

BAB III ANALISIS DAN PERANCANGAN

3.1 Metode Pengembangan Sistem

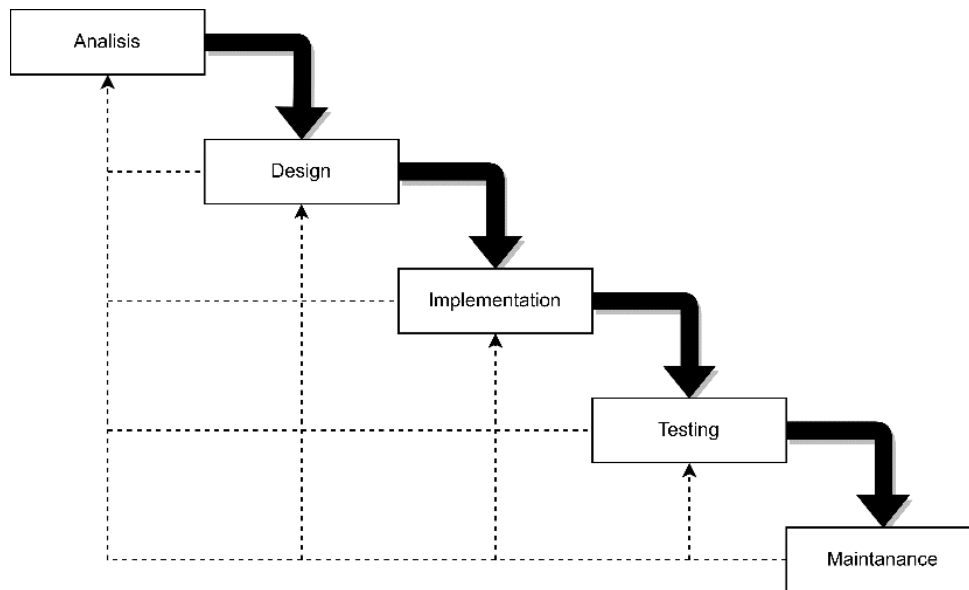
Dalam pengembangan sistem penelitian ini, digunakan model SDLC (*Software Development Life Cycle*). SDLC adalah proses pembuatan dan perubahan sistem yang melibatkan model dan metodologi untuk mengembangkan sebuah sistem. SDLC juga merupakan pola yang digunakan dalam pengembangan perangkat lunak, terdiri dari tahap-tahap seperti perencanaan, analisis, desain, implementasi, pengujian, dan pemeliharaan. Dalam penelitian ini menggunakan model *Waterfall*.

- *Waterfall Model* atau *Classic Life Cycle* adalah salah satu model yang paling umum digunakan dalam *Software Engineering* (SE). Menurut (Bassil, 2012) model ini mendefinisikan beberapa fase berurutan yang harus diselesaikan satu demi satu dan berpindah ke fase berikutnya hanya jika fase sebelumnya telah selesai. Dalam model ini, tahap-tahap pengembangan dilakukan secara linear dan bergantung satu sama lain. Tahap baru dimulai setelah tahap sebelumnya diselesaikan sepenuhnya.

Pada tahap perencanaan, semua persyaratan sistem ditentukan dengan jelas dan rencana proyek dibuat. Kemudian, pada tahap analisis, kebutuhan fungsional dan non-fungsional dikumpulkan dan dianalisis untuk merancang desain sistem yang sesuai.

Setelah itu, pada tahapan desain, spesifikasi teknis dari solusi yang akan diimplementasikan dibuat. Tahapan implementasi melibatkan pembuatan kode program berdasarkan desain yang telah ditetapkan.

Selanjutnya adalah fase pengujian di mana aplikasi atau sistem dikaji untuk memastikan bahwa ia bekerja sesuai dengan persyaratan awal. Terakhir adalah fase pemeliharaan di mana perbaikan atau peningkatan dilakukan jika ditemukan *bug* atau fitur baru yang diperlukan.



Gambar 2. 4 Metode Waterfall

Dengan pendekatan *waterfall* ini, setiap langkah mengikuti urutan linier tanpa ada kemungkinan kembali ke langkah sebelumnya. Ini memberi struktur proyek yang terorganisir tetapi juga bisa membatasi fleksibilitas dalam hal perubahan kebutuhan selama proses pengembangan.

3.1.1 Analisis Sistem Terdahulu

Permodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diimplementasikan ke dalam bentuk perangkat lunak. Hal tersebut menjadi penting, mengingat perangkat lunak harus bisa berinteraksi dengan elemen-elemen yang lain seperti perangkat keras dan *database*. Tahap ini sering disebut dengan Definisi Proyek.

Pada tahap ini peneliti melakukan observasi dan berdiskusi langsung dengan pengguna dalam hal ini adalah HSE PT.XYZ. Penjelasan lebih lengkap dari aktifitas tersebut terdapat pada sub bab 3.3. Pada Observasi ini terfokus pada permasalahan yang dialami oleh pengguna. Setelah mendapatkan beberapa data, peneliti menyimpan data tersebut sebagai bahan acuan dalam tahap analisis.

3.1.2 Analisis Kebutuhan Perangkat Lunak

Proses pencarian kebutuhan diintensifkan secara menyeluruh, terutama pada perangkat lunak. Untuk memahami karakteristik program yang akan dikembangkan, *software engineer* perlu memahami domain informasi dari

perangkat lunak, termasuk fungsi yang dibutuhkan, antarmuka pengguna, dan elemen-elemen lainnya. Hasil penentuan kebutuhan sistem dan perangkat lunak perlu didokumentasikan dan dipresentasikan kepada pengguna dalam hal ini adalah divisi HSE PT XYZ.

Berdasarkan hasil observasi, peneliti menganalisis data keluhan dari pengguna. Analisis tersebut bertujuan untuk menentukan kebutuhan fungsional perangkat lunak yang dapat mengatasi masalah yang saat dialami oleh pengguna. Dokumen analisis yang dihasilkan kemudian digunakan sebagai referensi untuk presentasi kepada pengguna. Penjelasan lebih rinci mengenai kegiatan tersebut dapat ditemukan pada sub bab 3.4 sub bab Spesifikasi Perangkat Lunak

3.1.3 Perancangan Sistem

Proses ini bertujuan untuk menyusun kebutuhan-kebutuhan yang telah diidentifikasi menjadi penggambaran yang terstruktur dalam bentuk cetak biru perangkat lunak sebelum proses pemrograman dimulai. Desain ini harus mampu mengimplementasikan kebutuhan yang telah dibahas pada tahap sebelumnya. Sebagaimana halnya dengan dua aktivitas sebelumnya, proses ini juga harus didokumentasikan sebagai konfigurasi dari perangkat lunak. Informasi lebih lanjut mengenai kegiatan ini dapat ditemukan pada sub bab 3.5.

Setelah peneliti memperoleh dokumentasi dari hasil analisis, langkah selanjutnya melibatkan transformasi kebutuhan fungsional perangkat lunak menjadi bentuk cetak biru perangkat lunak yang terstruktur. Hasil desain ini akan menjadi panduan bagi peneliti dalam membangun aplikasi tersebut.

3.1.4 Implementasi

Agar dapat dipahami oleh mesin, khususnya komputer, desain yang telah disusun harus diubah ke dalam bentuk yang dapat diproses oleh mesin, yakni melalui proses *coding* dalam bahasa pemrograman *python*. Tahap ini mewakili implementasi dari desain dan biasanya dilakukan oleh para *programmer* secara teknis.

Pada langkah ini, peneliti mulai membangun aplikasi berdasarkan cetak biru yang telah dirancang sebelumnya. Pengembangan aplikasi ini mencakup tahap awal

hingga aplikasi siap untuk dijalankan. Mulai dari fungsi-fungsi yang diperlukan hingga antarmuka pengguna diimplementasikan. Penjelasan yang lebih rinci mengenai kegiatan ini dapat ditemukan pada Bab IV

3.1.5 Pengujian

Setiap hasil pengembangan harus melewati pengujian, sebagaimana halnya dengan perangkat lunak. Semua fungsi yang terdapat dalam perangkat lunak harus diuji untuk memastikan bahwa tidak ada kesalahan, dan hasilnya harus sepenuhnya sesuai dengan kebutuhan yang telah ditentukan sebelumnya. Setelah menyelesaikan pembangunan aplikasi, peneliti menjalani serangkaian uji coba pada tahap ini. Aplikasi diuji menggunakan metode kotak hitam untuk menilai sejauh mana komponen sistem mencapai tingkat keberhasilan. Selain itu, peneliti juga melibatkan pengujian langsung dengan melibatkan pengguna yang merupakan bagian dari HSE PT XYZ. Penjelasan yang lebih rinci mengenai kegiatan ini dapat ditemukan

3.1.6 Pemeliharaan

Pemeliharaan perangkat lunak sangat penting, termasuk dalam hal pengembangan, karena perangkat lunak yang telah dibuat tidak selalu berada pada keadaan yang statis. Saat dioperasikan, mungkin masih terdapat kesalahan kecil yang tidak terdeteksi sebelumnya atau perlu penambahan fitur yang belum ada pada perangkat lunak tersebut. Pengembangan diperlukan ketika terjadi perubahan dari faktor eksternal perusahaan, seperti pergantian sistem operasi atau perangkat keras lainnya.

Peneliti belum mencapai tahap ini, oleh karena itu, tahap ini belum diimplementasikan. Rencana peneliti adalah melakukan perbaikan pada beberapa tahapan, terutama pada tahapan sebelum terjadinya kesalahan. Dengan demikian, peneliti tidak perlu memulai kembali dari awal hingga akhir dalam proses pengembangan perangkat lunak.

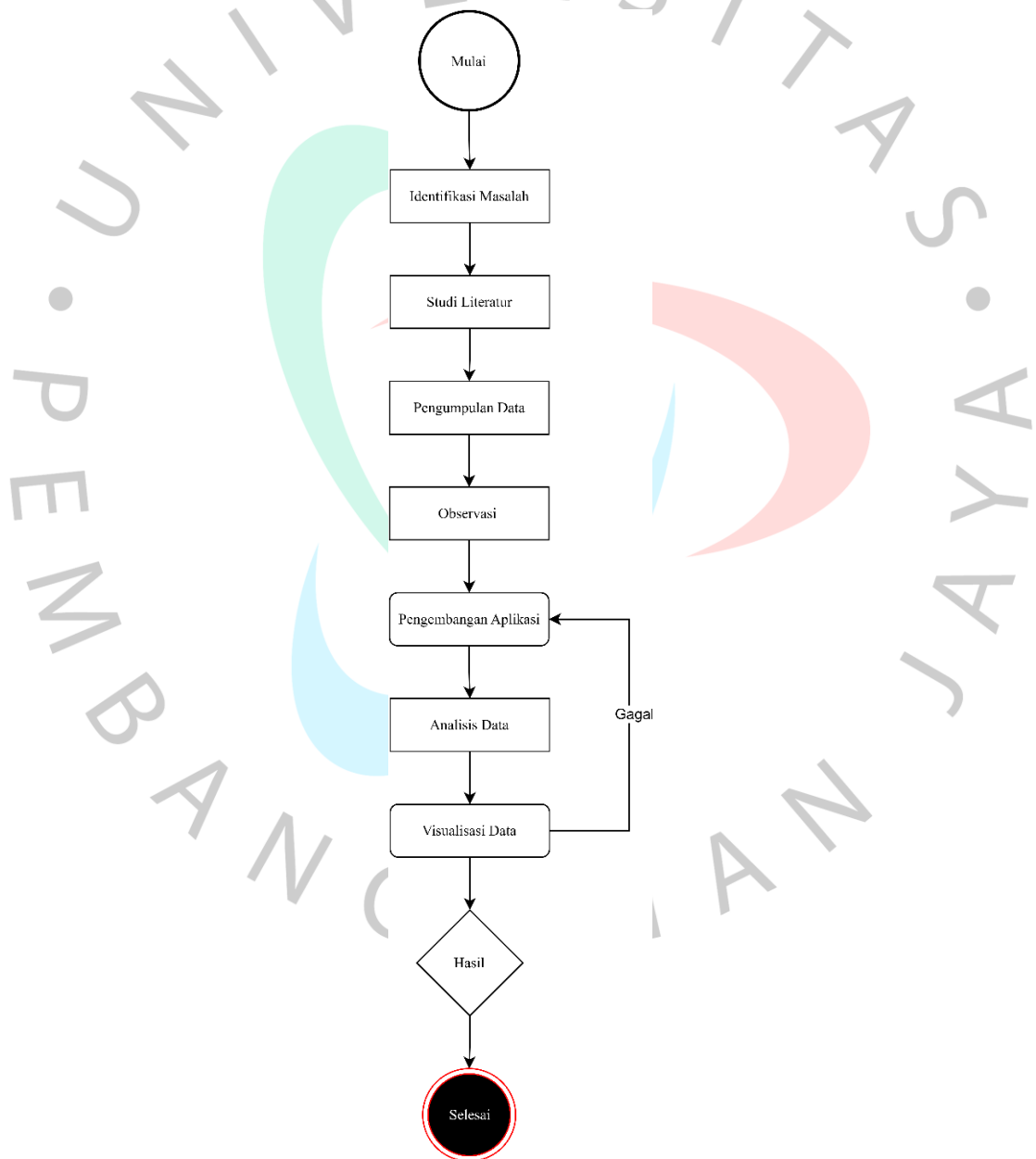
3.2 Sumber Data

Data yang digunakan dalam penelitian ini adalah data laporan kejadian penting PT. XYZ periode tahun 2022 sampai dengan September 2023. Data tersebut mencakup beberapa aspek penting, seperti jumlah kejadian kecelakaan, jenis-jenis

kecelakaan, lokasi kejadian, tingkat keparahan, korban dan kerugian yang timbul dari kejadian kecelakaan kerja.

