IMPLEMENTASI *TEXT MINING* DENGAN TF-IDF UNTUK KLASIFIKASI JENIS PERBAIKAN KENDARAAN DAN EVALUASI AKURASI ALGORITMA *MACHINE LEARNING*

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata-1
Program Studi Teknik Industri - Fakultas Teknologi Industri
Universitas Islam Indonesia

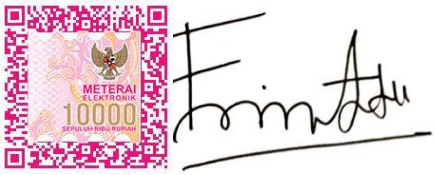
Nama : Firza Faturrahman
No. Mahasiswa : 21522338

PROGRAM STUDI TEKNIK INDUSTRI PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2025

PERNYATAAN KEASLIAN

Saya mengakui bahwa tugas akhir ini adalah hasil karya saya sendiri kecuali kutipan dan ringkasan yang seluruhnya sudah saya jelaskan sumbernya. Jika dikemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah maka saya bersedia ijazah yang telah saya terima ditarik kembali oleh Universitas Islam Indonesia.

Yogyakarta, 22 September 2025



(Firza Faturrahman)
21522338

SURAT BUKTI PENELITIAN

Reference Letter
Surat Keterangan Kerja
No: 012/SKet/ASSA-HC/IV/2025

Here by certifying that,
Dengan ini menerangkan

Name Nama	: Firza Faturrahman
Classification Jabatan	: Business Process Analyst Intern
Department Departemen	: Fleet Management
Branch Cabang	: Head Office
Period Of Service Masa Kerja	: September 6th, 2024 – December 31st, 2024

Has served a period of work as an intern at PT. Adi Sarana Armada Tbk. May success be with you in the future

Adalah benar telah menjalani masa kerja sebagai karyawan magang di PT. Adi Sarana Armada Tbk.
Semoga kesuksesan selalu menyertai anda dimasa mendatang

Jakarta, April 21st, 2025

PT Adi Sarana Armada Tbk

Adam Prasetya Adi
Human Capital Department

LEMBAR PENGESAHAN PEMBIMBING

IMPLEMENTASI *TEXT MINING* DENGAN TF-IDF UNTUK KLASIFIKASI JENIS PERBAIKAN KENDARAAN DAN EVALUASI AKURASI ALGORITMA *MACHINE LEARNING*

TUGAS AKHIR

Disusun Oleh :

Nama : Firza Faturrahman
No. Mahasiswa : 21522338

Yogyakarta, 22 September 2025
Dosen Pembimbing



(Dr. Drs. Imam Djati Widodo, M.Eng.Sc.)

LEMBAR PENGESAHAN DOSEN PENGUJI

IMPLEMENTASI *TEXT MINING* DENGAN TF-IDF UNTUK KLASIFIKASI JENIS PERBAIKAN KENDARAAN DAN EVALUASI AKURASI ALGORITMA *MACHINE LEARNING*

TUGAS AKHIR

Disusun Oleh :
Nama : Firza Faturrahman
No. Mahasiswa : 21522338

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata-1 Teknik Industri Fakultas Tekonologi Industri Universitas Islam Indonesia

Yogyakarta, 22 September 2025

Tim Penguji

Dr. Drs. Imam Djati Widodo, M.Eng.Sc.	
Ketua	
Nashrullah Setiawan, S.T., M.Sc., Ph.D.	
Anggota I	
Dr. Harwati, S.T., M.T.	
Anggota II	

Mengetahui,
Ketua Program Studi Teknik Industri Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

Ir. Muhammad Rizwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM

NIK: 015220101

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur ke hadirat Allah SWT atas rahmat dan karunia-Nya, Tugas Akhir ini saya persembahkan kepada kedua orang tua yang dengan doa, cinta, dan pengorbanannya menjadi sumber kekuatan terbesar dalam hidup penulis. Serta kepada Bapak Dr. Drs. Imam Djati Widodo, M.Eng.Sc. selaku dosen pembimbing yang telah membimbing dan memberikan arahan dengan sabar. Tak lupa penulis mengucapkan terima kasih kepada Fatiyya Ariza S.Tr.J.M.P, yang setia menemani dalam setiap langkah proses ini, serta kepada pihak yang tidak bisa saya sebutkan satu persatu. Terima kasih juga saya ucapkan untuk diri saya sendiri atas kerja keras dalam menghadapi segala tantangan. Semoga penelitian ini dapat menjadi manfaat bagi semua pihak yang membacanya.

MOTTO

"Tidaklah Allah membebani seseorang melainkan sesuai dengan kesanggupannya."
(Q.S. Al-Baqarah: 286)

"Sesungguhnya Allah tidak akan mengubah keadaan suatu kaum sampai mereka mengubah keadaan yang ada pada diri mereka sendiri."
(Q.S. Ar-Ra'd: 11)

.

KATA PENGANTAR

Assalamualaikum Warrahmatullahi Wabarakatuh

Alhamdulillahirobil'alamin, puji syukur kehadirat Allah SWT, yang telah melimpahkan rahmat, hidayah, serta karunia-Nya kepada kita semua. Shalawat serta salam senantiasa tercurah kepada junjungan kita Nabi Muhammad SAW, yang telah menjadi teladan terbaik bagi umat manusia, sehingga penulis dapat menyelesaikan penyusunan tugas akhir ini dengan lancar yang berjudul "Implementasi *Text Mining* Dengan TF-IDF Untuk Klasifikasi Jenis Perbaikan Kendaraan dan Evaluasi Akurasi Algoritma *Machine Learning".*

Penyusunan tugas akhir ini dilakukan dalam rangka memenuhi salah satu syarat kelulusan untuk mencapai gelar Sarjana Teknik Program Studi Teknik Industri Fakultas Teknologi Industri Universitas Islam Indonesia. Penulis menyadari bahwa dalam penulisan tugas akhir ini banyak mengalami kendala, namun berkat bantuan dan bimbingan berbagai pihak dan ridho dari Allah SWT sehingga kendala yang penulis hadapi dapat terselesaikan. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., IPU., ASEAN.Eng. selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Dr. Drs. Imam Djati Widodo, M.Eng.Sc. selaku Ketua Jurusan Teknik Industri Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Bapak Ir. Muhammad Ridwan Andi Purnomo, ST., M.Sc., Ph.D., IPM., selaku Ketua Program Studi Teknik Industri Program Sarjana, Fakultas Teknologi Industri, Universitas Islam Indonesia.
4. Bapak Dr. Drs. Imam Djati Widodo, M.Eng.Sc. selaku Dosen Pembimbing Tugas Akhir yang memberikan saran, masukan, dan bimbingannya dengan sabar dalam penyelesaian laporan ini.
5. Kedua orang tua yang selalu mendoakan, menguatkan, memberikan kasih sayang, mendukung secara materi serta meyakinkan agar penulis dapat menyelesaikan laporan ini dengan baik.
6. Putri Ika Wahyu Retno Juwita selaku mentor MSIB serta rekan – rekan karyawan PT Adi Sarana Armada Tbk yang telah membimbing dan memberikan kesempatan pada penulis untuk dapat belajar di ASSA
7. Kepada Fatiyya Ariza yang senantiasa memberikan dukungan, semangat, serta motivasi yang tiada henti selama proses penyusunan proposal ini. Kehadiran dan bantuannya menjadi dorongan berharga yang membuat penulis lebih kuat dalam menyelesaikan setiap tahapan dengan penuh keyakinan.
8. Serta semua pihak yang tidak dapat penulis sebutkan namanya satu persatu. Terimakasih dan semoga Allah SWT membalas kebaikan dengan sesuatu yang lebih.
9. *Last but not least, I wanna thank me. I wanna thank me for believing in me, i wanna thank me for doing all this hard work, i wanna thank me for having no days off. I wanna thank me for never quiting. I wanna thank me for always being a giver and tryna give more than i recive. I wanna thank me for tryna do more right than wrong. I wanna thank me for just being me at all times.*

Demikian penulisan laporan Tugas Akhir ini, penulis menyadari bahwa masih terdapat kekurangan dan ketidaksempurnaan dalam penyusunannya. Sehingga kritik serta saran dapat

diberikan sebagai perbaikan dalam penyusunan laporan pada penelitian berikutnya. Semoga laporan ini dapat memberikan manfaat kepada penulis maupun pembacanya.
Wassalamu'alaikum Warrahmatullahi Wabarakatuh.

Yogyakarta, 22 September 2025



Firza Faturrahman

ABSTRAK

Pemeliharaan kendaraan menjadi salah satu aspek penting dalame menjamin kelancaran operasional layanan penyewaan kendaraan. PT Adi Sarana Armada merupakan perusahaan penyedia layanan penyewaan kendaraan yang perlu memastikan proses identifikasi jenis perbaikan kendaraan berjalan cepat, akurat, dan konsisten. Permasalahan yang muncul adalah deskripsi kerusakan pada Surat Perintah Kerja (SPK) bersifat tidak terstruktur dan bervariasi sehingga menyulitkan pengelompokan jenis perbaikan kendaraan. Penelitian ini bertujuan untuk membangun model klasifikasi berbasis *text mining* menggunakan pendekatan *Term Frequency–Inverse Document Frequency* (TF-IDF) untuk mengelompokkan jenis perbaikan kendaraan ke dalam tiga kategori perbaikan yaitu Perbaikan Berkala (PB), Perbaikan Berkala Tidak Teratur (PBTT), dan Adhoc. Metode yang digunakan meliputi pengumpulan data historis SPK, tahap *preprocessing* seperti *cleaning data, case folding data*, *tokenize data*, *stopword data*, dan *stemming data.* Kemudian transformasi data teks menjadi numerik dengan TF-IDF. Data dibagi menjadi data latih dan uji dengan rasio 80:20, lalu dilatih menggunakan tiga algoritma *machine learning* yaitu *Naive Bayes, Logistic Regression,* dan *Random Forest.* Evaluasi dilakukan menggunakan *confusion matrix* untuk menilai akurasi klasifikasi. Hasil penelitian menunjukkan bahwa *Random Forest* memiliki akurasi tertinggi yaitu 0,7549, *Logistic Regression* sebesar 0,7441, dan *Naive Bayes* sebesar 0,7351. Sehingga, model *Random Forest* dapat dikatakan model yang cocok untuk klasifikasi jenis perbaikan kendaraan. Sistem klasifikasi ini diharapkan dapat mendukung proses identifikasi jenis perbaikan kendaraan secara lebih cepat, objektif, dan konsisten sehingga meningkatkan efisiensi layanan perbaikan kendaraan.

Kata Kunci: *Text Mining, Machine Learning,* TF-IDF, Klasifikasi

DAFTAR ISI

PERNYATAAN KEASLIAN ii
SURAT BUKTI PENELITIAN iii
LEMBAR PENGESAHAN PEMBIMBING iv
LEMBAR PENGESAHAN DOSEN PENGUJI v
HALAMAN PERSEMBAHAN vi
MOTTO vii
KATA PENGANTAR viii
ABSTRAK x
DAFTAR ISI xi
DAFTAR TABEL xiii
DAFTAR GAMBAR xiv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 7
1.3 Tujuan Penelitian 7
1.4 Manfaat Penelitian 8
1.5 Batasan Penelitian 8
BAB II TINJAUAN PUSTAKA 9
2.1 Kajian Literatur 9
2.1.1 *Text Mining* 9
2.1.2 Machine Learning 11
2.1.3 Posisi Penelitian 13
2.2 Landasan Teori 15
2.2.1 Surat Perintah Kerja/Laporan Teknisi 15
2.2.2 *Knowledge Discovery in Database* (KDD) 15
2.2.3 *Data Mining* 16
2.2.4 *Pre-processing Data* 17
2.2.5 *Text Mining* 18
2.2.6 TF-IDF 19
2.2.7 *Machine Learning* 21
2.2.8 *Naïve bayes* 22
2.2.9 *Logistic Regression* 22
2.2.10 *Random Forest* 22
2.2.11 *Confusion Matrix* 23
BAB III METODE PENELITIAN 24
3.1 Jenis Data 24
3.2 Pengumpulan Data 24
3.3 Pengolahan Data 24
3.4 Alur Penelitian 25
3.5 Kerangka Konseptual Penelitian 29
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA 33
4.1 Pengumpulan Data 33
4.2 Pengolahan Data 33

4.2.1 *Preprocessing Data* 34
4.2.2 Eksplorasi Data dan Visualisasi 38
4.2.3 Pembobotan dan Pemisahan Data 41
4.2.4 Penerapan Algoritma *Machine Learning* 42
4.3 Evaluasi Model Klasifikasi 45
4.3.1 Evaluasi Model *Naive Bayes* 45
4.3.2 Evaluasi Model *Logistic Regression* 46
4.3.3 Evaluasi Model *Random Forest* 47
BAB V PEMBAHASAN 49
5.1 Analisis Evaluasi Model 49
5.1.1 Akurasi Model 49
5.1.2 Interpretasi Confusion Matrix 52
5.2 Perbandingan Hasil Prediksi Model 54
5.3 Perbandingan Kinerja Antar Model 57
5.4 Implementasi Model Machine Learning 58
5.4.1 Pemrosesan Data Baru 58
5.4.2 Prediksi Model Yang Terpilih 59
5.4.3 Penyajian Hasil Prediksi 60
BAB VI PENUTUP 63
6.1 Kesimpulan 63
6.2 Saran 64
DAFTAR PUSTAKA 66
LAMPIRAN A-1

DAFTAR TABEL

Tabel 2. 1 Posisi Penelitian 13
Tabel 4. 1 Proses *Case Folding* 36
Tabel 4. 2 Proses *Tokenize Data* 36
Tabel 4. 3 Proses *Stopwrod Data* 37
Tabel 4. 4 Proses *Steeming Data* 37
Tabel 4. 5 *Code* Pembobotan dan Pemisahan Data 41
Tabel 4. 6 *Code Algoritma Machine Learning* 43
Tabel 5. 1 Hasil Prediksi Model *Naive Bayes* 54
Tabel 5. 2 Hasil Prediksi Model *Logistic Regression* 55
Tabel 5. 3 Hasil Prediksi Model *Random Forest* 56
Tabel 5. 4 Perbandingan Akurasi Model Klasifikasi 57
Tabel 5. 5 *Dataset* Baru 58
Tabel 5. 6 *Code* Pemrosesan Data Baru 59
Tabel 5. 7 *Code* Prediksi Model 60
Tabel 5. 8 *Code* Penyajian Hasil Prediksi 60
Tabel 5. 9 Data Hasil Prediksi 61

