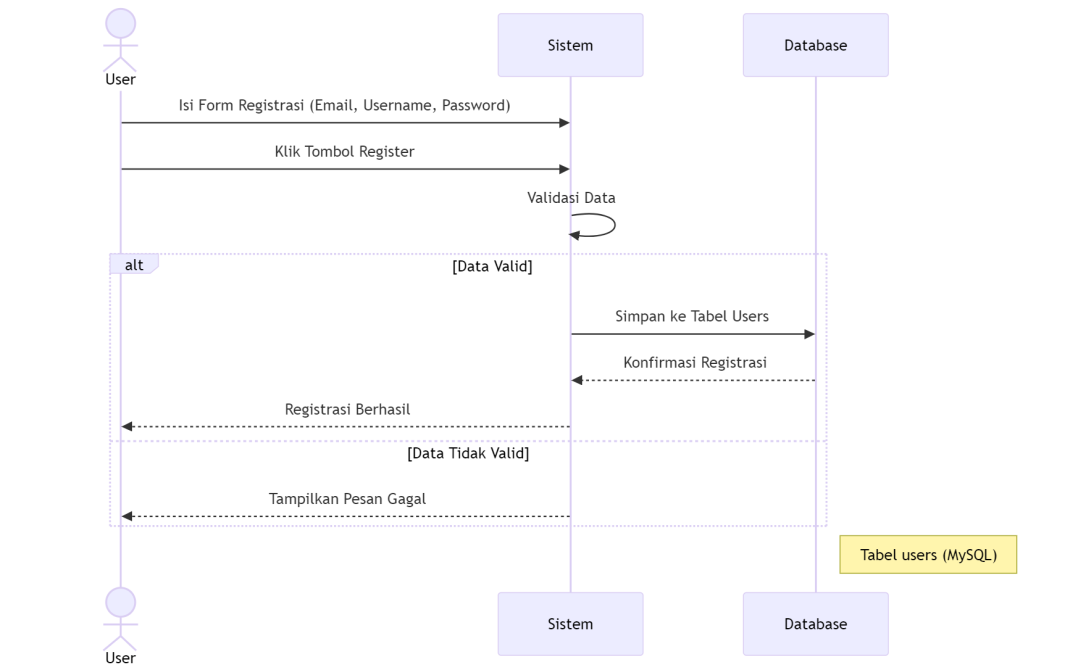


Gambar 3. *Sequence diagram* registrasi admin

Diagram 3.18 di atas menunjukkan alur registrasi Admin ke dalam sistem. Setelah Admin mengisi data registrasi dan mengirimkan permintaan ke sistem, sistem akan melakukan validasi kelengkapan dan duplikasi data. Jika data dianggap valid, sistem menyimpan informasi ke tabel users dalam database MySQL dan mengonfirmasi keberhasilan registrasi. Sebaliknya, jika terdapat kesalahan (misalnya input kosong atau username sudah terdaftar), sistem akan memberikan notifikasi kegagalan. Proses ini menjadi fondasi utama dalam pembatasan akses peran dalam sistem.

1. *Sequence diagram* registrasi user

Registrasi User dilakukan melalui halaman publik sistem. Pengguna diminta mengisi form registrasi dengan email, username, dan password. Setelah itu, sistem akan memvalidasi data yang dimasukkan. Jika data valid dan belum pernah terdaftar, informasi pengguna akan disimpan ke dalam basis data. Setelah registrasi berhasil, pengguna dapat melanjutkan proses login untuk mengakses dashboard hasil analisis publik. Proses ini menjadi bagian dari kontrol akses sistem untuk membedakan antara pengguna publik dan pengguna internal (Admin).

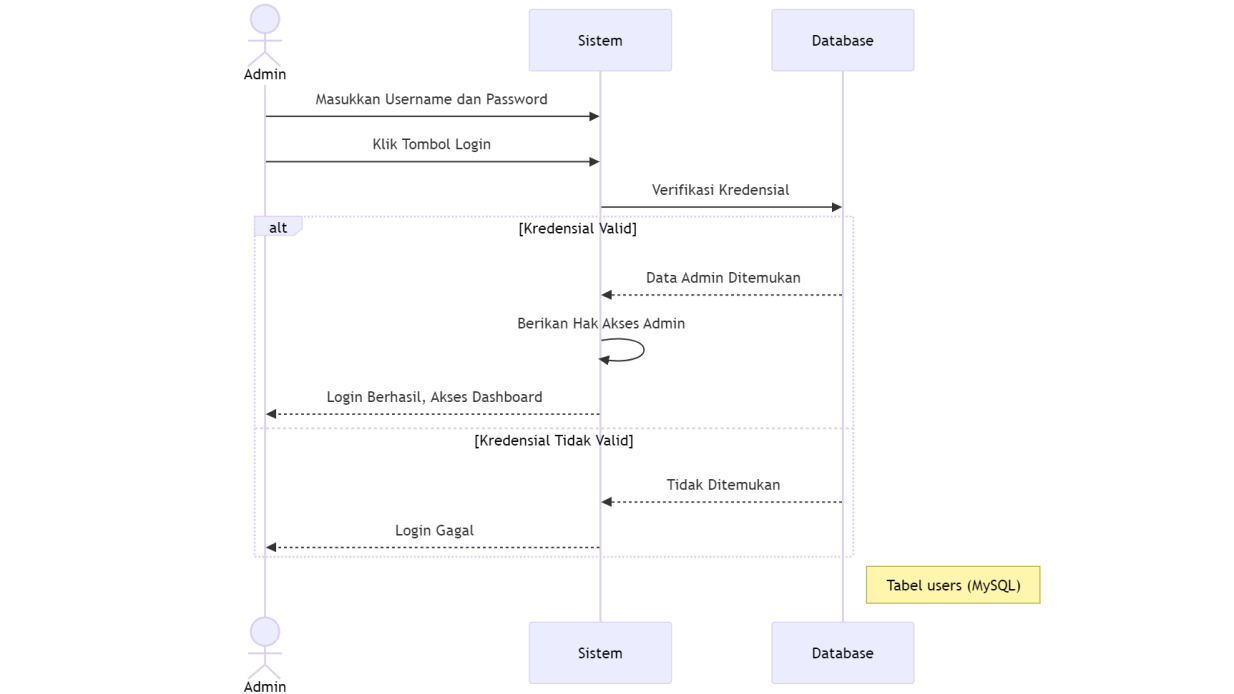


Gambar 3. *Sequence diagram* registrasi user

Diagram Gambar 3.19 di atas menunjukkan bahwa User mengisi form registrasi melalui tampilan publik. Setelah data dikirim, sistem akan mengecek validitas seperti panjang password, format email, dan duplikasi username. Jika lolos, data disimpan ke dalam tabel users pada database MySQL dan sistem memberikan notifikasi keberhasilan. Jika gagal, sistem akan menampilkan pesan kesalahan seperti "Email sudah terdaftar" atau "Password terlalu pendek". Dengan registrasi ini, sistem dapat membatasi akses User agar hanya melihat fitur-fitur yang sesuai dengan perannya, seperti visualisasi hasil sentimen dan unduhan data terbatas.

1. *Sequence diagram* login admin

Proses login Admin dilakukan melalui halaman khusus backend sistem. Admin memasukkan username dan password yang telah terdaftar sebelumnya. Sistem akan memverifikasi data tersebut dengan mencocokkannya pada tabel users di basis data. Jika login berhasil, sistem akan mengidentifikasi peran sebagai Admin dan menampilkan fitur dashboard secara lengkap, termasuk akses ke fitur scraping, analisis sentimen, dan pengelolaan data. Jika kredensial tidak valid, sistem akan menampilkan pesan kesalahan.

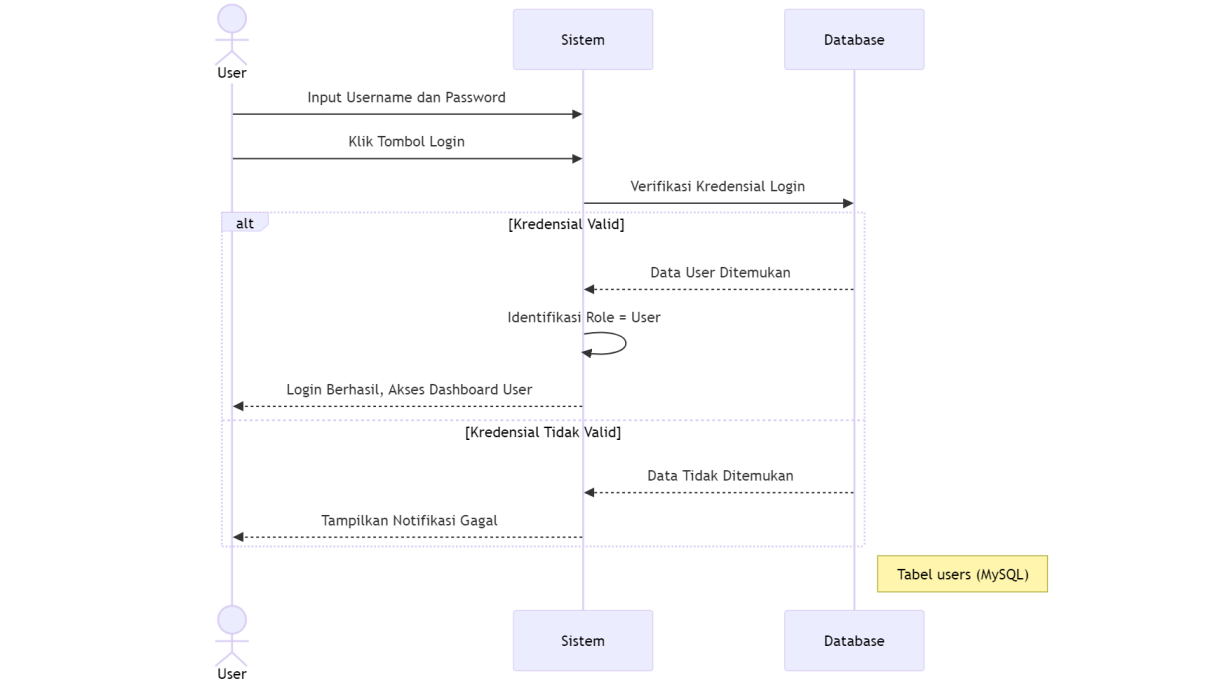


Gambar 3.  *Sequence diagram* login admin

Diagram Gambar 3.20 di atas menggambarkan alur autentikasi Admin. Setelah Admin memasukkan data login, sistem melakukan pengecekan pada database. Jika data valid dan peran Admin terverifikasi, maka sistem mengarahkan Admin ke dashboard dengan tampilan fitur penuh. Jika login gagal karena kredensial salah atau akun tidak terdaftar, sistem menampilkan pesan kesalahan. Proses ini merupakan bagian dari pengamanan sistem dan kontrol akses, yang memastikan bahwa fitur-fitur penting hanya dapat dijalankan oleh pihak yang memiliki otoritas sesuai peran.

1. *Sequence diagram* login user

Login User dilakukan melalui halaman publik sistem setelah pengguna berhasil melakukan registrasi. Pengguna memasukkan username dan password, lalu sistem akan memverifikasi kredensial tersebut ke dalam basis data. Jika informasi valid, sistem akan memberikan hak akses User dan menampilkan dashboard publik berisi visualisasi hasil analisis sentimen. Namun, jika login gagal, sistem akan memberikan notifikasi bahwa login tidak berhasil. Proses ini juga menjadi bagian dari implementasi kontrol akses berbasis peran (*role-based access control*).

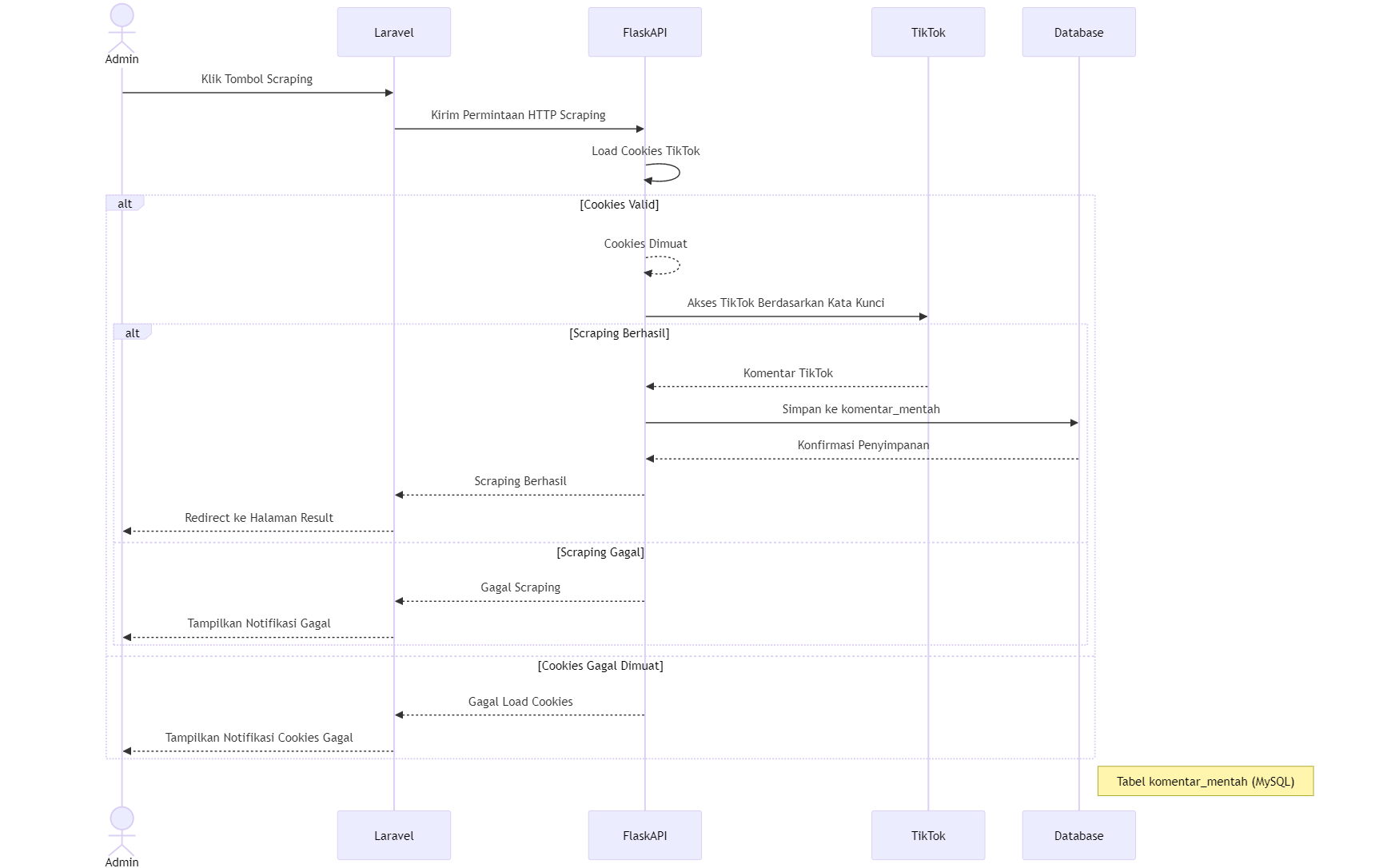


Gambar 3. *Sequence diagram* login user

Diagram Gambar 3.21 ini menunjukkan bahwa proses login User hampir sama dengan login Admin, namun hak akses yang diberikan berbeda. Setelah verifikasi berhasil, sistem hanya akan menampilkan dashboard dengan fitur terbatas (visualisasi dan unduhan hasil tanpa data sensitif). Jika kredensial salah atau tidak ditemukan di database, sistem akan memberikan peringatan login gagal. Login ini bertujuan untuk membedakan akses pengguna dan menjaga keamanan sistem secara menyeluruh.

1. *Sequence diagram scraping* komentar Tiktok (admin)

*Scraping* komentar TikTok dilakukan oleh Admin melalui dashboard sistem. Setelah tombol "Scraping" ditekan, sistem Laravel akan mengirim permintaan HTTP ke Flask API. Flask akan mencoba memuat cookies TikTok yang telah disiapkan sebelumnya. Jika *cookies* valid, maka proses scraping dimulai berdasarkan kata kunci tertentu. Komentar yang berhasil diambil akan disimpan ke database. Namun, jika *cookies* gagal dimuat atau scraping tidak berhasil, sistem akan memberikan notifikasi kegagalan kepada Admin. Proses ini mencerminkan bagaimana sistem menangani keberhasilan maupun kegagalan scraping secara otomatis dan responsif.



Gambar 3. *Sequence diagram scraping* komentar Tiktok (admin)

Gambar 3.22 di atas menunjukkan urutan proses *scraping* komentar TikTok yang dilakukan oleh Admin melalui antarmuka Laravel. Setelah tombol "Scraping" ditekan, sistem Laravel mengirim permintaan HTTP ke Flask API. Tahapan pertama yang dilakukan oleh Flask adalah memuat file *cookies* TikTok dalam format .json. File *cookies* ini penting karena berfungsi sebagai autentikasi agar dapat mengakses halaman TikTok tanpa login manual secara real-time.

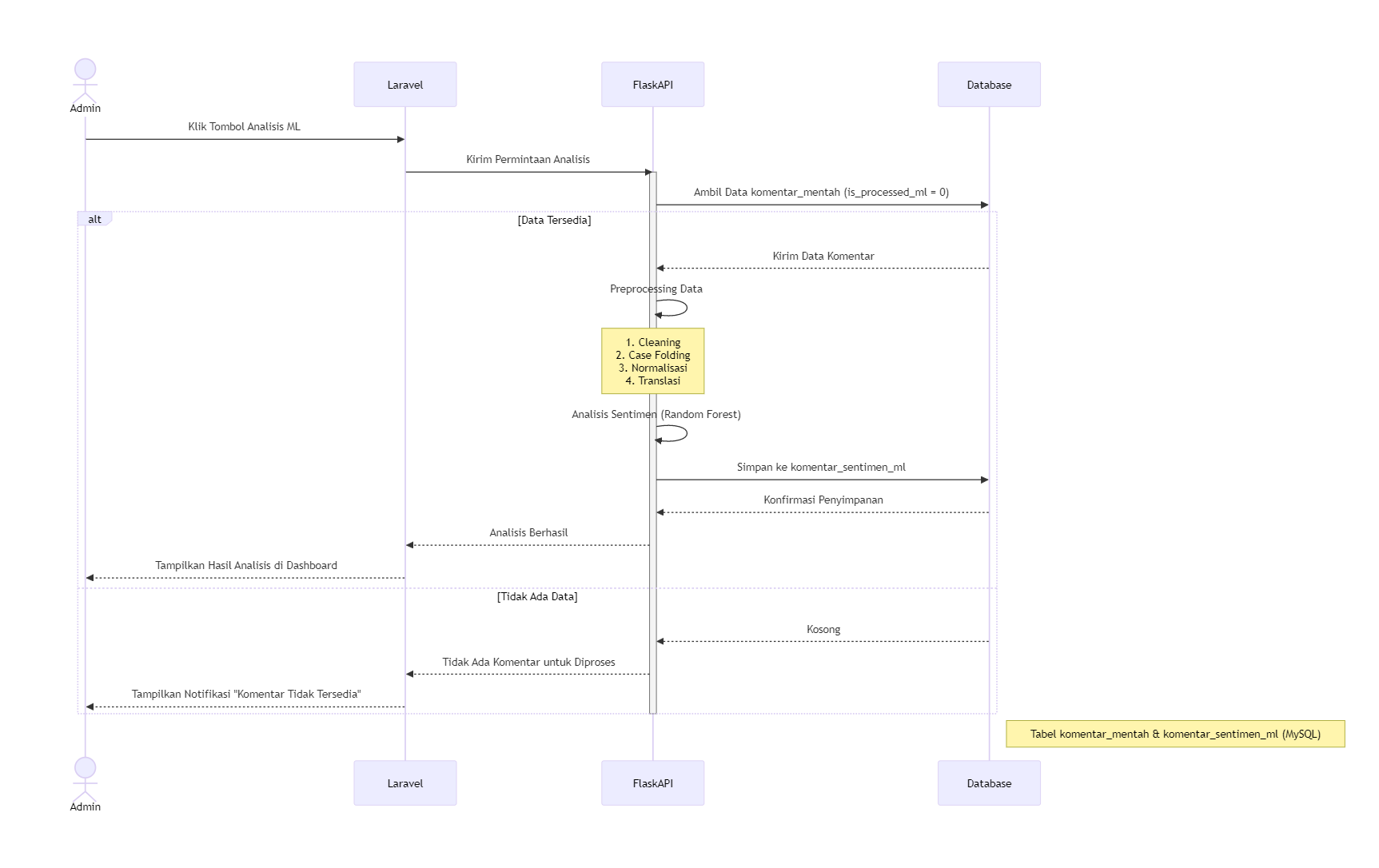
Flask kemudian melakukan pengecekan terhadap validitas cookies. Jika cookies berhasil dimuat, maka proses *scraping* dilanjutkan dengan membuka halaman TikTok berdasarkan kata kunci tertentu, seperti "kejaksaan agung" atau "kinerja kejaksaan agung". Selanjutnya, script akan mengekstrak komentar dari video-video yang ditemukan. Namun, jika *cookies* tidak valid atau gagal dimuat, sistem langsung menghentikan proses dan mengirimkan notifikasi "Gagal Memuat Cookies" kepada Admin melalui Laravel.