3.3 Tahapan Pelaksanaan

Langkah-langkah pelaksanaan dalam Pengembangan Aplikasi Web untuk pengelolaan dan Analisis data *Health Safety and Environment* (HSE) terdapat beberapa tahapan. Berikut adalah diagram alir yang memvisualisasikan langkah-langkah pelaksanaan pada penelitian ini:



Gambar 3. 1 Tahapan Pelaksanaan

Tahapan pelaksanaan proyek ini dimulai dengan Identifikasi Masalah, di mana masalah atau tantangan yang akan diatasi dalam penelitian diidentifikasi secara jelas. Setelahnya, dilakukan Pengumpulan Data melalui metode observasi dan studi literatur untuk memperoleh informasi yang relevan dan mendalam terkait masalah yang diidentifikasi.

Langkah selanjutnya adalah pengembangan Web, di mana dilakukan pembangunan aplikasi web untuk menangani masalah tersebut. Setelah aplikasi web dikembangkan, dilanjutkan dengan tahap Analisis Data, *Machine Learning*, NLP (*Natural Language Processing*), dan Visualisasi Data. Tahapan ini bertujuan untuk mengolah dan menganalisis data menggunakan berbagai teknik, termasuk penerapan *machine learning* dan NLP, serta menampilkan hasilnya melalui visualisasi data.

- Dalam skenario jika terjadi kegagalan dalam tahapan Analisis Data, *Machine Learning*, NLP, atau Visualisasi Data, proses akan kembali ke tahap pengembangan aplikasi untuk melakukan perbaikan atau penyesuaian yang diperlukan. Namun, jika berhasil, maka proses dianggap selesai.

3.3.1 Studi Literatur

Studi literatur ini dirancang untuk mendapatkan dan menghimpun berbagai referensi yang relevan dengan tugas akhir. Hal ini mencakup penelitian sebelumnya, literatur, jurnal ilmiah, dan dokumen terkait lainnya. Fokus utama dari studi literatur adalah untuk memberikan panduan dalam melakukan penelitian dengan memperhatikan batasan masalah yang telah ditetapkan.

Lebih lanjut, pentingnya studi literatur juga terkait dengan kebutuhan pemahaman mendalam terhadap teknik yang diterapkan dalam tugas akhir. Ini mencakup pemahaman menyeluruh tentang proses pengambilan data hingga analisis model yang dihasilkan dari data pelatihan. Referensi yang dikumpulkan diharapkan dapat memberikan wawasan yang komprehensif mengenai metode klustering, pengembangan HSE (*Health, Safety, Environment*) untuk *platform* web, serta setiap tahap proses dan tantangan yang terkait dengan *Natural Language Processing*.

Studi literatur juga berkaitan erat dengan konsep klustering, aspek HSE dalam pengembangan web, dan permasalahan yang mungkin timbul dalam pemrosesan bahasa alami. Dengan merinci dan memahami literatur-literatur tersebut, diharapkan penelitian ini dapat ditempatkan dalam kerangka kerja dan mendukung implementasi metode yang tepat.

3.3.2 Pengumpulan Data

Data *Health Safety and Environment* (HSE) adalah kumpulan informasi yang sangat vital dalam konteks lingkungan kerja. Data ini mencakup beberapa aspek penting, seperti jumlah kejadian kecelakaan, jenis-jenis kecelakaan, lokasi kejadian, tingkat keparahan, dan data mengenai korban-korban kecelakaan. Jumlah kejadian kecelakaan mencerminkan tren yang ada, sehingga kita dapat mengevaluasi apakah ada peningkatan atau penurunan kecelakaan dari waktu ke waktu. Sementara informasi mengenai jenis-jenis kecelakaan membantu dalam mengidentifikasi area atau situasi kerja yang lebih berisiko terhadap kecelakaan bisa dilihat pada Tabel 4.1.

Tabel 4. 1 Isi Laporan Insiden

No	Elemen	Deskripsi
1	Date	Tanggal kejadian insiden kecelakaan kerja.
2	Regional Subsidiaries	Regional atau wilayah di mana kejadian terjadi
3	Location	Lokasi spesifik insiden
4	Sub Location	Sub-lokasi atau detail lokasi insiden.
5	Provinsi	Provinsi tempat kejadian terjadi
6	Location Category	Kategori lokasi insiden
7	Sub Location Category	Kategori sub-lokasi insiden
8	Incident Owner	Pemilik atau unit yang bertanggung jawab atas insiden.
9	Incident Description	Deskripsi rinci tentang bagaimana insiden terjadi.
10	Reportable/Non-Reportable	Status laporan insiden (Reportable atau Non-Reportable).
11	Recordable	Apakah insiden tersebut direkam atau tidak
12	Work Related	Apakah insiden tersebut terkait dengan pekerjaan.
13	Incident Severiy	Tingkat keparahan insiden
14	Incident Category	Kategori insiden

No	Elemen	Deskripsi
15	Incident Type	Jenis insiden
16	Loss Rp	Kerugian finansial dalam Rupiah
17	Spill Liter	Jumlah liter tumpahan (jika ada)
18	Wounded Victims	Jumlah korban luka
19	Dead Victims	Jumlah korban meninggal.
20	Causes	Penyebab insiden
21	LSR Category	Kategori risiko yang terkait
22	Process Safety Category	Kategori keselamatan proses
23	HIPO Category	Kategori insiden tinggi.
24	Pelaksana Investigasi	Orang yang bertanggung jawab melakukan investigasi.
25	Pelaksanaan Investigasi	Status pelaksanaan investigasi.
26	Moda	Kategori Pekerja

3.3.3 Pra-Pemrosesan Data

Pra-pemrosesan data (*pre-processing*) merupakan tahapan dalam pengelolaan dataset sebelum dilakukan analisis lebih lanjut. Pada tahap ini, data dari sumbernya diuji dan diperiksa untuk memastikan integritasnya sebelum disajikan dalam aplikasi. Proses ini mencakup pengecekan tipe data, identifikasi nilai yang hilang, serta penanganan *outlier* atau data yang tidak sesuai. Melalui *Streamlit*, pengguna dapat dengan mudah melakukan pemeriksaan terperinci terhadap struktur data, mengidentifikasi tipe data dari setiap kolom, dan mengevaluasi distribusi nilai-nilai dalam *dataset*.

Selain itu, pra-pemrosesan data juga mencakup langkah-langkah untuk mengatasi masalah kekosongan data atau *missing values*. *Streamlit* memungkinkan para pengguna untuk melihat visualisasi yang jelas tentang seberapa banyak data yang hilang dan memilih strategi penanganan yang sesuai, seperti pengisian nilai rata-rata atau menggunakan metode lainnya. Dengan memanfaatkan fitur-fitur *Streamlit*, pengguna dapat menjalankan operasi pra-pemrosesan data secara interaktif, membuatnya menjadi alat yang efektif dalam memastikan kualitas dan integritas data sebelum dilibatkan dalam proses analisis lebih lanjut.

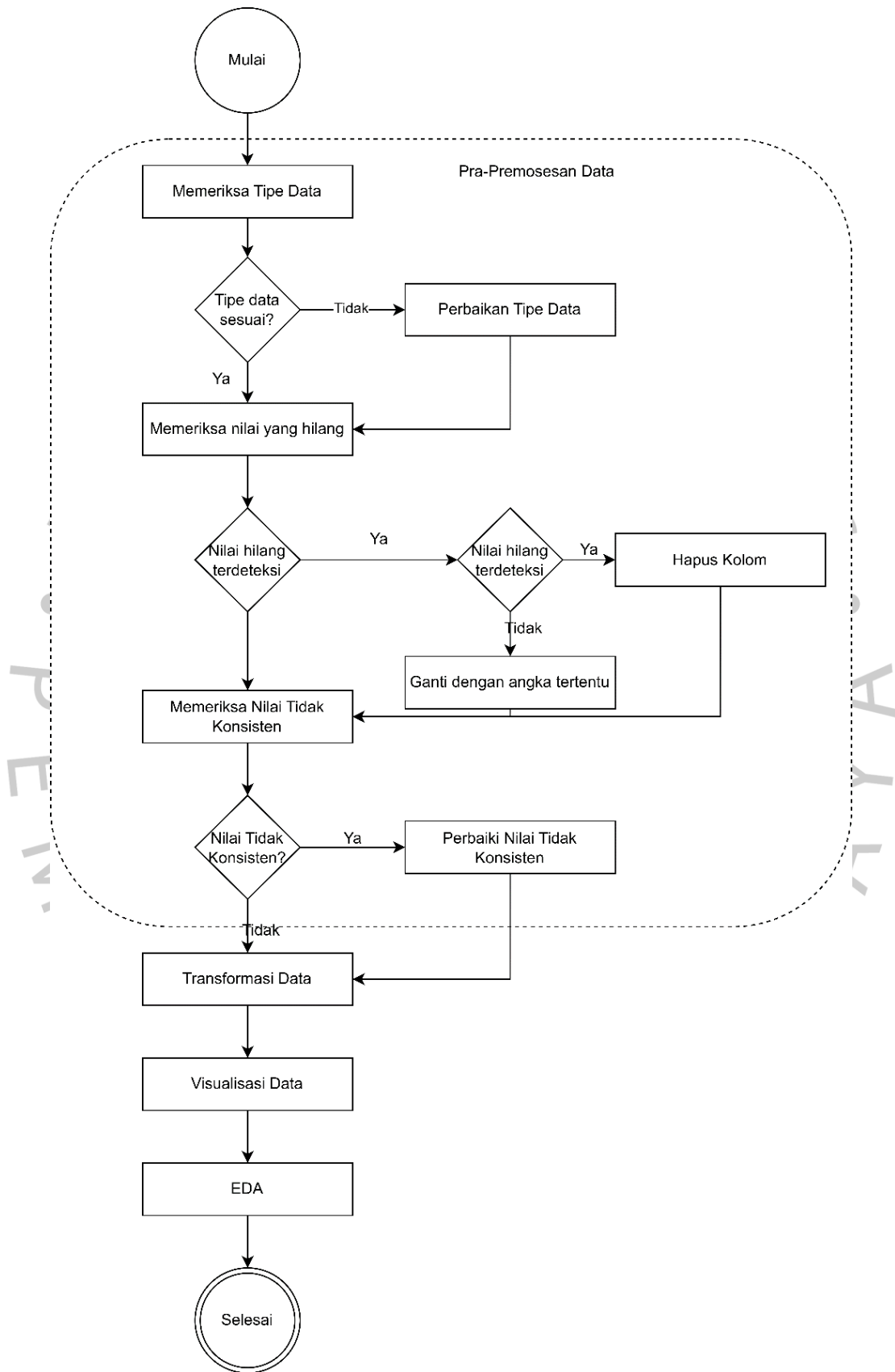
3.3.4 Pengembangan Aplikasi

Pengembangan aplikasi untuk pengelolaan dan analisis data menggunakan *Streamlit* memberikan solusi dan *user-friendly* bagi para pengguna yang ingin mengakses dan menganalisis data dengan mudah. *Streamlit* memungkinkan pengembang untuk dengan cepat membuat antarmuka pengguna yang responsif dan menarik tanpa memerlukan pengetahuan mendalam tentang pemrograman web. Dengan bantuan *Streamlit*, aplikasi dapat dibangun dengan fitur-fitur seperti pemrosesan dan visualisasi data, pengelolaan dataset, serta integrasi dengan berbagai alat analisis, sehingga memudahkan para analis data untuk mendapatkan wawasan yang signifikan dari dataset yang mereka tangani.