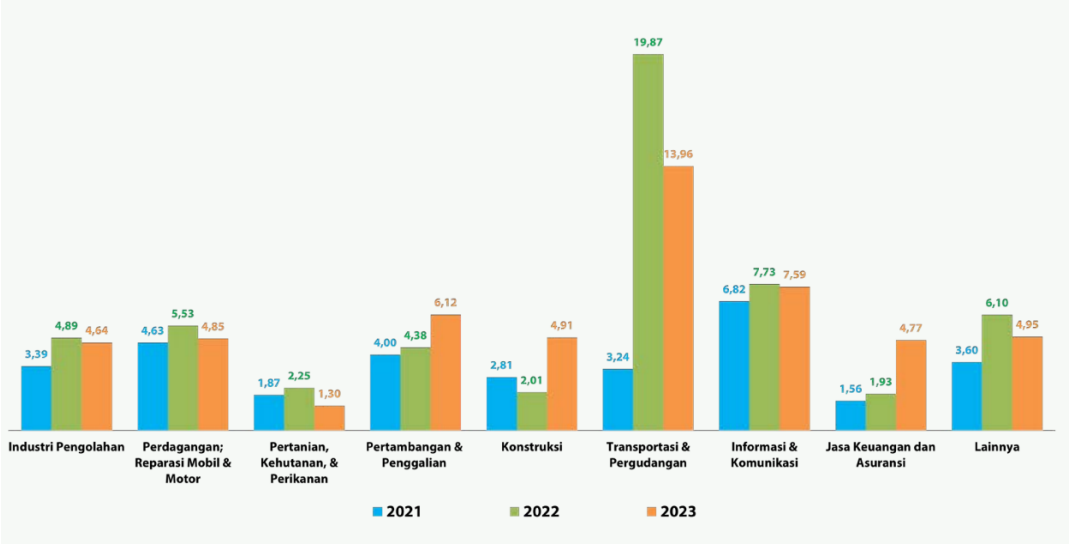
DAFTAR GAMBAR

Gambar 1. 1 Pertumbuhan PDB Beberapa Lapangan Usaha 1
Gambar 1. 2 Perkembangan Jumlah Mobil Penumpang / *Passenger Car* 3
Gambar 1. 3 *Global Car Rental Market,* 2020 – 2030 3
Gambar 3. 1 Alur Penelitian 26
Gambar 3. 2 Kerangka Konseptual Penelitian 30
Gambar 4. 1 Hasil Pengumpulan Data 33
Gambar 4. 2 Hasil *Cleaning Data* 34
Gambar 4. 3 Distribusi Jenis Perbaikan 35
Gambar 4. 4 Hasil Eksplorasi Informasi Umum *Dataset* 38
Gambar 4. 5 Hasil Eksplorasi *Data Null* 39
Gambar 4. 6 Hasil Eksplorasi *Unique Values* 39
Gambar 4. 7 Visualisasi *Word Cloud* 40
Gambar 4. 8 Hasil Evaluasi *Naïve Bayes* 46
Gambar 4. 9 Hasil Evaluasi Model *Logistic Regression* 47
Gambar 4. 10 Hasil Evaluasi Model *Random Forest* 48

BAB I
PENDAHULUAN

1.1 Latar Belakang

Transportasi merupakan salah satu elemen fundamental dalam kehidupan manusia yang mendukung mobilitas individu serta distribusi barang dan jasa. Sistem transportasi yang efisien berperan besar dalam meningkatkan produktivitas ekonomi, konektivitas sosial, dan pengembangan wilayah. Mengutip dari laporan World Economic Forum (2024), sektor transportasi berkontribusi lebih dari 6% terhadap PDB global, dengan tingkat pertumbuhan yang terus meningkat seiring dengan meningkatnya urbanisasi dan digitalisasi di berbagai sektor (Khodadadi et al., 2022). Selain itu, kemajuan teknologi dalam bidang transportasi juga memungkinkan pengembangan moda yang cepat, efisien, serta ramah lingkungan, termasuk transportasi berbasis listrik dan otonom (Arteaga et al., 2020). Berdasarkan gambar 1.1 menunjukkan PDB menurut lapangan pekerjaan, dimana ekonomi di Indonesia pada tahun 2023 tumbuh sebesar 5,05 persen. Pertumbuhan tercatat seluruh lapangan usaha, dengan sektor Transportasi dan Pergudangan mencatat kenaikan tertinggi yaitu sebesar 13,96 %. Hal ini menunjukkan bahwa pertumbuhan industri transportasi setiap tahunnya semakin meningkat dan memiliki banyak peminat.

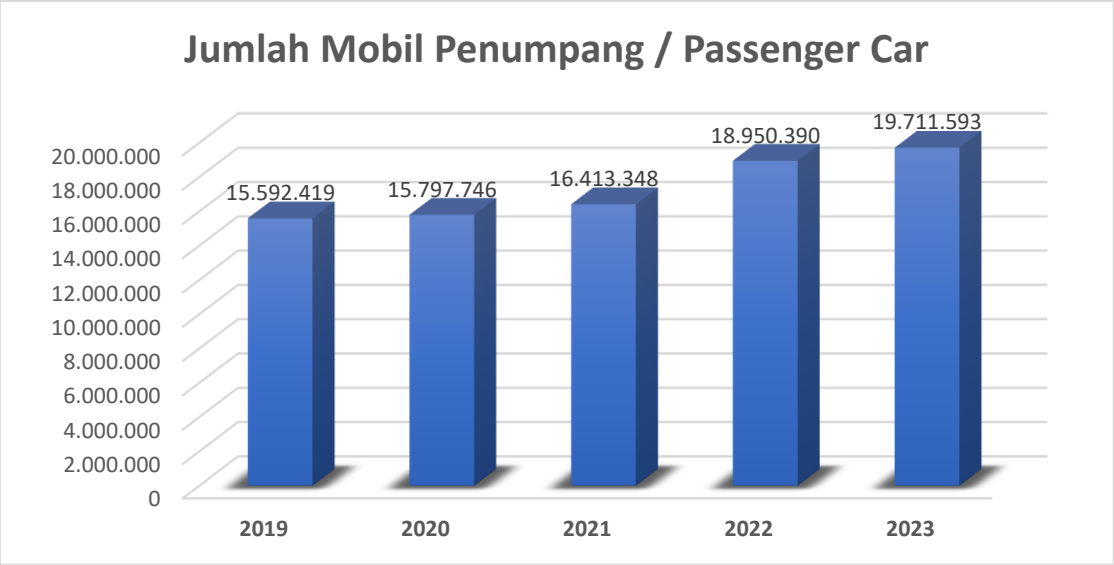


Gambar 1. 1 Pertumbuhan PDB Beberapa Lapangan Usaha

Sumber: (Badan Pusat Statistik, 2023)

Dalam perkembangan industri transportasi, berbagai moda telah dikembangkan untuk memenuhi kebutuhan mobilitas manusia, termasuk transportasi darat, laut, udara, dan rel. Dari berbagai moda tersebut, transportasi darat menjadi yang paling dominan karena fleksibilitasnya dalam mobilitas jarak pendek hingga menengah. Mobilitas berbasis kendaraan pribadi maupun umum menjadi pilihan utama bagi masyarakat di berbagai negara, terutama di kawasan perkotaan yang memiliki kepadatan tinggi. Dikutip dari laman ALG Global (2024) yang menyatakan transportasi kendaraan pribadi secara konsisten mendominasi pembagian moda di sebagian besar wilayah metropolitan di seluruh dunia, karena faktor – faktor seperti kenyamanan, kemandirian, fleksibilitas, kenyamanan, kecepatan, dan keandalan, yang dikombinasikan dengan investasi historis dalam infrastruktur jalan, telah berkontribusi pada peningkatan pangsa moda kendaraan pribadi dibandingkan dengan transportasi umum.

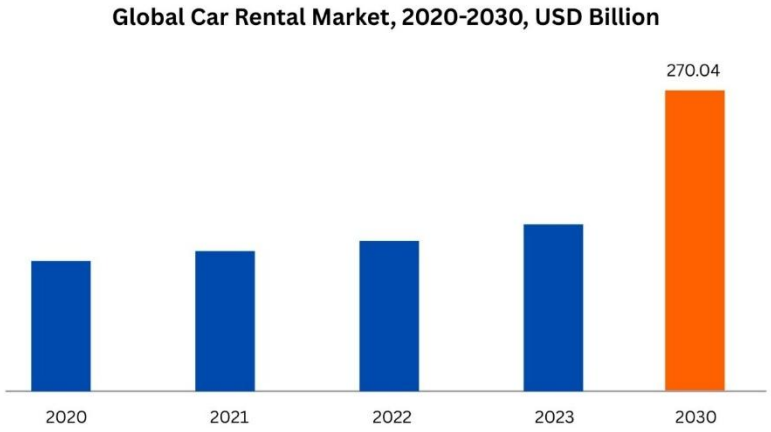
Transportasi darat memiliki kontribusi besar terhadap perkembangan perekonomian suatu negara. Kendaraan roda empat seperti mobil pribadi, taksi, dan kendaraan niaga ringan tidak hanya berfungsi sebagai alat mobilitas individu, tetapi juga mendukung berbagai sektor industri seperti logistik dan transportasi umum (Ren et al., 2022). Berdasarkan data Gaikindo (Gabungan Industri Kendaraan Bermotor Indonesia) di Indonesia, jumlah kendaraan roda empat yang beroperasi setiap hari mencapai lebih dari 40 juta unit. Berdasarkan Gambar 1.2 menunjukkan perkembangan jumlah mobil penumpang, dimana pada periode 2019 – 2023 terjadi peningkatan jumlah kendaraan khususnya mobil penumpang / *passenger car.* Kenaikan jumlah mobil penumpang terjadi sebesar 6,04 persen pertahun. Faktor pendorong pertumbuhan ini antara lain meningkatnya kebutuhan mobilitas, perkembangan sektor pariwisata, dan pertumbuhan ekonomi yang stabil (Chowdhury & Alzarrad, 2023).



Gambar 1. 2 Perkembangan Jumlah Mobil Penumpang / *Passenger Car*

Sumber: (Badan Pusat Statistik, 2023)

Dengan meningkatnya jumlah kendaraan maka industri penyewaan kendaraan telah menjadi komponen vital dalam sektor transportasi global, menawarkan solusi mobilitas yang fleksibel dan ekonomis bagi individu maupun perusahaan. Layanan ini memungkinkan akses kendaraan tanpa komitmen kepemilikan jangka panjang, sehingga menjadi alternatif menarik bagi banyak pihak (LPS Konstruksi, 2024).



Gambar 1. 3 *Global Car Rental Market,* 2020 – 2030

Sumber: (exactitudeconsultancy.com)

Di Indonesia, industri penyewaan kendaraan juga mengalami perkembangan pesat. Berdasarkan Gambar 1.3 Laporan dari Exactitude Consultancy (2023) menunjukkan bahwa pasar penyewaan kendaraan secara global akan mencapai 270,04 USD Billion pada tahun 2030.

Faktor pendorong pertumbuhan ini antara lain meningkatnya kebutuhan mobilitas, perkembangan sektor pariwisata, dan pertumbuhan ekonomi yang stabil. Selain itu, perubahan preferensi konsumen yang lebih memilih layanan fleksibel tanpa harus memiliki kendaraan sendiri turut berkontribusi pada peningkatan permintaan jasa sewa kendaraan.

DKI Jakarta merupakan kota besar dengan mobilitas yang tinggi. Hal ini yang dimanfaatkan oleh perusahaan rental mobil di DKI Jakarta untuk mendirikan rental mobil, salah satu perusahaan rental mobil tersebut adalah PT Adi Sarana Armada Tbk (ASSA). PT Adi Sarana Armada Tbk (ASSA) merupakan perusahaan terkemuka di sektor penyewaan kendaraan di Indonesia. Sejak tahun 2021, ASSA memiliki armada sebanyak 26.000 unit, dengan 22,63% merupakan kendaraan komersial dan 77,37% kendaraan penumpang. ASSA berfokus pada layanan sewa transportasi yang ditujukan untuk korporat atau ditujukan untuk perusahaan dengan jangka waktu yang panjang (Long Term). Dengan jumlah armada yang besar, perusahaan ini menghadapi tantangan dalam pengelolaan kendaraan, termasuk aspek perawatan dan perbaikan untuk memastikan operasional yang optimal.

Sebagai perusahaan yang bergerak di industri penyewaan kendaraan, ASSA tidak hanya berfokus pada penyediaan kendaraan tetapi juga harus memastikan bahwa setiap unit yang disewakan berada dalam kondisi prima. Oleh karena itu, layanan perbaikan dan perawatan kendaraan menjadi aspek krusial dalam operasional perusahaan. Untuk mendukung hal ini, ASSA memiliki layanan purna jual berupa bengkel perawatan dan perbaikan kendaraan yang memastikan armada selalu dalam kondisi layak jalan.

Namun, dengan skala operasi yang besar, sering kali terdapat aspek yang terlupakan. Salah satu aspek penting adalah kebutuhan akan perbaikan / perawatan kendaraan. Tanpa perawatan yang memadai, kendaraan dapat mengalami penurunan performa, meningkatkan risiko kecelakaan, serta berkontribusi terhadap pencemaran lingkungan. Sejalan dengan penelitian (Mena et al., 2024) yang menyatakan perawatan preventif kendaraan memiliki peran penting dalam menurunkan risiko kecelakaan hingga 20% serta mengurangi waktu henti operasional yang tidak perlu hingga 75%. Kerusakan pada kendaraan roda empat dapat dikategorikan ke dalam beberapa jenis utama, yaitu kerusakan pada mesin, sistem kelistrikan, sistem rem, transmisi, serta suspensi dan kemudi. Kerusakan mesin sering kali disebabkan oleh *overheat,* kebocoran oli, atau kerusakan pada komponen mekanis (Syifah, 2024a). Gangguan pada sistem kelistrikan mencakup masalah pada baterai, alternator, atau sistem pengapian. Masalah pada sistem rem bisa meliputi rem blong atau keausan kampas rem yang berlebihan,

sementara transmisi mengalami kendala pada perpindahan gigi atau sistem kopling (Dokter Mobil, 2025). Selain itu, kerusakan pada suspensi dan kemudi dapat mengganggu kenyamanan serta keselamatan berkendara, terutama jika terdapat masalah pada *shock absorber, tie rod,* atau *rack steer* (Zhao et al., 2024).

Untuk menangani berbagai jenis kerusakan tersebut, terdapat beberapa kategori perbaikan pada PT Adi Sarana Armada Tbk diantaranya, Perbaikan Berkala (PB) merupakan perbaikan yang dilakukan secara rutin berdasarkan jadwal yang telah ditentukan sesuai dengan standar pemeliharaan kendaraan, mencakup penggantian oli, penggantian kampas rem, atau penyetelan mesin (Syifah, 2024b). Perbaikan Berkala Tidak Teratur (PBTT) merupakan perbaikan yang seharusnya dilakukan secara berkala tetapi mengalami ketidakteraturan dalam pelaksanaanya, hal ini disebabkan oleh beberapa faktor meliputi jadwal pemakaian kendaraan yang padat, kendala operasional, atau keterlambatan dalam pemeliharaan. Dan Adhoc merupakan perbaikan atau perawatan yang dilakukan di luar jadwal pemeliharaan rutin dan berkala, Adhoc dilakukan sebagai respons terhadap kondisi tertentu, seperti kerusakan mendadak, kebutuhan operasional mendesak, atau perintah khusus yang tidak masuk dalam skema pemeliharaan reguler. Seiring dengan kemajuan teknologi otomotif, penggunaan sistem berbasis komputer untuk mendiagnosis kerusakan kendaraan semakin umum. Salah satu teknologi yang sering digunakan adalah *On-Board Diagnostics* (OBD), yang memungkinkan mekanik untuk mengidentifikasi masalah kendaraan lebih cepat dan akurat (Medashe et al., 2020).

Permasalahan yang dialami PT Adi Sarana Armada Tbk. Salah satunya ketidakakuratan evaluasi yang terjadi pada kantor pusat akibat ketidaksesuaian informasi dalam mengklasifikasikan jenis perbaikan kendaraan secara cepat, tepat dan konsisten akibat deskripsi kerusakan yang bersifat tidak terstruktur dan bervariasi. Ketidaksesuaian muncul karena terjadi kesalahan diagnosis jenis perbaikan dengan kegiatan aktual yang dilakukan. Hal ini juga disebabkan oleh sistem pencatatan yang hanya memungkinkan tercantumnya satu kategori jenis perbaikan untuk satu laporan SPK. Hal ini menyebabkan proses pengambilan keputusan strategis menjadi terhambat, bahkan berpotensi menimbulkan kesalahan dalam menentukan kebijakan maupun alokasi sumber daya perusahaan. Dampak yang lebih jauh dapat terlihat pada menurunnya efisiensi operasional, meningkatnya risiko kesalahan administratif, serta menurunnya kepercayaan stakeholder terhadap keandalan sistem informasi perusahaan. Hal ini juga dapat menimbulkan beberapa tantangan, seperti evaluasi kurang akurat, waktu yang

dibutuhkan untuk identifikasi masalah yang relatif lama, kemungkinan kesalahan manusia dalam menentukan jenis perbaikan, serta variasi bahasa yang digunakan dalam laporan teknisi yang dapat menyebabkan ketidaksesuaian dalam diagnosis (Torres et al., 2021). Keterbatasan ini menunjukkan perlunya pengembangan sistem klasifikasi yang mampu mengenali dan mengelompokkan berbagai jenis perbaikan secara lebih akurat dan lengkap berdasarkan deskripsi kerusakan yang diberikan.