Apabila *scraping* berhasil, komentar-komentar yang diperoleh akan disimpan ke dalam tabel komentar\_mentah di database MySQL. Sistem memastikan bahwa proses penyimpanan dilakukan dengan aman dan akan memberikan respons berhasil jika penyimpanan data tidak mengalami kendala. Namun, apabila proses *scraping* gagal — misalnya karena TikTok gagal dimuat, koneksi terputus, atau komentar tidak tersedia — maka sistem akan langsung memberikan notifikasi "Scraping Gagal" kepada Admin dan tidak melakukan penyimpanan data.

Setelah proses selesai, Admin akan diarahkan ke halaman "Result" untuk melihat komentar yang berhasil diambil. Proses *scraping* ini tidak hanya bersifat otomatis dan responsif, tetapi juga dilengkapi dengan penanganan error (*error handling*) di setiap tahapannya, mulai dari pemuatan cookies hingga penyimpanan data. Hal ini membuat sistem lebih stabil dan dapat diandalkan dalam skenario dunia nyata, di mana kemungkinan gagal *scraping* sangat mungkin terjadi.

1. *Sequence diagram* analisis sentimen *machine learning* (admin)

Proses analisis sentimen dilakukan oleh Admin melalui tombol “Analisis ML” di dashboard. Saat tombol ditekan, Laravel akan mengirim permintaan HTTP ke Flask API. Flask kemudian mengambil data komentar dari database yang belum diproses oleh model *Machine Learning* (is\_processed\_ml = 0). Setelah data diambil, Flask menjalankan tahapan *preprocessing* yang terdiri dari pembersihan teks, normalisasi kata, translasi ke bahasa Inggris, hingga vektorisasi. Data yang sudah diproses lalu dianalisis menggunakan model Random Forest. Hasil klasifikasi (positif, negatif, netral) disimpan ke dalam tabel komentar\_sentimen\_ml. Jika tidak ada data yang tersedia untuk diproses, atau terjadi error saat pemrosesan, sistem akan memberikan notifikasi kegagalan kepada Admin.



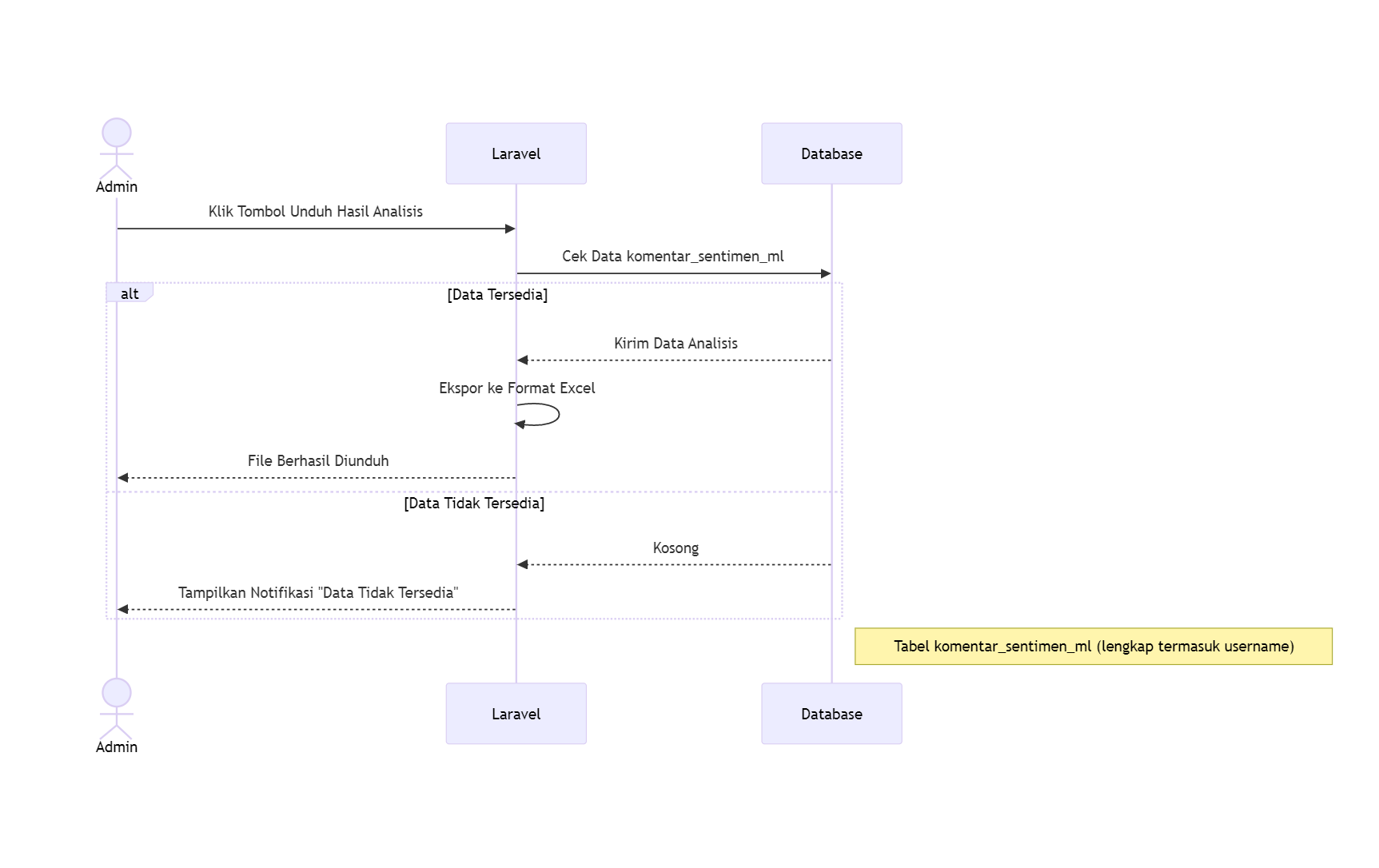
Gambar 3. *Sequence diagram* analisis sentimen ML (admin)

Diagram Gambar 3.23 di atas menggambarkan alur proses analisis sentimen yang dilakukan oleh Admin melalui sistem. Setelah Admin menekan tombol "Analisis ML" di dashboard, sistem Laravel akan mengirimkan permintaan HTTP ke Flask API. Flask kemudian mengambil data komentar mentah dari database yang belum diproses oleh model *Machine Learning*, ditandai dengan status “is\_processed\_ml = 0”. Jika data tersedia, Flask akan menjalankan serangkaian tahapan *preprocessing* sebelum dilakukan analisis.

Tahapan *preprocessing* terdiri dari beberapa langkah utama, yaitu: *cleaning* (menghapus karakter tidak relevan seperti URL, mention, dan simbol), *case folding* (mengubah seluruh huruf menjadi huruf kecil), normalisasi kata menggunakan kamus normalisasi berbasis JSON, serta translasi komentar ke bahasa Inggris agar sesuai dengan model *Random Forest* yang telah dilatih sebelumnya. Seluruh tahapan ini bertujuan untuk memastikan bahwa teks yang dianalisis bersih, terstandarisasi, dan dapat dikenali oleh algoritma pembelajaran mesin. Setelah *preprocessing* selesai, data komentar akan diubah ke bentuk numerik menggunakan TF-IDF dan kemudian dianalisis menggunakan model klasifikasi *Random Forest*. Model ini akan menghasilkan label sentimen untuk setiap komentar, yaitu positif, netral, atau negatif. Hasil prediksi tersebut, bersama dengan informasi tambahan seperti komentar awal, komentar yang telah diproses, dan waktu proses, disimpan ke dalam tabel “komentar\_sentimen\_ml”. Jika seluruh proses berhasil, sistem akan memberikan notifikasi sukses dan menampilkan hasil analisis pada dashboard Admin. Namun, jika tidak ada data yang tersedia untuk diproses, sistem akan menginformasikan kepada Admin bahwa tidak ada komentar baru yang dapat dianalisis. Dengan alur ini, sistem dirancang tidak hanya untuk akurasi analisis, tetapi juga untuk memberikan respon yang adaptif terhadap kondisi data dan menjaga keandalan proses secara keseluruhan.

1. *Sequence diagram* unduh hasil analisis sentimen (admin)

Fitur ini memungkinkan Admin mengunduh seluruh data hasil analisis sentimen yang telah diproses oleh model Machine Learning. Proses dimulai saat Admin menekan tombol "Unduh Hasil" di dashboard. Sistem Laravel akan memeriksa ketersediaan data hasil analisis dari database. Jika data tersedia, sistem mengekspor data ke format Excel atau CSV yang berisi komentar, label sentimen, username, serta informasi lainnya. Jika data tidak tersedia, sistem akan memberikan notifikasi bahwa data belum tersedia untuk diunduh. Fitur ini hanya tersedia untuk Admin, guna menjaga integritas dan privasi informasi pengguna.

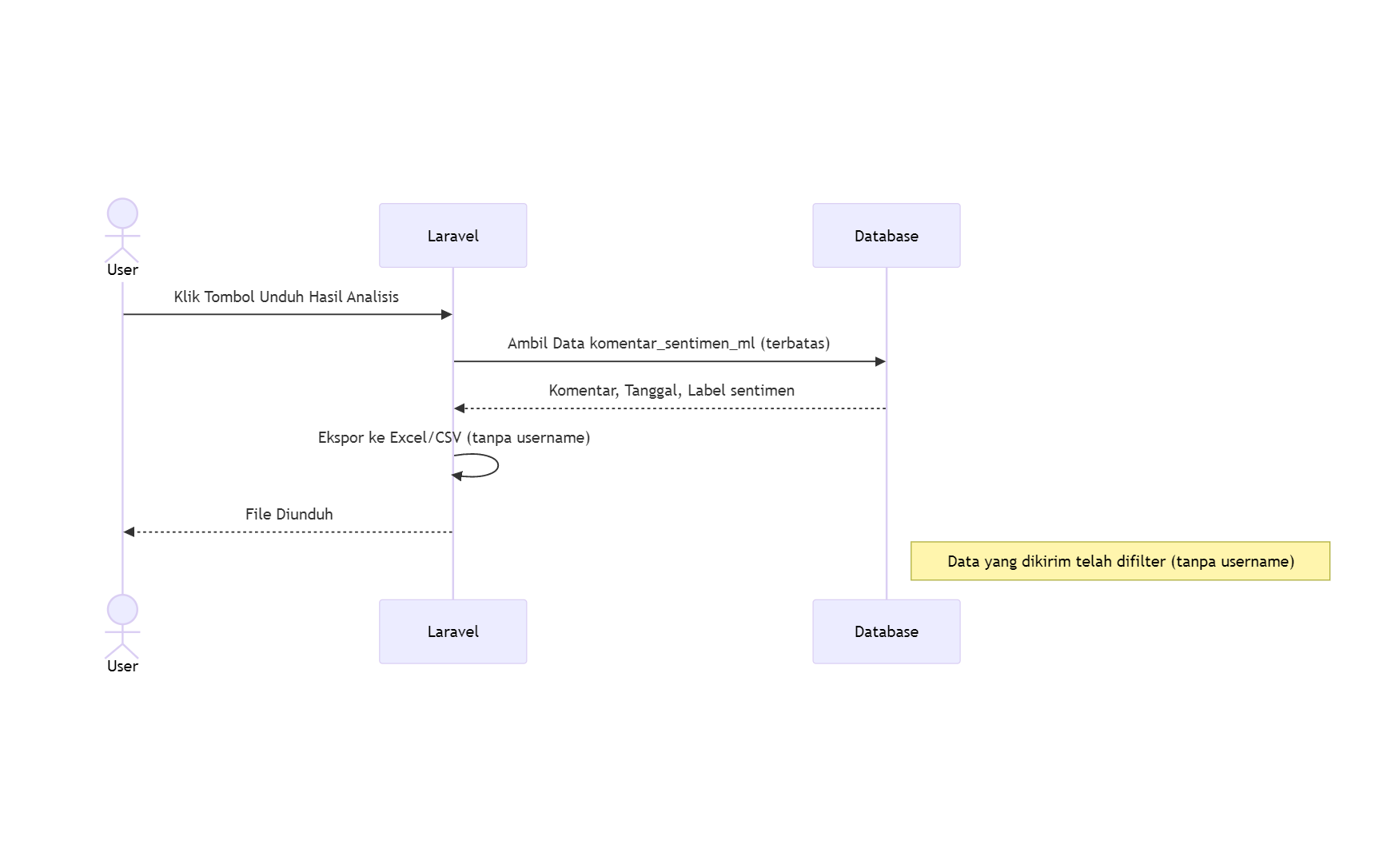


Gambar 3. *Sequence diagram* unduh hasil analisis sentimen (admin)

Diagram Gambar 3.24 di atas menggambarkan alur interaksi sistem saat Admin melakukan pengunduhan data hasil analisis sentimen. Ketika tombol “Unduh Hasil” ditekan, sistem akan terlebih dahulu mengecek ketersediaan data di tabel “komentar\_sentimen\_ml”. Jika data tersedia, sistem Laravel akan mengambil seluruh informasi yang relevan, termasuk username, komentar asli, hasil *preprocessing*, serta label sentimen yang telah diprediksi oleh model *Machine Learning*. Data tersebut kemudian diekspor ke dalam format file .xlsx dan diberikan kepada Admin sebagai file unduhan. Proses ekspor ini dilakukan secara otomatis tanpa intervensi manual tambahan. Sebaliknya, jika sistem mendeteksi bahwa belum ada data hasil analisis yang tersedia, maka sistem akan menampilkan notifikasi kepada Admin bahwa data belum bisa diunduh. Fitur ini menjadi salah satu alat bantu utama Admin dalam melakukan evaluasi atau pelaporan hasil analisis, dan dirancang agar fleksibel namun tetap menjaga kontrol akses — karena hanya Admin yang memiliki hak untuk melihat dan mengunduh data secara lengkap.

1. *Sequence diagram* melihat visualisasi hasil analisis sentimen (user)

Fitur ini memungkinkan pengguna dengan peran User untuk mengunduh hasil analisis sentimen dalam format .xlsx. File yang diunduh berisi informasi terbatas yang sama seperti yang ditampilkan pada tabel di halaman dashboard User, yaitu: komentar, waktu komentar, dan label sentimen. Informasi seperti username, ID video, jumlah suka, jumlah replay maupun hasil *preprocessing* tidak disertakan untuk menjaga privasi data. Fitur ini bertujuan memberikan transparansi data kepada publik tanpa membuka informasi sensitif.



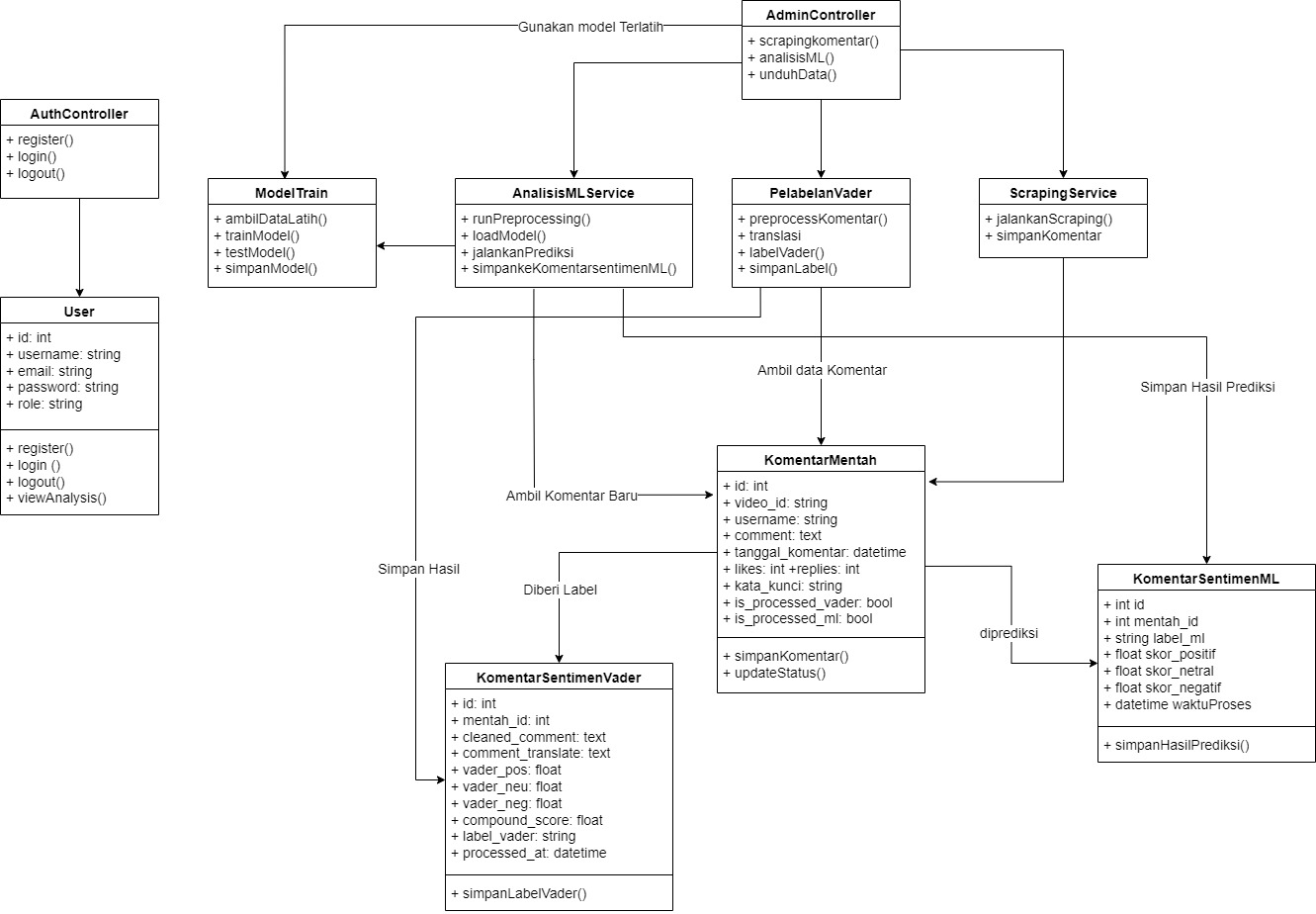
Gambar 3. *Sequence diagram* melihat visualisasi hasil analisis sentimen (user)

Diagram Gambar 3.25 di atas memperlihatkan alur sistem ketika User menekan tombol "Unduh Hasil" di dashboard publik. Sistem Laravel akan mengambil data dari tabel “komentar\_sentimen\_ml”, namun hanya dengan kolom yang diperbolehkan untuk ditampilkan ke User, yaitu: komentar, waktu komentar, dan label sentimen. Setelah data diperoleh, Laravel akan memprosesnya menjadi file .xlsx lalu memberikan file tersebut kepada pengguna sebagai hasil unduhan. Fitur ini merupakan bentuk keterbukaan hasil analisis sistem kepada publik, sekaligus bentuk pengendalian akses data. Informasi sensitif seperti nama pengguna TikTok, video ID, jumlah suka, jumlah replay bahkan skor prediksi model tidak disertakan dalam file. Dengan demikian, sistem tetap memberikan manfaat analisis kepada publik tanpa melanggar prinsip privasi dan keamanan data.

Hal ini dilakukan sebagai bentuk perlindungan privasi dan bagian dari implementasi *role-based access control*, di mana pengguna hanya mendapatkan akses terbatas sesuai haknya. Fitur unduhan ini memungkinkan User mendapatkan dokumentasi hasil analisis secara personal, namun tetap menjaga batasan etis dalam distribusi data hasil *scraping* dan klasifikasi sentimen.