Kelebihan *Streamlit* terletak pada kemampuannya menyederhanakan proses pengembangan aplikasi web, sehingga pengembang dapat fokus pada logika aplikasi dan analisis data tanpa harus terjebak dalam detail teknis pengembangan. Antarmuka pengguna yang dapat diakses melalui browser memungkinkan pengguna untuk dengan mudah berinteraksi dengan aplikasi, menjadikannya solusi yang efektif bagi yang ingin mengoptimalkan pengelolaan dan analisis data tanpa harus menghadapi kompleksitas pengembangan web.

3.3.5 Analisis Data

Analisis data merupakan serangkaian langkah yang mencakup pengumpulan, pemrosesan, interpretasi, dan penyajian data dengan tujuan mendukung pengambilan keputusan. Dalam konteks ini, penerapan *Python* sebagai bahasa pemrograman menjadi bagian integral dari proses analisis data, melibatkan tahapan-tahapan tersebut untuk mencapai tujuan pengambilan keputusan.



Gambar 3.2 Tahapan Analisis Data

Pada Gambar 3.3, proses dilakukan dengan memanfaatkan berbagai *library* seperti *Pandas* untuk data dalam format CSV atau Excel. Setelah data dimuat, langkah berikutnya adalah melakukan pemeriksaan data untuk menjamin integritas dan kualitasnya.

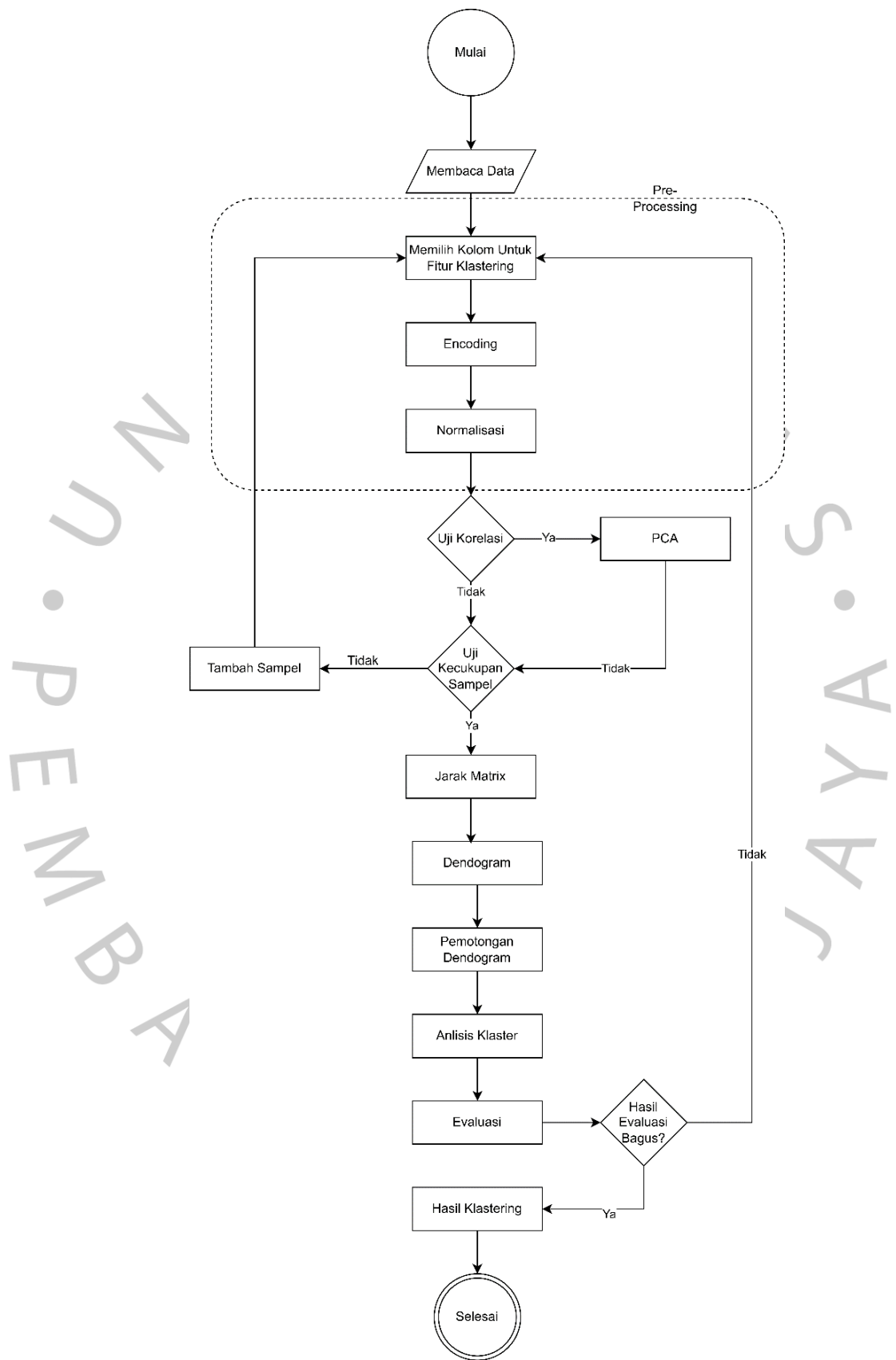
Pemeriksaan data melibatkan serangkaian langkah, termasuk memeriksa ukuran data, jumlah baris dan kolom, tipe data, nilai-nilai yang hilang, nilai-nilai yang tidak valid, dan nilai-nilai yang tidak konsisten. Tujuan dari tahap ini adalah untuk mengidentifikasi potensi masalah atau anomali dalam data. Jika selama pemeriksaan data ditemukan masalah, langkah berikutnya adalah membersihkan data. Pembersihan data melibatkan tindakan seperti mengisi data yang hilang, mengubah data yang tidak valid, atau menggabungkan data yang tidak konsisten.

Setelah data dibersihkan, langkah berikutnya adalah transformasi data, jika diperlukan. Transformasi data dapat mencakup penggabungan data dari beberapa sumber, pemisahan data menjadi beberapa bagian, atau konversi data dari satu format ke format lain. Transformasi data bertujuan untuk menyesuaikan struktur data agar sesuai dengan kebutuhan analisis. Dengan data yang telah dimuat, diperiksa, dibersihkan, dan diubah (jika perlu), tahap terakhir adalah eksplorasi data (EDA).

Eksplorasi data (EDA) merupakan langkah penting dalam analisis data karena memberikan wawasan mendalam tentang karakteristik dan pola data. EDA melibatkan analisis deskriptif untuk memahami statistik dasar, visualisasi data untuk mengidentifikasi pola dan tren, serta analisis statistik untuk menguji hipotesis tentang data.

1. *Machine Learning*

Implementasi *Machine Learning* yang memungkinkan analisis data HSE yang lebih efisien dan mendalam. Salah satu pendekatan yang efektif dalam menganalisis data HSE adalah menggunakan *Agglomerative Hierarchical Clustering*. Mulai dari tahap *pre-processing* data hingga evaluasi kinerja model, pada Gambar 3.3 merupakan langkah-langkah menerapkan *Machine Learning* menggunakan *Agglomerative Hierarchical Clustering*.



Gambar 3. 3 Tahapan Proses Machine Learning

a. *Encoding*

Sebelum melibatkan data dalam proses analisis kluster, langkah kritis pertama adalah persiapan data yang melibatkan *encoding* dan normalisasi. *Encoding* diperlukan untuk mengubah variabel kategori atau *string* menjadi format numerik agar dapat dimengerti oleh algoritma klusterisasi. Misalnya, dengan menggunakan metode *Label Encoding*, kategori dapat diwakili sebagai nilai numerik.

b. Pengujian Korelasi

Korelasi adalah suatu metrik yang mengukur sejauh mana hubungan yang erat antara variabel-variabel. Dalam konteks analisis *cluster*, penting untuk memastikan bahwa variabel yang digunakan sebagai *input* analisis *cluster* bersifat independen satu sama lain. Oleh karena itu, dilakukan pengujian korelasi Pearson sesuai dengan rumus-rumusnya (2.2), sebagai salah satu syarat yang harus dipenuhi dalam analisis *cluster*.

Ketika nilai korelasi mendekati 0, hal ini menandakan bahwa terdapat korelasi yang lemah atau bahkan tidak ada korelasi sama sekali antara dua variabel. Uji korelasi Pearson digunakan untuk menguji secara statistik apakah korelasi antara variabel tersebut berbeda secara signifikan dari nol atau tidak. Hipotesis nol (H_0) menyatakan bahwa tidak ada korelasi antara variabel, sedangkan hipotesis alternatif (H_1) menyatakan bahwa ada korelasi antara variabel tersebut.

Melalui uji hipotesis ini, kita dapat menentukan apakah nilai korelasi antara variabel signifikan secara statistik. Jika nilai p yang dihasilkan dari uji hipotesis lebih kecil dari tingkat signifikansi yang ditetapkan (biasanya 0.05), kita dapat menolak hipotesis nol dan menyimpulkan bahwa terdapat korelasi yang signifikan antara variabel tersebut. Sebaliknya, jika nilai p lebih besar dari tingkat signifikansi, kita tidak memiliki cukup bukti statistik untuk menyatakan bahwa terdapat korelasi yang signifikan antara variabel tersebut.

c. Normalisasi Data

Selanjutnya, normalisasi menjadi langkah penting untuk memastikan bahwa variabel memiliki skala yang sebanding dalam analisis kluster. L1 Normalisasi dan

L2 Normalisasi adalah metode yang digunakan untuk mengubah setiap nilai dalam suatu baris data dengan menggambarkan data tersebut sebagai vektor unit. L1 Normalisasi menghasilkan vektor dengan jumlah nilai absolut setiap baris sama dengan 1, sementara L2 Normalisasi menghasilkan vektor dengan panjang *Euclidean* 1.

Selain itu, terdapat metode normalisasi seperti *Min-Max Scaling*, di mana nilai-nilai dalam suatu variabel diubah ke dalam rentang tertentu, seringkali antara 0 dan 1. *Z-Score*, atau *Standard Scaling*, memusatkan dan menskala data sehingga memiliki rata-rata nol dan standar deviasi satu.

d. Analisis Komponen Utama

Analisis Komponen Utama (*Principal Component Analysis* atau PCA) digunakan untuk mengubah variabel yang saling berkorelasi menjadi variabel baru yang tidak memiliki korelasi satu sama lain. Menurut Richard A. Johnson, (2001), analisis komponen utama digunakan untuk mengidentifikasi variabel baru yang mendasari data variabel ganda, mengurangi jumlah variabel yang banyak dan saling berkorelasi menjadi variabel baru yang tidak berkorelasi, sambil tetap mempertahankan sebanyak mungkin keragaman data dan menghapus variabel asal yang tidak memberikan informasi yang signifikan.

Variabel baru yang terbentuk memiliki beberapa karakteristik, seperti merupakan kombinasi linier dari variabel asal, jumlah kuadrat koefisien dalam kombinasi linier tersebut sama dengan satu, tidak saling berkorelasi, dan ragamnya diurutkan dari yang terbesar hingga yang terkecil. Variabel-variabel baru ini memanfaatkan informasi dari variabel-variabel asal, dan nilai yang diperoleh untuk setiap objek mencerminkan ordinat objek-objek tersebut dalam variabel baru, yang bertindak sebagai sumbu-sumbu koordinat.

e. Pembentukan Matriks Jarak

Menghitung matriks jarak adalah langkah kunci dalam analisis kluster, digunakan untuk mengukur seberapa dekat atau jauh setiap pasangan data dalam kumpulan data. Konsep dasarnya adalah mengukur "jarak" antara dua titik data,

mencerminkan seberapa mirip atau berbeda keduanya. Beberapa metode umum digunakan untuk mengukur jarak antara dua titik data.