Metode *Text Mining* dapat menjadi solusi dalam mengidentifikasi proses klasifikasi perbaikan kendaraan. *Text mining* merupakan teknik untuk melakukan pengklasifikasian data melalui identifikasi pola - pola dalam kumpulan teks berukuran besar (Pratama & Lestika Atmi, 2020). Dengan pendekatan ini, sistem dapat mengklasifikasikan jenis perbaikan secara otomatis berdasarkan pola dalam teks yang dimasukkan oleh teknisi atau pelanggan. Salah satu metode yang cukup umum diterapkan dalam *Text Mining* yaitu *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF berfungsi menilai tingkat kepentingan sebuah kata di dalam kumpulan dokumen serta telah terbukti efektif pada berbagai aplikasi klasifikasi teks, seperti *sentiment analysis*, *spam detection*, serta pemrosesan bahasa alami lainnya. TF-IDF telah terbukti mampu meningkatkan efektifitas dan akurasi dalam mendiagnosis permasalahan kendaraan berdasarkan teks deskriptif yang diberikan oleh pengguna dalam, termasuk untuk analisis jenis perawatan dan perbaikan kendaraan (Xiao et al., 2024). TF-IDF bekerja dengan perbandingan tiap kalimat topik hasil *text mining,* lalu menentukan bobot dengan menggunakan algoritma TF-IDF (Sari et al., 2021).

Dalam konteks klasifikasi jenis perbaikan kendaraan, TF-IDF berguna untuk mengubah laporan teknisi ke dalam bentuk numerik, sehingga proses pengelompokkan jenis perbaikan menjadi lebih akurat. Beberapa algoritma *machine learning* telah banyak dimanfaatkan dalam klasifikasi teks berbasis TF-IDF antara lain *Naive Bayes* (NB), *Logistic Regression* (LR), dan *Random Forest* (RF). Masing-masing algoritma memiliki keunggulan tersendiri, seperti efektivitas dalam mengidentifikasi pola pada teks, kemampuan menangani data dengan jumlah fitur yang besar, efisiensi dalam mengolah teks dengan jumlah kata kunci yang tinggi.

Namun, penelitian yang membandingkan efektivitas berbagai algoritma dalam klasifikasi perbaikan kendaraan masih terbatas. Selain itu, masih terdapat inkonsistensi dari penelitian terdahulu yang menjadi celah peneliti. Beberapa kesenjangan yang ingin diatasi dalam penelitian ini meliputi kurangnya studi yang menguji efektivitas TF-IDF dalam mengklasifikasikan laporan teknisi kendaraan dan minimnya perbandingan akurasi algoritma

machine learning dalam klasifikasi perbaikan kendaraan. Penelitian yang dilakukan oleh Hidayah dan Dodiman (2024) penerapan dari hasil berbagai algoritma *machine learning* pada data monitoring, dan menunjukkan hasil akurasi yang cukup tinggi pada model klasifikasi tersebut, sehingga model dengan algoritma *machine learning* tersebut dapat diimplementasikan. Oleh karena itu, penelitian sebelumnya telah mengintegrasikan berbagai algoritma *machine learning* pada data monitoring. Melalui penelitian ini diharapkan model *machine learning* dapat diterapkan pada data SPK agar dapat meningkatkan keakuratan evaluasi kinerja perusahaan.

Dengan demikian, penelitian ini perlu dilakukan untuk menghadirkan solusi yang mampu meningkatkan akurasi informasi, memperbaiki mekanisme evaluasi perusahaan, dan mendukung proses pengambilan keputusan yang lebih tepat, cepat, dan berbasis data yang valid. Objek dari penelitian ini adalah PT Adi Sarana Armada sebagai kebaruan penelitian. Data dalam penelitian ini bersumber dari *database* PT Adi Sarana Armada Tbk, khususnya pada data Surat Perintah Kerja (SPK). Secara teoritis, penelitian akan memperkaya literatur mengenai penerapan *Text Mining* dalam klasifikasi dokumen teknis. Secara praktis, hasil penelitian digunakan oleh bengkel dan perusahaan otomotif untuk mengoptimalkan layanan perbaikan kendaraan secara lebih efisien dan akurat. Sehingga penelitian ini diberi judul "Implementasi *Text Mining* Dengan TF-IDF Untuk Klasifikasi Jenis Perbaikan Kendaraan dan Evaluasi Akurasi Algoritma *Machine Learning*"

1.2 Rumusan Masalah

Dari pokok permasalahan yang telah dijelaskan dalam latar belakang, dapat diberikan rumusan masalah dapat disusun sebagai berikut:

1. Apa perbedaan karakteristik *rule* yang dihasilkan oleh algoritma *machine learning* dalam menginterpretasikan data teks?
2. Bagaimana ketidaksesuaian informasi antara data lapangan dengan laporan yang diterima mempengaruhi akurasi evaluasi kinerja?
3. Bagaimana efektivitas TF-IDF dan kinerja model *machine learning* dalam klasifikasi jenis perbaikan kendaraan?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang terbentuk maka tujuan yang ingin dicapai dalam penelitian ini disusun sebagai berikut:

1. Menganalisis perbedaan karakteristik *rule* yang dihasilkan oleh algoritma *machine learning* dalam mengklasifikasikan jenis perbaikan kendaraan.
2. Menganalisis pengaruh ketidaksesuaian informasi antara data lapangan dengan laporan yang diterima akurasi evaluasi kinerja.
3. Menganalisis efektivitas metode TF-IDF dalam merepresentasikan data teks dan mengevaluasi kinerja model *machine learning* dalam klasifikasi jenis perbaikan kendaraan untuk menentukan model dengan akurasi terbaik.

1.4 Manfaat Penelitian

Adapun kontribusi penelitian ini ditujukan kepada perusahaan maupun bidang akademis:

1. Bagi Perusahaan

   Penelitian ini diharapkan dapat menjadi acuan dalam mengklasifikasikan jenis perbaikan kendaraan berbasis teks, sehingga setiap kerusakan dapat teridentifikasi lebih akurat dan cepat. Dengan demikian, proses perbaikan dapat berjalan lebih efisien serta meningkatkan efektivitas manajemen armada kendaraan. Selain itu, algoritma klasifikasi yang digunakan dapat dijadikan alat bantu dalam menentukan jenis perbaikan kendaraan di bengkel, sehingga memudahkan identifikasi berdasarkan kerusakan kendaraan. Hal ini diharapkan dapat mengurangi tingkat ketidaksesuaian informasi pada Surat Perintah Kerja (SPK).
2. Bagi Akademis

   Penelitian ini diharapkan mampu memperkaya wawasan di bidang data sekaligus memberikan gambaran nyata mengenai penerapan ilmu Teknik Industri, khususnya pada konsentrasi studi riset operasional.

1.5 Batasan Penelitian

Adapun batasan yang diterapkan oleh peneliti pada penelitian yang dilakukan.

1. Data Surat Perintah Kerja (SPK) yang digunakan antara lain *Order Desc, Item Text*, dan Jenis Perbaikan.
2. *Tools* yang digunakan untuk menganalisis data adalah *Microsoft Excel* dan *Google Colaboratory* dengan menggunakan metode *text mining* pendekatan TF-IDF dan algoritma *machine learning.*
3. Data Surat Perintah Kerja yang digunakan yaitu dalam jangka waktu 1 Januari 2024 hingga 31 Maret 2024.

BAB II
TINJAUAN PUSTAKA

2.1 Kajian Literatur

2.1.1 *Text Mining*

Dari beberapa penelitian sebelumnya yang memiliki keterkaitan dengan metode *text mining*. Pada penelitian yang dilakukan oleh Sari et al., (2021) mengenai penerapan *text mining* pada aplikasi repository STMIK Budi Darma. Dalam penjelasan yang telah disampaikan dalam laporan ini, dapat disimpulkan bahwa algoritma text mining dengan metode Term Frequency Inverse Document Frequency (TF-IDF) efektif dalam melakukan klasifikasi terhadap dokumen skripsi. Penerapan cosine similarity menjadi langkah penting untuk menilai tingkat kesamaan antar abstrak. Semakin banyak kata kunci yang terdapat pada sebuah abstrak, maka hasil pengelompokan dokumen akan semakin optimal.

Penelitian yang dilakukan oleh Chamira (2022) mengenai implementasi *text mining* untuk monitoring diskusi online. Hasil penelitian tersebut, dapat disimpulkan bahwa metode *Term Frekuensi - Inverse Document Frekuensi* (TF-IDF), kesamaan *cosinus*, dan *Maximal Marginal Relevance* (MMR) berhasil menerapkan dalam proses menyederhanakan dokumen untuk keperluan pemantauan diskusi berani. Penerapan nilai MMR terbukti efektif dalam mengidentifikasi dan menyaring komentar yang relevan, sehingga dapat dipertahankan dalam forum diskusi online serta memberikan manfaat bagi seluruh pengguna. Selain itu, hasil pengujian dengan metode *black box* dan *User Acceptance Test* (UAT) menunjukkan bahwa sistem yang dikembangkan telah memenuhi kriteria kelayakan dan siap digunakan sebagai alat bantu dalam mengawasi penyiaran diskusi secara berani.

Penelitian yang dilakukan oleh Arsad et al., (2024) mengenai penerapan *text mining* untuk menentukan kesamaan dokumen skripsi. Penerapan algoritma *text mining* dan metode *cosine similarity* efektif dalam mendeteksi tingkat kesamaan antar dokumen skripsi. Proses *preprocessing* seperti *tokenisasi, stopword removal*, dan *stemming* menjadi tahapan penting sebelum dilakukan perhitungan kesamaan dokumen. Semakin tinggi nilai *cosine similarity* yang diperoleh antara dua dokumen, maka semakin besar tingkat kemiripan konten di antara keduanya. Hal ini menunjukkan bahwa pendekatan ini dapat digunakan sebagai alat bantu

dalam proses deteksi plagiarisme dan evaluasi dokumen akademik secara lebih efisien dan objektif.

Penelitian yang dilakukan oleh Farhan et al., (2023) mengenai *text mining* untuk klasifikasi UU ITE. Penelitian ini memanfaatkan proses *preprocessing data*, pembobotan kata, dan pelatihan model untuk mengelompokkan 245 kronologi pelanggaran ke dalam kategori pasal 27 dan 28. Sistem dikembangkan menggunakan *framework flask* sebagai antarmuka berbasis *web*. Dengan akurasi sebesar 80%, hasil penelitian menunjukkan bahwa pendekatan ini cukup efektif dalam mengidentifikasi potensi pelanggaran hukum dari teks yang tersedia. Temuan ini menegaskan bahwa teknologi klasifikasi teks dapat dijadikan sebagai sarana edukatif dalam meningkatkan kewaspadaan pengguna internet terhadap regulasi digital yang berlaku.

Penelitian yang dilakukan Timmisetty et al., (2024) mengenai *text mining* untuk klasifikasi tingkat keparahan kendaraan**.** Penelitian ini menggunakan *machine learning* untuk mengklasifikasikan tingkat keparahan masalah mobil, dengan tujuan meningkatkan kepuasan pelanggan dan efisiensi biaya. Deskripsi masalah pelanggan diubah menjadi format terstruktur menggunakan *text mining*. Model pohon keputusan (*decision* tree) dipilih karena lebih akurat dalam klasifikasi. Teknisi akan mengklasifikasikan masalah sebagai parah atau tidak, dan memberikan saran sesuai tingkat keparahan, meskipun model ini tidak dapat menjelaskan masalah spesifik mobil.

Penelitian oleh Jue Li dan Chang Wu (2023) berfokus pada klasifikasi informasi kecelakaan konstruksi. Penelitian ini menerapkan teknik *deep learning* dan *text mining* untuk mengklasifikasikan dan mengekstrak informasi penting dari narasi kecelakaan konstruksi. Menggunakan model *Convolutional Neural Network* (CNN) untuk klasifikasi teks, penelitian ini menunjukkan akurasi 84% dalam memproses laporan kecelakaan. Selain itu, algoritma *Term Frequency-Inverse Document Frequency* (TF-IDF) digunakan untuk mengekstrak informasi kunci dan menghasilkan visualisasi *word cloud*. Temuan ini memberikan wawasan berharga dalam manajemen keselamatan proyek konstruksi dan dapat diterapkan untuk mencegah kecelakaan di lapangan.

2.1.2 Machine Learning

Penelitian yang dilakukan oleh Nurul Hidayah dan Dodiman (2024) mengenai pengklasifikasian saran monitoring dan evaluasi mahasiswa. Pada penelitian ini menerapkan algoritma *Multinomial Naïve Bayes,* metode bobot *Term Frequency-Inverse Document Frequency* (TF-IDF), dan evaluasi melalui *Confusion Matrix* terbukti efektif dalam proses klasifikasi saran mahasiswa. Tahapan *preprocessing* seperti pembersihan teks, normalisasi, dan ekstraksi fitur menjadi langkah penting sebelum dilakukan proses klasifikasi. Hasil akurasi yang mencapai 85% menunjukkan bahwa sistem ini dapat digunakan sebagai alat bantu dalam menganalisis sentimen atau masukan mahasiswa secara terstruktur. Dengan demikian, pendekatan ini berpotensi untuk diterapkan dalam mendukung peningkatan kualitas proses pembelajaran dan evaluasi kinerja dosen di lingkungan pendidikan tinggi.

Penelitian yang dilakukan Khodadadi et al., (2022). mengenai diagnostik kendaraan otomatis. Penelitian ini menerapkan pembelajaran mesin untuk meningkatkan diagnostik kendaraan dengan menggunakan NLP untuk mengekstrak informasi dari laporan kegagalan. Algoritma pembelajaran mendalam memvalidasi klaim, dan algoritma klasifikasi mengarahkan permintaan layanan ke departemen yang sesuai. Model BiLSTM-CNN meningkatkan akurasi validasi permintaan layanan 18% lebih tinggi dibandingkan teknisi, dengan peningkatan akurasi (25%), sensitivitas (39%), spesifisitas (11%), dan presisi (11%) pada model GTB, serta ROC-AUC 0,82.

Penelitian yang dilakukan Fang Li (2020) berfokus mengevaluasi algoritma klasifikasi. Penelitian ini mengembangkan model klasifikasi teks ucapan dalam kendaraan untuk meningkatkan pemahaman bahasa alami. Penelitian ini menerapkan berbagai algoritma pembelajaran mesin seperti *Naive Bayes*, KNN, dan pohon keputusan (*decision tree*) untuk mengekstrak serta mengklasifikasikan fitur dari ucapan pengguna. Hasil penelitian menunjukkan bahwa model klasifikasi berbasis SVM memberikan kinerja terbaik dalam hal akurasi dan kecepatan, sedangkan pendekatan berbasis aturan juga efektif dalam memahami makna perintah pengguna melalui kombinasi klasifikasi dan ekstraksi informasi.

Penelitian yang dilakukan Deming Li (2024) berfokus mengevaluasi model *machine learning***.** Hasil penelitian menunjukkan bahwa model klasifikasi berbasis algoritma pembelajaran mesin (*Machine Learning*) efektif untuk evaluasi pengajaran desain interaktif di perguruan tinggi. Dalam percobaan ini, algoritma NB, KNN, LR, dan SVM digunakan sebagai

meta-pembelajaran, dengan peringkat klasifikasi SVM > LR > KNN > NB. SVM mencapai akurasi klasifikasi 92,1%, menunjukkan model ini sangat cocok untuk konstruksi sistem evaluasi pengajaran interaktif di pendidikan emosional.