1. Class Diagram

*Class diagram* merupakan representasi visual dari struktur sistem dalam bentuk kelas-kelas (class) beserta atribut, method, serta relasi antar objek yang saling berinteraksi. Diagram ini digunakan untuk menggambarkan arsitektur sistem secara statis, sehingga memudahkan pemahaman terhadap peran masing-masing komponen serta alur data yang terjadi di antara mereka. Pada penelitian ini, class diagram disusun untuk merepresentasikan keseluruhan proses mulai dari registrasi, login, scraping komentar, analisis sentimen dengan VADER dan Machine Learning, hingga penyimpanan hasil ke dalam database. Class diagram ini menjadi fondasi penting dalam proses pengembangan sistem, karena berfungsi sebagai blueprint dalam implementasi program menggunakan Laravel, Flask, dan Python.



Gambar 3. *Class diagram*

Berdasarkan Gambar 3.26 di atas, terdapat beberapa class utama yang saling terhubung sesuai dengan fungsi dan proses dalam sistem. Class User mewakili pengguna sistem yang dapat melakukan registrasi, login, logout, dan melihat hasil analisis. Proses autentikasi diatur melalui class AuthController yang memiliki method register(), login(), dan logout() untuk memastikan keamanan akses pengguna.

Selanjutnya, class AdminController bertindak sebagai pengontrol utama untuk admin, dengan method seperti scrapingKomentar(), analisisML(), dan unduhData(). Proses scraping dilakukan melalui class ScrapingService yang bertugas menjalankan proses pengambilan komentar dari TikTok menggunakan Selenium serta menyimpannya ke dalam class Komentar\_mentah.

Komentar yang telah terkumpul kemudian diproses dan dilabeli menggunakan class PelabelanVader, yang mencakup tahap-tahap penting seperti preprocessKomentar(), translasi, serta pelabelan sentimen menggunakan VADER. Hasil pelabelan disimpan ke dalam class KomentarSentimenVader yang menyimpan skor sentimen (positif, netral, negatif, dan compound) beserta label klasifikasinya. Data dari pelabelan ini digunakan oleh class ModelTrain untuk melatih model *Machine Learning* dengan metode *Random Forest*. Class ini menjalankan method trainModel() dan testModel() lalu menyimpan model terlatih untuk digunakan pada prediksi selanjutnya.

Prediksi komentar baru dilakukan oleh class AnalisisMLService, yang memanfaatkan model terlatih dari ModelTrain. Komentar yang belum diproses diambil dari class Komentar\_mentah, kemudian dilakukan *preprocessing*, pemuatan model, dan prediksi.

Seluruh relasi antara class dijelaskan secara terstruktur, seperti satu komentar pada Komentar\_mentah memiliki satu hasil label di KomentarSentimenVader, beserta satu hasil prediksi pada KomentarSentimenML. Relasi antar class telah diberi keterangan seperti "ambil data komentar", "diberi label", "diprediksi", dan "simpan hasil", sehingga memberikan gambaran utuh terhadap interaksi antar komponen dalam sistem. Dengan demikian, class diagram ini mampu menjelaskan secara lengkap alur pengolahan data dari awal hingga akhir pada sistem analisis sentimen berbasis Laravel dan Flask.

1. *Entity Relationship Diagram* (*ERD*)

*Entity Relationship Diagram* (ERD) digunakan untuk menggambarkan hubungan antar entitas dalam basis data sistem yang dibangun. ERD merupakan alat penting dalam proses perancangan database karena dapat membantu mengidentifikasi entitas utama, atribut yang dimiliki, serta relasi yang terjadi antar entitas tersebut. Pada penelitian ini, ERD disusun untuk memodelkan struktur penyimpanan data hasil *scraping* TikTok, proses pelabelan sentimen menggunakan VADER, dan hasil prediksi menggunakan model *Machine Learning* yaitu *random forest*. Diagram ini mencerminkan integrasi sistem *backend* yang melibatkan Laravel, Flask API, serta basis data MySQL sebagai penyimpan utama.



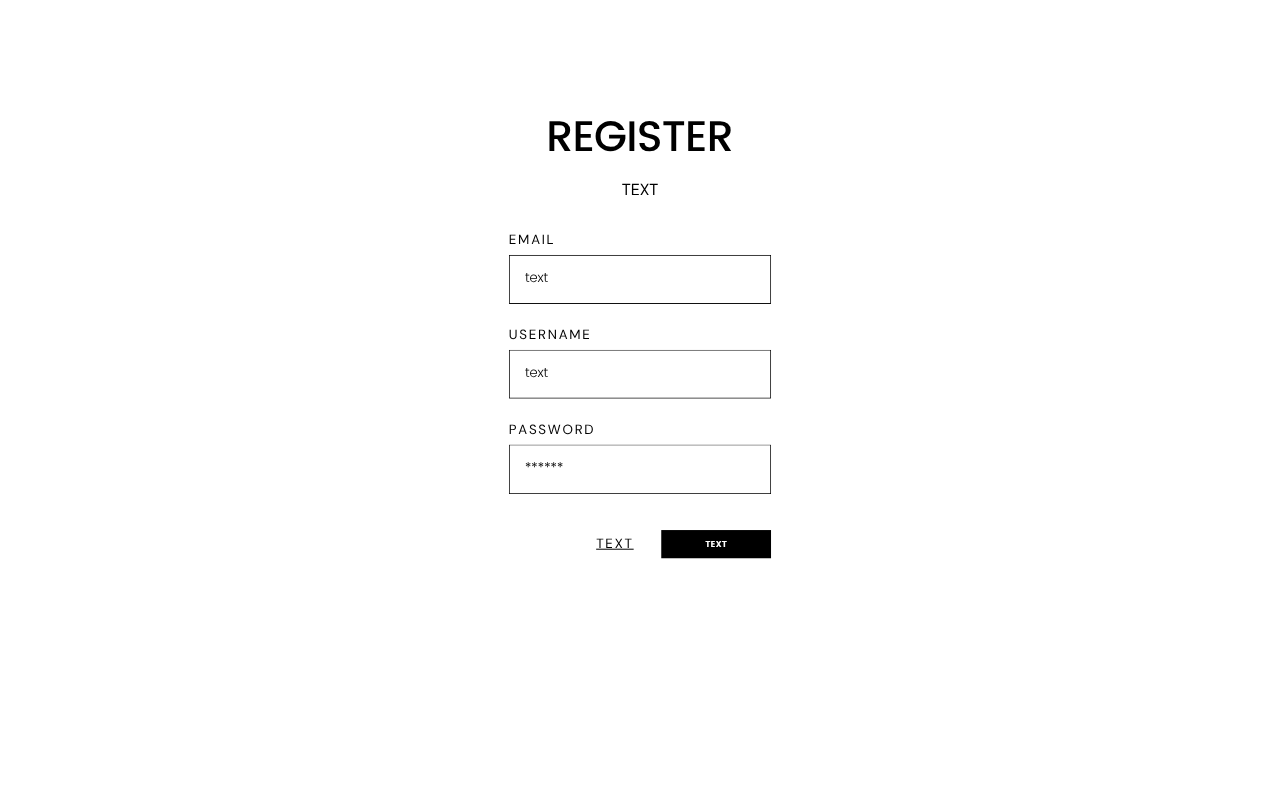
Gambar 3. *Entity Relationship Diagram* (ERD)

Berdasarkan Gambar 3.27 di atas, terdapat empat entitas utama dalam sistem, yaitu users, komentar\_mentah, komentar\_sentimen\_vader, dan komentar\_sentimen\_ml. Entitas users merepresentasikan pengguna sistem yang terdiri dari admin dan user biasa. Atribut yang dimiliki oleh entitas ini meliputi name, email, password, dan role, yang digunakan untuk mengatur hak akses terhadap fitur sistem. Walaupun tidak memiliki foreign key langsung ke entitas komentar, entitas ini secara logis memiliki relasi 1-to-many terhadap komentar\_mentah, karena hanya admin yang dapat memicu proses *scraping* data dan mengakses data mentah. Entitas komentar\_mentah menyimpan hasil komentar dari TikTok yang diperoleh melalui proses *scraping* otomatis. Data yang disimpan mencakup video\_id, kata\_kunci, username, comment, likes, replies, dan tanggal\_komentar. Terdapat pula atribut boolean is\_processed\_vader dan is\_processed\_ml yang berfungsi sebagai penanda status pemrosesan komentar tersebut dalam pipeline VADER maupun *Machine Learning*. Selanjutnya, entitas komentar\_sentimen\_vader merepresentasikan hasil pelabelan komentar menggunakan pendekatan VADER. Setiap data di sini merujuk ke komentar awal melalui atribut mentah\_id yang berperan sebagai foreign key. Atribut lainnya seperti cleaned\_comment, comment\_translate, vader\_pos, vader\_neu, vader\_neg, dan compound\_score digunakan untuk menyimpan skor sentimen hasil analisis VADER. Label klasifikasi disimpan dalam bentuk enum label\_vader. Sedangkan entitas komentar\_sentimen\_ml menyimpan hasil klasifikasi sentimen dari model *Machine Learning* (*Random Forest*). Sama seperti sebelumnya, komentar mentah dihubungkan melalui mentah\_id, dan prediksi disimpan pada field predicted\_label beserta confidence\_score. Field tambahan seperti cleaned\_comment dan comment\_translate digunakan untuk menyimpan teks setelah *preprocessing* sebelum diproses oleh model. Relasi antar entitas ditunjukkan dengan garis koneksi yang dilengkapi keterangan, seperti “memiliki”, “diprediksi”, dan “mengakses”, yang bertujuan untuk memberikan pemahaman konteks antar tabel. ERD ini menjadi representasi penting dari keseluruhan alur data pada sistem, mulai dari pengumpulan data mentah hingga penyimpanan hasil analisis yang ditampilkan dalam aplikasi berbasis web Laravel.

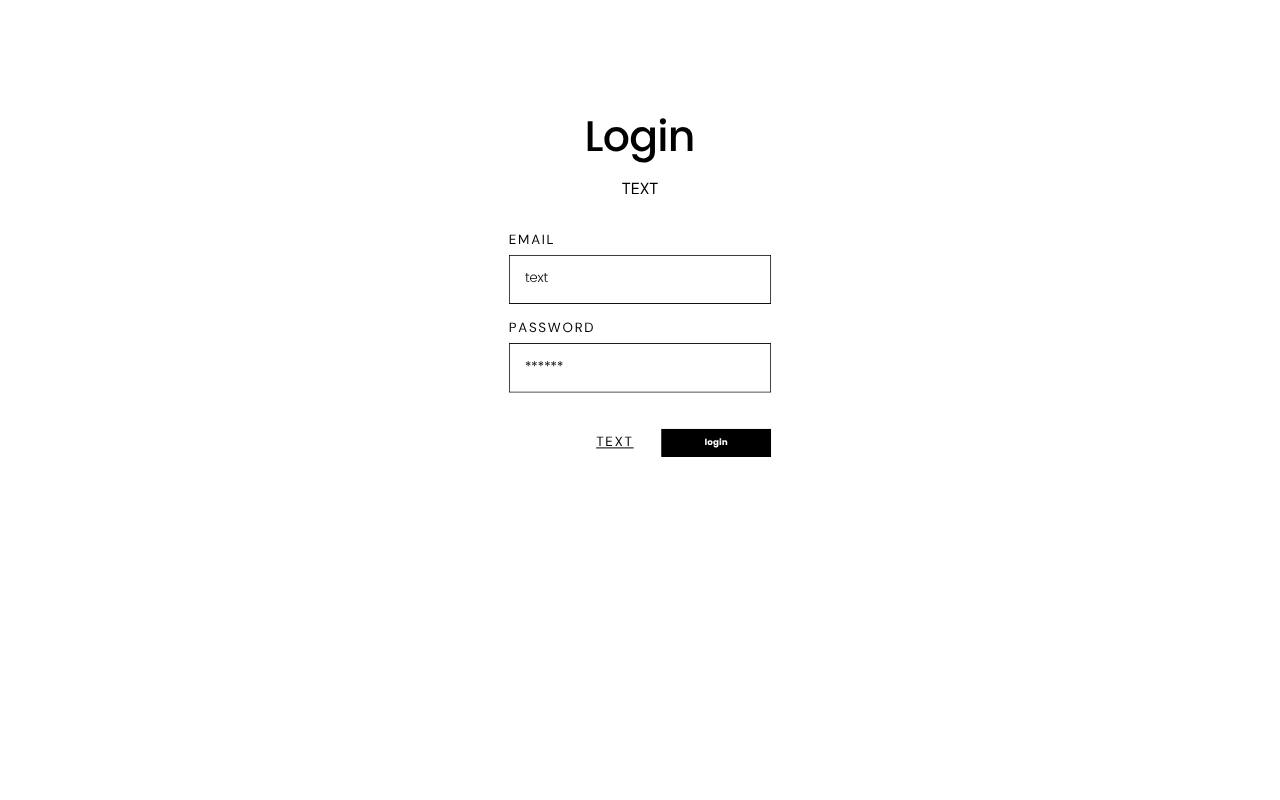
1. Perancangan Antar Muka Pengguna

Perancangan antarmuka pengguna (*User Interface*/UI) pada sistem ini berorientasi pada penyediaan pengalaman interaksi yang mudah diakses, intuitif, dan adaptif terhadap kebutuhan pengguna, baik untuk peran administrator maupun pengguna umum. Pendekatan desain UI mengedepankan kesederhanaan tanpa mengorbankan kelengkapan informasi, guna memfasilitasi efisiensi akses dan pengelolaan fitur sistem.

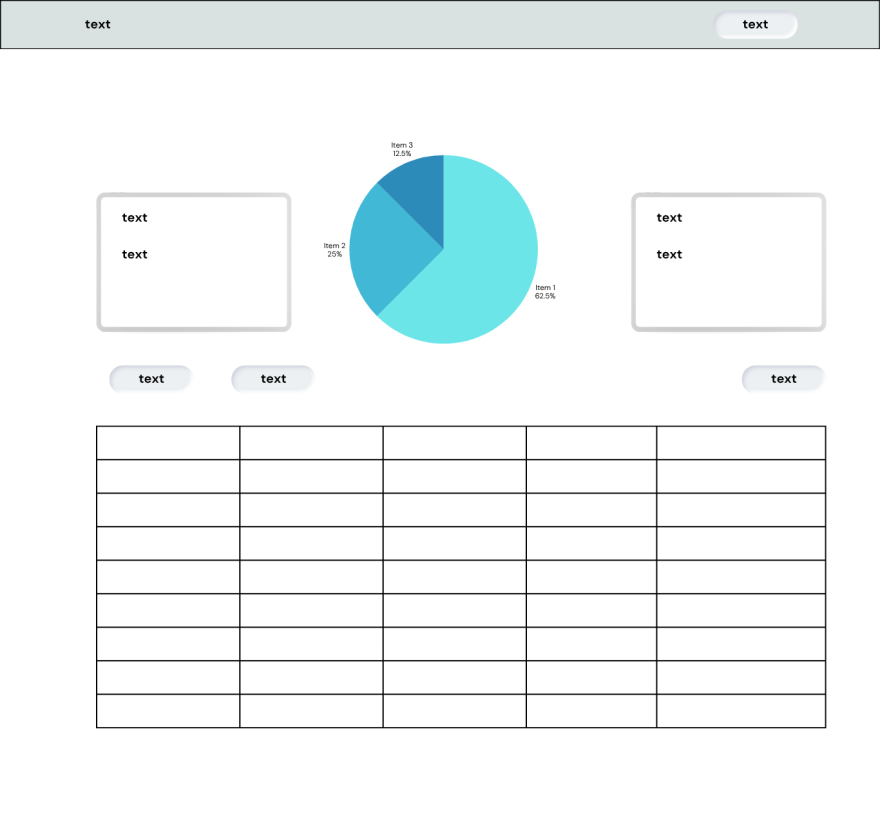
Komponen antarmuka yang dirancang meliputi halaman registrasi dan halaman login sebagai pintu masuk sistem. Untuk peran administrator, disediakan tampilan dasbor yang mengintegrasikan fitur pengambilan data (*scraping*), analisis sentimen, dan visualisasi data. Selanjutnya, halaman hasil *scraping* (result) berfungsi untuk menampilkan data komentar mentah yang berhasil dihimpun dari platform TikTok. Seluruh elemen UI dirancang secara kohesif untuk mendukung alur kerja sistem yang telah ditetapkan, dengan mempertimbangkan aspek fungsionalitas dan kemudahan navigasi.



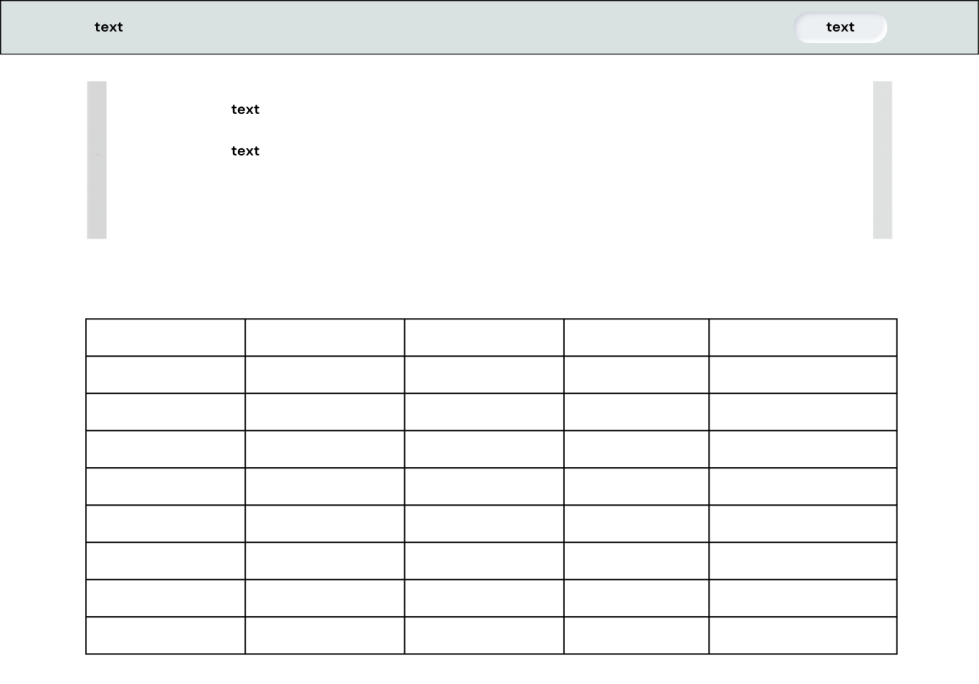
Gambar 3. *User interface* (register)



Gambar 3. *User interface* (login)



Gambar 3. *User interface* (dashboard)



Gambar 3. *User interface* (result)

Gambar 3.28, Gambar 3.29, Gambar 3.30 dan Gambar 3.31 di atas mengilustrasikan beberapa komponen esensial antarmuka sistem. Dimulai dengan halaman registrasi, pengguna baru dapat mendaftarkan akun melalui formulir input yang mencakup nama pengguna, alamat email, dan kata sandi. Setelah itu, halaman login berfungsi sebagai gerbang autentikasi, memvalidasi kredensial untuk membedakan hak akses antara administrator dan pengguna biasa sebelum mereka memasuki sistem.

Pasca-login, pengguna akan diarahkan ke dasbor yang disesuaikan dengan peran mereka. Dasbor administrator dilengkapi dengan tombol untuk menjalankan proses *scraping* data komentar TikTok dan tombol "Analisis ML" yang akan memicu proses analisis sentimen melalui Flask API. Selain itu, sistem menyajikan visualisasi hasil sentimen dalam bentuk grafik dan tabel. Terakhir, halaman "Result" secara khusus menampilkan komentar-komentar hasil scraping dari TikTok, memungkinkan administrator untuk memverifikasi data sebelum melangkah ke tahap analisis sentimen. Keseluruhan desain antarmuka ini dirancang dengan konsistensi dan fungsi yang jelas, menunjang kelancaran proses analisis sentimen dalam sistem secara holistik.