Euclidean Distance mengukur panjang garis lurus antara dua titik dalam ruang *Euclidean*. Ini sering diartikan sebagai "jarak sejati" antara dua titik, seperti panjang garis lurus yang menghubungkan titik koordinat dalam ruang dua dimensi. Selanjutnya, *Manhattan Distance* diukur sebagai jumlah perbedaan absolut antara koordinat titik dalam setiap dimensi. Ini menggambarkan jarak sepanjang *grid* jalan kota, contohnya adalah jumlah jarak horizontal dan vertikal di kota untuk mencapai satu titik dari yang lain.

Cosine Similarity mengukur sudut antara dua vektor, dengan sudut yang lebih kecil menunjukkan kesamaan yang lebih besar. Sebagai contoh, jika kita memiliki vektor yang mewakili properti numerik dari dua objek, *cosine similarity* antara vektor-vektor tersebut menunjukkan sejauh mana kedua objek tersebut serupa dari segi properti numerik.

Matriks jarak ini memberikan dasar pemahaman tentang seberapa mirip atau berbeda pasangan data. Informasi ini selanjutnya dapat dimanfaatkan dalam proses klusterisasi untuk membentuk kelompok data yang serupa.

i. Pembentukan *Dendrogram*

Pembentukan *dendrogram* adalah langkah berikutnya dalam proses analisis klaster yang melibatkan pembuatan representasi visual dari hierarki hubungan antara data. *Dendrogram* adalah sebuah struktur pohon yang memperlihatkan bagaimana setiap elemen data atau klaster terkait satu sama lain.

Dalam konteks analisis klaster, *dendrogram* digunakan untuk menggambarkan hubungan hierarkis antara data atau klaster. Langkah-langkahnya melibatkan penyusunan dan pengelompokan data berdasarkan tingkat kesamaan mereka. Pada tingkat awal, setiap data atau observasi dianggap sebagai klaster terpisah. Selanjutnya, dua klaster yang paling mirip atau sering disebut sebagai "anak" dari suatu "induk" dihubungkan dalam struktur *dendrogram*. Proses ini terus berlanjut, dan cabang-cabang *dendrogram* merepresentasikan tingkat kesamaan antara data atau klaster yang lebih tinggi. Semakin dekat dua cabang pada

dendrogram, semakin mirip atau dekat hubungan antara kluster tersebut. Sebaliknya, semakin jauh dua cabang, semakin berbeda atau jauh hubungan antara kluster tersebut.

Dendrogram membentuk visualisasi yang intuitif dan informatif tentang cara data dikelompokkan secara hierarkis. Pemahaman struktur *dendrogram* dapat membantu pemilihan jumlah kluster yang optimal dalam tahap pemotongan *dendrogram* selanjutnya, sehingga menjadi langkah penting dalam analisis kluster.

f. Pemotongan *Dendrogram*

Setelah pembentukan *dendrogram*, langkah selanjutnya dalam analisis kluster adalah pemotongan *dendrogram*. Pemotongan *dendrogram* dilakukan untuk menentukan jumlah kluster yang dihasilkan dari data atau observasi yang ada. Ini merupakan tahap kritis dalam proses klusterisasi karena menentukan bagaimana data akan dikelompokkan ke dalam kluster.

Pemotongan *dendrogram* dilakukan dengan cara memilih tingkat atau cabang tertentu pada struktur *dendrogram*. Tingkat pemotongan ini akan menentukan seberapa banyak kluster yang akan dihasilkan. Semakin tinggi tingkat pemotongan, semakin sedikit kluster yang terbentuk, dan sebaliknya.

Keputusan pemotongan *dendrogram* dapat dipengaruhi oleh berbagai faktor, termasuk tujuan analisis dan konteks aplikasi. Misalnya, jika tujuan adalah mendapatkan kluster yang lebih halus dan spesifik, tingkat pemotongan dapat dipilih lebih rendah. Sebaliknya, jika kebutuhan aplikasi adalah memperoleh kluster yang lebih besar dan umum, tingkat pemotongan dapat ditingkatkan.

Pemotongan *dendrogram* juga dapat melibatkan penggunaan metode atau kriteria tertentu untuk membantu menentukan jumlah kluster yang optimal. Penggunaan indeks validitas seperti *Silhouette Index* atau *Davies-Bouldin Index* dapat menjadi panduan dalam mengevaluasi hasil pemotongan *dendrogram*. Pemotongan *dendrogram* adalah langkah krusial dalam membentuk kluster dan memberikan landasan untuk tahap evaluasi selanjutnya dalam analisis kluster.

g. Evaluasi

Evaluasi merupakan langkah dalam analisis kluster, bertujuan untuk mengukur dan menilai kualitas hasil klusterisasi yang telah terbentuk. Terdapat beberapa metode evaluasi yang digunakan untuk mengevaluasi sejauh mana kluster mencerminkan struktur sebenarnya dalam data. Salah satu metode yang umum digunakan adalah Indeks Silhouette, yang memberikan Gambaran seberapa baik setiap objek dalam klusternya dan seberapa berbeda kluster tersebut satu sama lain. Nilai indeks Silhouette berkisar antara -1 hingga 1, di mana nilai lebih tinggi menandakan hasil klusterisasi yang lebih baik.

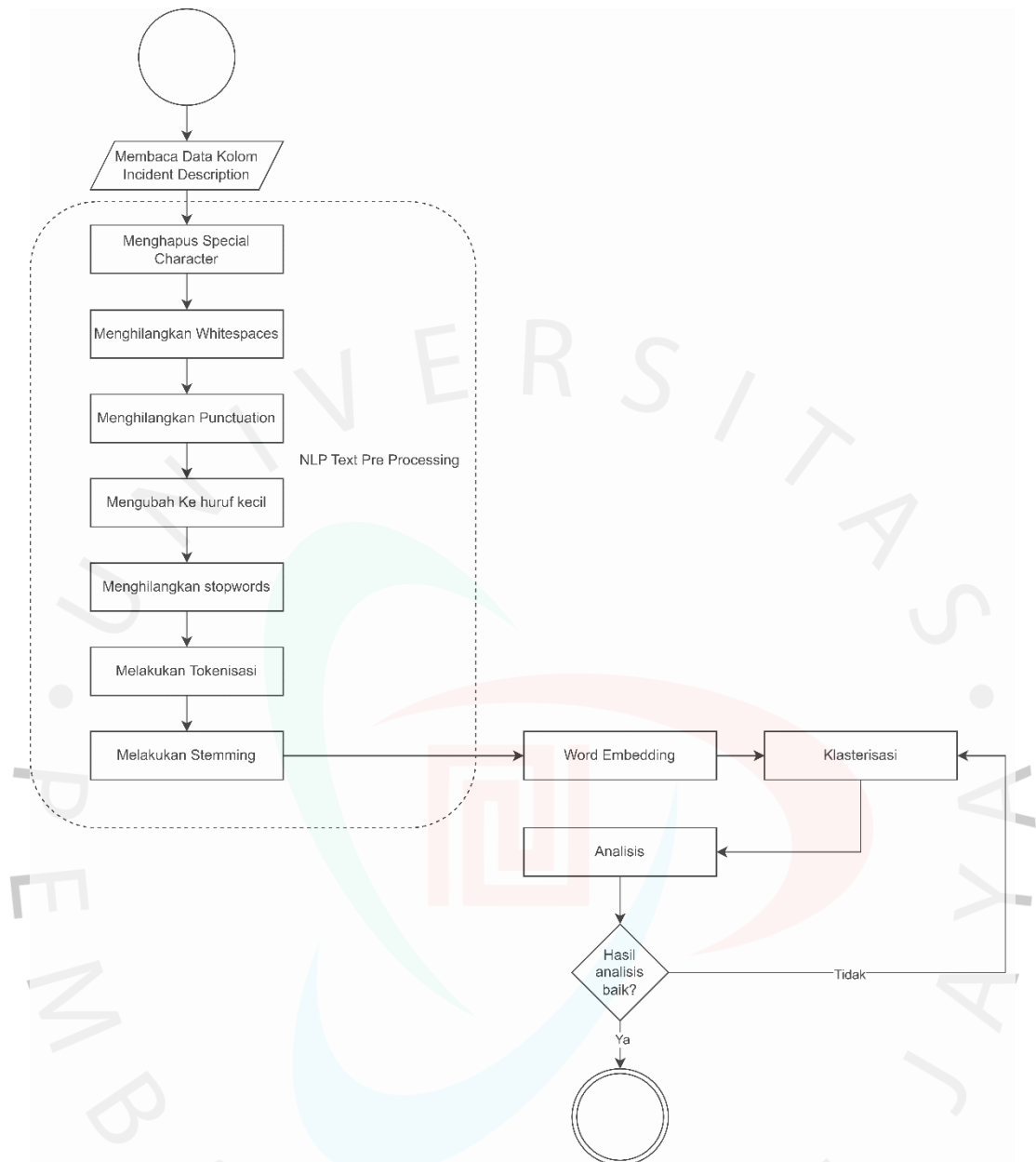
Selain itu, Indeks *Davies-Bouldin* digunakan untuk mengevaluasi sejauh mana kluster terpisah dan tidak saling tumpang tindih. Nilai rendah pada indeks ini mengindikasikan pembentukan kluster yang lebih baik. Sementara Indeks *Calinski-Harabasz* mengukur kesejajaran dan kompaknya kluster, dengan nilai tinggi menunjukkan hasil klusterisasi yang baik.

Koefisien Korelasi *Cophenetic* digunakan untuk mengevaluasi seberapa baik *dendrogram* mempertahankan jarak antara data asli. Nilai tinggi pada koefisien ini menandakan *dendrogram* yang lebih baik. Selain metrik kuantitatif, analisis visual juga merupakan aspek penting dalam evaluasi. Visualisasi kluster dan distribusi data dalam kluster, seperti *scatter plot* atau *box plot*, dapat memberikan pemahaman yang lebih mendalam tentang struktur hasil klusterisasi.

Pemilihan metode evaluasi yang tepat bergantung pada konteks dan tujuan analisis kluster. Dengan memanfaatkan beberapa metode evaluasi, kita dapat mendapatkan pemahaman yang lebih holistik tentang keberhasilan klusterisasi. Evaluasi yang cermat memastikan bahwa hasil klusterisasi sesuai dengan kebutuhan dan tujuan analisis yang telah ditentukan.

2. *Natural Language Processing (NLP)*

Implementasi *Natural Language Processing (NLP)* untuk menggali informasi dari kolom *Incident Description*. Dengan memanfaatkan NLP memungkinkan kita untuk menyelami makna dan konteks di balik setiap deskripsi kejadian. Penerapan NLP sebagaimana ditunjukkan pada Gambar 3.4.



Gambar 3. 4 Tahapan Analisis Menggunakan NLP

a. Membaca Data

Pertama-tama, dalam pemrosesan bahasa alami (NLP), langkah awalnya adalah membaca data dari kolom *Incident Description*. Proses ini melibatkan pengambilan teks atau informasi dari kolom tersebut untuk kemudian diolah lebih lanjut. Langkah ini menjadi landasan awal dalam menerapkan teknik-teknik NLP, seperti tokenisasi, pemrosesan kata, dan analisis teks. Dengan mengakses dan

membaca data dari kolom Incident Description, kita dapat menjalankan serangkaian langkah selanjutnya dalam rangkaian pemrosesan bahasa alami untuk mendapatkan wawasan dan informasi yang berharga dari teks yang ada.

b. *Pre-Processing Data*

Tahapan ini melibatkan serangkaian langkah untuk membersihkan dan mempersiapkan data teks sebelum dilakukan analisis atau pemodelan lebih lanjut. Berikut adalah langkah-langkah *pre-processing* yang dapat diidentifikasi dari deskripsi:

1) Menghapus *Special Character*

Menghilangkan karakter khusus dari teks, seperti tanda baca atau simbol tertentu, yang mungkin tidak memberikan nilai tambah pada analisis.