Penelitian yang dilakukan Sathishkumar & Ibrahim (2025) berfokus mengevaluasi algoritma *machine learning* pada operasi mesin. Penelitian ini menggunakan algoritma regresi pembelajaran mesin (*Machine Learning*) untuk memperkirakan kinerja dan emisi pada mesin hidrogen diesel dengan pengapian kompresi muatan homogen (HDHCCI). Data mencakup tiga parameter masukan dan lima parameter keluaran. Sebanyak 26 algoritma pembelajaran mesin diuji, dengan Matern 5/2 GPR, *Wide Neural Network,* dan *Fine Tree* menunjukkan hasil terbaik dalam memprediksi BTE dan emisi lainnya, dengan nilai R2 mendekati 1. Namun, untuk estimasi emisi HC, hanya GPR yang menunjukkan kinerja baik.

Penelitian yang dilakukan Gupta et al., (2022) berfokus membandingkan kinerja algortima *machine learning* berdasarkan akurasi. Penelitian ini membandingkan kinerja berbagai algoritma pembelajaran mesin menggunakan alat R dan model perbandingan yang berbeda. Setelah perbandingan dilakukan, algoritma terbaik diterapkan untuk membuat prediksi. Penelitian ini mengevaluasi algoritma seperti *Decision Tree* (CART), *Support Vector Machine* (SVM), *K-Nearest Neighbors* (KNN), *Random Forest* (RF), dan *Naïve Bayes* (NB) menggunakan validasi silang berulang dengan 10 lipatan dan 3 pengulangan. Metode evaluasi termasuk metrik akurasi dan kappa, serta visualisasi seperti *plot* kepadatan dan *boxplot.* Hasilnya menunjukkan hasil *Random Forest* dengan akurasi serta nilai Kappa tertinggi (akurat: 83,3–91,6%; Kappa: 0,750–0,877), diikuti oleh CART dan SVM. KNN dan NB menunjukkan kinerja yang lebih rendah. Penelitian ini menyimpulkan bahwa pendekatan *trial-and-error* dalam memilih dan menyetel algoritma dapat meningkatkan akurasi prediksi, dengan *Random Forest* dan CART sebagai pilihan utama untuk dataset yang digunakan.

Penelitian yang dilakukan Nie et al., (2025). Studi ini mengembangkan model prediksi klasifikasi *reservoir* menggunakan pembelajaran mesin dengan akurasi tinggi berdasarkan data pengeboran real-time. Kontribusi utama meliputi metode pemilihan fitur untuk data pengeboran seperti resistivitas, GR, pengukuran gas, dan lainnya, serta pembangunan basis data pengeboran skala besar untuk klasifikasi reservoir.

Penelitian yang dilakukan Kordbagheri et al., (2025) mengenai prediksi menggunakan *machine learning***.** Hasil menunjukkan bahwa model ML tingkat lanjut mengungguli regresi

logistik pada data pelatihan dan pengujian. Model seperti kNN, GBE, dan RF memiliki skor tertinggi berdasarkan pseudo-R2 dan AUC. Nilai pseudo-R2 bervariasi, dengan kNN mencatat nilai tertinggi sebesar 0,099, sementara regresi logistik terendah di 0,022. Subskala keramahan paling berpengaruh dalam memprediksi penyelesaian pendidikan universitas.

Penelitian oleh Maximilian Lowin (2024) mengusulkan pendekatan berbasis penambangan aturan asosiasi (ARM) dan model bahasa besar (LLM) untuk pemeliharaan prediktif dalam manajemen fasilitas. Dengan menggabungkan ARM temporal dan kesamaan semantik dari LLM, metode ini dapat mengidentifikasi kebutuhan pemeliharaan yang akan datang dari permintaan layanan berbasis teks. Hasil studi kasus menunjukkan bahwa model bahasa Jerman besar memberikan hasil terbaik, dan penerapan filter temporal lift memungkinkan penyaringan aturan yang lebih relevan. Pendekatan ini efektif meskipun tanpa data berlabel, memberikan wawasan berharga bagi pengambil keputusan dalam manajemen fasilitas.

2.1.3 Posisi Penelitian

Berikut merupakan posisi penelitian berdasarkan penelitian terdahulu yang menjadi acuan referensi:

Tabel 2. 1 Posisi Penelitian

No	Penulis	Objek	Research Focus		
			Text Mining	***TF-IDF***	***Machine Learning***
1	(Sari et al., 2021)	Skripsi Mahasiswa	√	√	
2	(Chamira, 2022)	Konten Diskusi Online	√	√	
3	(Arsad et al., 2024)	Dokumen Skripsi	√		
4	(Farhan et al., 2023)	Pasal UU ITE	√		√
5	(Hidayah & Dodiman, 2024)	Data Monitoring	√	√	
6	(Timmisetty et al., 2024)	Tingkat Keparahan Masalah Mobil	√		√

No	Penulis	Objek	Research Focus		
			Text Mining	***TF-IDF***	***Machine Learning***
7	(Khodadadi et al., 2022)	Laporan Layanan Pelanggan			√
8	(J. Li & Wu, 2023)	Laporan Kecelakaan Konstruksi	√		
9	(F. Li, 2020)	Data Ucapan & Teks Kendaraan			√
10	(D. Li, 2024)	Data Evaluasi Pengajaran Pendidikan			√
11	(Sathishkumar & Ibrahim, 2025)	Data Operasi Pembelajaran Mesin			√
12	(Gupta et al., 2022)	Data Kinerja Mesin			√
13	(Nie et al., 2025)	Data Pengeboran			√
14	(Kordbagheri et al., 2025)	Data Pendidikan			√
15	(Lowin, 2024)	Data Pemeliharaan Manajemen Fasilitas			√
16	**Usulan**	*Database Maintenance* (Jenis Perbaikan Kendaraan)	√	√	√

Berdasarkan tabel 2.1 yang menjelaskan *research gap*, keterbaruan penelitian ini yaitu terletak pada tujuan penelitian, kondisi data, jenis data perbaikan kendaraan yang digunakan, lokasi penelitian, serta waktu penelitian yang dilakukan. Fokus utama penelitian ini adalah

mengidentifikasi jenis perbaikan kendaraan dengan menggunakan data Surat Perintah Kerja (SPK). Data Surat Perintah Kerja (SPK) mengalami ketidaksesuaian (*nonconfermity*) antara kategori jenis perbaikan dengan kerusakan aktual pada kendaraan sehingga diperlukan metode untuk mengatasi permasalahan tersebut. Selain itu, penelitian ini juga akan membandingkan tingkat akurasi performa model algoritma *machine learning*.

2.2 Landasan Teori

2.2.1 Surat Perintah Kerja/Laporan Teknisi

Surat Perintah Kerja (SPK) atau laporan teknisi adalah dokumen resmi yang berfungsi sebagai panduan pelaksanaan pekerjaan oleh teknisi. Di bidang otomotif, SPK biasanya mencatat detail seperti keluhan pelanggan, pemeriksaan awal kendaraan, tindakan perbaikan yang dilakukan, dan catatan tambahan terkait kondisi kendaraan. SPK menjadi sumber data penting karena berisi informasi tekstual yang menggambarkan jenis masalah, tingkat keparahan, serta riwayat perawatan kendaraan. Dalam penelitian berbasis text mining, data SPK digunakan untuk mengekstraksi pola-pola kerusakan kendaraan dan mendukung sistem klasifikasi perbaikan.

2.2.2 *Knowledge Discovery in Database* (KDD)

Knowledge Discovery in Database (KDD) adalah proses sistematis untuk menemukan pola tersembunyi, hubungan, atau informasi baru dari kumpulan data besar (Safii, 2018). Proses *Knowlegde Discovery in Database* (KDD) mencakup tahapan *data mining* (proses pengekstrak kecenderungan pada pola data), kemudian mengubah hasil tersebut secara tepat menjadi informasi yang mudah dipahami (Riandari & Simangunsong, 2019). KDD sangat penting dalam penelitian ini karena membantu mengubah data laporan teknisi yang tidak terstruktur menjadi informasi terklasifikasi yang bisa digunakan untuk pengambilan keputusan, seperti menentukan jenis kerusakan kendaraan atau menyarankan langkah perbaikan.

Inti dari *Knowledge Discovery in Database* (KDD) dapat dijelaskan sebagai berikut (Riandari & Simangunsong, 2019):

1. *Data Selection*

   Sebelum memasuki tahap inti dalam KDD, perlu dilakukan pemilihan data berdasarkan sekumpulan data operasional. Data yang sudah diseleksi kemudian disimpan dalam sebuah berkas tersendiri, terpisah dari basis data operasional.
2. *Pre-processing / Cleaning*

Tahap ini dilakukan sebelum proses *data mining* dimulai, yaitu dengan membersihkan data yang akan digunakan. Pembersihan meliputi penghapusan data ganda, pemeriksaan inkonsistensi, serta perbaikan kesalahan dalam data, seperti kesalahan penulisan (tipografi).

3. *Transformation*

   Transformasi atau *coding* dilakukan pada telah diseleksi guna sesuai untuk kebutuhan *data mining*. Tahap ini bersifat kreatif serta bergantung pada pola maupun jenis informasi yang ingin diperoleh dari basis data.

4. *Data Mining*

   Data mining merupakan tahap pencarian pola atau informasi berharga dari data yang telah diproses, dengan menggunakan metode atau algoritma tertentu. Pemilihan metode yang tepat sangat ditentukan oleh tujuan dan keseluruhan proses KDD.

5. *Interpretation / Evaluation*

   Informasi atau pola diperoleh dari data mining kemudian disajikan dalam bentuk yang mudah dipahami oleh pihak yang berkepentingan. Pada proses ini juga dilakukan evaluasi guna memastikan bahwa pola yang ditemukan tidak bertentangan dengan fakta maupun hipotesis sebelumnya.

2.2.3 *Data Mining*

Data mining merupakan proses untuk memperoleh informasi penting dari sekumpulan data berukuran besar. Dalam penerapannya, data mining banyak memanfaatkan metode statistika, matematika, serta kecerdasan buatan (*Artificial Intelligence*). Tahapan yang dilalui dalam proses ini mencakup pembersihan data, integrasi, seleksi, transformasi, penambangan data, evaluasi pola, hingga penyajian data.

Data mining adalah proses menemukan informasi baru dengan cara mengidentifikasi pola atau aturan dalam kumpulan data berukuran besar. Proses ini terdiri dari beberapa tahapan yang bertujuan mengekstrak pengetahuan yang sebelumnya tidak dapat diperoleh secara manual. Aktivitas ini juga sering disebut sebagai *knowledge discovery in databases* (KDD), yang mencakup pengumpulan serta analisis data historis untuk mengungkap pola maupun hubungan dalam suatu *dataset.*

Data mining berfokus pada identifikasi pola menarik pada data dalam volume besar, yang bisa disimpan dalam *database*, *data warehouse*, atau sistem penyimpanan informasi

lainnya. Bidang ini berhubungan dengan ilmu lain seperti sistem basis data, *data warehouse*, statistik, *machine learning*, pengambilan informasi, dan komputasi tingkat tinggi.

Menurut Yang et al. (2020), data mining merupakan metode untuk menggali informasi serta pengetahuan tersembunyi dari kumpulan data berukuran besar. Informasi yang diperoleh melalui proses ini harus memenuhi tiga karakteristik, yaitu bersifat baru (belum pernah diketahui sebelumnya), efektif, dan praktis. Data mining sendiri terbagi menjadi dua kategori utama, yaitu predictive dan descriptive. Sementara itu, Hamonangan Nasution & Iqbal Sofyan (2020) menjelaskan bahwa istilah "*mining*" berasal dari bahasa Inggris yang berarti menambang atau menggali sumber daya di dalam tanah. Dengan demikian, data mining dapat dipahami sebagai proses menambang informasi tersembunyi dari kumpulan data dengan volume yang sangat besar. Lebih jauh, data mining berkaitan erat dengan berbagai disiplin ilmu, seperti kecerdasan buatan (Artificial Intelligence), machine learning, statistika, dan basis data.

2.2.4 *Pre-processing Data*

Pre-processing data merupakan tahapan awal dalam mengolah data yang harus dilakukan sebelum dapat dilakukan perhitungan oleh algoritma yang bertujuan untuk membersihkan data dari anomali sehingga data dapat menjadi terstruktur (Farhan et al., 2023b). *Pre-processing* dapat didefinisikan sebagai proses mempersiapkan data mentah sebelum dilakukan proses selanjutnya. Data teks mentah yang diperoleh biasanya merupakan data tidak terstruktur serta memiliki banyak data *noise* seperti tanda baca, imbuhan, angka, karakter – karakter khusus serta lain sebagainya. Berikut merupakan beberapa tahapan *Pre-processing data*:

1. *Cleaning Data*

   Tahap ini guna membersihkan teks pada elemen yang tidak relevan, misalnya karakter khusus maupun data ganda. Pembersihan dilakukan untuk mengurangi *noise* yang dapat mengganggu proses klasifikasi.
2. *Folding Data*

   Pada tahap ini, teks mengalami transformasi format, misalnya dengan mengubah huruf kapital menjadi huruf kecil serta menghapus konten yang bukan berupa karakter alfabet. Hanya huruf A sampai Z yang dipertahankan, sedangkan karakter lainnya dihilangkan (Afida et al., 2021).
3. *Tokenize Data*

Pada tahap ini, dilakukan dengan memecah struktur teks menjadi kata-kata yang lebih kecil. Kalimat diuraikan menjadi kata per kata agar dapat dianalisis lebih mendalam. (Afida et al., 2021). Tahapan ini memiliki tujuan untuk memisahkan setiap kata dalam teks sehingga struktur dapat dianalisis lebih dalam.

4. *Stopwords Data*

   Tahap ini berfokus pada penghapusan kata-kata yang dianggap tidak memiliki makna signifikan. Prosesnya mengikuti daftar kata yang tercantum dalam *stopwords list (Afida et al., 2021)*.

5. *Stemming Data*

   Stemming dilakukan guna mengembalikan kata ke bentuk dasarnya dengan cara menghapus imbuhan, baik berupa awalan (*prefix*), akhiran (*suffix*), maupun gabungan (*konfiks*). Tujuannya adalah untuk menyamakan bentuk kata sehingga lebih konsisten saat dianalisis (Arsad et al., 2024).

2.2.5 *Text Mining*

Text mining merupakan proses penggalian informasi dari kumpulan data berbentuk teks, yang biasanya bersumber dari berbagai dokumen. Tujuan utama *text mining* adalah mengidentifikasi kata – kata yang mampu merepresentasikan isi dokumen secara keseluruhan untuk memungkinkan dilakukannya analisis keterkaitan antar dokumen. Tidak seperti data dalam bentuk rekaman basis data, *text mining* berfokus pada informasi yang berasal dari teks tidak terstruktur. Proses awal yang penting dalam *text mining* adalah prapengolahan data, yakni tahap mempersiapkan teks agar bisa diproses lebih lanjut. Dalam tahap ini, rangkaian karakter dalam bentuk teks diuraikan menjadi bagian-bagian yang lebih kecil dan bermakna, misalnya menjadi bab, sub-bab, paragraf, kalimat, kata, atau bahkan suku kata. Metode ini dimulai dengan mengidentifikasi istilah - istilah utama dalam dokumen, kemudian melanjutkan ke analisis hubungan antar dokumen dengan menggunakan teknik statistik seperti *clustering*, *classification*, dan *association* (Aditya et al., 2015).