1. Rancangan Metode Penelitian

Penelitian ini mengadopsi pendekatan rekayasa sistem yang mengintegrasikan pengumpulan data otomatis, analisis sentimen berbasis *Natural Language Processing* (NLP), dan algoritma *Machine Learning* untuk klasifikasi komentar publik mengenai kinerja Kejaksaan Agung. Metodologi penelitian mencakup beberapa tahapan:

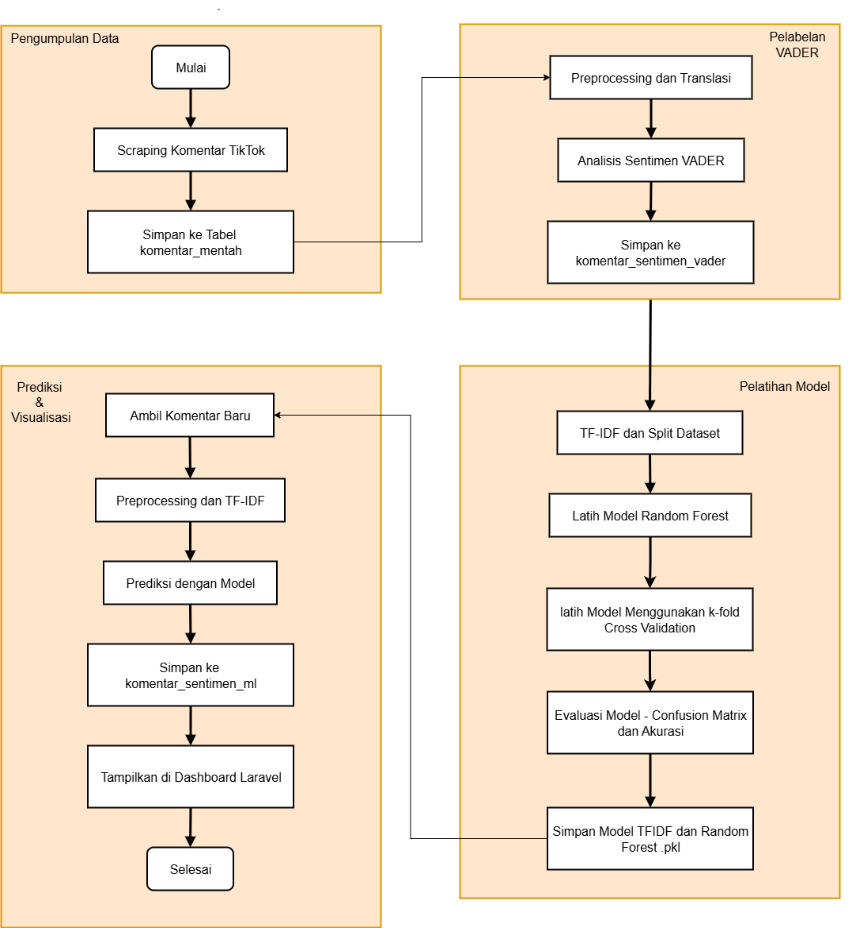
1. Pengambilan data: Komentar dari platform Tiktok dikumpulkan dengan memanfaatkan teknik web *scraping*.
2. Pelabelan data: Proses pelabelan sentimen pada data dilakukan dengan memanfaatkan algoritma VADER (*Valence Aware Dictionary and sEntiment Reasoner*).
3. Pelatihan model: Model klasifikasi *random forest* dilatih berdasarkan data yang telah dilabeli.

Seluruh tahapan tersebut diintegrasikan ke dalam sebuah sistem berbasis web yang dibangun dengan kerangka kerja Laravel, dengan layanan analisis yang dijalankan melalui Flask API pada sisi backend. Setiap proses dirancang secara otomatis, terstruktur, dan dapat diulang oleh pengguna tanpa memerlukan intervensi manual pada kode program.

Pada subbab ini, akan dijelaskan secara rinci mengenai alur penelitian, metode pengolahan data dan pelabelan sentimen, proses pelatihan dan evaluasi model klasifikasi, hingga tahapan prediksi yang menghasilkan output sentimen akhir. Selain itu, akan dipaparkan pula rumus evaluasi model dan prinsip kerja algoritma yang digunakan untuk memperkuat aspek ilmiah penelitian ini.

1. Alur penelitian

Penelitian ini dilaksanakan melalui serangkaian tahapan yang saling terintegrasi, dimulai dari pengumpulan data hingga evaluasi hasil prediksi sentimen. Tujuan utama dari alur ini adalah untuk membangun sistem analisis sentimen berbasis data TikTok yang mampu mengklasifikasikan opini publik terhadap kinerja Kejaksaan Agung secara otomatis. Setiap tahapan dirancang untuk saling melengkapi, baik dari sisi implementasi teknis maupun keilmuan.



Gambar 3. Alur penelitian

Tahapan awal dimulai dengan pengumpulan data komentar secara otomatis dari platform TikTok menggunakan Selenium. Komentar yang berhasil dikumpulkan kemudian disimpan dalam basis data sebagai data mentah. Tahap selanjutnya adalah pelabelan otomatis menggunakan algoritma VADER, yang akan memberikan label sentimen (positif, netral, negatif) berdasarkan perhitungan skor leksikal. Label ini kemudian digunakan sebagai data latih dan uji dalam proses pelatihan model klasifikasi menggunakan algoritma *Random Forest*. Setelah model selesai dilatih, dilakukan evaluasi menggunakan confusion matrix untuk menghitung akurasi, presisi, recall, dan F1-scoree.

Model yang telah dievaluasi dan disimpan dalam format .pkl kemudian digunakan untuk memprediksi komentar baru yang diperoleh dari proses scraping selanjutnya. Hasil prediksi disimpan dan ditampilkan pada dashboard Laravel melalui integrasi API dengan backend Flask. Proses ini dapat dijalankan ulang secara otomatis oleh administrator kapan saja melalui antarmuka sistem.

1. Tahapan *Preprocessing* data Komentar

*Preprocessing* data merupakan fase krusial dalam pengolahan data teks untuk mempersiapkan data sebelum analisis lebih lanjut. Data komentar TikTok, yang diperoleh melalui proses web *scraping*, seringkali mengandung berbagai elemen tidak standar, informal, dan kurang relevan, seperti emotikon, akronim, mention, dan variasi struktur kalimat. Oleh karena itu, serangkaian tahapan *preprocessing* menjadi esensial untuk meningkatkan kualitas data, yang kemudian akan menjadi masukan bagi metode analisis sentimen, baik yang berbasis leksikon (VADER) maupun algoritma *Machine Learning* (*Random Forest*).

Dalam penelitian ini, tahapan *preprocessing* yang diterapkan meliputi tiga langkah utama: pembersihan teks (*text cleaning*), normalisasi kata (*word normalization*), dan translasi ke bahasa Inggris. Ketiga tahapan ini diimplementasikan sebagai fungsi terintegrasi dalam skrip Python dan diaplikasikan secara konsisten pada proses analisis sentimen menggunakan VADER serta prediksi dengan model *Random Forest*.

1. Pembersihan Teks (*Text Clening*)

Tahap awal dalam proses preprocessing data adalah pembersihan teks (text cleaning), yang bertujuan untuk menghilangkan elemen-elemen tidak relevan dari komentar. Proses ini meliputi beberapa sub-tahapan sebagai berikut:

1. Penghapusan URL atau Tautan: Mengidentifikasi dan menghapus Uniform Resource Locators (URL) atau tautan yang sering muncul dalam komentar dan tidak berkontribusi pada analisis sentimen.
2. Penghapusan Mention (@username): Menghilangkan penyebutan pengguna lain (mention) yang diawali dengan simbol '@' karena informasi ini umumnya tidak relevan untuk analisis sentimen.
3. Penghapusan Hashtag (#tagar): Menghapus hashtag yang diawali dengan simbol '#' untuk menghindari bias atau kebisingan dalam data.
4. Penghapusan Angka dan Karakter Non-Alfabet: Menghilangkan semua karakter numerik dan simbol non-alfabetis (kecuali yang relevan seperti tanda baca tertentu jika dibutuhkan) yang dapat mengganggu analisis.
5. Penghapusan Spasi Ganda: Mengeliminasi spasi berlebih atau spasi ganda untuk memastikan konsistensi dalam pemisahan kata.
6. Konversi Huruf Menjadi Huruf Kecil (*Lowercase*): Mengubah seluruh teks menjadi huruf kecil untuk menyeragamkan data dan mencegah interpretasi kata yang sama sebagai entitas berbeda (misalnya, "Senang" dan "senang").

Contoh hasil sebelum dan sesudah pembersihan:

Sebelum: @kejaksaan\_agung Mantap brooo!! 👍🔥🔥 Cek https://bit.ly/test

Sesudah: mantap brooo

1. Normalisasi Kata

Tahap berikutnya dalam preprocessing adalah normalisasi kata (*word normalization*). Proses ini krusial untuk mengganti kata-kata tidak baku, singkatan, atau variasi penulisan informal dengan bentuk baku yang sesuai. Normalisasi dilakukan dengan memanfaatkan sebuah kamus normalisasi yang disimpan dalam format JSON (seperti, kamus\_normalisasi.json). Kamus ini berisi pemetaan antara kata-kata tidak baku yang sering ditemukan dalam data (seperti, "gpp", "bgt", "bgs") dengan bentuk baku atau resminya (seperti, "tidak apa-apa", "banget", "bagus"). Implementasi normalisasi ini memastikan konsistensi leksikal dalam korpus data, yang pada gilirannya akan meningkatkan akurasi analisis selanjutnya.

Contoh normalisasi:

1. “gk”→ “tidak”
2. “bgt” → “banget”
3. “smngt” → “semangat”

Dengan menerapkan proses normalisasi, variasi penulisan informal atau tidak baku dalam komentar dapat diseragamkan secara efektif. Hal ini krusial agar sistem analisis sentimen dapat mengenali makna sesungguhnya dari setiap kata dengan lebih akurat dan konsisten, mengurangi ambiguitas yang mungkin timbul dari penggunaan bahasa informal.

1. Translasi Kedalam Bahasa Inggris

Setelah tahapan pembersihan dan normalisasi data selesai, langkah krusial selanjutnya adalah penerjemahan teks ke dalam Bahasa Inggris. Proses ini esensial karena analisis sentimen VADER dan representasi fitur untuk model *Machine Learning* berbasis TF-IDF secara *inheren* dirancang untuk bekerja dengan korpus teks berbahasa Inggris.

Sebelum proses penerjemahan umum dilakukan, sistem mengimplementasikan penanganan khusus untuk beberapa *idiom* atau frasa khas Bahasa Indonesia yang mungkin tidak memiliki terjemahan literal yang sesuai atau dapat mengubah makna sentimen. Tujuannya adalah untuk memastikan sentimen yang terkandung dalam *idiom* tersebut dapat tertangkap dengan akurat setelah diterjemahkan.

Contoh translasi akhir:

1. Komentar: “gass lanjut pak jaksa”
2. Setelah idion: “go lanjut pak jaksa”
3. Setelah translasi: “go ahead mr. prosecutor”

Seluruh tahapan *preprocessing* mulai dari pembersihan teks, normalisasi kata, hingga penerjemahan ke Bahasa Inggris diintegrasikan secara kohesif dalam sebuah pipeline sistem. Fungsi *preprocessing* ini dipanggil secara otomatis setiap kali sistem akan menganalisis komentar. Ini berlaku baik untuk proses pelabelan sentimen menggunakan metode VADER maupun untuk prediksi sentimen melalui model *Random Forest*. Dengan adanya *preprocessing* yang seragam dan terotomatisasi, sistem mampu menghasilkan input data yang bersih, baku, dan siap dianalisis. Konsistensi ini krusial karena secara langsung meningkatkan akurasi hasil analisis sentimen, memastikan bahwa model menerima data dalam format optimal untuk kinerja terbaiknya.

1. Pelabelan Komentar dengan VADER

Pelabelan komentar merupakan tahap penting dalam membentuk dataset berlabel yang diperlukan untuk melatih model Machine Learning. Pada penelitian ini, pelabelan dilakukan secara otomatis menggunakan pendekatan lexicon-based, yaitu VADER (*Valence Aware Dictionary and sEntiment Reasoner*). VADER merupakan metode analisis sentimen berbasis kamus (lexicon) yang telah dirancang untuk menganalisis teks pendek dan informal seperti komentar media sosial.

Sebelum dianalisis menggunakan VADER, komentar yang dikumpulkan melalui proses scraping terlebih dahulu diproses melalui tahap *preprocessing* meliputi: pembersihan teks (*cleaning*), *case folding*, normalisasi menggunakan kamus, serta translasi dari Bahasa Indonesia ke Bahasa Inggris. Translasi dilakukan karena kamus lexicon bawaan VADER hanya mengenali kata-kata dalam Bahasa Inggris.

1. Tahapan proses
2. *Preprocessing*

Setiap komentar hasil scraping dari TikTok dibersihkan melalui proses cleaning, case folding, dan normalisasi. Kata tidak baku seperti "kejagung" akan diubah menjadi "kejaksaan agung" menggunakan kamus normalisasi internal.

1. Translasi

Karena VADER hanya mendukung Bahasa Inggris, komentar akan diterjemahkan menggunakan *GoogleTranslator*. Sebelum translasi, sistem juga menggunakan kamus idiom sederhana untuk mengganti istilah lokal (seperti “jos”, “gass”, atau “maung”) menjadi padanan makna internasional (misalnya “awesome”, “go”, atau “hero”).

1. Analisis sentimen

Komentar yang telah diterjemahkan dianalisis menggunakan fungsi *SentimentIntensityAnalyzer* dari pustaka nltk.sentiment.vader.

Fungsi ini menghasilkan:

1. vader\_pos → proporsi sentimen positif
2. vader\_neu → proporsi sentimen netral
3. vader\_neg → proporsi sentimen negatif
4. compound →skor keseluruhan yang menunjukkan polaritas utama komentar.
5. Rumus perhitungan *compound score*

Nilai *compound* dihitung dari total skor valensi kata-kata dalam komentar, dengan rumus normalisasi:

Compount =

1. “”: merupakan jumlah total valensi positif dan negatif berdasarkan leksikon VADER.
2. “”: merupakan konstanta penyesuaian dengan nilai default “15”

Hasil *compound* akan selalu berada dalam rentang -1 sampai +1 dan digunakan untuk menentukan label sentimen. Berdasarkan nilai *compound*, label sentimen yang ditentukan sebgai berikut:

|  |  |
| --- | --- |
| Nilai *Compound* | Label |
| ≥ 0.3 | Positif |
| ≤ -0.3 compound ≤ 0.3 | Netral |
| ≤ -0.3 | Negatif |

Ambang batas ini digunakan karena dinilai paling representatif untuk data informal, dan juga didukung oleh penelitian sebelumnya seperti N. Sham et. al. (2016) dengan judul penelitian “*Climate Change Sentiment Analysis Using Lexicon, Machine Learning and Hybrid Approaches*” [7].

1. Contoh implementasi

Komentar asli: “kejagung mantap, kami bangga memiliki kalian, bagus nya kalian yg ddk di MK”

Hasil normalisasi: “kejaksaan agung mantap kami bangga memiliki kalian bagus nya kalian yang duduk di mk”

Hasil translasi kedalam bahasa inggris: “The Attorney General's Office is great, we are proud to have you guys who are sitting in the MK”

Setelah dianalisis oleh VADER, akan diperoleh hasil:

Tabel 3. Contoh implementasi

|  |  |
| --- | --- |
| Metrik | Nilai |
| Vader\_pos | 0.298 → Sekitar 29.8% dari teks dianggap positif |
| Vader\_neu | 0.702 Sekitar 70.2% dari teks dianggap netral |
| Vader\_neg | 0.000 → Tidak ada bagian komentar yang bersifat negatif. |
| Compound\_score | 0.802 → Skor akhir keseluruhan gabungan, hasil dari rumus normalisasi |
| Label\_vader | positif |

Penjelasan:

1. Kata “great” dan “proud” dikenali oleh VADER sebagai kata positif, masing-masing memiliki valensi sekitar +3.1 dan +2.1.
2. Kata-kata lainnya netral, seperti “the”, “you”, “guys”, “attorney”.
3. Maka total valensi = 5.2

Menggunakan rumus:

Compount =

1. Karena nilai ≥ 0.3, maka sistem memberi label: Positif
2. Penyimpanan kedalam database

Hasil pelabelan disimpan kedalam tabel komentar\_sentimen\_vader dengan struktur data berikut:

1. ID Komentar
2. Komentar asli (comment)
3. Komentar bersih setelah preprocessin (cleaned\_commend)
4. Komentar trjemahan (comment\_translate)
5. Skor vader\_pos, vader\_neu, vader\_neg, compound\_score
6. Label akhir (label\_vader)
7. Waktu pemrosesan (processed\_at)
8. Contoh hasil dalam database:



Gambar 3. Contoh hasil dalam database

Tahapan pelabelan komentar dengan VADER pada penelitian ini berfungsi sebagai dasar pembentukan dataset berlabel yang akan digunakan untuk proses pelatihan dan pengujian model klasifikasi *Machine Learning* (*Random Forest*). Pemilihan VADER sebagai metode pelabelan memberikan keunggulan dari sisi kecepatan, otomatisasi, dan keakuratannya dalam menangani bahasa informal yang umum ditemukan dalam komentar TikTok.

1. Prediksi Sentimen Menggunakan model *Random Forest*

Tahap prediksi merupakan fase implementatif dalam sistem analisis sentimen yang dikembangkan. Pada tahap ini, model *Machine Learning* (*Random Forest*) yang telah berhasil dilatih dan dievaluasi pada tahap sebelumnya, diaplikasikan untuk melakukan klasifikasi sentimen secara otomatis terhadap data komentar TikTok terbaru. Tujuan utama dari proses ini adalah untuk mengklasifikasikan komentar-komentar baru ke dalam kategori sentimen positif, netral, atau negatif secara konsisten dan efisien. Proses ini dirancang untuk berjalan secara otomatis, menggantikan metode pelabelan awal (VADER) dengan model prediktif yang telah terlatih, sehingga mendukung keberlanjutan sistem dalam menganalisis data secara real-time.

1. Sumber data untuk prediksi

Data yang digunakan dalam tahap prediksi adalah komentar-komentar baru yang diperoleh dari hasil scraping pada platform TikTok. Secara teknis, data ini bersumber dari tabel komentar\_mentah dalam database MySQL. Kriteria spesifik untuk data yang akan diproses adalah baris yang memiliki nilai pada kolom is\_processed\_ml adalah 0 dan is\_processed\_vader adalah 0. Status ganda ini memastikan bahwa komentar yang diproses oleh model ML adalah data yang benar-benar baru dan belum pernah dianalisis oleh metode apa pun sebelumnya.

1. Arsitektur dan Alur kerja prediksi

Proses prediksi sentimen dibangun di atas arsitektur sistem yang mengintegrasikan framework web Laravel dengan Flask API. Flask API berperan sebagai backend yang secara khusus menangani eksekusi skrip Python untuk pemrosesan Machine Learning. Alur kerja prediksi, sebagaimana diimplementasikan dalam skrip ml\_predict.py, berjalan secara otomatis ketika dipicu oleh pengguna dan dapat diuraikan melalui tahapan-tahapan berikut:

1. Pemicuan proses prediksi oleh Admin

Proses diawali oleh tindakan pengguna dengan peran Admin yang memicu sistem Laravel untuk mengirimkan permintaan HTTP. Permintaan ini memanggil fungsi run\_ML\_analysis dalam skrip ml\_predict.py, yang berfungsi sebagai titik masuk untuk memulai keseluruhan proses analisis.

1. Akuisisi data komentar baru

Setelah dipicu, skrip akan menjalankan fungsi exec\_main\_script. Fungsi ini melakukan kueri ke database MySQL untuk mengambil data komentar dari tabel komentar\_mentah yang memenuhi kriteria is\_processed\_ml = 0 dan is\_processed\_vader = 0. Sesuai dengan implementasi pada skrip, pengambilan data dibatasi hingga maksimal 200 komentar terbaru pada setiap eksekusi untuk menjaga efisiensi dan performa sistem. Jika tidak ada data yang memenuhi kriteria, skrip akan berhenti.

1. Tahap *preprocessing* data

Setiap komentar yang telah diambil akan melewati serangkaian tahap preprocessing yang identik dengan tahap persiapan data pada saat pelatihan model. Hal ini bertujuan untuk memastikan konsistensi format data antara data latih dan data baru. Tahapan tersebut meliputi:

1. Case Folding: Mengonversi seluruh teks komentar menjadi format huruf kecil (lowercase).
2. Pembersihan Karakter (Cleaning): Menghapus karakter-karakter yang tidak relevan seperti URL, mention (@username), tagar (#hashtag), angka, dan tanda baca menggunakan ekspresi reguler.
3. Normalisasi Kata: Mengubah kata-kata tidak baku atau singkatan menjadi bentuk baku sesuai dengan kamus kamus\_normalisasi.json yang telah dimuat.
4. Penggantian Idiom: Menerjemahkan istilah-istilah informal atau idiom seperti "maung" atau "jos" menjadi padanan kata berbahasa Inggris menggunakan kamus yang telah didefinisikan di dalam fungsi apply\_idioms.
5. Translasi Bahasa: Menerjemahkan komentar dari Bahasa Indonesia ke Bahasa Inggris menggunakan *GoogleTranslator*. Tahap ini krusial karena model Random Forest dilatih menggunakan data berbahasa Inggris.
6. Ekstraksi fitur dengan TF-IDF (*Term Frequency-Inverse Document Frequency*)

Komentar yang telah melalui seluruh tahap preprocessing selanjutnya ditransformasikan dari format teks menjadi representasi vektor numerik. Proses ini menggunakan model TF-IDF Vectorizer yang dimuat dari berkas public/scripts/tfidf\_model.pkl.