2) Menghilangkan *Whitespaces*

Membersihkan *whitespaces* atau spasi yang tidak diinginkan dalam teks.

3) Menghilangkan *Punctuation*

Menghapus tanda baca dari teks untuk memfasilitasi pemrosesan lebih lanjut.

4) Mengubah Ke Huruf Kecil

Merubah seluruh teks menjadi huruf kecil agar konsistensi dalam representasi kata.

5) Menghilangkan *StopWords*

Menghapus kata-kata stop (kata umum yang sering muncul dan biasanya diabaikan karena kurangnya informasi)

6) Melakukan Tokenisasi

Tokenisasi adalah langkah dalam pemrosesan teks yang mengimplikasikan pemisahan dokumen menjadi unit-unit kecil yang disebut token, yang biasanya merupakan kata-kata. Proses tokenisasi ini bertujuan untuk menciptakan representasi vektor dari setiap kata, di mana setiap token memiliki nilai numerik yang menggambarkan atribut atau makna tertentu. Dengan cara ini, tokenisasi memungkinkan kata-kata dalam teks direpresentasikan secara terstruktur dalam bentuk vektor untuk keperluan analisis lebih lanjut atau pemodelan data teks.

c. *Word Embedding*

Mengubah kata-kata menjadi representasi vektor numerik dalam ruang multidimensional. Dengan menggunakan model Word2Vec, setiap kata diberikan vektor yang mencerminkan hubungan semantiknya dengan kata-kata lain.

d. Klasterisasi Text

Klasterisasi teks melibatkan pengelompokan dokumen atau teks berdasarkan kemiripan isinya. Metode klasterisasi seperti *K-Means* atau *Hierarchical Clustering* digunakan untuk membentuk kelompok kata-kata atau dokumen yang serupa.

e. Analisis

Hasil dari klasterisasi teks memberikan pemahaman tentang pola atau kelompok yang muncul dalam dataset teks. Klasterisasi bertujuan untuk membentuk kelompok kata-kata atau dokumen yang memiliki kemiripan isinya. Dengan menganalisis hasil klasterisasi, kita dapat mengidentifikasi tematik atau topik utama yang muncul dalam teks.

Setiap klaster mewakili sekelompok kata atau dokumen yang memiliki kesamaan konten, sehingga dapat membantu dalam mengelompokkan informasi yang serupa atau saling terkait. Analisis lebih lanjut pada setiap klaster dapat memberikan wawasan mendalam terkait dengan topik atau pola tertentu yang ada dalam data teks. Selain itu, memahami distribusi dan karakteristik setiap klaster dapat memberikan konteks yang diperlukan untuk menafsirkan hasil klasterisasi.

Dengan demikian, analisis hasil klasterisasi teks menjadi kunci untuk mendapatkan pemahaman yang lebih baik terhadap struktur dan makna yang terkandung dalam dataset teks, membantu pengguna untuk mengambil keputusan atau mengeksplorasi informasi dengan lebih efektif.

3.3.6 EDA

Exploratory Data Analysis (EDA) adalah suatu pendekatan kritis dalam analisis data yang bertujuan untuk memahami sifat-sifat dasar dari dataset yang dimiliki. Tujuan utama dari EDA adalah mengungkap pola-pola menarik,

mengidentifikasi anomali, dan memahami hubungan antar variabel sebelum melibatkan analisis statistik yang lebih mendalam atau membangun model prediktif. Dalam proses EDA, kita secara rinci memeriksa struktur data, mencari anomali atau kesalahan dalam *dataset*, menemukan pola dan tren, serta mengidentifikasi hubungan antar variabel. Analisis ini memberikan pemahaman mendalam tentang karakteristik data, membantu dalam pengambilan keputusan terkait pemilihan variabel untuk analisis lebih lanjut, dan memvalidasi asumsi yang mungkin telah dibuat. Temuan dari EDA disajikan melalui visualisasi dan laporan, mempermudah komunikasi temuan kepada pemangku kepentingan atau rekan tim. Selain itu, EDA juga membantu dalam persiapan data dengan memberikan panduan terkait penanganan nilai-nilai yang hilang, pengelolaan *outlier*, dan transformasi variabel. Melalui pendekatan ini, EDA menjadi langkah awal yang esensial dalam proses analisis data.

3.3.7 Visualisasi Data

Dalam konteks Insiden Kecelakaan Kerja dan data HSE (*Health, Safety, Environment*), visualisasi data menjadi suatu teknik penting untuk menyajikan informasi terkait kejadian insiden dengan lebih jelas dan efektif. Teknik ini memanfaatkan grafis seperti grafik, diagram, atau peta untuk menggambarkan pola, tren, dan hubungan dalam data kecelakaan kerja. Tujuannya adalah untuk memudahkan pemahaman dan analisis, sehingga para pemangku kepentingan di bidang HSE dapat dengan lebih intuitif mengidentifikasi pola kompleks dan karakteristik dari *dataset* insiden.

Keuntungan utama dari penggunaan visualisasi data dalam konteks HSE termasuk kemampuan untuk memahami struktur data insiden yang kompleks, mendeteksi pola dan tren kecelakaan kerja, berkomunikasi efektif dengan pihak-pihak terkait di lingkungan kerja, membuat presentasi hasil analisis insiden yang menarik, serta membandingkan kuantitas insiden antar waktu, lokasi, atau jenis kejadian dengan lebih mudah.

Jenis-jenis visualisasi data yang relevan dalam analisis insiden kecelakaan kerja meliputi grafik garis untuk melihat tren kecelakaan seiring waktu, diagram

batang untuk membandingkan jumlah insiden antar lokasi atau jenis kejadian, diagram lingkaran untuk memvisualisasikan proporsi jenis kecelakaan.

Dalam konteks HSE, visualisasi data tidak hanya membantu dalam langkah awal eksplorasi data insiden, tetapi juga menjadi alat yang efektif dalam menyampaikan hasil analisis insiden kepada pihak-pihak terkait dengan latar belakang pengetahuan yang beragam di bidang kesehatan, keselamatan, dan lingkungan kerja.

3.4 Teknik Analisis Data

Dalam upaya merancang aplikasi web untuk pengelolaan dan analisis data *Health Safety and Environment* (HSE), proses tersebut melibatkan beberapa tahapan. Tahap pertama melibatkan analisis data dengan memanfaatkan *machine learning*, *Natural Language Processing* (NLP), dan visualisasi data. Tujuan penggunaan *machine learning* pada tahap ini adalah untuk mengidentifikasi pola dan tren signifikan dalam data HSE. Algoritma *Agglomerative Hierarchical Clustering* dipilih untuk klusterisasi pada regional subsidiaries dan lokasi berdasarkan tingkat keparahan insiden (Dharma Nugraha dkk., 2023). Langkah ini diharapkan dapat memberikan wawasan mendalam terkait sebaran insiden di berbagai wilayah, memungkinkan perusahaan untuk mengambil tindakan preventif yang lebih tepat.

Dalam konteks NLP, tahapan *pre-processing* menjadi kunci, di mana *incident description* melibatkan serangkaian langkah seperti *pre-processing*, tokenisasi, *embedding*, dan klasifikasi (Chang dkk., 2021). Sejalan dengan itu, visualisasi data menjadi elemen penting, dengan pembuatan *C&T Dashboard*, *Fleet Dashboard*, dan *Trendline Incident Report* sebagai langkah konkret. Hasil dari visualisasi tersebut nantinya akan dilaporkan secara periodik kepada senior analisis HSE dalam bentuk *weekly report*, *monthly report*, dan *annual report*.

Demi menjaga kelancaran perancangan dan pengembangan aplikasi, keberadaan spesifikasi sistem yang jelas dan komprehensif sangat diperlukan. Spesifikasi tersebut mencakup aspek-aspek vital seperti proses, data, pengguna, kebutuhan perangkat keras, kebutuhan perangkat lunak, hingga berbagai dokumen pendukung lainnya. Dengan adanya kerangka spesifikasi yang kokoh, diharapkan

pengembang dapat lebih efisien dalam merancang aplikasi web yang responsif dan sesuai dengan kebutuhan manajemen HSE.

3.4.1 Spesifikasi Proses

Dalam merancang aplikasi *Health Safety and Environment* (HSE) berbasis web, langkah-langkah yang ditempuh sejalan dengan kerangka penelitian. Tahap awal melibatkan Identifikasi Masalah, di mana masalah potensial dalam pengelolaan dan analisis data HSE diidentifikasi. Pengumpulan data, observasi, dan studi literatur dilakukan untuk memahami konteks dan tantangan yang dihadapi.

Tahapan berikutnya adalah pembuatan website, diikuti analisis data dan penerapan *machine learning*. Algoritma *Agglomerative Hierarchical Clustering* dipilih untuk klusterisasi regional subsidiaries dan lokasi berdasarkan tingkat keparahan insiden.

- Dalam konteks *Natural Language Processing* (NLP), tahapan *preprocessing* menjadi kunci untuk mengolah *incident description*. Ini melibatkan langkah-langkah seperti *preprocessing*, *tokenisasi*, *embedding*, dan klasifikasi. Selanjutnya, visualisasi data menjadi elemen penting, dengan *C&T Dashboard*, *Fleet Dashboard*, dan *Trendline Incident Report* sebagai langkah-langkah konkret.

Dengan hasil visualisasi tersebut, laporan secara periodik dalam bentuk *weekly report*, *monthly report*, dan *annual report* akan disampaikan kepada senior analisis HSE. Keseluruhan kerangka penelitian ini dirancang untuk menyusun aplikasi HSE yang responsif dan efektif dalam pengelolaan dan analisis data.

3.4.2 Analisis Pengguna

Dalam pengembangan aplikasi ini, proses identifikasi hak akses pengguna menjadi elemen penting yang membedakan interaksi mereka dengan aplikasi. Setiap pengguna memiliki hak akses yang berbeda sesuai dengan peran dan tanggung jawab mereka. Tabel ini mencantumkan jenis hak akses yang dimiliki oleh setiap pengguna, mulai dari akses penuh hingga akses terbatas. Untuk memberikan pemahaman yang lebih mendalam tentang hal ini, kami telah merinci spesifikasi hak akses pengguna sebagaimana di tunjukan pada Tabel 4.2.

Tabel 4. 2 Spesifikasi Pengguna

No.	Hak Akses	Deskripsi
1	Mengelola Data	Admin dapat mengakses, mengelola, dan mengedit data terkait kecelakaan kerja.
2	Menerapkan Analisis	Admin dapat menggunakan fitur analisis aplikasi untuk menganalisis data kecelakaan kerja menggunakan teknik ML.
3	Mengelola <i>Privilege</i>	Admin dapat mengatur hak akses dan <i>privilege</i> pengguna lain dalam aplikasi.
4	Melihat Laporan	Admin dapat melihat laporan hasil analisis data kecelakaan kerja.
5	Menyimpan Data	Admin dapat menyimpan data kecelakaan kerja yang telah dianalisis.
6	Mengelola Pengaturan	Admin dapat mengatur pengaturan aplikasi, seperti konfigurasi sistem dan preferensi pengguna.

Dalam aplikasi ini, hanya ada satu pengguna yaitu Admin yang juga sebagai data analis. Admin memiliki hak akses penuh terhadap seluruh menu untuk menjaga keamanan dan kerahasiaan data, serta mencegah penyalahgunaan hak akses yang dapat merusak integritas sistem. Sistem akan melakukan autentikasi identitas pengguna untuk memastikan perbedaan hak akses dan menerapkan hak akses pengguna sesuai yang telah ditetapkan.

3.4.3 Analisis Perangkat Keras

Dalam proses pengembangan aplikasi ini, pengembang membutuhkan spesifikasi perangkat keras yang direkomendasikan untuk pengembangan web pengelolaan dan analisis data HSE. Berikut adalah detail spesifikasi perangkat keras sebagaimana di tunjukan pada Tabel 4.3

Tabel 4. 3 Spesifikasi Perangkat Keras

Komponen	Spesifikasi
Prosesor	Intel Core i5 atau setara
RAM	Minimal 8 GB, Rekomendasi 16 GB atau lebih
Penyimpanan	Minimal 256 GB SSD, Rekomendasi 512 GB SSD atau lebih
Kartu Grafis	Integrated Graphics (Intel HD Graphics atau setara) atau Dedicated Graphics (Nvidia GeForce GTX atau setara) dengan minimal 4 GB VRAM
Layar	Resolusi HD (1280x720) atau lebih tinggi
Sistem Operasi	Windows 10
Koneksi Internet	Koneksi internet dengan kecepatan stabil

3.4.4 Analisis Perangkat Lunak

Selain perangkat keras, aplikasi juga memerlukan dukungan perangkat lunak untuk menggunakan berbagai alat dan teknologi seperti database, bahasa pemrograman, dan lainnya. Untuk melihat spesifikasi perangkat lunak yang diperlukan bisa dilihat pada Tabel 4.4.