Metode ini berguna untuk menganalisis opini, evaluasi, sikap, penilaian, serta emosi seseorang guna mengetahui keterkaitannya dengan suatu topik, layanan, organisasi, ataupun kegiatan tertentu (Wulandari et al., n.d.). Proses kerja *text mining* pada dasarnya banyak mengadaptasi konsep dari penelitian *data mining*. Perbedaan utama antara keduanya terletak pada sumber pola yang digunakan, di mana *text mining* mengekstraksi pola dari bahasa alami yang bersifat tidak terstruktur, sedangkan *data mining* mengambil pola dari basis data yang

terstruktur. Secara umum, tahapan algoritma *text mining* diawali dengan proses *pre-processing teks* (Ridwansyah, 2022).

Tujuan dilakukannya *text mining* adalah guna mendapatkan informasi dalam sekumpulan dokumen sehingga dapat digunakan oleh peneliti untuk mendapatkan informasi yang berguna serta guna kepentingan pengambilan keputusan. Menurut Ervita Kusuma Putri & Tedy Setiadi (Putri & Setiadi, 2014), *text mining* memiliki tugas khusus seperti pengkategorisasian teks dan juga pengelompokan teks. Dalam memberikan solusi atas suatu permasalahan *text mining* memberikan solusi dalam bentuk *machine learning, information retrieval* dan *data mining.* Langkah – langkah dalam *text mining* adalah yang pertama *preprocessing* teks, *core mining operations, presentation layer components* dan langkah yang keempat sebagai langkah terakhir adalah *refinement techniques.*

2.2.6 TF-IDF

Penelitian ini menggunakan metode pembobotan *Term Frequency Inverse Document Frequency* (TF-IDF) karena memiliki sejumlah keunggulan dibandingkan metode pembobotan lain. TF-IDF mampu memberikan bobot yang lebih besar pada kata – kata yang spresifik dan relevan dengan dokumen tertentu, sekaligus mengurangi bobot kata- kata umum yang muncul dibanyak dokumen. Hal ini menjadikannya sangat sesuai untuk klasifikasi laporan perbaikan kendaraan, di mana istilah teknis tertentu memiliki makna penting dalam membedakan kategori perbaikan. Dibandingkan dengan *Bag-of-Words* yang hanya menghitung frekuensi kata tanpa mempertimbangkan pentingnya kata dalam keseluruhan dokumen, TF-IDF terbukti menghasilkan akurasi klasifi yang lebih tinggi. Sementara itu, metode representasi berbasis *word of embeddings* seperti *Word2Vec* atau *FastText* memang mampu menangkap hubungan semantik antar kata, tetapi membutuhkan data yang lebih sulit.

Selain itu, metode pembobotan lain seperti BM25 atau variasi *neutrosophic* TF-IDF memang dapat memberikan peningkatan akurasi, namun cenderung lebih kompleks dalam implementasi dan memerlukan penyesuaian parameter tambahan. Oleh karena itu, pemilihan TF-IDF dalam penelitian ini didasarkan pada pertimbangan keseimbangan antara akurasi, efisiensi komputasi, kemudahan implementasi, serta interpretabilitas hasil, sehingga dapat mendukung kebutuhan praktis di kantor pusat dalam meningkatkan akurasi evaluasi kinerja.

Pembobotan *Term Frequency Inverse Document Frequency* (TF-IDF) merupakan metode umum yang digunakan untuk mengukur skala keterhubungan suatu kata (*term*) terhadap

dokumen atau kalimat, dengan cara memberikan bobot atau nilai tertentu pada setiap kata (Arifin et al., n.d.). Tujuan dari TF-IDF adalah untuk mengidentifikasi kata-kata yang paling penting dalam suatu dokumen atau kumpulan dokumen. *Term Frequency* (TF) adalah nilai frekuensi kemunculan suatu kata dalam sebuah dokumen (Wati et al., 2023). *Invers document frequency* akan mengidentifikasi bahwa kemunculan suatu istilah dengan frekuensi yang tinggi pada suatu data atau dokumen merupakan hal yang sangat penting. Begitu juga sebaliknya dengan kemunculan kata atau istilah yang rendah dianggap tidak penting pada suatu dokumen karena pembahasannya tidak mewakili dokumen tersebut (Yu et al., 1982). TF-IDF berasal dari perkalian antara *term frequency* (TF) dengan *inverse document frequency* (IDF).

Term Frequency dan *Inverse Document Frequency* (TF-IDF) merupakan pembobotan yang sering digunakan dalam penelusuran informasi dan *text mining* (Agusvina & Santoso, 2018).

Cara kerja dalam mencari nilai *term frequency* melalui beberapa persamaan yaitu:

$$\mathrm{Tf_{t,d}} = 1 + {}^{10}\mathrm{Log\ tf} \tag{2.1}$$

Dimana setiap variabel dijelaskan sebagai berikut:

Tf : *term frekuensi atau banyaknya kata pada dokumen*

$\mathrm{Tf_{t,d}}$: term frekuensi atau banyaknya kata t pada dokumen d atau pembobotan local

Mencari nilai *inverse document-frequenct* melalui persamaan 2:

$$\mathrm{Idf_t} = {}^{10}\mathrm{Log}\ {}^{\mathrm{n}}/\mathrm{df_t} \tag{2.2}$$

$\mathrm{Idf_t}$: i*nverse document-frequency* atau pembobotan global

n : banyaknya dokumen

$\mathrm{df_t}$: banyaknya dokumen yang memiliki kata t

berdasarkan persamaan 1 dan persamaan 2 baru dapat ditentukan nilai bobotnya ($\mathrm{W_{t,d}}$) dengan mengalikan kedua persamaan sehingga menjadi persamaan 3

$$\mathrm{W_{t,d}} = tf_{t,d} x\ idf_t \tag{2.3}$$

$\mathrm{Tf_{t,d}}$: term frekuensi atau banyaknya kata t pada dokumen d atau pembobotan local

$\mathrm{Idf_t}$: i*nverse document-frequency* atau pembobotan global

$W_{\mathrm{t,d}}$: nilai bobot akhir kata

Kemudian lakukan normalisasi pada bobot yang telah didapatkan dengan menggunakan persamaan 4.

$$\frac{W_{t,d}}{\sqrt{\sum_{t=1}^{n}(W_{t,s})^2}} \tag{2.4}$$

Dimana setiap variabel dijelaskan sebagai berikut:

n : banyaknya dokumen

t : iterasi kata ke-

$W_{t,d}$: nilai bobot akhir kata

Teknik ini membantu mengidentifikasi pentingnya suatu kata dalam dokumen tertentu maupun dalam keseluruhan koleksi dokumen, dengan cara menghitung bobot berdasarkan seberapa sering kata tersebut muncul (Alessa & Faezipour, 2019).

2.2.7 *Machine Learning*

Machine Learning merupakan salah satu metode pembelajaran yang termasuk dalam cabang kecerdasan buatan, di mana sistem mampu belajar secara mandiri tanpa campur tangan manusia secara langsung. *Machine learning* didefinisikan sebagai penerapan algoritma matematika dan aplikasi komputer yang belajar dari data, kemudian menghasilkan prediksi untuk masa mendatang (Goldberg & Holland, 1988).

Algoritma dalam *Machine Learning* membentuk model matematika dengan mengandalkan data pelatihan (*training data*) yang digunakan untuk membuat prediksi atau mengambil keputusan. Fokus utama dari pendekatan ini adalah pengembangan program komputer yang mampu secara otomatis memperbaiki kinerjanya berdasarkan pengalaman yang terus bertambah.

Terdapat berbagai jenis algoritma dalam *Machine Learning* yang secara umum terbagi menjadi tiga kategori utama.

1. *Supervised Learning*, yaitu jenis algoritma yang mempelajari hubungan antara data masukan dan keluaran melalui proses pelabelan. Algoritma ini banyak digunakan untuk menyelesaikan permasalahan klasifikasi maupun regresi.
2. *Unsupervised Learning*, yakni algoritma yang digunakan untuk mengevaluasi dan mengelompokkan data tanpa adanya label atau target keluaran, biasanya dipakai untuk analisis dan klasterisasi data
3. *Reinforcement Learning*, yaitu metode pembelajaran di mana agen belajar melalui interaksi langsung dengan lingkungan. Algoritma ini memungkinkan sistem

menentukan strategi terbaik secara mandiri melalui proses percobaan dan kesalahan dalam lingkungan yang terus berubah.

2.2.8 *Naïve bayes*

Naïve Bayes adalah salah satu metode klasifikasi berbasis teori probabilitas *Bayes*. Algoritma ini mengasumsikan bahwa fitur-fitur input bersifat independen satu sama lain (*naïve assumption*) (Vernanda et al., 2024). Metode *Naïve Bayes* dikenal memiliki akurasi serta kecepatan yang baik ketika diterapkan pada basis data berukuran besar. Algoritma ini banyak digunakan dalam penyelesaian masalah machine learning karena mampu memberikan hasil akurat dengan perhitungan yang relatif sederhana (Mayasari & Indarti, 2022). Dengan pendekatan ini*, Naïve Bayes* mampu menghitung kemungkinan kelas tertentu berdasarkan fitur yang ada, sehingga cocok digunakan untuk tugas-tugas klasifikasi teks meliputi deteksi spam, analisis sentimen, dan pengelompokan laporan teknisi. Kelebihan metode ini adalah sederhana, cepat, dan tetap efektif meskipun data berukuran besar.

2.2.9 *Logistic Regression*

Logistic Regression adalah metode statistik yang digunakan untuk memprediksi kategori hasil berdasarkan satu atau lebih variabel input. Walaupun namanya "*regression*", prediksi nilai kontinu (Mayasari & Indarti, 2022). Metode *logistic regression* digunakan untuk menganalisis hubungan antara variabel respon yang bersifat dikotomus (berbentuk ordinal atau nominal dengan dua kategori) maupun polikotomus (memiliki skala ordinal atau nominal dengan lebih dari dua kategori) (Susetyoko et al., 2022). Metode ini populer dalam prediksi biner, seperti menentukan apakah suatu laporan kerusakan termasuk kategori parah atau ringan.

2.2.10 *Random Forest*

Random Forest merupakan salah satu dari sekian banyak metode *ensemble* yang dikembangkan oleh Leo Breiman pada tahun 2001. *Random Forest* adalah metode *ensemble learning* yang membangun banyak pohon keputusan (*decision tree*) untuk melakukan klasifikasi atau regresi. Setiap pohon dalam *Random Forest* dilatih menggunakan subset acak dari data, dan hasil akhir ditentukan berdasarkan voting dari semua pohon (pada klasifikasi) atau rata-rata prediksi (pada regresi) (Finda & Utomo, 2024). *Random Forest* unggul dalam mengatasi masalah *overfitting* yang sering terjadi pada *decision tree* tunggal, serta mampu menangani data berukuran besar dengan akurasi tinggi. Dalam konteks klasifikasi laporan teknisi, *Random Forest* bisa menghasilkan prediksi jenis kerusakan dengan lebih akurat.

2.2.11 *Confusion Matrix*

Confusion Matrix adalah alat evaluasi model klasifikasi yang menampilkan hasil prediksi model dibandingkan dengan label asli data (Hakim et al., 2025). Tabel ini memiliki empat komponen utama:

- *True Positive* (TP): Prediksi benar pada kelas positif.
- *True Negative* (TN): Prediksi benar pada kelas negatif.
- *False Positive* (FP): Prediksi salah di mana model memprediksi positif padahal negatif.
- *False Negative* (FN): Prediksi salah di mana model memprediksi negatif yang seharusnya positif.

Dari *Confusion Matrix*, berbagai metrik kinerja seperti akurasi, presisi, *recall*, dan *F1-score* dapat dihitung untuk menilai seberapa baik performa model. Ini penting agar kita tahu bukan hanya berapa banyak prediksi yang benar, tapi juga seberapa baik model mengenali pola penting di dalam data.

BAB III
METODE PENELITIAN

3.1 Jenis Data

Data pada penelitian ini termasuk ke dalam data sekunder. Data sekunder adalah data yang diperoleh bukan melalui pengumpulan langsung, melainkan melalui pihak atau sumber perantara. Pada penelitian ini, data sekunder yang digunakan berasal dari literatur serta hasil riset terdahulu yang membahas topik sejenis dengan penelitian yang sedang dilakukan. Selain itu, data yang digunakan pada penelitian ini yaitu data Surat Perintah Kerja (SPK) yang diambil dari *database maintenance* PT Adi Sarana Armada Tbk.

3.2 Pengumpulan Data

Data yang diambil untuk selanjutnya akan diolah dalam penelitian ini yaitu sebagai berikut:

1. Dokumentasi merupakan metode pengumpulan data dengan cara mengkaji, mengumpulkan, mencatat, serta menelaah dokumen – dokumen atau arsip yang berkaitan dengan objek penelitian. Dokumen ini dapat berupa data tertulis, laporan, catatan, maupun bentuk arsip digital yang tersedia sebelumnya. Dalam penelitian ini teknik yang diterapkan yaitu pengambilan data Surat Perintah Kerja (SPK) dari *database maintenance* PT Adi Sarana Armada Tbk, yang berisi informasi deskripsi pekerjaan atau keluhan kendaraan, yang kemudian menjadi bahan utama untuk analisis.
2. Studi Literatur, dilakukan sebagai metode pengumpulan data yang mendukung penelitian dengan menelusuri berbagai literatur yang relevan. Sumber literatur dapat berasal dari buku, jurnal, situs web, tesis, maupun referensi lainnya. Dalam penelitian ini, studi literatur mencakup topik-topik seperti *text mining, Term Frequency – Inverse Document Frequency* (TF-IDF), *machine learning*, serta penelitian terdahulu yang berkaitan.

3.3 Pengolahan Data

Langkah – langkah sistematis pengolahan pada penelitian ini dapat dijabarkan sebagai berikut:

1. *Pre-processing data*

   Preprocessing data dilakukan sebagai tahap pembersihan dan persiapan data untuk meminimalisir variasi item teks, sehingga *machine learning* dapat mempelajari data

secara mudah dan akurat. Dalam penelitian ini, tahapan *preprocessing data* yang diterapkan meliputi *cleaning, folding, tokenize, stopword,* serta *steeming data.*

2. Representasi teks dengan TF-IDF

   Representasi teks dengan TF-IDF digunakan untuk merubah data teks menjadi representasi numerik agar dapat diproses oleh algoritma *machine learning.* Dalam penelitian ini, digunakan metode *Term Frequency - Inverse Document Frequency* (TF-IDF) sebagai Teknik konversi teks.