1. Klasifikasi sentimen dengan model *random forest*

Vektor numerik yang dihasilkan dari tahap TF-IDF kemudian diumpankan sebagai masukan (input) ke dalam model klasifikasi Random Forest yang dimuat dari public/scripts/ml\_model.pkl. Model ini akan memproses vektor tersebut untuk menghasilkan prediksi label sentimen ('positif', 'netral', atau 'negatif') serta skor probabilitas untuk setiap kelas.

1. Persistensi hasil prediksi kedalam database

Hasil klasifikasi dari model disimpan ke dalam database untuk keperluan analisis lebih lanjut. Proses ini dilakukan sebagai berikut:

1. Data hasil prediksi, yang mencakup mentah\_id, video\_id, username, comment (komentar asli), tanggal\_komentar, cleaned\_comment (hasil normalisasi), comment\_translate (hasil translasi), predicted\_label (label prediksi), dan confidence\_score (skor kepercayaan), disimpan sebagai entri baru di dalam tabel komentar\_sentimen\_ml.
2. Status komentar pada tabel komentar\_mentah diperbarui dengan mengubah nilai kolom is\_processed\_ml dari 0 menjadi 1 untuk setiap ID komentar yang telah berhasil diproses. Pembaruan ini berfungsi sebagai penanda untuk mencegah terjadinya pemrosesan ulang di masa mendatang.
3. Status keluaran dan penanganan eksepsi

Skrip ml\_predict.py dirancang untuk memberikan umpan balik yang jelas mengenai status eksekusinya, yang penting untuk integrasi dengan sistem Laravel.

1. Eksekusi Berhasil: Jika keseluruhan proses berjalan tanpa kesalahan, fungsi run\_ML\_analysis akan mengembalikan sebuah objek JSON dengan status: "success" beserta pesan dan log keluaran dari konsol.
2. Tidak Ada Data: Apabila tidak ditemukan komentar baru untuk dianalisis (LIMIT 200 tidak mengembalikan data), skrip akan keluar dan fungsi pembungkusnya akan mengembalikan objek JSON dengan status: "warning" dan pesan "Tidak ada data untuk diproses".
3. Penanganan Kesalahan Lain: Jika terjadi kesalahan lain selama eksekusi (misalnya, koneksi database gagal atau berkas model tidak ditemukan), try-except akan menangkap kesalahan tersebut dan menghentikan proses untuk mencegah perilaku tak terduga, kemudian melaporkan kesalahan tersebut ke sistem pemanggil.
4. Perancangan Teknik Pengujian
   1. Pelatihan model *Random forest*

Setelah proses pelabelan otomatis komentar TikTok menggunakan pendekatan VADER berbasis leksikon, tahapan selanjutnya adalah pembangunan model *Machine Learning* untuk memprediksi sentimen komentar baru. Model ini dilatih menggunakan dataset yang telah dilabeli sebelumnya. Dalam penelitian ini, algoritma yang digunakan adalah *Random Forest Classifier*. Algoritma ini merupakan algoritma *ensemble* yang mengintegrasikan berbagai pohon keputusan (*decision tree*) guna menghasilkan prediksi yang lebih akurat dan stabil.

1. Transformasi komentar menjadi representasi numerik (TF-IDF)

Komentar TikTok yang awalnya berbentuk teks mentah tidak dapat secara langsung diinterpretasikan oleh algoritma *Machine Learning*. Oleh karena itu, diperlukan tahapan vektorisasi teks untuk mengubahnya menjadi representasi numerik. Metode yang digunakan dalam penelitian ini adalah TF-IDF (*Term Frequency–Inverse Document Frequency*). TF-IDF bekerja dengan memberikan bobot pada setiap kata. Bobot ini ditentukan berdasarkan dua faktor utama: frekuensi kemunculan kata tersebut dalam suatu dokumen spesifik (*Term Frequency*) dan jarang atau seringnya kata tersebut muncul di seluruh korpus dokumen (*Inverse Document Frequency*). Dengan demikian, kata-kata yang relevan dan memiliki daya diskriminasi tinggi akan mendapatkan bobot yang lebih besar.

Rumus perhitungan TF-IDF:

*tf-idf(t, d) = tf(t, d) . log*

Keterangan:

1. *tf(t, d)*: merupakan jumlah kemunculan kata *t* dalam dokumen *d*.
2. *df(t)*: jumlah dokumen yang mengandung kata *t*
3. *N*: total jumlah dokumen

Sebagai hasil dari proses vektorisasi teks menggunakan TF-IDF, data komentar TikTok yang semula berbentuk narasi kini direpresentasikan dalam bentuk matriks fitur numerik. Setiap baris dalam matriks ini merepresentasikan satu komentar dan setiap kolom menunjukkan bobot TF-IDF dari suatu kata dalam komentar tersebut, yang secara efektif mengubah setiap komentar menjadi vektor numerik.

Rumus voting pada *random forest*:

Keterangan:

1. : prediksi dari pohon ke-i
2. : hasil akhir prediksi
3. *mode*: label yang paling sering muncul
4. Simulasi perhitungan dan contoh Implementasi

Contoh komentar: “mantap bangga, kejagung hebat”

Hasil *preprocessing* dan normalisasi: “mantap bangga kejaksaan agung hebat”

Translasi kedalam bahasa inggris: “great proud attorney general great”

1. Perhitungan tf-idf

Contohnya nilai-nilai berikut diketahui:

1. N (jumlah total komentar): 1000
2. Kata “great” muncul 2 kali dikomentar dan di 100 komentar df = 100
3. Kata “proud” muncul 1 kali dikomentar dan di 10 komentar df = 10
4. Kata “attorney general” muncul 1 kali dan di 200 komentar df = 200

Tabel 3. Contoh implementasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kata | TF | IDF | IDF (*log* ) | TF × IDF |
| Great | 2 | 100 | log(10)=1.0 | 2 × 1.0 = 2.0 |
| Proud | 1 | 10 | log(100)=2.0 | 1 × 2.0 = 2.0 |
| Attorney general | 1 | 200 | log(5)≈0.7 | 1 × 0.7 = 0.7 |

1. Normalisasi (L2 Normalization)

Lalu :

1. great: 2.0/2.91 = 0.687 → 0.65
2. proud: 2.0 / 2.91 = 0.687 → 0.70
3. attorney general: 0.7 / 2.91 = 0.24 → 0.50

dengan vektor akhir komentar: 0.65, 0.70, 0.50

1. prediksi model

vektor di atas diberikan ke model random forest. Misal terdapat 5 decision tree:

|  |  |
| --- | --- |
| Tree | Prediksi |
| 1 | Positif |
| 2 | Positif |
| 3 | Netral |
| 4 | Positif |
| 5 | Positif |

Melalui proses majority voting, 4 dari 5 pohon memprediksi “positif”, maka hasil akhir adalah: positif

1. penyimpanan dan integrasi sistem

Hasil prediksi kemudian disimpan ke dalam database tabel komentar\_sentimen\_ml, dan ditampilkan di dashboard Laravel secara real-time. Model dan vectorizer disimpan dalam file .pkl untuk digunakan kembali saat melakukan analisis komentar baru:

1. ml\_model.pkl → model Random Forest
2. tfidf\_vectorizer.pkl → vectorizer TF-IDF yang digunakan

Tahapan pelatihan model dilakukan dengan memanfaatkan dataset hasil pelabelan dari VADER, yang kemudian diubah menjadi fitur numerik menggunakan TF-IDF. Nilai-nilai TF-IDF dihitung berdasarkan kemunculan kata dan jarak penyebarannya dalam seluruh korpus, kemudian dinormalisasi agar dapat digunakan secara optimal oleh algoritma *Random Forest*. Proses pelatihan menghasilkan model klasifikasi yang dapat memprediksi sentimen komentar baru secara otomatis, sehingga dapat mempercepat dan mengotomatisasi analisis opini publik di media sosial.

* 1. *K-Fold Cross validation*

K-Fold Cross Validation adalah metode validasi yang digunakan untuk mengevaluasi performa model Machine Learning dengan membagi dataset menjadi beberapa bagian (fold). Pada setiap iterasi, satu fold digunakan sebagai data uji, sementara fold lainnya digunakan untuk data latih. Proses ini diulang hingga setiap fold menjadi data uji satu kali. Teknik ini membantu memastikan bahwa evaluasi model tidak bergantung pada satu pembagian data saja, sehingga hasilnya lebih akurat dan dapat diandalkan.

Metode ini dipilih karena mampu mengurangi risiko overfitting dan memberikan estimasi performa model yang lebih stabil. Setiap data dalam dataset memiliki kesempatan yang sama untuk menjadi data latih maupun data uji. Selain itu, metode ini memaksimalkan penggunaan data yang tersedia, sehingga model dapat dilatih dengan data sebanyak mungkin pada setiap iterasi.

Dalam penelitian ini, digunakan K-Fold Cross Validation dengan nilai K=5. Artinya, dataset dibagi menjadi lima bagian. Model dijalankan sebanyak lima kali, dimana setiap kali satu fold digunakan sebagai data uji dan empat fold lainnya digunakan sebagai data latih. Hasil dari kelima percobaan kemudian dirata-rata untuk mendapatkan nilai akhir dari performa model. Proses ini diterapkan pada dataset yang telah dilabeli menggunakan metode VADER. Model Random Forest dilatih dan divalidasi menggunakan pembagian data berdasarkan metode K-Fold ini, sebelum akhirnya model terbaik digunakan untuk melakukan prediksi komentar TikTok.

Penerapan K-Fold Cross Validation pada penelitian ini memberikan beberapa keuntungan, yaitu:

* + 1. Menjamin hasil evaluasi lebih general dan tidak bias terhadap pembagian data tertentu.
    2. Meminimalkan kesalahan estimasi performa model.
    3. Memastikan model yang dihasilkan benar-benar mampu melakukan generalisasi terhadap data baru.
    4. Memaksimalkan penggunaan seluruh data yang tersedia baik untuk pelatihan maupun validasi.

1. Ilustrasi proses

Secara sederhana, proses K-Fold pada penelitian ini berjalan seperti berikut:

1. Iterasi 1: Data latih (Fold 2,3,4,5), Data uji (Fold 1)
2. Iterasi 2: Data latih (Fold 1,3,4,5), Data uji (Fold 2)
3. Iterasi 3: Data latih (Fold 1,2,4,5), Data uji (Fold 3)
4. Iterasi 4: Data latih (Fold 1,2,3,5), Data uji (Fold 4)
5. Iterasi 5: Data latih (Fold 1,2,3,4), Data uji (Fold 5)

Setiap iterasi menghasilkan nilai akurasi, presisi, recall, dan F1-score, yang kemudian dirata-rata untuk mendapatkan hasil akhir evaluasi model.

* 1. *Confusion Matrix*

Evaluasi model klasifikasi bertujuan untuk mengukur performa model *Machine Learning* (*Random Forest*) yang telah dibangun berdasarkan data komentar TikTok. Model ini dievaluasi menggunakan beberapa metrik umum untuk klasifikasi multikelas agar dapat mengetahui efektivitas dan akurasi prediksi yang dihasilkan. Hasil evaluasi akan menjadi acuan dalam menentukan model terbaik yang akan digunakan pada tahap prediksi komentar baru.

1. Metode evaluasi

Proses evaluasi dilakukan dengan menggunakan beberapa variasi rasio data latih dan data uji, yaitu:

Tabel 3. pembagian data latih dan data uji

|  |  |  |
| --- | --- | --- |
| No | Data latih | Data uji |
| 1 | 70% data latih | 30% data uji |
| 2 | 80% data latih | 20% data uji |
| 3 | 90% data latih | 10% data uji |

Tujuan dari variasi ini adalah untuk mengidentifikasi skenario terbaik yang memberikan performa optimal dalam klasifikasi. Model dievaluasi menggunakan beberapa metrik, yaitu:

1. Confusion matrix
2. Accuracy
3. Precision
4. Recall
5. F1-Score
6. Confusion matrix

Confusion Matrix digunakan untuk menggambarkan performa klasifikasi dalam bentuk tabel, dengan membandingkan nilai prediksi model terhadap nilai aktual pada dataset uji. Pada kasus klasifikasi tiga kelas (positif, netral, negatif), confusion matrix berbentuk matriks 3x3 sebagai berikut:

Tabel 3. Rancangan confusion matrix

|  |  |  |  |
| --- | --- | --- | --- |
|  | Pred:positif | Pred:netral | Pred:negatif |
| Aktual positif | TP | FN1 | FN2 |
| Aktual netral | FP1 | TN | FN3 |
| Aktual negatif | FP2 | FP3 | TN2 |

Berikut merupakan Rumus – rumus metrik evaluasi yang digunakan:

1. Akurasi
2. Presisi

Recall

1. F1-Score

Confusion Matrix mencakup beberapa elemen utama, antara lain:

1. *True Positive* (TP): jumlah instansi positif yang diprediksi dengan benar.
2. *True Negative* (TN): jumlah instansi negatif yang diprediksi dengan benar.
3. *False Positive* (FP): jumlah instansi negatif yang salah diprediksi sebagai positif.
4. *False Negative* (FN): jumlah instansi positif yang salah diprediksi sebagai negatif.
5. Prosedur evaluasi

Langkah-langkah evaluasi model dalam penelitian ini meliputi:

1. Mengambil dataset hasil pelabelan sentimen menggunakan VADER dari tabel komentar\_sentimen\_vader
2. Melakukan dan transformasi komentar ke bentuk numerik menggunakan TF-IDF Vectorizer.
3. Membagi dataset menjadi data latih dan data uji.
4. Melatih model *random forest* menggunakan data latih
5. Menguji model terhadap data uji untuk setiap skenario.
6. Menghitung nilai metrik evaluasi untuk setiap skenario menggunakan confusion matrix.
7. Membandingkan hasil evaluasi dari setiap skenario yang dilakukan.
   1. Perancangan Teknik pengujian *Black Box*

Pengujian black box bertujuan untuk menguji fungsi-fungsi utama dalam sistem berdasarkan skenario penggunaan aktual tanpa melihat struktur internal kode program. Fokus pengujian adalah memastikan bahwa sistem dapat merespons masukan pengguna secara benar dan menghasilkan keluaran yang sesuai harapan. Pengujian dilakukan pada fitur-fitur utama seperti registrasi, login, scraping komentar, analisis sentimen, serta proses unduh hasil, baik untuk peran Admin maupun User.

Tabel 3. Perancangan skenario pengujian black box

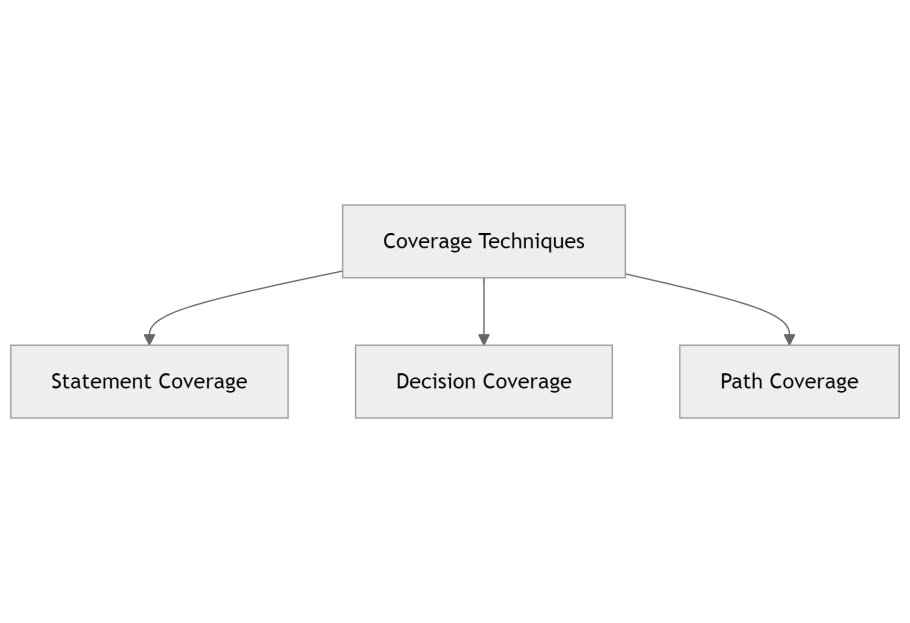
|  |  |  |  |
| --- | --- | --- | --- |
| No | Pengujian | Test Case | Hasil yang Diharapkan |
| 1 | Registrasi Pengguna | Melakukan registrasi dengan data yang valid | Sistem menampilkan pesan “Registrasi Berhasil” |
| 2 | Registrasi Pengguna | Melakukan registrasi dengan input kosong atau format email tidak valid | Sistem menampilkan pesan error dan registrasi ditolak |
| 3 | Login Pengguna (Admin) | Login dengan username dan password yang valid | Pengguna diarahkan ke halaman dashboard sesuai peran |
| 4 | Login Pengguna (Admin/User) | Login dengan username atau password yang salah | Sistem menampilkan pesan “Login Gagal” |
| 5 | Scraping Komentar TikTok | Admin menekan tombol “Scraping” | Sistem menjalankan proses scraping dan menyimpan data |
| 6 | Scraping Komentar TikTok | Proses scraping gagal (misalnya koneksi ke API gagal) | Sistem menampilkan notifikasi gagal |
| 7 | Analisis Sentimen ML | Admin menekan tombol “Analisis ML” saat data tersedia | Sistem menjalankan analisis dan menyimpan hasil |
| 8 | Analisis Sentimen ML | Admin menekan tombol “Analisis ML” saat tidak ada komentar baru | Sistem menampilkan pesan “Tidak ada komentar untuk dianalisis” |
| 9 | Lihat Hasil Analisis (Admin) | Admin membuka halaman analisis sentimen | Sistem menampilkan tabel hasil analisis lengkap |
| 10 | Lihat Hasil Analisis (User) | User membuka halaman analisis sentimen | Sistem menampilkan tabel hasil tanpa informasi username |
| 11 | Unduh Hasil Analisis (Admin) | Admin mengunduh hasil analisis melalui dashboard | Sistem menghasilkan file .xlsx berisi seluruh hasil analisis |
| 12 | Unduh Hasil Analisis (User) | User mengunduh hasil analisis | Sistem menghasilkan file .xlsx tanpa menampilkan username |
| 13 | Logout | Pengguna menekan tombol logout | Sistem mengakhiri sesi login dan mengarahkan kembali ke halaman login |

* 1. Perancangan Teknik Pengujian White Box

Pengujian white-box adalah metode yang digunakan untuk menguji logika internal dan struktur kontrol program secara menyeluruh. Dalam pengujian ini, penguji harus memahami alur kode dan mencakup setiap bagian logika yang ada. Tujuan utamanya adalah memastikan semua alur program telah diuji. Dalam penelitian ini, rancangan pengujian white-box difokuskan pada proses prediksi komentar. Pengujian dilakukan dengan pendekatan coverage testing yang terdiri dari:

* + 1. *Statement Coverage*: Memastikan seluruh baris kode dieksekusi minimal satu kali.
    2. *Decision Coverage*: Memastikan setiap cabang percabangan (if-else) diuji.
    3. *Path Coverage*: Memastikan seluruh jalur logika program diuji dari awal hingga akhir.

Berikut merupakan diagram struktur dari teknik coverage yang digunakan dalam pengujian white-box:



Gambar 3. Struktur teknik coverage pada pengujian white blox

Gambar 3.34 di atas menggambarkan tiga jenis teknik utama yang digunakan dalam pengujian white-box, yang menjadi dasar perancangan pengujian sistem dalam penelitian ini. Pengujian ini akan diterapkan pada proses klasifikasi komentar untuk memastikan kualitas dan ketepatan eksekusi logika program sebelum sistem digunakan secara menyeluruh. Hasil dari pengujian ini akan dijabarkan lebih rinci pada Bab IV.