Tabel 4. 4 Spesifikasi Perangkat Lunak

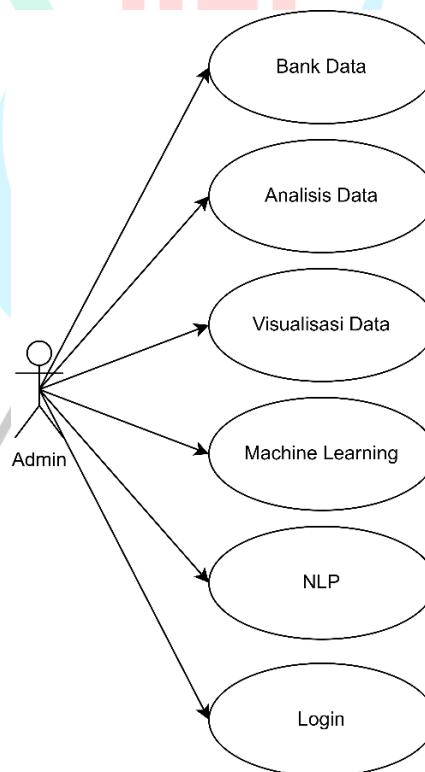
Komponen	Spesifikasi
Database	MySQL
Bahasa Pemrograman	Python
Framework	Streamlit
Editor Pemrograman	Visual Studio Code
Sistem Operasi	Windows 10
Tools dan Libraries	Visual Studio Code, Pustaka Python

3.5 Perancangan Sistem

Dalam proses pengembangan aplikasi, tahap perancangan memainkan peran yang sangat krusial untuk memahami dan menggambarkan mekanisme kerja sistem secara menyeluruh. *Unified Modeling Language (UML)* sebagai alat yang sangat berharga bagi para pengembang dalam konteks ini. Penggunaan UML memungkinkan mereka merinci proses-proses dan interaksi antara *class-class* dengan lebih terstruktur dan jelas. Dengan visualisasi yang diberikan oleh UML, pengembang dapat dengan mudah mengidentifikasi hubungan antar *class*, implementasi fungsi-fungsi kunci, dan alur data dalam aplikasi.

a. Use Case Diagram

Pemanfaatan kasus penggunaan (*use case*) memiliki kegunaan dalam menjelaskan interaksi antara aktor yang terlibat dengan sistem yang sedang dikembangkan, serta menjelaskan secara rinci fungsi-fungsi yang dimiliki oleh sistem tersebut. Ilustrasi pada Gambar 3.5 berupa diagram kasus penggunaan memberikan representasi visual tentang aktivitas yang dilakukan oleh aktor dan fitur-fitur yang dapat diakses dalam sistem.

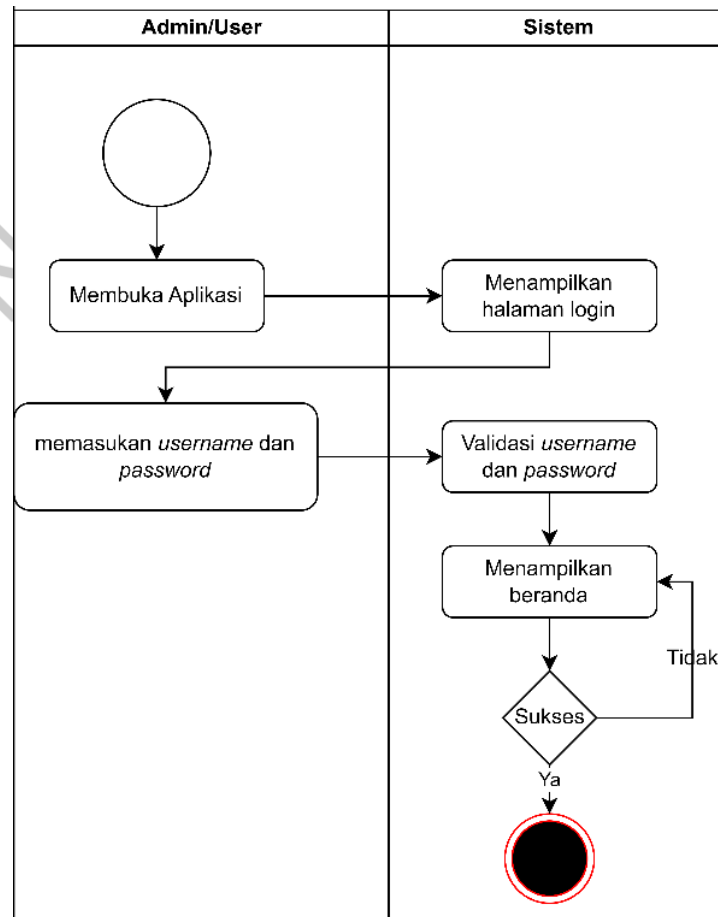


Gambar 3. 5 Use Case Diagram

b. *Activity Diagram*

1) *Activity Diagram Login*

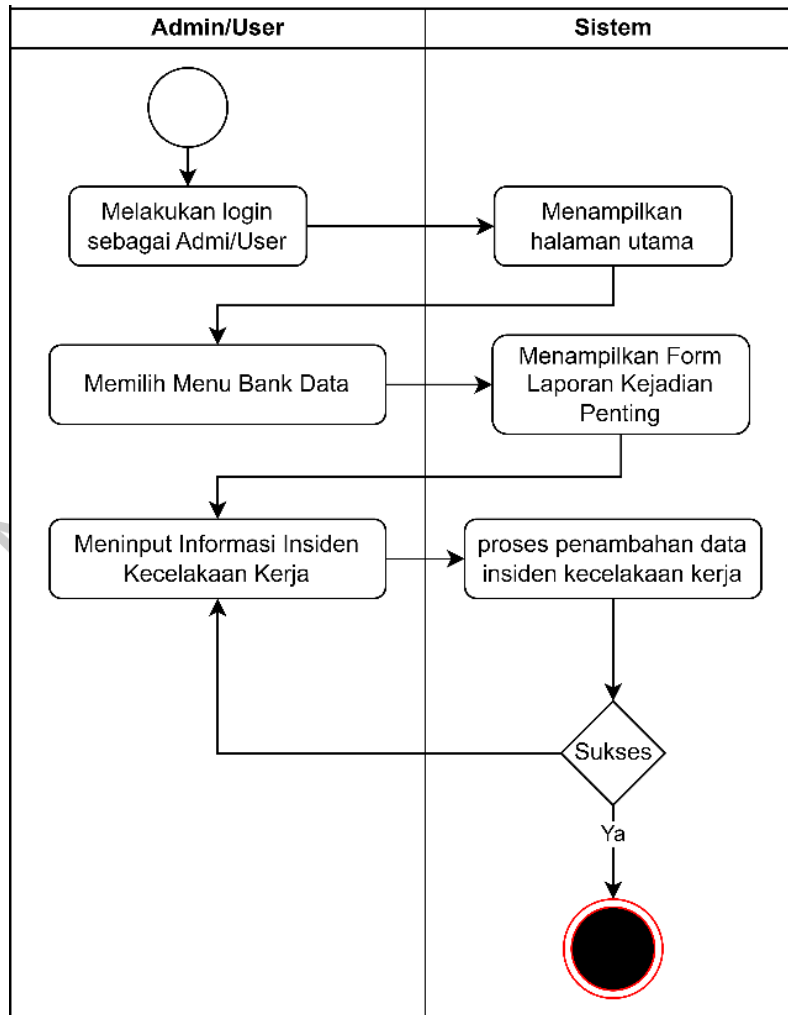
Activity diagram untuk login menggambarkan alur kerja atau aktivitas yang terlibat dalam proses *login* pada sistem sebagaimana ditunjukkan pada Gambar 3.6.



Gambar 3. 6 Activity Diagram Login

2) *Activity Diagram Bank Data*

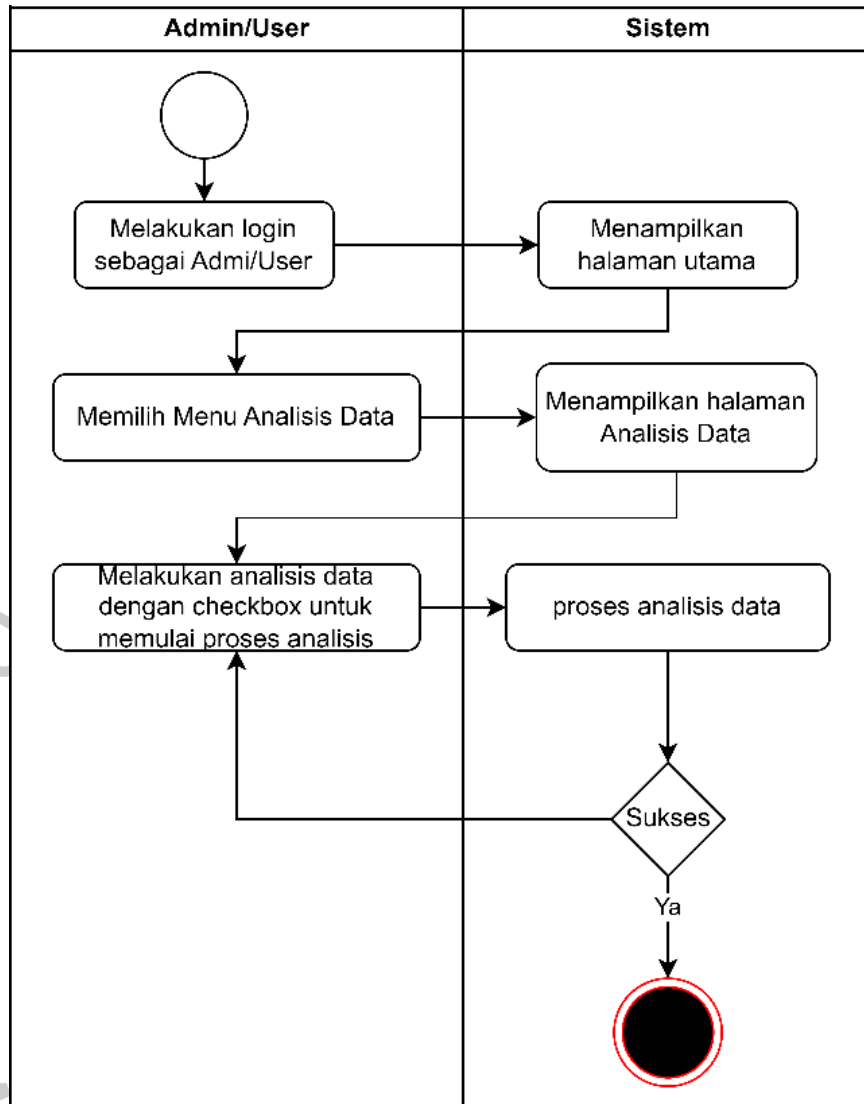
Aktivitas Diagram Bank Data adalah alur di mana admin dapat mengelola data kecelakaan kerja, seperti yang ditunjukkan pada Gambar 3.7.