3. *Splitting data*

   Sebelum membangun model, data terlebih dahulu dibagi menjadi dua bagian, yaitu *training set* dan *testing set* untuk menguji kinerjanya. Dalam penelitian ini, pembagian data dilakukan dengan rasio umum yaitu, 80% *training data* dan 20% *testing data.*

4. Pelatihan model

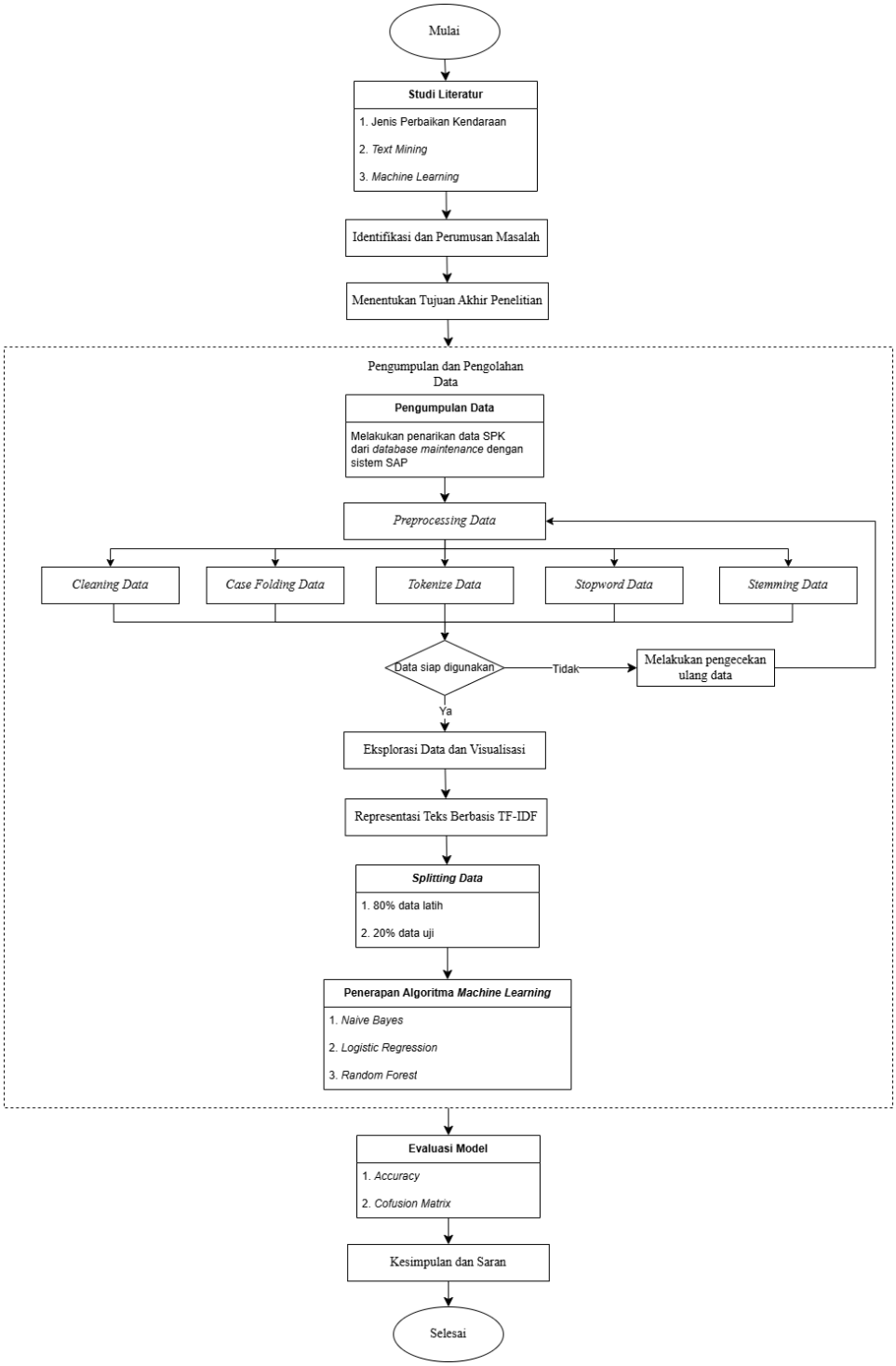
   Setelah dilakukan pembagian data, selanjutnya akan dilatih dengan model klasifikasi. Algoritma *machine learning* yang digunakan dalam penelitian ini antara lain *Naive Bayes* (NB), *Logistic Regression* (LR), dan *Random Forest* (RF).

5. Evaluasi model

   Untuk menilai kinerja model yang telah dilatih maka, dilakukan evaluasi menggunakan *confusion matrix* untuk menampilkan jumlah prediksi yang benar dan salah dari model klasifikasi berdasarkan kategorinya.

3.4 Alur Penelitian

Proses yang diterapkan dalam penelitian ini digambarkan melalui alur penelitian yang ditampilkan pada Gambar 3.1 sebagai berikut:



Gambar 3. 1 Alur Penelitian

Berikut merupakan uraian dari alur penelitian pada gambar 3.1 sebagai berikut:

1. Studi Literatur

   Tahap awal dalam penelitian ini yakni melakukan studi literatur, di mana penulis menelaah penelitian – penelitian terdahulu yang memiliki keterkaitan serta dapat

dijadikan sebagai landasan dan pendukung bagi penelitian yang sedang dilakukan, penelitian tersebut mengenai jenis perbaikan, *text mining,* dan *machine learning*.

- Jenis perbaikan kendaraan : Mengkaji karakteristik perbaikan kendaraan berdasarkan data historis, pada penelitian ini yaitu data Surat Perintah Kerja (SPK)
- *Text Mining* : Pemahaman terhadap teknik – teknik *text mining* yang digunakan guna mengekstrak informasi dari data teks tidak terstruktur.
- *Machine Learning* : Kajian terhadap algoritma *supervised learning* yang umum digunakan dalam klasifikasi teks, meliputi *naïve bayes, logistic regression,* serta *random forest.*

2. Identifikasi dan Perumusan Masalah

   Identifikasi masalah yang dilakukan pada tahap ini bertujuan untuk menemukan dan memahami permasalahan yang ada dalam sistem perbaikan kendaraan. Masalah utama yang akan diidentifikasi adalah bagaimana menganalisis jenis perbaikan kendaraan yang dilakukan, berdasarkan deskripsi yang terdapat dalam Surat Perintah Kerja (SPK). Dengan identifikasi yang dilakukan, maka dapat disusun rumusan masalah yang kemudian akan diteliti.

3. Menentukan Tujuan Akhir Penelitian

   Tahapan ini bertujuan untuk memberikan arah dan fokus terhadap keseluruhan proses penelitian. Dalam konteks ini, tujuan akhir dirumuskan berdasarkan identifikasi masalah serta hasil studi literatur yang telah dilakukan sebelumnya.

4. Pengumpulan Data

   Pengumpulan data sekunder dilakukan dengan mengumpulkan data Surat Perintah Kerja (SPK) yang terdapat dalam *database maintenance* dengan sistem SAP. Data ini berupa teks deskriptif yang mencakup jenis perbaikan kendaraan atau perawatan yang terjadi. Data diambil dalam format *excel,* kemudian diorganisir untuk mempermudah analisis lebih lanjut.

5. *Pre-processing Data*

   Tahap *preprocessing data* dilakukan untuk membersihkan data teks yang telah dikumpulkan guna mempersiapkan untuk dilakukan pengolahan atau analisis, sehingga mesin akan lebih mudah untuk mengenali data. Adapun tahapan *preprocessing data* yang dilakukan pada penelitian ini yaitu sebagai berikut:

   a. *Cleaning Data*

Menghapus karakter yang tidak relevan, seperti tanda baca atau angka.

b. *Case Folding Data*

Tahapan untuk menyeragamkan seluruh huruf, untuk menghindari perbedaan penulisan kata.

c. *Tokenize Data*

Memisahkan teks menjadi kata – kata atau kalimat yang lebih kecil, sehingga hanya menyisakan kata – kata yang dibutuhkan saja.

d. *Stopword Removal Data*

Menghilangkan kata – kata yang tidak bermakna atau tidak memberikan informasi penting misalnya, kata hubung atau kata umum lainnya.

e. *Steeming Data*

Mengubah kata ke bentuk dasarnya misalnya, ”perbaikan” menjadi ”perbaiki”.

6. Data Siap Digunakan

Tahapan ini merupakan proses validasi untuk memastikan bahwa data hasil *preprocessing* sudah cukup bersih dan representatif. Jika data belum memenuhi kriteria, akan dilakukan pengecekan data kembali untuk memastikan data tersebut valid untuk digunakan, kemudian akan dilakukan pengulangan proses *preprocessing* hingga data dinyatakan siap.

7. Eksplorasi Data dan Visualisasi

Setelah data dinyatakan siap, maka dilakukan eksplorasi data untuk memahami struktur, karakteristik, dan pola yang terdapat dalam data. Aktivitas ini meliputi analisis sstatistik deskriptif, visualisasi frekuensi kata (*word cloud* dan *histogram*).

8. Representasi Teks Berbasis TF-IDF

Setelah teks dibersihkan, dilakukan konversi menggunakan *Term Frequency – Inverse Document Frequency* (TF-IDF). Proses ini mengubah data teks menjadi representasi numerik, di mana setiap kata diberikan bobot berdasarkan frekuensinya dalam sebuah dokumen.

9. *Splitting Data*

Data yang telah diproses, selanjutnya dibagi menjadi dua bagian yaitu, data latih (*training data*) dan data pengujian (*testing data*). Data latih digunakan untuk membangun model klasifikasi, sedangkan data pengujian digunakan untuk menguji

seberapa baik model tersebut bekerja pada data baru. Pembagian data pada penelitian ini yaitu 80% data latih dan 20% data pengujian.

10. Penerapan Algoritma *Machine Learning*

    Analisis pada penelitian ini dilakukan dengan algoritma *Naive Bayes, Logistic Regression,* dan *Random Forest.* Setiap algoritma memiliki karakteristik yang berbeda dalam menangani data teks, sehingga penggunaan beberapa model memungkinkan dilakukan perbandingan kinerja antar algoritma dalam mengklasifikasikan jenis perbaikan kendaraan.

11. Evaluasi Model

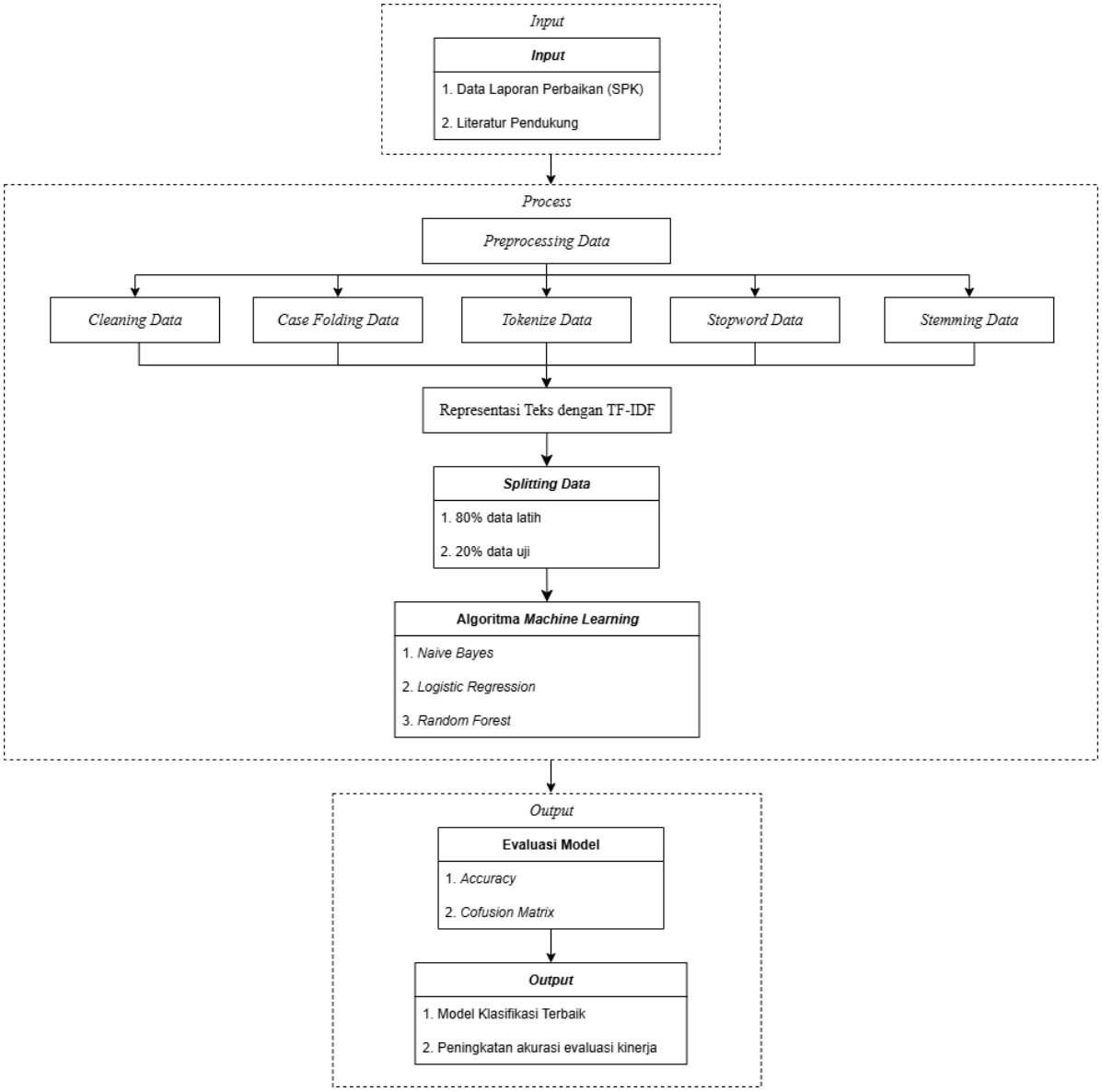
    Evaluasi model dilakukan untuk mengukur performa masing – masing model klasifikasi. Evaluasi ini menggunakan *confusion matrix* yang menunjukkan jumlah prediksi benar dan salah untuk tiap kelas, termasuk nilai *True Positive, False Positive, True Negative,* dan *False Negative.* Sedangakan *acuracy* guna mengukur seberapa banyak prediksi yang benar dari seluruh data uji.

12. Kesimpulan dan Saran

    Berdasarkan hasil pengolahan data yang dilakukan, maka dapat ditarik kesimpulan dan saran yang dapat membantu penelitian selanjutnya. Kesimpulan mencakup pencapaian tujuan penelitian serta jawaban atas permasalahan yang telah dirumuskan. Selain itu, diberikan saran untuk penelitian lanjutan, misalnya memperluas data, melakukan *preprocessing* yang lebih kompleks, atau mencoba algoritma *machine learning* yang lebih sesuai

3.5 Kerangka Konseptual Penelitian

Kerangka konseptual penelitian merupakan gambaran alur berpikir yang digunakan dalam menyelesaikan permasalahan. Kerangka ini menjelaskan bagaimana data penelitian diproses melalui tahapan – tahapan tertentu hingga menghasilkan *output* yang dapat menjawab rumusan masalah. Berikut merupakan kerangka konseptual sebagai berikut:



Gambar 3. 2 Kerangka Konseptual Penelitian

Berikut merupakan uraian kerangka konseptual pada gambar 3.2 sebagai berikut:

1. *Input*
    a. Data laporan perbaikan kendaraan (SPK)

       Data SPK yang berasal dari *database maintenance.* Data ini berisikan catatan deskriptif mengenai jenis perbaikan kendaraan yang dilakukan oleh *service advisor.* Data inilah yang menjadi objek utama untuk dianalisis dan diklasifikasikan.
    b. Literatur pendukung

Literatur mencakup teori terkait *text mining,* pembobotan teks (TF-IDF), dan algoritma *machine learning,* yang digunakan sebagai landasan konseptual dalam penerapan model

2. *Process*
   a. *Preprocessing data*
      - *Cleaning Data*: menghapus karakter yang tidak relevan (angka, tanda baca, simbol).
      - *Case Folding*: menyeragamkan format huruf menjadi huruf kecil.
      - *Tokenizing*: memecah teks menjadi kata-kata.
      - *Stopword Removal*: menghapus kata-kata umum yang tidak bermakna penting.
      - *Stemming*: mengembalikan kata ke bentuk dasar.
   b. Representasi teks dengan TF-IDF

      Pada tahap ini, teks yang sudah diproses akan dikonversi menjadi representasi numerik dengan metode TF-IDF. Metode ini dipilih karena mampu memberikan bobot yang lebih proporsional antara kata yang sering muncul dengan kata yang jarang tetapi penting.
   c. *Splitting data*

      Data yang telah direpresentasikan dibagi menjadi data latih (80%) dan data uji (20%) untuk keperluan pembelajaran model dan pengujian performa.
   d. Penerapan algortima *machine learning*

      Model klasifikasi dibangun menggunakan tiga algoritma, yaitu Naive Bayes, Logistic Regression, dan Random Forest. Ketiga algoritma ini dipilih untuk dibandingkan performanya dalam mengklasifikasikan jenis perbaikan kendaraan.
3. *Output*
   a. Evaluasi model

      Model klasifikasi dibangun menggunakan tiga algoritma, yaitu Naive Bayes, Logistic Regression, dan Random Forest. Ketiga algoritma ini dipilih untuk dibandingkan performa akurasi dalam mengklasifikasikan jenis perbaikan kendaraan.
   b. *Output*
      - Model klasifikasi terbaik

Model *machine learning* yang memiliki performa akurasi tertinggi berdasarkan hasil evaluasi. Model ini diharapkan mampu mengklasifikasikan laporan perbaikan kendaraan dengan lebih akurat.