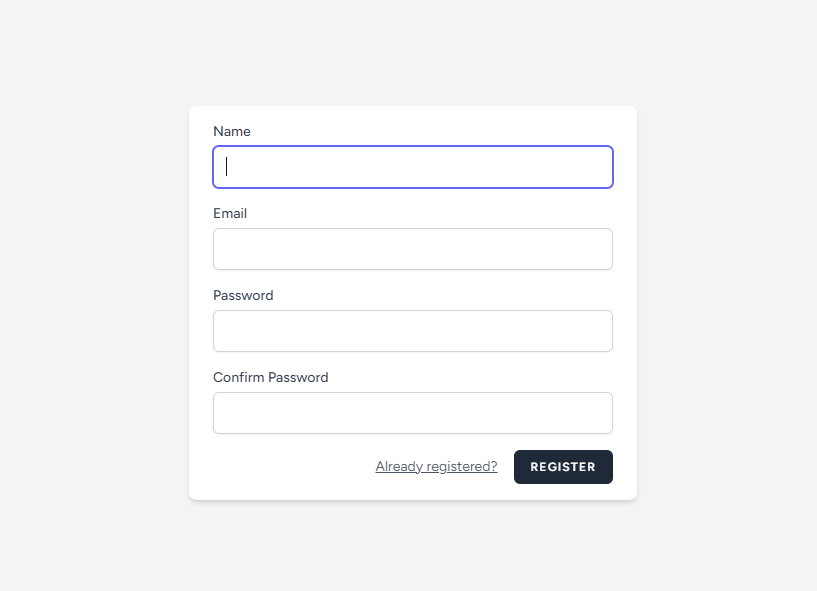
# BAB IV

# HASIL DAN PEMBAHASAN

1. Hasil Tampilan Website

Pada subbab ini menjalaskan mengenai Sistem yang telah dibangun memiliki beberapa antarmuka (*user interface*) utama yang digunakan oleh admin dalam mengelola proses *scraping* dan analisis sentimen terhadap komentar TikTok dan *user* dalam tampilan hasil analisis sentimen yang telah divisualisasi. Berikut ini adalah tampilan halaman-halaman penting yang terdapat pada website yang dibangun.

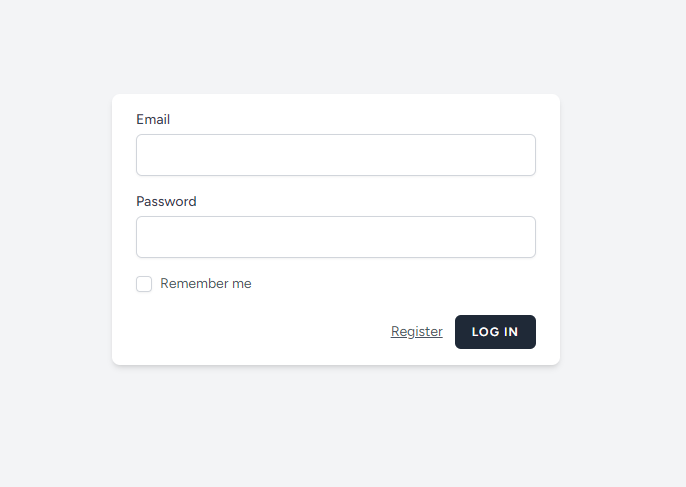
1. Tampilan *Register*



Gambar 4. Hasil tampilan *register*

Pada Gambar 4.1 menampilkan tampilan *register*. Bagi pengguna baru, sistem menyediakan halaman registrasi untuk membuat akun. Form registrasi mencakup nama, *email*, *password*, dan konfirmasi *password*. Jika registrasi berhasil, pengguna akan masuk kehalaman Login.

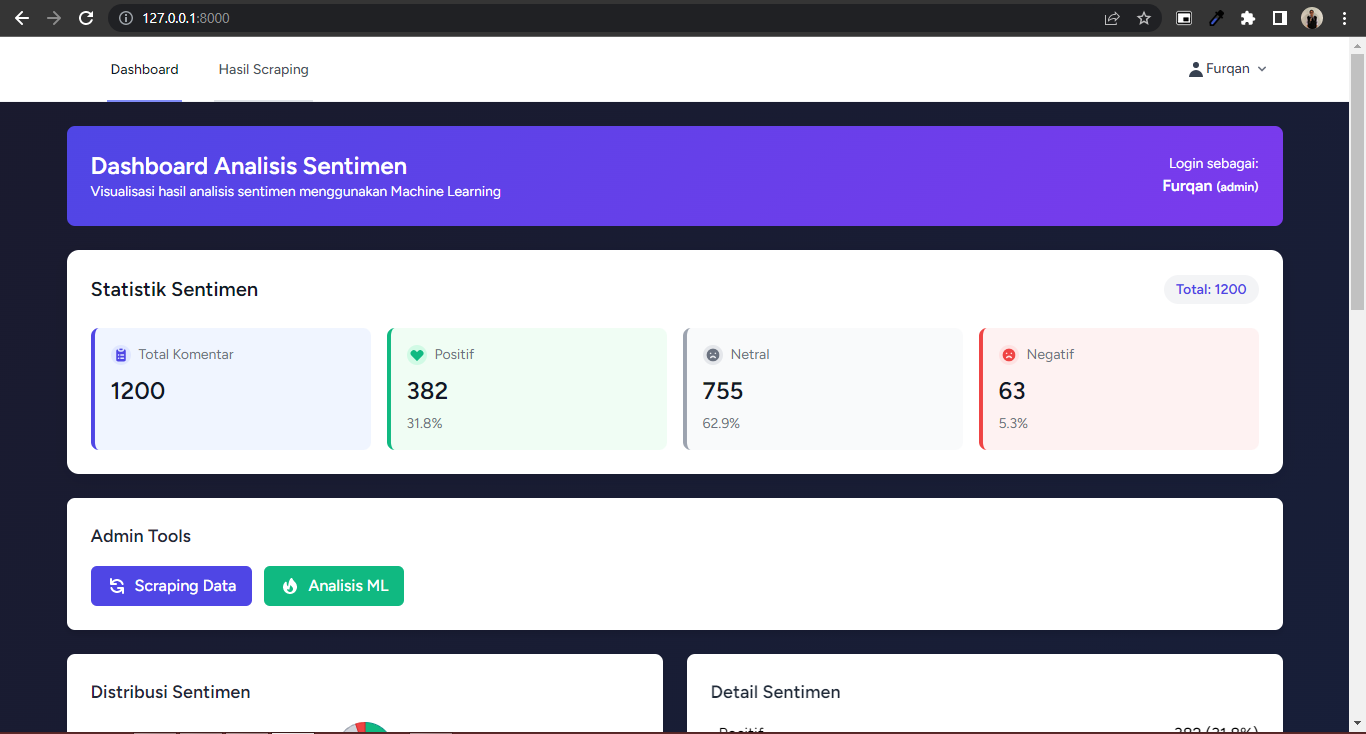
1. Tampilan *Login*



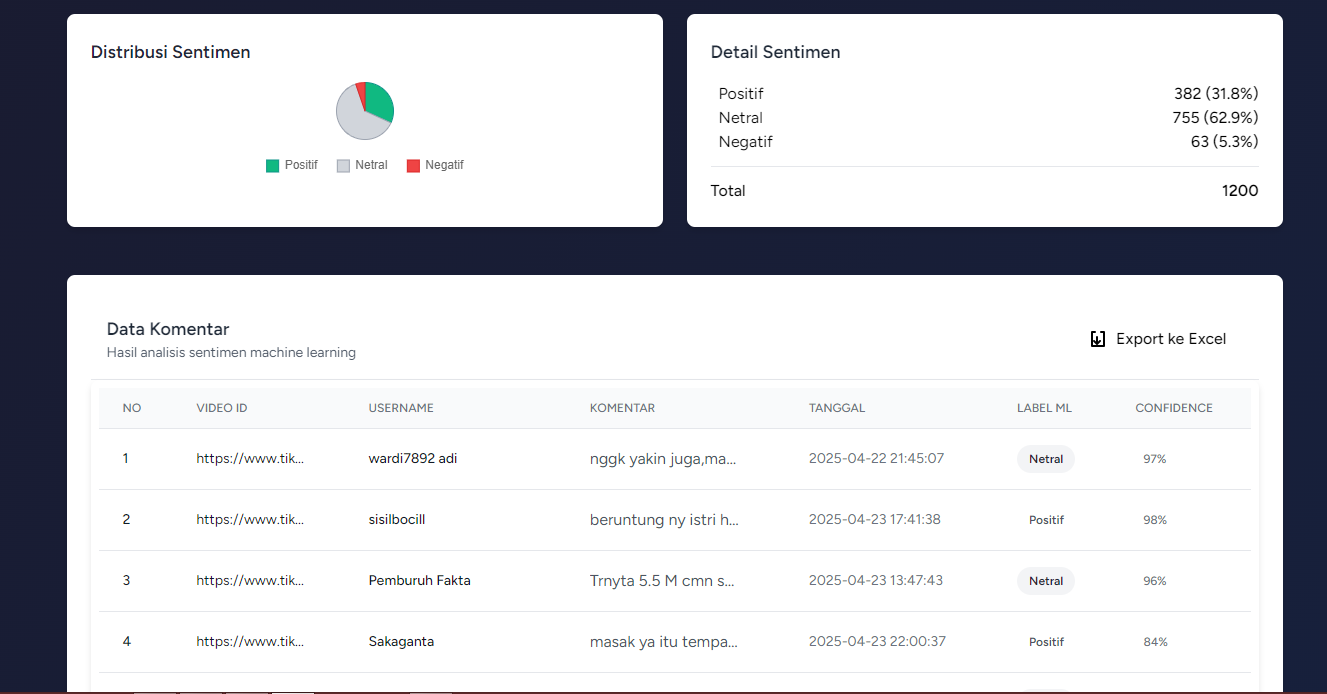
Gambar 4. Tampilan halaman *login*

Pada Gambar 4.2 menampilkan halaman *login*. Sebelum mengakses sistem, pengguna harus melakukan login menggunakan akun yang telah terdaftar. Halaman login terdiri dari form *email* dan *password*.

1. Tampilan Halaman *Dashboard* (Admin)



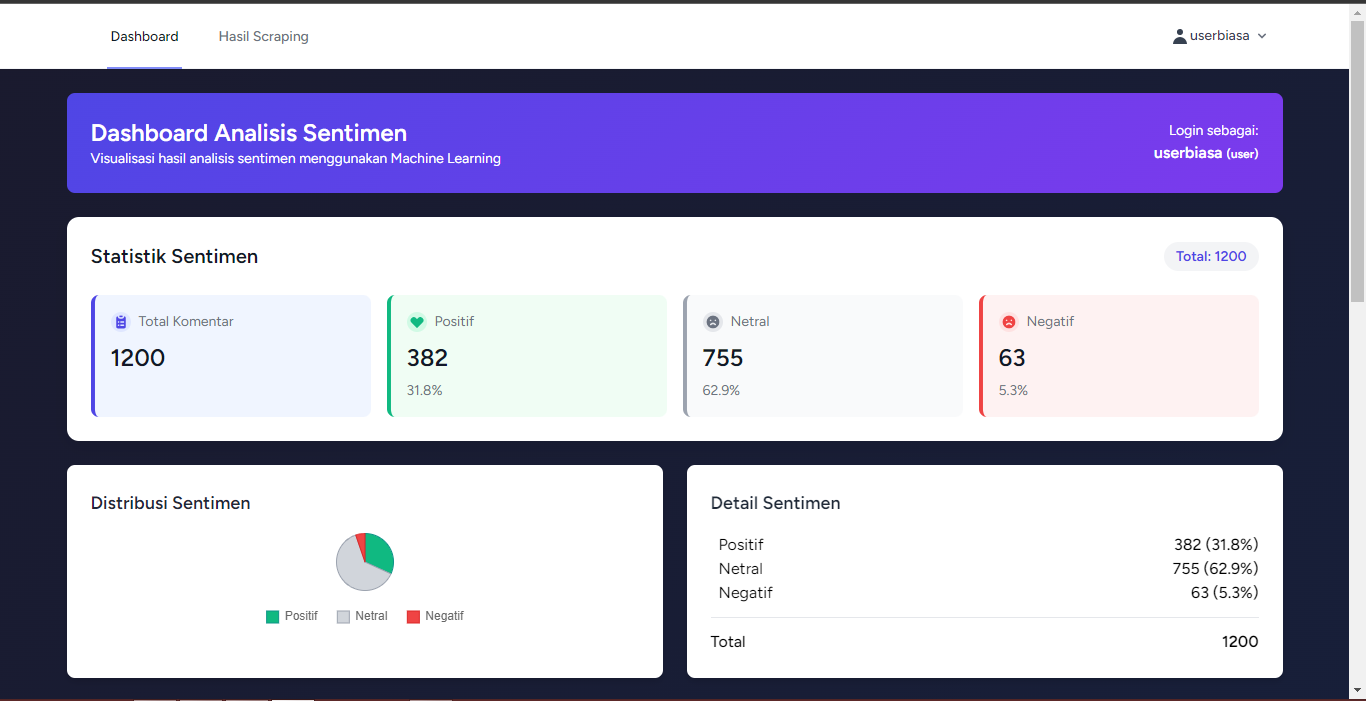
Gambar 4. Tampilan halaman *dashboard* 1(Admin)



Gambar 4. Tampilan halaman *dashboard* 2(Admin)

Pada Gambar 4.3 menampilkan halaman *dashboard* untuk admin, admin memiliki fitur penuh termasuk untuk memicu proses scraping komentar TikTok dan menjalankan analisis sentimen menggunakan model *Machine Learning* (ML). Terdapat dua tombol utama: “*Scraping*” dan “Analisis ML”. Selain itu, hasil analisis ditampilkan dalam bentuk diagram *pie chart* serta tabel komentar beserta label sentimen dan *confidence score*.

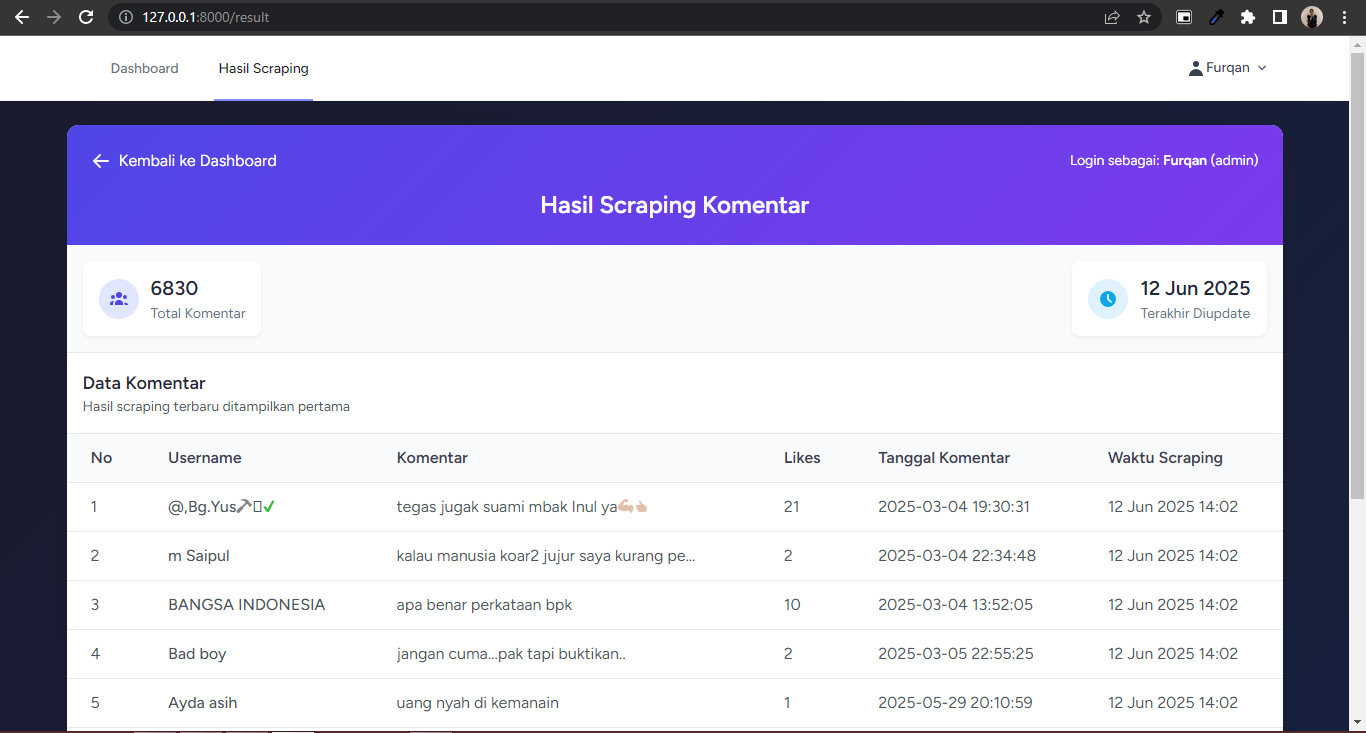
1. Tampilan Halaman *Dashboard* (*user*)



Gambar 4. Tampilan halaman *dashboard* (*user*)

Pada Gambar 4.4 menampilkan halaman *dashboard* (*user*), padaTampilan dashboard untuk *user* biasa hanya menyajikan informasi hasil analisis sentimen tanpa memberikan akses terhadap fitur *scraping* atau pemrosesan model. User hanya dapat melihat visualisasi pie chart dan tabel komentar yang mencakup *username*, isi komentar, tanggal, label hasil analisis ML, serta *confidence score*. Peran user bersifat *read-only* terhadap data hasil analisis.

1. Tampilan Halaman Hasil *Scraping* komentar



Gambar 4. Tampilan halaman hasil *scraping* komentar

Pada Gambar 4.6 menampilkan halaman hasil *scraping* komentar serta pemanfaatan dari fitur tombol *scraping* komentar yang terdapat pada halaman dashboard, Halaman ini menampilkan hasil scraping komentar TikTok secara rinci. Setiap baris berisi ID komentar, tautan video TikTok, username pengguna, isi komentar, jumlah likes dan tanggal komentar yang diperoleh oleh komentar tersebut. Halaman ini mempermudah admin dalam mengecek data mentah yang berhasil dikumpulkan.

1. Hasil *Preprocessing* Data Komentar TikTok

Proses *preprocessing* merupakan tahap penting yang harus dilalui oleh data sebelum dianalisis lebih lanjut, baik dengan algoritma VADER maupun *Machine Learning* (*RandomForest*). Berdasarkan implementasi aktual pada *script* *Python* yang digunakan, *preprocessing* terdiri dari beberapa langkah utama, yaitu:

1. Penghapusan *URL*, *Mention*, dan *Hastag*. Langkah pertama dalam preprocessing adalah menghapus teks yang tidak relevan seperti URL, mention pengguna, dan hashtag.

Tabel 4. Kode penghapusan *URL*, *Mention*, dan *Hastag*

|  |
| --- |
| teks = re.sub(r"http\S+|@\S+|#\S+", "", teks) |

1. Penghapusan simbol, angka, dan karakter khusus. Tahapan berikutnya adalah menghilangkan karakter khusus, simbol, angka, spasi, spasi berlebih dan tanda baca untuk menjaga konsistensi data.

Tabel 4. Kode penghapusan simbol, angka, dan karakter khusus

|  |
| --- |
| teks = re.sub(r"\d+", "", teks)  teks = re.sub(r"[^\w\s]", " ", teks)  teks = re.sub(r'\s+', ' ', teks) |

1. *Lowercasing* (Mengubah ke Huruf Kecil), Semua teks diubah menjadi huruf kecil untuk mencegah duplikasi kata akibat perbedaan kapitalisasi.

Tabel 4. Kode *lowercasing* (mengubah ke huruf kecil)

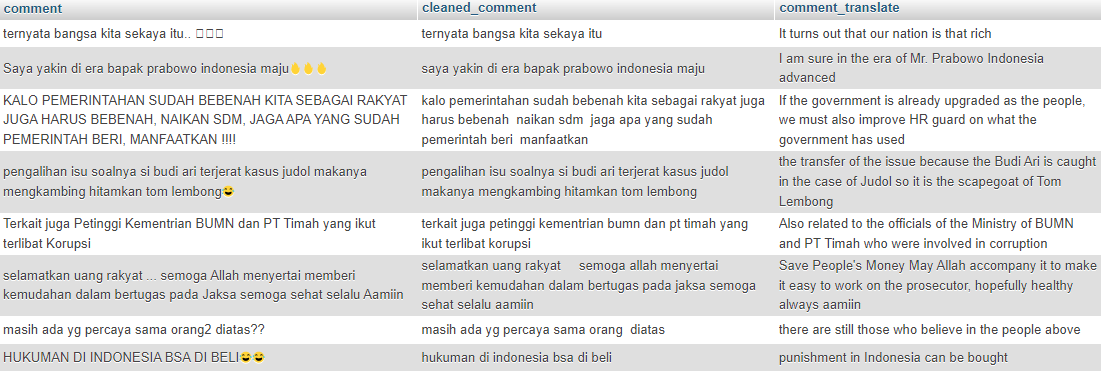
|  |
| --- |
| teks = teks.lower().strip() |

1. Penerjemahan kedalam Bahasa Inggris Khusus untuk analisis menggunakan VADER. Teks yang sudah dibersihkan harus diterjemahkan ke bahasa Inggris agar dapat diproses oleh algoritma VADER yang berbasis *lexicon* bahasa Inggris, untuk *library* yang digunakan yaitu *deep\_translator*.