Gambar 3. 7 Activity Diagram Kelola Data

3) *Activity Diagram Analisis Data*

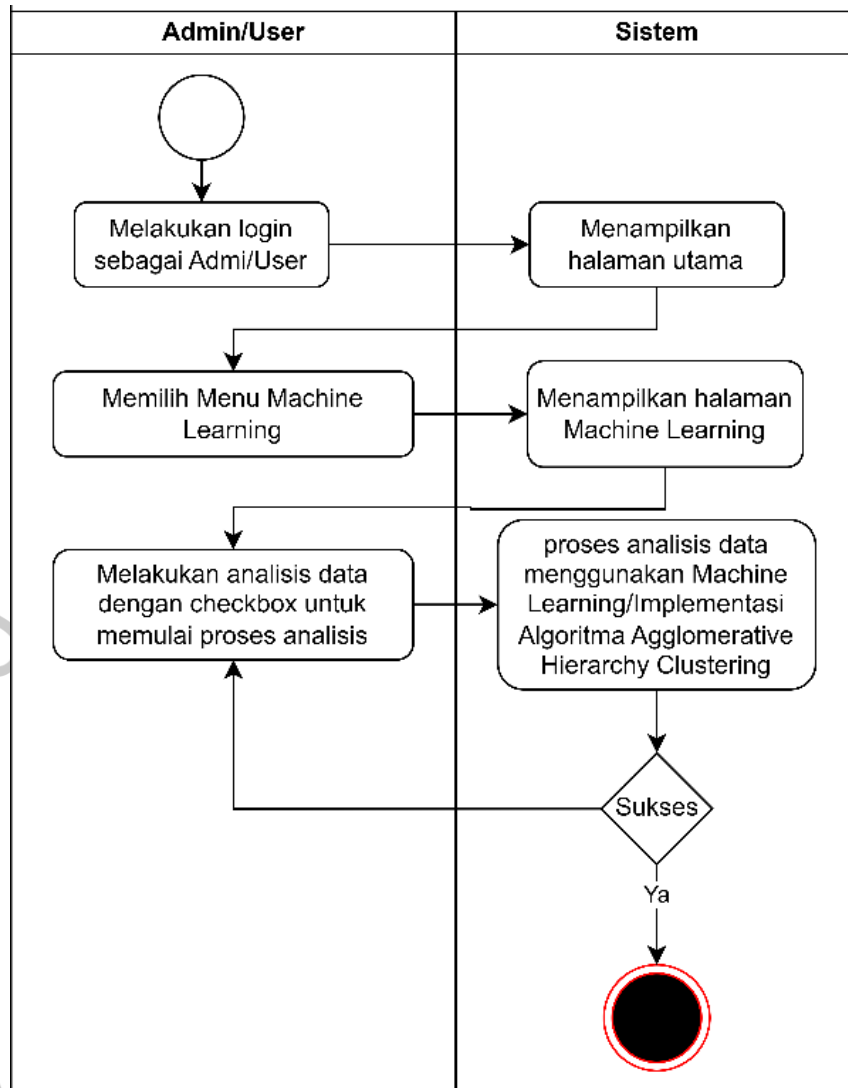
Aktivitas Diagram Analisis Data adalah alur di mana admin dapat melakukan proses analisis terhadap data yang ada, seperti yang ditunjukkan pada Gambar 3.8.



Gambar 3. 8 Activity Diagram Analisis Data

4) *Activity Diagram Machine Learning*

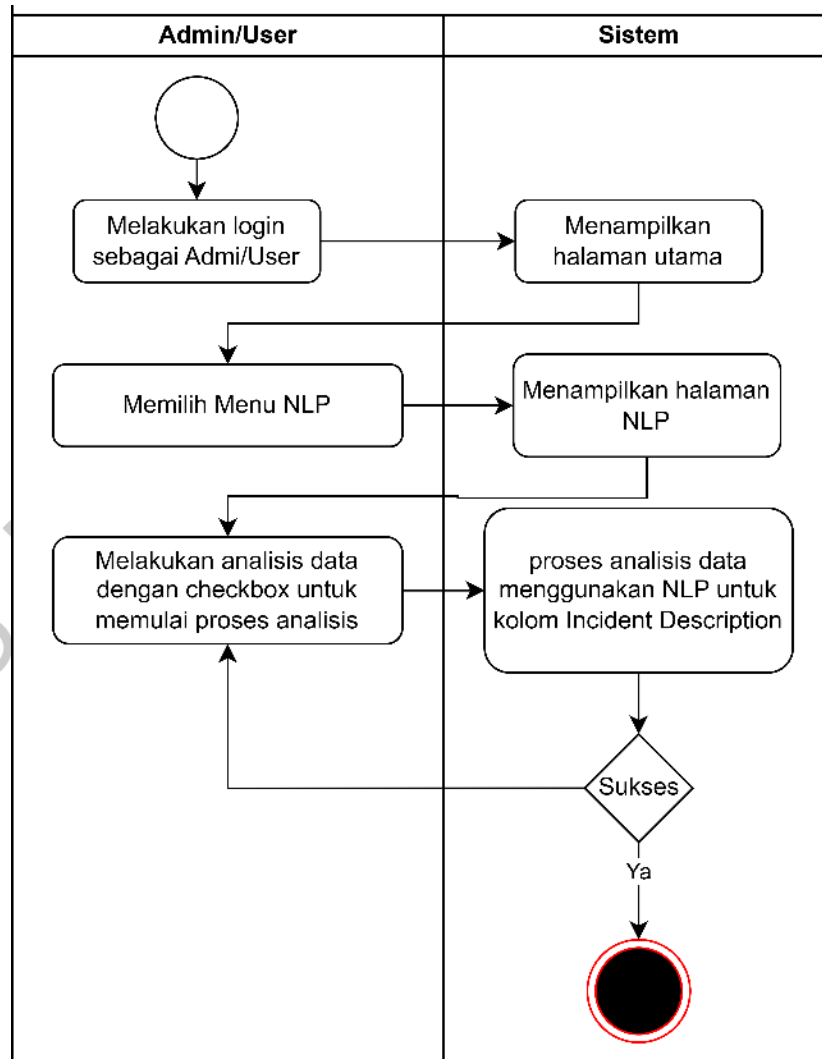
Aktivitas Diagram *Machine Learning* adalah alur di mana proses pembelajaran mesin dilakukan, mencakup tahap pemuatan data, pelatihan model, evaluasi, dan penggunaan model untuk prediksi, seperti yang ditunjukkan pada Gambar 3.9.



Gambar 3. 9 Activity Diagram Machine Learning

5) Activity Diagram NLP

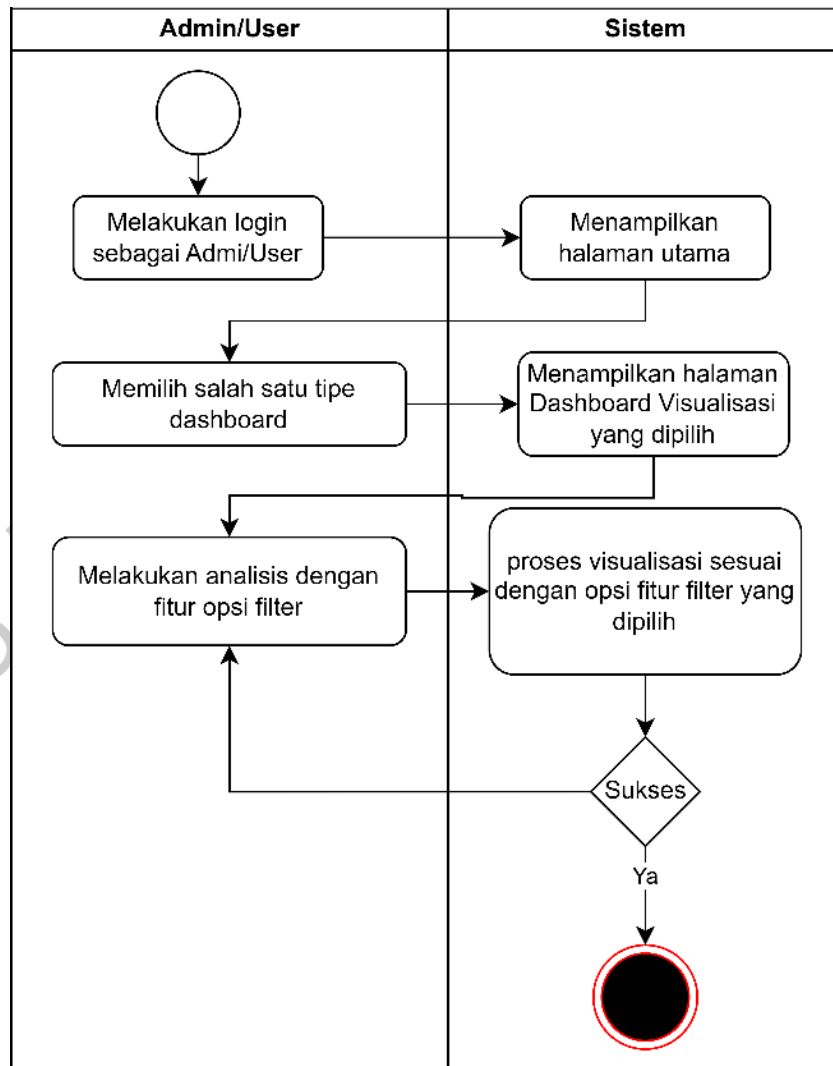
Aktivitas Diagram *Natural Language Processing (NLP)* adalah alur di mana sistem NLP mengelola dan memproses teks atau bahasa alami. Ini mencakup tahap pengolahan teks, pemahaman konteks, ekstraksi informasi, dan tugas-tugas NLP lainnya, seperti yang ditunjukkan pada Gambar 3.10.



Gambar 3. 10 Activity Diagram NLP

6) *Activity Diagram Home (Visualisasi Data)*

Aktivitas Diagram "Home (Visualisasi Data)" adalah alur di mana pengguna dapat memvisualisasikan data pada halaman utama (*Home*). Ini mencakup langkah-langkah seperti memilih data, memilih jenis visualisasi, dan menampilkan visualisasi data, seperti yang ditunjukkan pada Gambar 3.11.

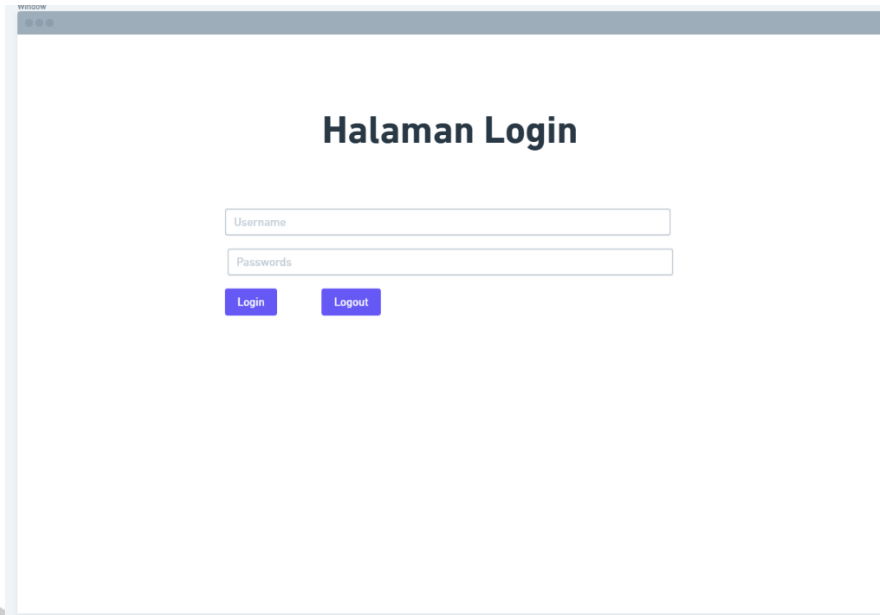


Gambar 3. 11 Activity Diagram Visualisasi Data

c. Rancangan Antar Muka

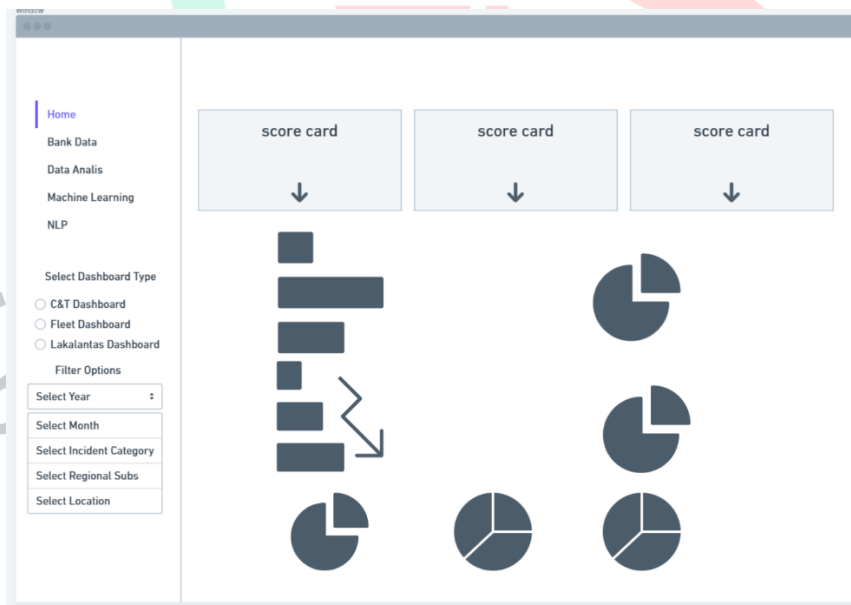
Rancangan antarmuka adalah komponen desain yang memvisualisasikan bagaimana pengguna akan berinteraksi pada suatu aplikasi atau sistem. Seperti rancangan antarmuka untuk *login* yang ditunjukkan pada Gambar 3.12, halaman utama (*Home*) pada Gambar 3.14, Rancangan Antarmuka Form LKP pada Gambar 3.15, Rancangan Antarmuka Data Insiden pada Gambar 3.16, Rancangan Antarmuka Unggah Data Insiden pada Gambar 3.17, Rancangan Antarmuka Analisis Data pada Gambar 3.18, Rancangan Antarmuka *Machine Learning* pada Gambar 3.19, dan Rancangan Antarmuka NLP pada Gambar 3.20.