- Peningkatan akurasi evaluasi kinerja

  Dengan adanya klasifikasi yang akurat, ketidaksesuaian informasi antara lapangan dengan kantor pusat dapat diminimalisir. Hal ini berdampak pada peningkatan akurasi evaluasi, efisiensi operasional, dan mendukung pengambilan keputusan yang lebih tepat.

BAB IV
PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Pengumpulan Data

Pengumpulan data dilakukan untuk memenuhi kebutuhan penelitian yang sedang dilaksanakan. Pada penelitian ini, data yang digunakan berupa data historis dari laporan Surat Perintah Kerja (SPK) sesuai dengan perawatan / perbaikan kendaraan yang dilakukan. Data yang digunakan secara menyeluruh berjumlah 250420 data dengan kolom sebanyak 3 kolom, yang mencakup periode waktu mulai 1 Januari 2024 hingga 31 Maret 2024.

Total data: 250420

	Order Desc	Item Text	Jenis Perbaikan
0	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	NO PO : 4102186285 (Internal)	Adhoc
1	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	Adhoc
2	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	Adhoc
3	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	V-Belt	Adhoc
4	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	V-Belt P/S	Adhoc
...	...	...	...
250415	PB,BE-9665-YB,KM.125586,PB.125	Lampu box	PB
250416	PBTT,BD-1316-EC,KM.30361,G/T WIPER	G/T WIPER AUS (Internal)	PBTT
250417	PBTT,BD-1316-EC,KM.30361,G/T WIPER	PO 4102251025 (Internal)	PBTT
250418	PBTT,BD-1316-EC,KM.30361,G/T WIPER	Wiper	PBTT
250419	PBTT,BD-1316-EC,KM.30361,G/T WIPER	Wiper	PBTT

250420 rows × 3 columns

Gambar 4. 1 Hasil Pengumpulan Data

4.2 Pengolahan Data

Pada tahapan ini dilakukan proses pengolahan data teks dari Surat Perintah Kerja (SPK) yang telah dikumpulkan dengan tujuan untuk mempersiapkan data agar dapat digunakan dalam model klasifikasi berbasis *machine learning.* Proses pengolahan dimulai dari tahapan *Preprocessing Data* hingga representasi data dalam bentuk numerik menggunakan *Term Frequncy – Invers Document Frequency* (TF-IDF).

Tahapan ini penting dilakukan guna mengurangi *noise* pada data teks, memperbaiki kualitas data, dan menyesuaikan dengan kebutuhan algoritma klasifikasi yang akan dilaksanakan pada penelitian ini. Jumlah data yang digunakan pada tahap ini dapat berubah dari jumlah awal, disebabkan karena penyesuaian dengan hasil dari proses pembersihan yang ditetapkan dalam penelitian. Adapun proses pengolahan data yang dilakukan adalah sebagai berikut:

4.2.1 *Preprocessing Data*

Preprocessing data merupakan tahapan awal dalam mempersiapkan data mentah agar menjadi data terstruktur dan layak untuk dianalisis lebih lanjut. *Preprocessing data* menggunakan *tools Python* dan *Google Colaboratory* yang dilakukan meliputi tahapan berikut:

a. *Cleaning Data*

Pada tahapan ini dilakukan pemilihan variabel yang difokuskan untuk penelitian ini. Dari sekian banyak informasi yang diberikan terdapat 3 variabel yang akan digunakan yaitu "*Order Desc", "Item Text",* dan "Jenis Perbaikan". Selain itu dilakukan pemilihan data berdasarkan waktu, data yang digunakan yaitu pada tanggal 1 Januari 2024 hingga tanggal 31 Maret.

Pada tahapan *cleaning data* akan dilakukan penghapusan *duplicate data* dan *noise data.* Hal tersebut dilakukan agar semua data yang dianalisis merupakan data yang tidak berulang dan hasil akan lebih akurat.

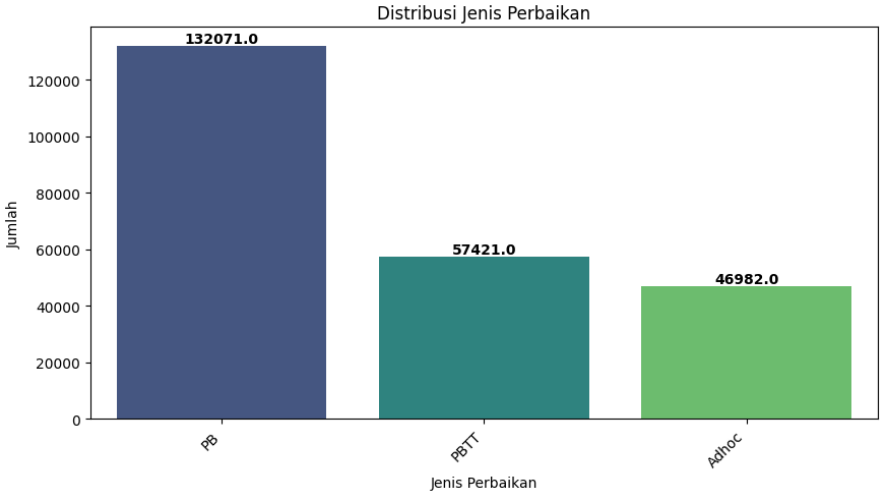
```
Total data after preprocessing: 236474
```

	Order Desc	Item Text	Jenis Perbaikan	Cleaned_Text
0	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	NO PO : 4102186285 (Internal)	Adhoc	
1	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	Adhoc	pabucukmgt vbelt
2	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	V-Belt	Adhoc	vbelt
3	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	V-Belt P/S	Adhoc	vbelt ps
4	PA,B-9215-UXA,KM.47940,STOORING	KUNJUNGAN ASTRA WORLD (Internal)	Adhoc	kunjung
...	...	...	...	...
236469	PB,BE-9665-YB,KM.125586,PB.125	Lampu	PB	lampu
236470	PB,BE-9665-YB,KM.125586,PB.125	Lampu box	PB	lampu box
236471	PBTT,BD-1316-EC,KM.30361,G/T WIPER	G/T WIPER AUS (Internal)	PBTT	wiper aus
236472	PBTT,BD-1316-EC,KM.30361,G/T WIPER	PO 4102251025 (Internal)	PBTT	
236473	PBTT,BD-1316-EC,KM.30361,G/T WIPER	Wiper	PBTT	wiper

236474 rows × 4 columns

Gambar 4. 2 Hasil *Cleaning Data*

Dapat dilihat pada Gambar 4.2 menunjukkan bahwa terdapat pengurangan data sebanyak 13.946 data akibat *cleaning data,* sehingga jumlah data yang sebelumya 250420 data menjadi 236474 data. Selain itu, terdapat penambahan *colomn* baru hasil dari tahapan *cleaning data* yaitu *Cleaned_Text.*



Gambar 4. 3 Distribusi Jenis Perbaikan

Pada Gambar 4.3 dapat dilihat bahwa distribusi data berdasarkan jenis perbaikan, di mana jenis perbaikan yang paling dominan adalah Perbaikan Berkala (PB) sebanyak 132.071 data. Hal ini menunjukkan bahwa sebagian besar kendaraan menjalani proses pemeliharaan rutin sesuai dengan jadwal yang ditentukan. Selanjutnya, Perbaikan Berkala Tidak Teratur (PBTT) berjumlah 57.421 data, menandakan bahwa masih terdapat sejumlah besar kendaraan yang menjalani perwatan di luar jadwal perbaikan. Sedangkan, kategori Adhoc sebanyak 46.982 data mencerminkan adanya kebutuhan perbaikan mendesak di luar jadwal yang telah direncanakan.

Distribusi ini memberikan gambaran awal terhadap komposisi label dalam data yang akan digunakan untuk klasifikasi. Dominasi kategori PB juga menunjukkan ketidakseimbangan data (*Imbalance Class*) yang perlu diperhatikan dalam proses pelatihan model *machine learning,* karena dapat memengaruhi performa prediksi untuk kelas minoritas seperti PBTT dan Adhoc.

b. *Case Folding Data*

Tahap *case folding* yakni mengubah seluruh huruf kapital yang ada di dalam data menjadi *lowercase.* Proses ini dilakukan karena tidak semua dokumen konsisten dalam menggunakan huruf kapital. Oleh sebab itu, dibutuhkan konversi seluruh abjad yang ada dalam dokumen menjadi huruf kecil. Proses *case folding* ditunjukkan pada Tabel 4.1.

Tabel 4. 1 Proses *Case Folding*

Sebelum *Case Folding*	**Sesudah *Case Folding***
G/T Oli Mesin + Oli Filter	g/t oli mesin + oli filter
Ganti Oli Mesin + Oli Filter (Internal)	ganti oli mesin & filter oli (internal)
G/T,Oli Mesin,SHELL,5.5L (Internal)	g/t,oli mesin,shell,5.5l (internal)
G/T Oli Mesin + Cek Remote Tidak Berfungsi	g/t oli mesin + cek remote tidak befungsi
G/T, ACCU, Solite,1PC, NS40ZL (Internal)	g/t, accu, solite,1pc, ns40zl (internal)
G/T Oli Filter (Internal)	g/t oli filter (internal)

c. *Tokenize Data*

Setelah data dibersihkan dari *duplicate data* kemudian dilakukan proses *tokenization* untuk menghapus tanda – tanda yang tidak digunakan seperti tanda kurung, koma, titik serta memecah kalimat menjadi sekumpulan kata atau *token*. Tahapan ini bertujuan agar pada tahap penghapusan *stop words,* mesin dapat mengenali tiap kata dalam teks dengan lebih baik. Proses serta hasil *tokenize data* ditampilkan pada Tabel 4.2

Tabel 4. 2 Proses *Tokenize Data*

Sebelum *Tokenize Data*	**Sesudah *Tokenize Data***
G/T Oli Mesin + Oli Filter	['g/t', 'oli', 'mesin', 'oli', 'filter']
Ganti Oli Mesin + Oli Filter (Internal)	['ganti', 'oli', 'mesin', 'filter', 'oli', 'internal']
G/T,Oli Mesin,SHELL,5.5L (Internal)	['g/t', 'oli', 'mesin', 'shell', '5.5l', 'internal']
G/T Oli Mesin + Cek Remote Tidak Berfungsi	['g/t', 'oli', 'mesin', 'cek', 'remote', 'tidak', 'befungsi']
G/T, ACCU, Solite,1PC, NS40ZL (Internal)	['g/t', 'accu', 'solite', '1pc', 'ns40zl', 'internal']
G/T Oli Filter (Internal)	['g/t', 'oli', 'filter', 'internal']

d. *Stopword Data*

Pada data hasil *tokenize* masih terdapat beberapa kata yang dapat dihapus atau tidak dibutuhkan untuk dianalisis lebih lanjut. Sehingga diperlukan *stop word removal* untuk

lebih mengefisiensi kalimat yang akan dianalisis. Kata yang dihilangkan seperti Internal, Eksternal, dan lain sebagainya. Adapun hasil dari penghapusan *stop word* ditampilkan pada tabel 4.3.

Tabel 4. 3 Proses *Stopwrod Data*

Sebelum *Stopword Data*	**Sesudah *Stopword Data***
G/T Oli Mesin + Oli Filter	['oli', 'mesin', 'oli', 'filter']
Ganti Oli Mesin + Oli Filter (Internal)	['oli', 'mesin', 'oli', 'filter']
G/T,Oli Mesin,SHELL,5.5L (Internal)	['oli', 'filter']
G/T Oli Mesin + Cek Remote Tidak Berfungsi	['oli', 'mesin', 'remote', 'befungs']
G/T, ACCU, Solite,1PC, NS40ZL (Internal)	['accu', ‘solite']
G/T Oli Filter (Internal)	['oli', ‘filter']

e. *Stemming Data*

Pada *preprocessing* sebelumnya, masih terdapat beberapa kalimat yang tidak langsung menuju pada kata dasar sehingga, dilakukan tahapan untuk menghilangkan kata imbuhan yang tidak diperlukan atau dapat dikatakan tahapan ini menjadikan kata – kata menjadi kata dasar tanpa imbuhan. Adapun hasil dari *steeming* ditunjukkan pada tabel 4.4.

Tabel 4. 4 Proses *Steeming Data*

Sebelum *Tokenize Data*	**Sesudah *Tokenize Data***
G/T Oli Mesin + Oli Filter	oli mesin oli filter
Ganti Oli Mesin + Oli Filter (Internal)	oli mesin oli filter
G/T,Oli Mesin,SHELL,5.5L (Internal)	oli mesin
G/T Oli Mesin + Cek Remote Tidak Berfungsi	oli mesin remote fungsi
G/T, ACCU, Solite,1PC, NS40ZL (Internal)	accu solite
G/T Oli Filter (Internal)	oli filter

4.2.2 Eksplorasi Data dan Visualisasi

Tahapan eksplorasi data (*Exploratory Data Analysis* atau EDA) merupakan proses yang penting dalam melakukan pengolahan data sebelum dilakukan pemodelan klasifikasi. Tujuan dari tahapan ini guna memperoleh pemahaman menyeluruh terhadap struktur data, memeriksa kualitas data hasil *preprocessing,* serta mengidentifikasi pola atau informasi penting yang terkandung dalam data teks. Analisis ini juga digunakan mengetahui distribusi kategori label (Jenis Perbaikan) dan frekuensi kata yang sering muncul dalam data.