Tabel 4. Kode penerjemahan kedalam bahasa inggris

|  |
| --- |
| def translate\_comment(teks):  try:  translated = GoogleTranslator(  source='auto', target='en').translate(teks)  return translated  except Exception as e:  print(f"X Terjadi kesalahan saat translate: {e}")  return teks |

1. Hasil Preprocessing



Gambar 4. Hasil *preprocessing*

Tahapan preprocessing ini digunakan secara konsisten dalam proses pelabelan data oleh VADER dan juga dalam pelatihan serta prediksi menggunakan model Machine Learning (RandomForest), memastikan bahwa data yang dianalisis memiliki format yang seragam dan berkualitas tinggi.

1. Hasil *Labeling* Data

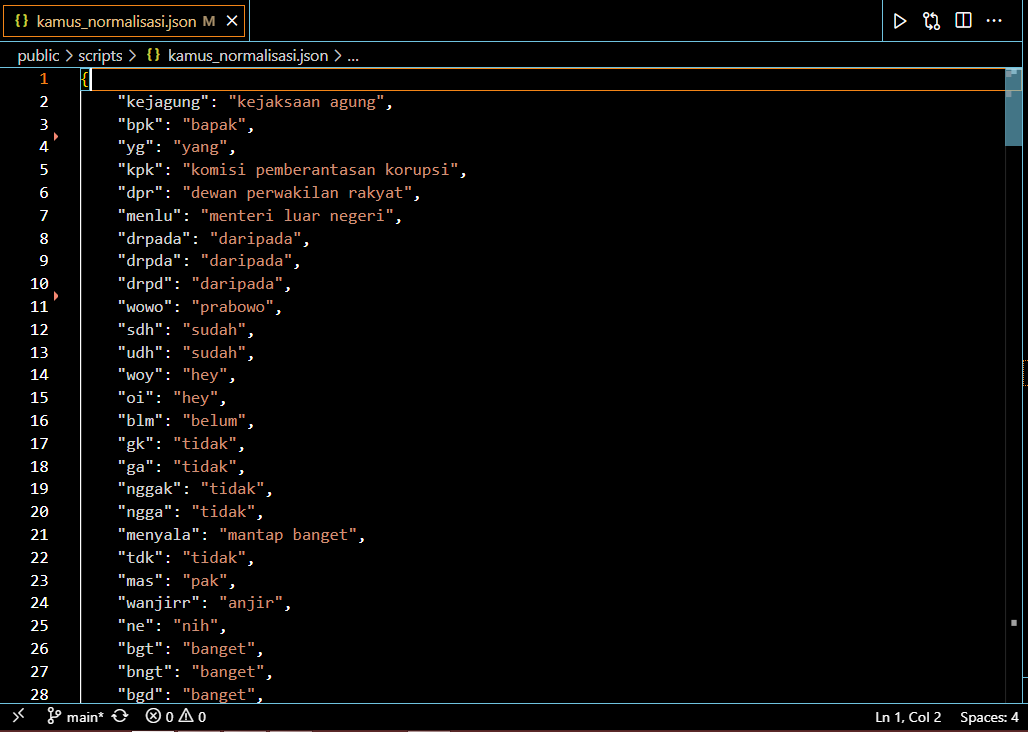
Setelah komentar TikTok berhasil dikumpulkan dan diproses melalui tahap *preprocessing*. Pada tahap ini, komentar-komentar hasil *preprocessing* dari TikTok diproses menggunakan metode analisis sentimen berbasis leksikon, yaitu VADER (*Valence Aware Dictionary and sEntiment Reasoner*). Proses pelabelan tidak dilakukan secara langsung, namun melalui beberapa tahapan, yaitu normalisasi kata, penggantian idiom, translasi ke bahasa Inggris, dan terakhir dilakukan *scoring* dan pelabelan sentimen.

1. Penerapan kamus normalisasi

Kamus normalisasi digunakan untuk mengubah kata-kata tidak baku atau slang menjadi kata baku agar proses translasi dan analisis dapat berjalan dengan baik.

1. Kamus normalisasi

Berikut merupakan potongan atau beberapa isi yang ada didalam file kamus normalisasi yan akan dimanfaatkan untuk VADER serta model *random forest* nantinya.



Gambar 4. Kamus normalisasi

1. Kode penerapan kamus normalisasi

Berikut merupakan potongan kode untuk pemanfaatan kamus normalisasi yang telah dibuat sebelumnya.

Tabel 4. Kode penerapan kamus normalisasi

|  |
| --- |
| try:      with open("public/scripts/kamus\_normalisasi.json", "r", encoding="utf-8") as f:          normalization\_dict = json.load(f)  except Exception as e:      logger.error(f"Gagal load kamus\_normalisasi.json: {e}")      normalization\_dict = {}  def normalisasi(teks):      kata\_list = teks.split()      return " ".join([normalization\_dict.get(k, k) for k in kata\_list])  def preprocessing(teks):      teks = bersihkan\_teks(teks)      teks = normalisasi(teks)      return teks |

1. Penggantian idiom (*idiom dictionary*)

Sebelum proses translasi, sistem melakukan penggantian idiom-idiom lokal ke padanan bahasa Inggris untuk meningkatkan akurasi hasil translasi dan sentimen.

Tabel 4. Kode penggantian idiom

|  |
| --- |
| idiom\_dict = {      "maung": "hero", "jos": "awesome", "gass": "go",      "di tangan": "in the hands of", "gokil": "crazy cool",      "gasskeun": "let's go", "sangar": "awesome", "kpk gunanya apa": "kpk is useless",      "kpk kerjanya apa": "kpk does nothing",      "ngapain aja": "what have they done",      "fungsi nya apa": "what's the function",      "kerjanya cuma": "only does",  }  def apply\_idioms(teks):  for k, v in idiom\_dict.items():  teks = teks.replace(k, v)  return teks |

1. Proses translasi

Setelah proses *preprocessin*g, kamus normalisasi, idiom, selanjutnya Komentar kemudian diterjemahkan ke dalam bahasa Inggris menggunakan modul *deep\_translator*.

Tabel 4. proses translasi komentar

|  |
| --- |
| from deep\_translator import GoogleTranslator  def translate(teks):  try:  teks = apply\_idioms(teks)  translated = GoogleTranslator(source='auto', target='en').translate(teks)  return translated  except Exception as e:  print(f"Translasi gagal: {e}")  return teks |

1. Setelah proses *preprocessin*g, kamus normalisasi, idiom, dan translasi selanjutnya proses analisis sentimen dilakukan dengan menggunakan VADER.

Tabel 4. Perhitungan Skor dan Label Sentimen VADER

|  |
| --- |
| From nltk.sentiment.vader import SentimentIntensityAnalyzer  nltk.download("vader\_lexicon")  vader = SentimentIntensityAnalyzer()  def analyze\_sentiment(teks):  scores = vader.polarity\_scores(teks)  compound = scores['compound']  if compound >= 0.3:  label = "positif"  elif compound <= -0.3:  label = "negatif"  else:  label = "netral"  return scores['pos'], scores['neu'], scores['neg'], compound, label |

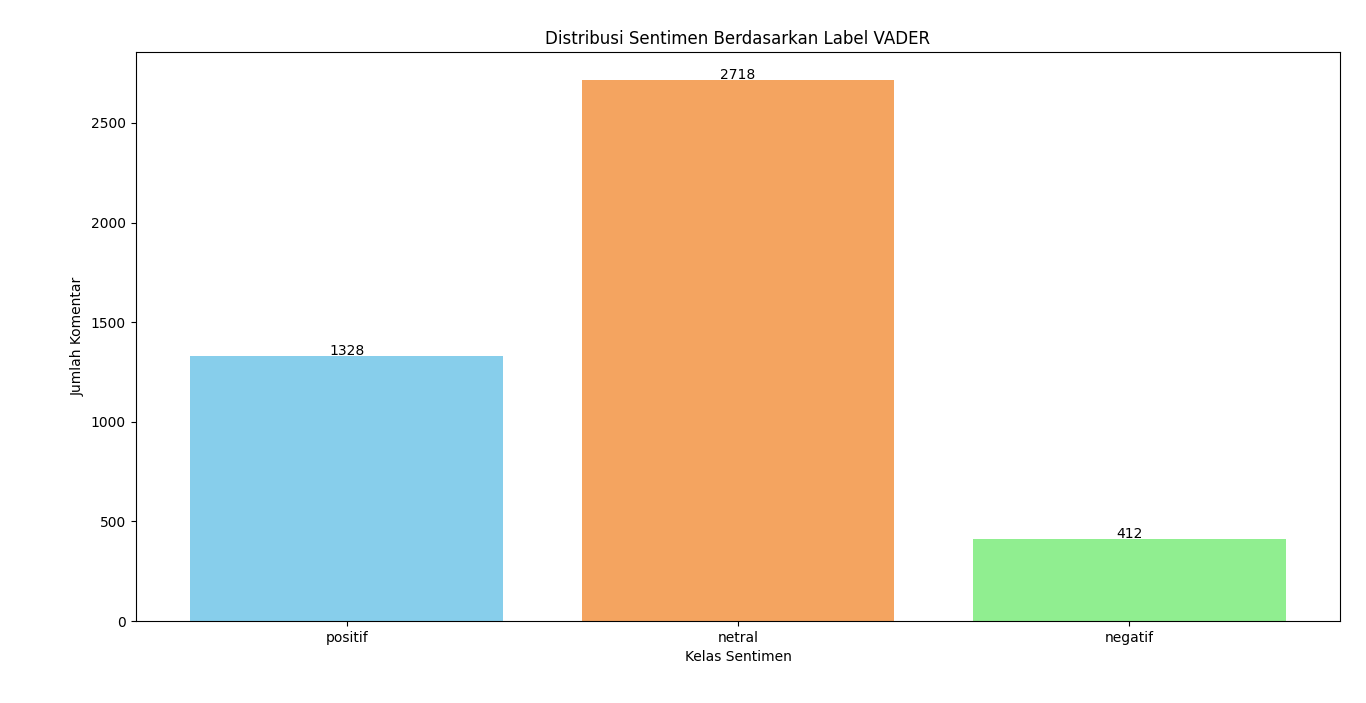
1. Hasil akhir proses pelabelan

Setelah proses proses *preprocessin*g, kamus normalisasi, idiom, translasi dan pelabelan berhasil dijalankan,. Berikut adalah hasil dari proses pelabelan komentar.

1. Tanpa menerapkan kamus normalisasi beserta idiom (*idiom dictionary*)



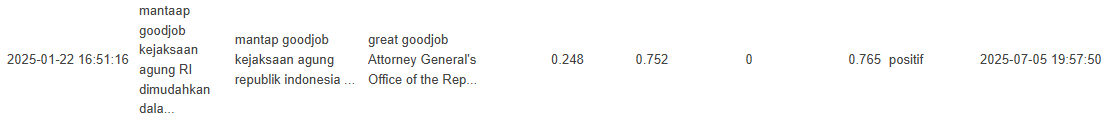
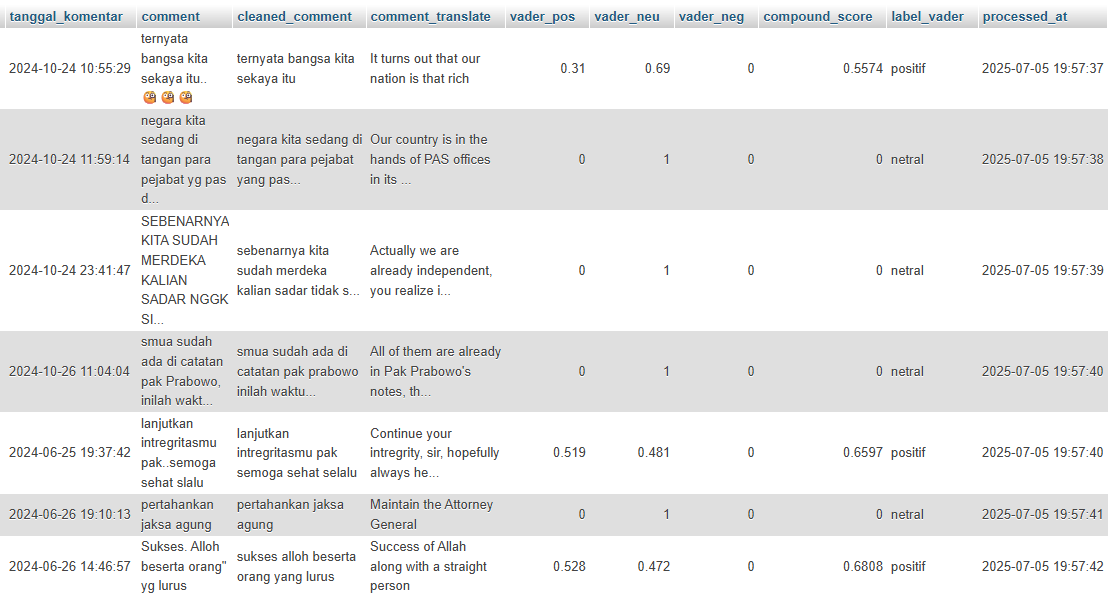
Gambar 4. Hasil *labeling* data tanpa menggunakan kamus



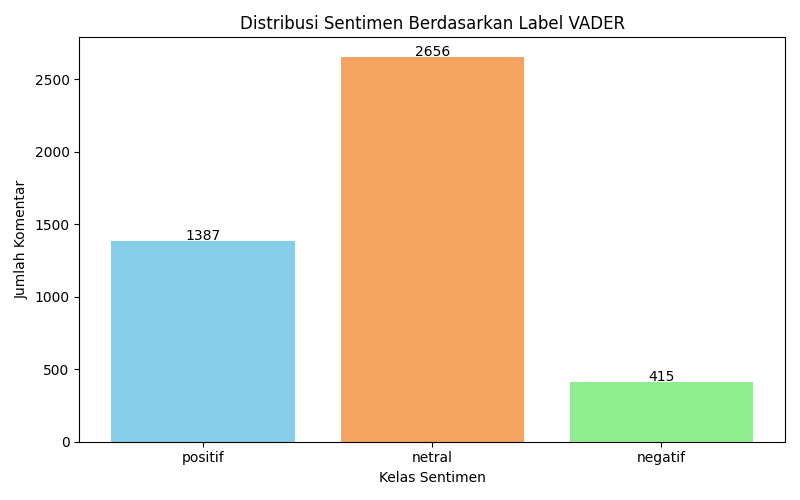
Gambar 4. Distribusi *labelling* data tanpa kamus

Berdasarkan Gambar 4. 9 dan Gambar 4. 10 tersebut, distribusi sentimen berdasarkan label VADER yang tidak menerapkan kamus normalisasi serta idiom menunjukkan jumlah sebagai berikut: sentimen netral sebanyak 2718, positif sebanyak 1328, dan negatif sebanyak 412 dengan total keseluruhan 4458 data.

1. menerapkan kamus normalisasi beserta idiom (*idiom dictionary*)



Gambar 4. Hasil *labeling* data menggunakan kamus



Gambar 4. distribus *labeling* data menggunakan kamus

Berdasarkan Gambar 4. 10 dan Gambar 4. 12, Distribusi data dalam penelitian ini mengacu pada hasil pelabelan sentimen menggunakan metode VADER yang diterapkan pada komentar TikTok. Dataset yang telah melalui proses normalisasi dengan kamus menghasilkan 1.387 komentar positif, 2.656 netral, dan 415 negatif dari total 4.458 data. Dataset ini dipilih karena proses normalisasi terbukti meningkatkan akurasi pelabelan, khususnya dalam mengenali kata-kata tidak baku atau slang yang umum digunakan di media sosial. Penggunaan kamus memungkinkan VADER menangkap nuansa emosional dengan lebih tepat, yang ditunjukkan dengan peningkatan proporsi kelas positif dan keseimbangan distribusi antar kelas.

Sebaliknya, dataset tanpa kamus menunjukkan dominasi kelas netral (2.718) dengan proporsi positif yang lebih rendah (1.328), menandakan bahwa banyak komentar bermuatan sentimen tidak terdeteksi secara akurat. Ketidakseimbangan ini berpotensi menyebabkan bias model terhadap kelas mayoritas dan menurunkan performa saat dihadapkan pada data baru. Oleh karena itu, dataset dengan kamus dipilih karena dinilai lebih representatif, seimbang, dan sesuai dengan karakteristik bahasa informal pada platform TikTok.

1. Hasil Pelatihan Model *Random Forest*
   * 1. *K-Fold Cross Validation*

Proses pengujian model *Machine Learning* dilakukan dengan metode *K-Fold Cross Validation*, yang bertujuan untuk mengevaluasi kestabilan dan konsistensi performa model terhadap berbagai pembagian data. Teknik ini membagi dataset menjadi beberapa bagian (fold), dimana pada setiap iterasi, satu fold digunakan sebagai data uji, sementara fold lainnya sebagai data latih.

Tabel 4. Potongan kode *K-Fold Cross Validation*

|  |
| --- |
| # ==== Load TF-IDF ====  tfidf = joblib.load('public/scripts/tfidf\_model.pkl')  X = tfidf.transform(df['comment\_translate'])  y = df['label\_vader'].str.lower().str.strip()  # ==== Model ====  model = RandomForestClassifier(      n\_estimators=100,      random\_state=42  )  # ==== Setup K-Fold ====  n\_splits = 5  n\_repeats = 3  cv = RepeatedStratifiedKFold(      n\_splits=n\_splits,      n\_repeats=n\_repeats,      random\_state=999  )  # ==== Jalankan K-Fold ====  scores = cross\_val\_score(      model, X, y,      scoring='accuracy',      cv=cv,      n\_jobs=-1  )  print(f"Akurasi tiap fold ({n\_splits} Fold x {n\_repeats} Repeat):")  print(scores)  # ==== Hitung Mean Tiap Pengulangan ====  scores\_reshaped = scores.reshape(n\_repeats, n\_splits)  mean\_per\_repeat = scores\_reshaped.mean(axis=1) |

Pengujian ini menggunakan konfigurasi *Repeated Stratified K-Fold* dengan 5 fold dan 3 kali pengulangan (5x3) untuk memastikan distribusi label yang seimbang di setiap fold. Pengujian dilakukan terhadap model *Random Forest* yang telah dilatih dengan data hasil pelabelan dari VADER.



Gambar 4. Hasil Grafik *K-Fold cross validation random forest*

Gambar 4.13 menunjukkan hasil visualisasi pengujian *K-Fold Cross Validation* pada model Random Forest. Garis berwarna oranye menunjukkan nilai akurasi pada masing-masing fold, sedangkan garis biru putus-putus menunjukkan rata-rata akurasi keseluruhan.

Tabel 4. Rata-rata akurasi tiap pengulangan

|  |  |
| --- | --- |
| Pengulangan | Mean Akurasi (%) |
| Repeat 1 | 87.93 |
| Repeat 2 | 88.11 |
| Repeat 3 | 87.64 |

Berdasarkan hasil yang ditampilkan pada Gambar 4.13 Hasil Grafik *K-Fold cross validation random forest*, nilai akurasi pada masing-masing fold menunjukkan fluktuasi yang tidak terlalu signifikan, dengan rentang akurasi antara 87.11% hingga 88.93%. Nilai ini menggambarkan bahwa model memiliki performa yang cukup stabil pada berbagai pembagian data. Berdasarkan hasil pengujian menggunakan Repeated Stratified K-Fold (5 Fold × 3 Repeat), diperoleh nilai akurasi yang relatif stabil. Akurasi per repeat masing-masing adalah 87.93%, 88.11%, dan 87.64%, dengan rata-rata keseluruhan sebesar 87.89% dan standar deviasi sebesar 0.0078. Nilai deviasi yang kecil ini menunjukkan bahwa model Random Forest memiliki konsistensi performa yang baik pada berbagai subset data.