1 *Login*



Gambar 3. 12 Rancangan Antarmuka Login

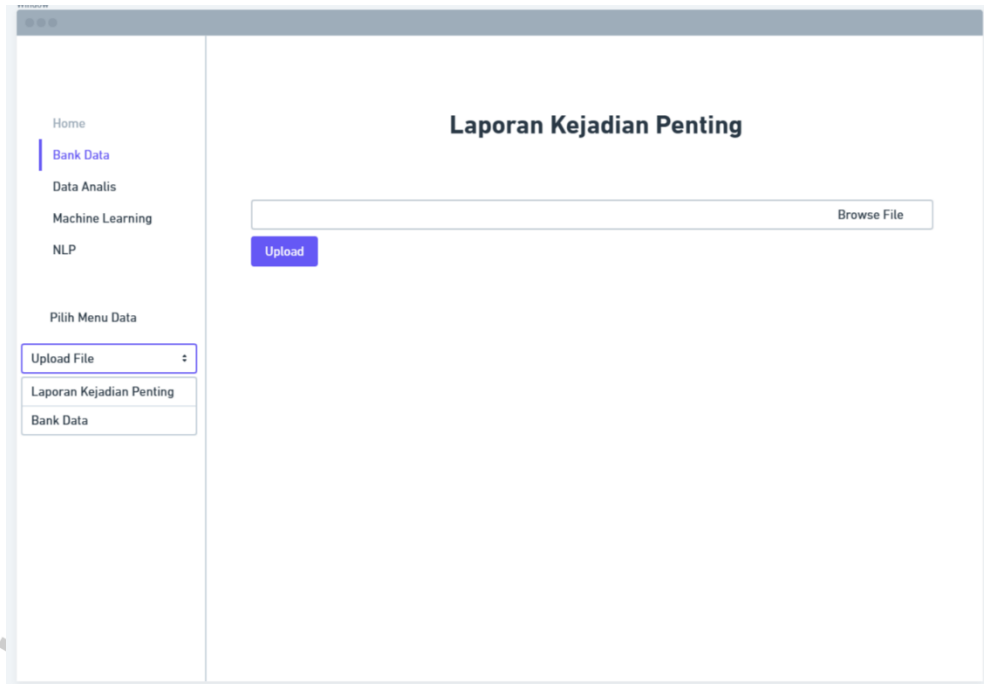
2. Home



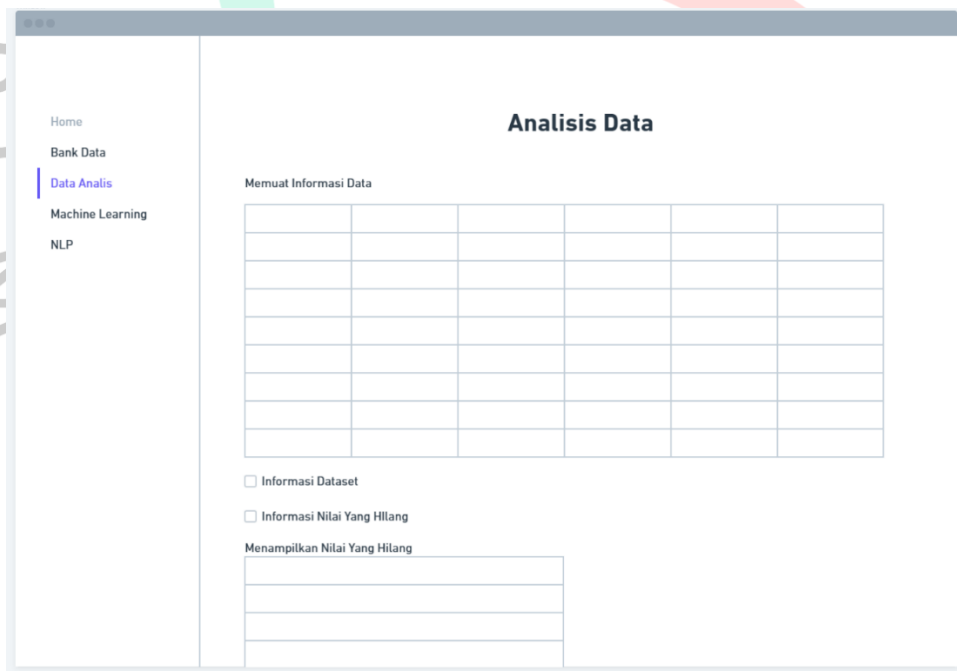
Gambar 3. 13 Rancangan Antarmuka Visualisasi Data

Gambar 3. 14 Rancangan Antarmuka Form LKP

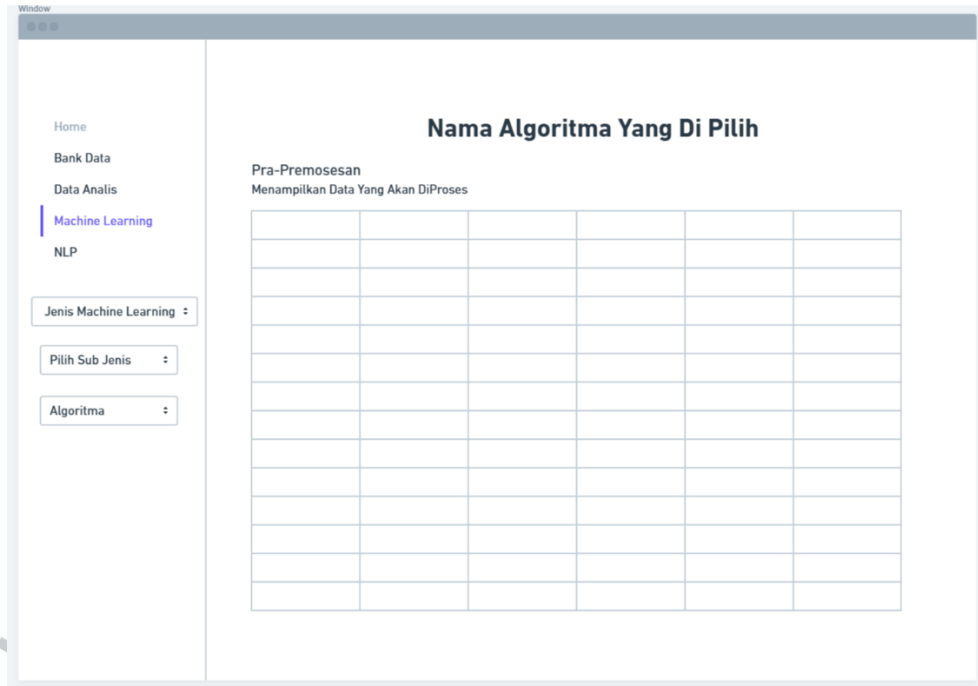
Gambar 3. 15 Rancangan Antarmuka Data Insiden



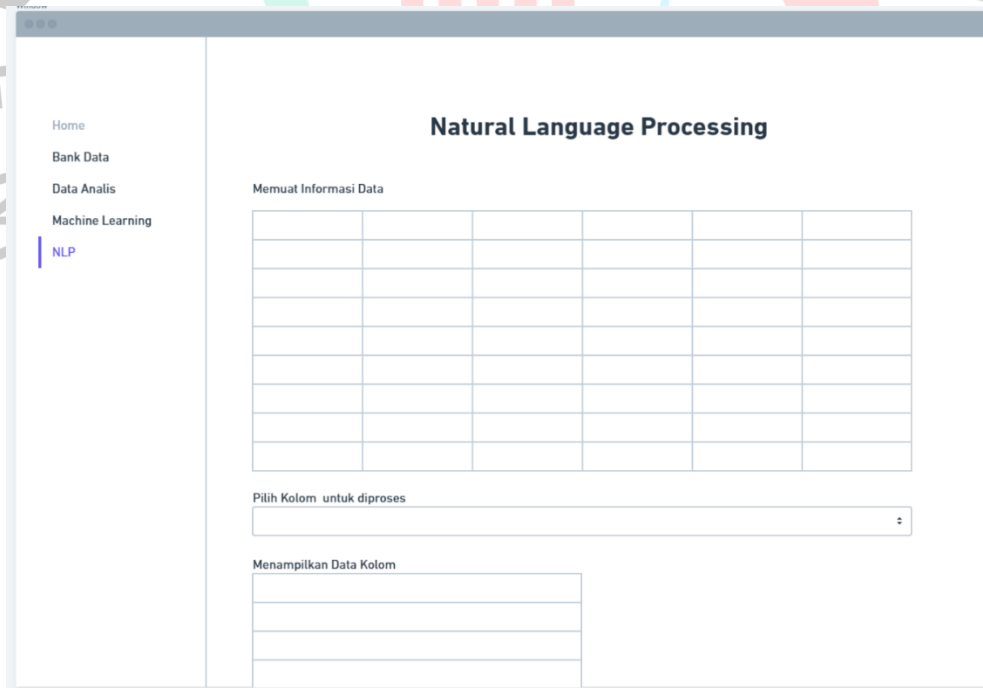
Gambar 3. 16 Rancangan Antarmuka Unggah Data Insiden



Gambar 3. 17 Rancangan Antarmuka Analisis Data



Gambar 3. 18 Rancangan Antarmuka Machine Learning



Gambar 3. 19 Rancangan Antarmuka NLP

3.6 Metode Pengujian

Dalam penelitian ini, dilakukan pemeriksaan data menggunakan dua pendekatan yang berbeda, yaitu pengujian kotak hitam dan pengujian kotak putih.

3.6.1 Metode Kotak Hitam

Dalam pengujian dan pengevaluasian sistem, metode pengujian perangkat lunak yang Berdasarkan evaluasi terhadap rincian teknis perangkat lunak tanpa referensi pada cara kerja internalnya(Khan & Khan, 2012). Maksud dari hal itu ialah untuk melaksanakan uji coba sejauh mana elemen sesuai dengan ketentuan yang dipublikasikan untuk elemen tersebut.

3.6.2 Metode Kotak Putih

Menurut Khan, Mohd.E. (2012), pengujian kotak putih berdasarkan analisis internal dan struktur dari sebuah perangkat lunak. Pengujian kotak putih adalah proses memberikan input ke sistem dan memeriksa bagaimana sistem memproses *input* tersebut untuk menghasilkan keluaran yang dibutuhkan. Hal ini diperlukan bagi seorang penguji untuk memiliki pengetahuan penuh tentang kode sumber.

Pengujian kotak putih dapat diterapkan pada tingkat integrasi, unit, dan sistem dalam proses pengujian perangkat lunak. Dalam kedua metode pengujian ini, terdapat beberapa cara yang dilakukan, seperti menginputkan berbagai pertanyaan dengan topik yang berbeda namun tetap fokus pada mendapatkan jawaban yang sesuai dengan pola-pola pembentukan yang telah ditentukan.