Tabel 4. Perbandingan Fold yang berbeda-beda

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pembagian Data (Fold) | Hasil | Standar Devisi | Jumlah Iterasi | Mean Akurasi |
| 66.7% latih dan 33.3% uji (3-Fold) |  | 0.008% | 9 | 86.78% |
| 80% latih dan 20% uji (5-Fold) |  | 0.007% | 15 | 87.89% |
| 90% latih dan 10% uji (10-Fold) |  | 0.0108% | 30 | 88.45% |

* + 1. Evaluasi Model Menggunakan *Confusion Matrix*

Evaluasi dilakukan dengan membandingkan performa model Random Forest berdasarkan dua pendekatan preprocessing, yaitu tanpa kamus normalisasi dan idiom serta dengan kamus. Selain itu, dilakukan pengujian dengan tiga skenario pembagian data: 70:30, 80:20, dan 90:10. Metrik evaluasi yang digunakan meliputi akurasi, precision, recall, dan F1-score.. Proses pelatihan model sekaligus evaluasi Confusion Matrix ini tidak hanya menghasilkan metrik evaluasi, tetapi juga menyimpan model hasil pelatihan (ml\_model.pkl) dan *TF-IDF vectorizer* (tfidf\_model.pkl) yang akan digunakan pada tahap prediksi selanjutnya.

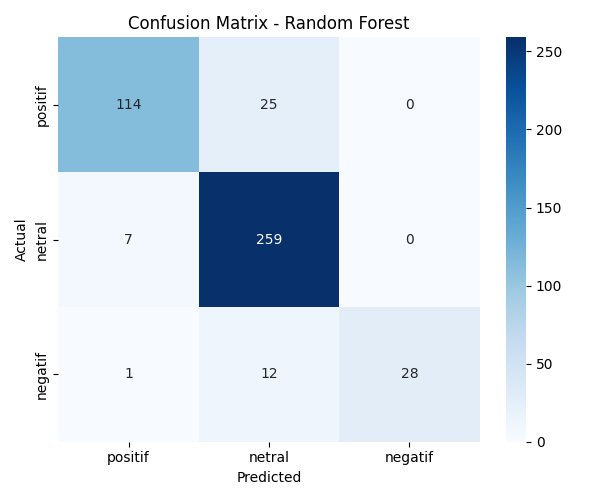
Tabel 4. Kode pengujian *confusion matrix*

|  |
| --- |
| import pandas as pd  from sqlalchemy import create\_engine  from sklearn.ensemble import RandomForestClassifier  from sklearn.feature\_extraction.text import TfidfVectorizer  from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score, f1\_score  from sklearn.model\_selection import train\_test\_split  import matplotlib.pyplot as plt  import seaborn as sns  import pickle  import numpy as np  # Load Data dari Database  query = """  SELECT comment\_translate, label\_vader  FROM komentar\_sentimen\_vader  WHERE comment\_translate IS NOT NULL AND label\_vader IS NOT NULL  """  df = pd.read\_sql(query, engine)  df['label\_vader'] = df['label\_vader'].str.lower().str.strip()  # TF-IDF  vectorizer = TfidfVectorizer(max\_features=1000)  X = vectorizer.fit\_transform(df['comment\_translate'])  y = df['label\_vader']  # Split Data  X\_train, X\_test, y\_train, y\_test = train\_test\_split(  X, y, test\_size=0.1, random\_state=42, stratify=y  )  # Training Model  model = RandomForestClassifier(n\_estimators=100, random\_state=42)  model.fit(X\_train, y\_train)  # === 6️⃣ Simpan model dan vectorizer ===  with open('public/scripts/tfidf\_model.pkl', 'wb') as f:      pickle.dump(vectorizer, f)  with open('public/scripts/ml\_model.pkl', 'wb') as f:      pickle.dump(model, f)  print("💾 Model & Vectorizer disimpan sebagai 'ml\_model.pkl' dan 'tfidf\_model.pkl'")  # Evaluasi Model  y\_pred = model.predict(X\_test)  print(classification\_report(y\_test, y\_pred))  print("Akurasi:", accuracy\_score(y\_test, y\_pred))  # Plot Confusion Matrix  matrix = confusion\_matrix(y\_test, y\_pred, labels=["positif", "netral", "negatif"])  sns.heatmap(matrix, annot=True, fmt='d', cmap='Blues')  plt.title('Confusion Matrix - Random Forest')  plt.show() |

Tabel 4. Distribusi label pada dataset yang digunakan (menggunakan kamus)

|  |  |
| --- | --- |
| Label | Jumlah |
| Netral | 2656 |
| Positif | 1387 |
| Negatif | 415 |
| Total | 4458 |

Dari total dataset 4458, dataset tersebut kemudian dibagi menjadi 80% data latih (3.566 data) dan 20% data uji (892 data). Split data ini digunakan berdasarkan perbandingan pada subbab sebelumnya



Gambar 4. Hasil Pengujian *confusion matrix*

Tabel 4. Evaluasi model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Label | Precision | Recall | F1-Score | Support |
| Negatif | 0.92 | 0.58 | 0.71 | 83 |
| Netral | 0.86 | 0.97 | 0.91 | 531 |
| Positif | 0.94 | 0.81 | 0.87 | 278 |
| Accuracy | | | 0.88 | 892 |
| Macro Avg | 0.91 | 0.79 | 0.83 | 892 |
| Weighted Avg | 0.88 | 0.88 | 0.88 | 892 |

Tabel 4. Perbandingan hasil pengujian *confusion matrix* (*data split*)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Split Data | Hasil | F1-Score (macro) | F1-Score (micro) | Akurasi |
| 70% latih dan 30% uji |  | 80% | 87% | 87% |
| 80% latih dan 20% uji |  | 83% | 88% | 88% |
| 90% latih dan 10% uji |  | 86% | 89% | 89% |

Berdasarkan hasil *Confusion Matrix* pada Gambar 4.14, Tabel 4.15 dan Tabel 4.16 di atas, Pada pembagian data 80% latih dan 20% uji, model Random Forest menunjukkan performa yang seimbang dengan akurasi sebesar 88.34% dan nilai F1-score (macro average) sebesar 83.01%. Kelas netral memiliki recall tertinggi sebesar 97%, menandakan model sangat baik dalam mengenali komentar netral. Sementara itu, kelas negatif masih menunjukkan tantangan dengan recall sebesar 58%, meskipun precision-nya tinggi (92%), yang mengindikasikan model cenderung berhati-hati dalam mengklasifikasikan komentar negatif. Secara keseluruhan, performa model dapat dikatakan optimal, terutama pada kelas mayoritas (netral dan positif).

1. Prediksi Sentimen Menggunakan *Machine Learning* (*Random Forest*)

Tahap ini merupakan proses implementasi dari model *machine learning* (*random forest*) yang telah dibangun dan dilatih sebelumnya. Model ini digunakan untuk mengklasifikasikan komentar-komentar TikTok terbaru ke dalam kategori sentimen positi, netral, ataupun negatif.

Proses prediksi dirancang berjalan secara otomatis, tidak lagi dilakukan secara manual, melainkan terintegrasi langsung ke dalam sistem melalui kombinasi *Laravel* dan *Flask API*. *Flask API* berfungsi sebagai *backend* yang menangani seluruh proses prediksi berbasis Python, sementara *Laravel* bertindak sebagai frontend yang menyediakan antarmuka bagi admin untuk menjalankan proses prediksi.

1. Integrasi dengan *Flask API*

Prediksi dilakukan melalui pemanfaatan tombol “Analisis ML” pada dashboard admin Laravel. Saat tombol ditekan, *Laravel* akan mengirim permintaan HTTP ke *endpoint Flask API*. *Flask API* kemudian akan menjalankan skrip Python (ml\_predict.py) yang memuat model *Machine Learning* (ml\_model.pkl) dan model TF-IDF (tfidf\_model.pkl) untuk melakukan prediksi pada komentar terbaru.

Keunggulan dari integrasi ini adalah seluruh proses dilakukan secara otomatis tanpa perlu berpindah aplikasi atau melakukan proses manual. Sistem akan:

1. Memeriksa apakah ada data komentar yang belum diprediksi (is\_processed\_ml = 0).
2. Melakukan preprocessing data secara otomatis.
3. Melakukan klasifikasi dengan Random Forest.
4. Menyimpan hasil ke database.
5. Memberikan umpan balik kepada Laravel untuk ditampilkan di dashboard.
6. Kode utama prediksi sentimen

Proses prediksi sentimen dilakukan dengan beberapa tahapan utama yang diimplementasikan dalam skrip ml\_predict.py. Berikut adalah potongan kode penting yang menggambarkan proses tersebut:

Tabel 4. Kode pada *laravel* untuk permintaan *endpoint*

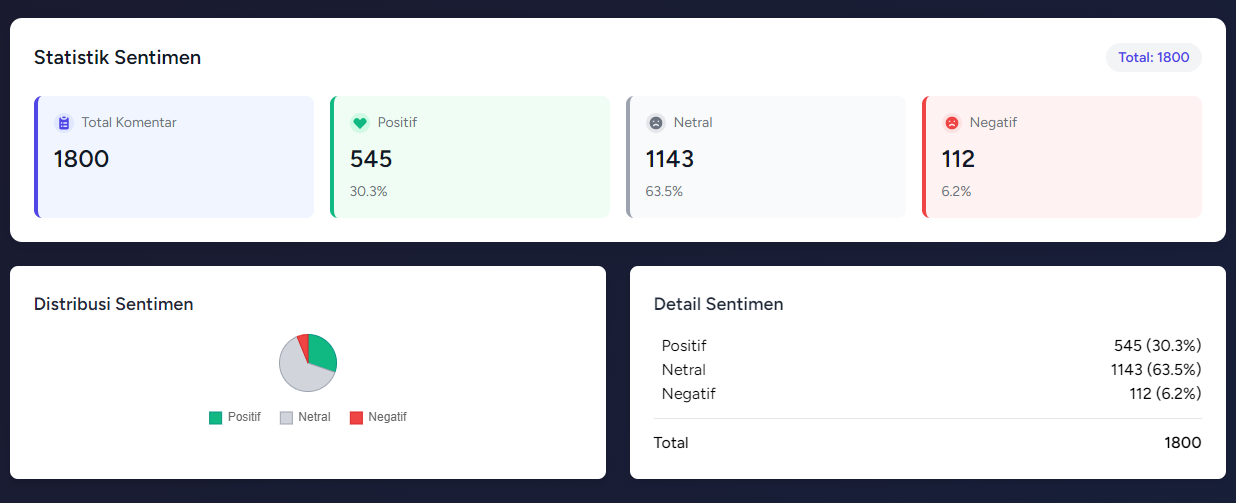
|  |
| --- |
| $.ajax({  url: "http://127.0.0.1:5000/analyze/analisis-ml",  type: "POST",  success: function(response) {  hideLoading();alert(response.message);  window.location.href = "{{ route('dashboard') }}";},  error: function(xhr, status, error)  {hideLoading();  alert("Terjadi kesalahan saat analisis ML. Error: " + error);} }); |

Tabel 4. Kode integrasi *flask API*

|  |
| --- |
| from flask import Flask, jsonify, request  from ml\_predict import run\_ML\_analysis  from flask\_cors import CORS  import os  app = Flask(\_\_name\_\_)  @app.route('/analyze/analisis-ml', methods=['POST'])  def analyze\_ml():  try:  result = run\_ML\_analysis()  return jsonify(result), 200  except Exception as e:  return jsonify({"status": "error", "message": str(e)}), 500  if \_\_name\_\_ == '\_\_main\_\_':  app.run(port=5000, debug=True) |

Kode pada Tabel 4.15 dan kode pada Tabel 4.16 di atas berfungsi sebagai endpoint untuk memicu proses analisis sentimen. Sistem Laravel akan mengirimkan permintaan HTTP POST ke endpoint /analyze/analisis-ml seperti yang ada pada Tabel 4. 15. Setelah menerima permintaan tersebut, Flask akan mengeksekusi fungsi run\_ML\_analysis() dari skrip ml\_predict.py. Selanjutnya, hasil analisis akan disimpan ke dalam database, dan Flask API akan memberikan respons berupa status berhasil atau gagal kepada sistem Laravel.

1. Hasil prediksi sentimen terhadap komentar baru



Gambar 4. Hasil prediksi sentimen terhadap komentar baru

1. Hasil Pengujian *BlackBox*

Pengujian blackbox dilakukan untuk memastikan bahwa setiap fungsi pada sistem berjalan sesuai dengan harapan berdasarkan skenario input dan output dari sudut pandang pengguna. Pengujian ini menguji fitur *login, scraping* data komentar, *show data,* analisis sentimen, *show data* analisis sentimen pada halaman dashboard, *public user,* dan *logout*. Pengujian ini tidak mempertimbangkan struktur kode internal, melainkan fokus pada fungsionalitas eksternal dari sistem.

Project Name : KEJAGUNG Sentiment Analysis Web App

Module Name : register, login, scraping data komentar, show data, analysis sentiment data , dashboard, public user, dan logout.

*References Document :* Spesifikasi Kebutuhan Perangkat Lunak.

Created By : Furqan

Date of creation : 25 Mei 2025

Date of review :

1. Pengujian *register*

Tabel 4. Pengujian *register* (*black box*)

| No | TEST CASE ID | TEST SCENARIO | TEST CASE | PRECONDITION | TES STEPS | TEST DATA | EXPECTED RESULT | POST CONDITION | ACTUAL RESULT | STATUS |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | TC-REG-01 | Menampilkan halaman registrasi | Klik tombol register | Berada pada halaman login | Klik tombol register | - | Form registrasi tampil | Form registrasi tampil | Form registrasi tampil | Lulus |
| 2 | TC-REG-02 | Melakukan registrasi akun admin | Mengisi username, email, password | Berada pada form registrasi | Isi form registrasi dan klik submit | Username, email, password | Sistem menampilkan pesan 'Registration Successful' | Sistem berhasil register | Sistem menampilkan pesan 'Registration Successful | Lulus |
| 3 | TC\_REG\_03 | Registrasi gagal (email sudah digunakan) | Input email yang sudah terdaftar | Email sudah pernah digunakan | 1. Masukkan email yang sama 2. Klik tombol Register | Email: test@mail.com | Menampilkan pesan 'Email sudah digunakan' | User tidak didaftarkan ulang | - | Lulus |

1. Pengujian login

Tabel 4. pengujian login (*black box*)

| No | TEST CASE ID | TEST SCENARIO | TEST CASE | PRECONDITION | TES STEPS | TEST DATA | EXPECTED RESULT | POST CONDITION | ACTUAL RESULT | STATUS |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | TC-LOG-01 | Menampilkan halaman login | Klik tombol login | Berada pada halaman utama | Klik tombol login | - | Form login tampil | Form login tampil | Form login tampil | Lulus |
| 2 | TC-LOG-02 | Melakukan login admin | Mengisi username dan password | Berada pada halaman login | Isi form login dan klik submit | Username dan password valid | Sistem menampilkan dashboard admin | Sistem berhasil login | Sistem menampilkan dashboard admin | Lulus |
| 3 | TC\_LOG\_03 | Login gagal (password salah) | Input password salah | Akun sudah terdaftar | 1. Masukkan email valid, password salah 2. Klik Login | Email: user@mail.com Password: salah | Menampilkan pesan 'Password salah' | Tidak login | - | Lulus |

1. Pengujian tombol *scraping* (admin)

Tabel 4. Pengujian tombol *scraping* (admin)

| No | TEST CASE ID | TEST SCENARIO | TEST CASE | PRECONDITION | TES STEPS | TEST DATA | EXPECTED RESULT | POST CONDITION | ACTUAL RESULT | STATUS |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | TC\_SCR\_01 | Scraping berhasil dijalankan | Klik tombol scraping | Admin sudah login | 1. Klik tombol Scraping di dashboard | - | Data berhasil diambil dan disimpan ke database | Komentar masuk ke tabel komentar\_mentah | - | Lulus |
| 2 | TC\_SCR\_02 | Scraping gagal (tidak ada data baru) | Klik tombol scraping tanpa data baru | Admin login, tidak ada video terkait | 1. Klik tombol Scraping | - | Muncul notifikasi 'Tidak ada data untuk diambil' | Tidak ada data disimpan | - | Lulus |

1. Pengujian Tombol analisis sentimen *machine learning* (admin)

Tabel 4. Pengujian Tombol analisis sentimen *machine learning* (admin)

| No | TEST CASE ID | TEST SCENARIO | TEST CASE | PRECONDITION | TES STEPS | TEST DATA | EXPECTED RESULT | POST CONDITION | ACTUAL RESULT | STATUS |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | TC\_ML\_01 | Analisis berhasil dijalankan | Klik tombol Analisis ML | Data komentar tersedia | 1. Klik tombol Analisis ML | - | Data diproses dan hasil prediksi disimpan ke tabel | Komentar berpindah ke tabel komentar\_sentimen\_ml | - | Lulus |
| 2 | TC\_ML\_02 | Analisis gagal (tidak ada data) | Klik tombol Analisis ML tanpa data baru | Tidak ada komentar baru | 1. Klik tombol Analisis ML | - | Muncul notifikasi 'Tidak ada data untuk diproses' | Tidak ada data diproses | - | Lulus |

1. Pengujian *White Box*

sadasdasdasdasd

# BAB V

# KESIMPULAN DAN SARAN

# DAFTAR PUSTAKA

[1] V. N. Prabowo, “Independensi Kejaksaan Agung Republik Indonesia,” vol. 12, no. 2, 2023, [Online]. Available: https://digilib.ubaya.ac.id/index.php?page=view/daftarpustaka\_detail&mode=&key=269328

[2] Imam Rahmaddani, “Pengawasan Kode Etik Jaksa Oleh Komisi Kejaksaan Guna Terwujudnya Jaksa Yang Profesional Dan Berintegritas,” *J. Presumption Law*, vol. 5, no. 1, pp. 18–34, 2023, doi: 10.31949/jpl.v5i1.4403.

[3] N. Apriliani, N. Suarna, and W. Prihartono, “Analisis Sentimen Review Penggunaan Tiktok Melalui Pendekatan Algoritma Naïve Bayes,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 6, pp. 3725–3731, 2024, doi: 10.36040/jati.v7i6.8299.

[4] W. N. Mufidati Nur Edma, E. N. Andini, and I. Widodo, “Analisis Sentimen Pada Pengguna Tiktok Menggunakan Metode Random Forest (Studi Kasus: Jessica-Mirna),” *J. Soc. Sci. Res.*, vol. 4, pp. 14477–14489, 2024.

[5] E. H. Muktafin, K. Kusrini, and E. T. Luthfi, “Analisis Sentimen pada Ulasan Pembelian Produk di Marketplace Shopee Menggunakan Pendekatan Natural Language Processing,” *J. Eksplora Inform.*, vol. 10, no. 1, pp. 32–42, 2020, doi: 10.30864/eksplora.v10i1.390.

[6] E. Yuniar, D. S. Utsalinah, and D. Wahyuningsih, “Implementasi Scrapping Data Untuk Sentiment Analysis Pengguna Dompet Digital dengan Menggunakan Algoritma Machine Learning,” *J. Janitra Inform. dan Sist. Inf.*, vol. 2, no. 1, pp. 35–42, 2022, doi: 10.25008/janitra.v2i1.145.

[7] N. M. Sham and A. Mohamed, “Climate Change Sentiment Analysis Using Lexicon, Machine Learning and Hybrid Approaches,” *Sustain.*, vol. 14, no. 8, pp. 1–28, 2022, doi: 10.3390/su14084723.

[8] D. Abimanyu, E. Budianita, E. P. Cynthia, F. Yanto, and Y. Yusra, “Analisis Sentimen Akun Twitter Apex Legends Menggunakan VADER,” *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 3, pp. 423–431, 2022, doi: 10.32672/jnkti.v5i3.4382.

[9] M. A. Tsaqif and W. Maharani, “Comparison of Random Forest and Decision Tree Methods for Emotion Classification based on Social Media Posts,” vol. 6, no. 4, pp. 2240–2248, 2025, doi: 10.47065/bits.v6i4.6677.

[10] M. Vakili, M. Ghamsari, and M. Rezaei, “Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification,” 2020, [Online]. Available: http://arxiv.org/abs/2001.09636

[11] T. Purwanto, “Analisa Perbandingan Kinerja Rest Api Dengan Framework Flask, Laravel, Dan Express Js,” *Sci. Sacra J. Sains*, vol. 3, no. 4, pp. 49–55, 2023.

[12] S. Andriansyah and Nurhasanah, “Perbandingan Performa Framework Laravel, Flask API Python, dan PHP Native untuk Aplikasi API pada Data AIS Polbeng,” *Konsep Desain Menentukan Hull Type, Mater. Dan Propulsi Unmanned Surf. Veh. Untuk Patroli Di Wil. Rokan Hiir Dengan Metod. Desicion Tree*, no. Lcm, pp. 478–486, 2020.