Sedangkan visualisasi berguna sebagai alat bantu untuk menyajikan hasil eksplorasi secara grafis agar informasi yang ditemukan dapat dengan mudah dianalisis dan dilakukan interpretasi. Hasil eksplorasi ini dapat menjadi dasar untuk menentukan strategi lanjutan pada tahap representasi data dan pemodelan *machine learning.*

Adapun informasi yang didapatkan dari proses eksplorasi dan visualisasi sebagai berikut:

a. Informasi umum *dataset*

Analisis awal dilakukan dengan menampilkan lima baris pertama dari *dataset* dengan menggunakan fungsi *df.head()* dan memeriksa struktur data dengan fungsi *df.info().* Dari hasil tersebut diketahui bahwa data terdiri dari beberapa kolom penting, seperti *Order Desc, Item Text,* Jenis Perbaikan, dan lain sebagainya. Adapun hasil dari eksplorasi ini sebagai berikut:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 236474 entries, 0 to 236473
Data columns (total 4 columns):
 #   Column           Non-Null Count   Dtype 
---  ------           --------------   ----- 
 0   Order Desc       236474 non-null  object
 1   Item Text        236474 non-null  object
 2   Jenis Perbaikan  236474 non-null  object
 3   Cleaned_Text     236474 non-null  object
dtypes: object(4)
```

Gambar 4. 4 Hasil Eksplorasi Informasi Umum *Dataset*

Pada Gambar 4.4 menunjukkan *dataset* yang digunakan terdiri dari 236.474 baris dan 4 kolom, yaitu ”*Order Desc*”, ”*Item Text*”, ”Jenis Perbaikan”, dan ”*Cleaned_Text*”. Seluruh kolom bertipe data *object,* yang menunjukkan bahwa data ini didominasi oleh teks atau *string.* Kolom ”*Order Desc*” dan ”*Item Text*” berisi informasi deskriptif mengenai order layanan dan item atau komponen kendaraan. Kolom ”Jenis Perbaikan” merupakan label klasifikasi yang telah ditentukan

sebelumnya, sedangkan ”*Cleaned_Text*” adalah hasil dari proses *preprocessing* yang digunakan sebagai basis untuk analisis *teks mining* lebih lanjut.

b. Pemeriksaan nilai kosong

Pemeriksaan jumlah nilai kosong pada setiap kolom dilakukan menggunakan fungsi *df.isnull().sum().* Dari hasil pengecekan tersebut tidak ditemukan nilai kosong pada kolom data yang digunakan dalam penelitian, sehingga tidak diperlukan penanganan khusus terhadap *missing values.* Adapun hasil dari peeriksaan ini sebagai berikut:

```
Missing values per column:
Order Desc         0
Item Text          0
Jenis Perbaikan    0
Cleaned_Text       0
dtype: int64
```

Gambar 4. 5 Hasil Eksplorasi *Data Null*

Berdasarkan gambar 4.5 menampilkan hasil pemeriksaan nilai kosong, keempat kolom dalam *dataset* tidak memiliki nilai kosong. Hal ini menunjukkan bahwa *dataset* memiliki kualitas yang cukup baik dari sisi kelengkapan data. Tidak adanya nilai yang hilang mempermudah proses analisis selanjutnya, karena tidak diperlukan proses penanganan khusus untuk mengatasi *missing value.*

c. Jumlah nilai unik

Untuk mengetahui variasi data pada masing – masing kolom, dilakukanlah analisis jumlah nilai dengan fungsi *df.unique().* Hasilnya menunjukkan bahwa kolom Jenis Perbaikan memiliki kategori unik, yaitu Perbaikan Berkala (PB), Perbaikan Berkala Tidak Teratur (PBTT), dan *Adhoc,* sesuai dengan batasan penelitian. Adapun kategori unik pada variabel lainnya yakni sebagai berikut:

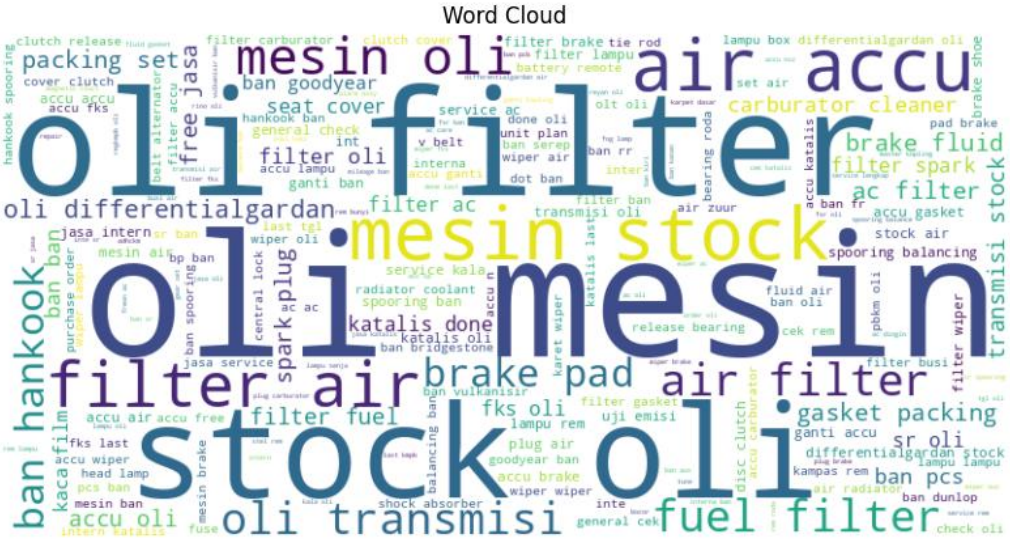
```
Number of unique values per column:
Order Desc         49868
Item Text          70871
Jenis Perbaikan        3
Cleaned_Text       29137
dtype: int64
```

Gambar 4. 6 Hasil Eksplorasi *Unique Values*

Setiap kolom dalam *dataset* memiliki jumlah nilai uni yang berbeda, menunjukkan tingkat keragaman informasi dalam *dataset.* Seperti pada *colomn* ”*Item Text*” mengandung 79.871 nilai unik, yang mencerimkan tingginya variasi komponen atau item kendaraan yang tercatat dalam layanan, begitupun dengan *colomn* lainnya.

d. Visualisasi *word cloud*

Visualisasi *word cloud* dibuat berdasarkan gabungan semua teks dalam kolom *cleaned_text. Word cloud* menampilkan kata – kata yang paling sering muncul dalam dataset, dimana ukuran kata menunjukkan tingkat frekuensi kemunculannya. Kata – kata yang muncul seperti ”oli”, ”mesin”, filter”, dan lain sebagainya, menandakan bahwa kata tersebut sering digunakan oleh *service advisor* untuk mendeskripsikan kerusakan kendaraan. Adapun hasil visualisasi *word cloud* yaitu sebagai berikut



Gambar 4. 7 Visualisasi *Word Cloud*

Pada gambar 4.7 terlihat bahwa kata – kata seperti ”oli”, ”mesin”, dan ”filter” muncul dengan ukuran besar, hal ini menandakan bahwa kata tersebut merupakan bagian yang paling sering menjadi fokus dalam proses perbaikan kendaraan. Kata ”oli mesin” dan ”oli filter” menjadi yang paling dominan, mengindikasikan bahwa penggantian oli merupakan aktivitas perbaikan yang sangat umum dilakukan. Kata lain seperti ”accu”, ”transmisi”, dan ”ban” juga menunjukkan bawa sistem lainnya turut menjadi perhatian dalam proses perbaikan kendaraan.

Secara keseluruhan, *word cloud* ini memberikan gambaran umum mengenai pola dan fokus perbaikan kendaraan yang paling sering dilakukan. Informasi ini dapat dimanfaatkan untuk memahami kebutuhan akan perbaikan kendaraan.

4.2.3 Pembobotan dan Pemisahan Data

Setelah data teks melalui tahapan *preprocessing* dan dilakukan eksplorasi, langkah selanjutnya adalah mengubah data teks tersebut menjadi representasi numerik yang dapat diproses oleh algoritma *machine learning.* Proses ini dikenal sebagai pembobotan teks, yang dalam penelitian ini dilakukan menggunakan metode *Term Frequency – Inverse Document Frequency* (TF-IDF). Setelah pembobotan dilakukan, data kemudian dibagi menjadi dua bagian, yaitu data latih (*training set*) dan data uji (*testing set*), untuk keperluan pelatihan dan evaluasi model. Pada tabel 4.5 menunjukkan proses pembobotan dan pemisahan data.

Tabel 4. 5 *Code* Pembobotan dan Pemisahan Data

Code	**Fungsi**
`from sklearn.feature_extraction.text import TfidfVectorizer` `vectorizer = TfidfVectorizer()` `X = vectorizer.fit_transform(df["Cleaned_Text"])` `y = df["Jenis Perbaikan"]`	*tfidfVectorizer()* dibuat guna mengubah teks dalam kolom *Cleaned_Text* menjadi representasi numerik berbasis bobot TF-IDF. Proses *fit_transform()* digunakan untuk menghitung bobot setiap kata berdasarkan frekuensinya dalam dokumen dan ketidakumumannya dalam seluruh korpus
`from sklearn.model_selection import train_test_split` `X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)`	Tahapan ini membagi data menjadi data latih (*x_train, y_train*) dan data uji (*X_test, y_test*) dengan rasio 80:20 menggunakan *train_test_split()*. Parameter *random_state* = 42 memastikan hasil pembagian konsisten setiap kali dijalankan, sementara *stratify=y* menjaga proporsi distribusi kelas pada data latih dan data uji tetap seimbang sesuai dengan distribusi awal dari target y

Adapun penjelasan dari pembobotan dan pemisahan data yang dilakukan yaitu sebagai berikut:

a. Pembobotan dengan TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) merupakan salah satu metode yang banyak digunakan dalam pengolahan teks untuk mengukur seberapa penting suatu kata dalam sebuah dokumen relatif terhadap kumpulan dokumen lainnya. Metode ini efektif untuk melihat kata – kata yang bersifat unik dan mengurangi pengaruh kata – kata yang terlalu sering muncul dalam seluruh dokumen. Dalam implementasi penelitian ini, pembobotan dilakukan dengan menggunakan fungsi *TfidVectorizer()* dari pustaka *scikit-learn* pada kolom *Cleaned_Text,* yaitu kolom teks yang telah diproses. Hasil dari pembobotan ini berupa matriks berdimensi (n, m), di mana *n* adalah jumlah dokumen dan *m* adalah jumlah fitur (kata unik) setelah *preprocessing.* Setiap nilai dalam matriks mewakili bobot TF-IDF dari suatu kata dalam sebuah dokumen.

Pada penelitian ini, pembobotan TF-IDF berguna untuk merepresentasikan data teks menjadi numerik yang dapat diproses oleh algoritma *machine learning.* Representasi ini menjadi input utama dalam proses klasifikasi jenis perbaikan kendaraan. Dangan TF-IDF, model mampu mengenali kata – kata penting yang menjadi ciri khas masing – masing kategori, sehingga membantu meningkatkan performa model dalam mengklasifikasikan jenis perbaikan kendaraan.

b. Pemisahan data (*Data Splitting*)

Setelah data teks dikonversi ke bentuk numerik menggunakan TF-IDF, langkah selanjutnya adalah melakukan pembagian data menjadi dua bagian yaitu data latih (*training set*) digunakan untuk melatih model dan data uji (*testing set*) digunakan untuk menguji kinerja model.

Pada penelitian ini pembagian data dilakukan menggunakan fungsi *train_test_split()* dari pustaka *scikit-learn,* dengan rasio 80% untuk data latih dan 20% untuk data uji. Pemilihan rasio ini didasarkan pada praktik umum dalam *machine learning,* agar model memiliki cukup data untuk mempelajari serta cukup data yang terpisah untuk dievaluasi secara objektif. Proses ini juga dilakukan dengan parameter *stratify* pada label Jenis Perbaikan untuk memastikan distribusi kelas tetap seimbang antara data latih dan data uji.

4.2.4 Penerapan Algoritma *Machine Learning*

Setelah data teks dikonversi menjadi bilangan numerik dengan menggunakan metode TF-IDF dan dilakukan pemisahan data latih dan data uji, tahapan berikutnya yaitu

membangun dan melatih model klasifikasi dengan menggunakan algoritma *machine learning.* Tujuan dari tahapan ini guna mengembangkan sistem yang mampu mengklasifikasi jenis perbaikan kendaraan, apakah termasuk kedalam kategori Perbaikan Berkala (PB), Perbaikan Berkala Tidak Teratur (PBTT), atau Adhoc.

Dalam penelitian ini, digunakan tiga algoritma klasifikasi berbasis *machine learning* yang umum digunakan dan terbukti efektif dalam pengolahan data berbasis teks, algoritma yang digunakan yakni *Naive Bayes, Logistic Regression,* dan *Random Forest.* Ketiga algoritma ini dipilih karena memiliki karakteristik yang berbeda dalam menangani data teks dan dapat dibandingkan performanya untuk menentukan model terbaik serta cocok digunakan dalam kasus pemeliharaan armada. Pada tabel 4.6 menunjukkan proses penerapan model *machine learning,* di mana setiap model dilatih menggunakan data latih yang telah dibagi sebelumnya untuk mengetahui performa masing – masing model secara obyektif.

Tabel 4. 6 *Code Algoritma Machine Learning*

Code	**Fungsi**
`from sklearn.naive_bayes import MultinomialNB` `model_nb = MultinomialNB()` `model_nb.fit(X_train, y_train)` `y_pred_nb = model_nb.predict(X_test)`	Mengimpor kelas *MultinomialNB,* dilatih dengan *fit()* menggunakan data lagi *x-train* dan *y-train,* lalu digunakan untuk memprediksi data uji *x_test* menggunakan *predict()* dan disimpan dalam *y_pred_nb*
`from sklearn.linear_model import LogisticRegression` `model_lr = LogisticRegression(max_iter=1000, random_state=42)` `model_lr.fit(X_train, y_train)` `y_pred_lr = model_lr.predict(X_test)`	*LogisticRegression* diimpor dan dibuat dengan batas iterasi 1000 serta *seed* random. Model dilatih dengan *fit()* pada data latih, dan prediksi dilakukan pada *x_test* dengan *predict(),* hasilnya disimpan di *y_pred_ir*
`from sklearn.ensemble import RandomForestClassifier`	*RandomForestClassifier* diimpor dan dibuat dengan 100 *tree* dan *seed* tetap.

Code	Fungsi
`model_rf = RandomForestClassifier(n_estimators=100, random_state=42)` `model_rf.fit(X_train, y_train)` `y_pred_rf = model_rf.predict(X_test)`	Model dilatih pada *x_train* dan *y_train* menggunakan *fit(),* lalu prediksi dilakukan pada data uji dan hasilnya disimpan dalam *y_pred_rf*

Adapun penjelasan penerapan ketiga algoritma *Machine Learning* yang digunakan yaitu sebagai berikut:

a. *Naive Bayes*

Algoritma *naive bayes* merupakan salah satu model klasifikasi paling sederhana tetapi cukup efektif, terutama untuk data berbasis teks. Model *naive bayes* ini bekerja berdasarkan prinsip *Teorema Bayes,* yang menghitung probabilitas suatu kelas berdasarkan fitur – fitur yang dimiliki oleh data tersebut. Adapun untuk kata ”*naive*” merujuk pada asumsi dasar bahwa setiap fitur dianggap saling independen terhadap fitur lainnya.

Dalam penelitian ini digunakan varian *Multinomial Naive Bayes,* karena tipe data yang diolah berasal dari pembobotan TF-IDF yang mencerminkan frekuensi relatif pada suatu kata. Di mana algoritma ini cocok digunakan untuk data teks yang telah direprentasikan dalam bentuk vektor.

b. *Logistic Regression*

Algoritma *logistic regression* merupakan model linier yang banyak digunakan dalam proses klasifikasi. Walaupun algoritma ini bernama ”*regression*”, algoritma ini bertugas memetakan input ke dalam probabilitas kelas, lalu mengambil keputusan klasifikasi berdasarkan ambang batas tertentu.

Dalam konteks teks pada penelitian ini, *logistic regression* berkerja cukup baik karena dapat menangani data yang cukup tinggi seperti pada hasil pemboboran TF-IDF, di mana ribuan kata dirubah menjadi fitur. Model *logistic regression* dilatih menggunakan pendekatan *One-vs-Rest* (OvR), di mana model membangun satu klasifikasi untuk setiap kelas terhadap kelas lainnya.

c. *Random Forest*

Algoritma *random forest* merupakan salah satu model *ensemble learning* yang bekerja dengna membangun banyak pohon keputusan (*decision tree*) secara paralel, lalu menggabungkan hasilnya melalui teknik *voting* mayoritas. Setiap *tree* dalam *forest* dilatih dengan *subset* acak dari data dan fitur, sehingga dapat menangkap keragaman pola dalam *dataset.*

Dalam pengolahan teks, *random forest* cukup efektif karena dapat mempelajari pola kompleks yang mungkin tidak dapat ditangkap oleh model linear lainnya. Di mana model ini akan memberikan *feature importance,* yang bisa membantu mengetahui kata mana yang paling berpengaruh terhadap prediksi.

4.3 Evaluasi Model Klasifikasi

Setelah dilakukan pelatihan model menggunakan tiga algoritma klasifikasi yang berbeda, yaitu *naive bayes, logistic regression,* dan *random forest.* Tahapan selanjutnya adalah melakukan evaluasi performa masing – masing model terhadap data uji. Evaluasi ini bertujuan untuk mengetahui sejauh mana model mampu mengklasifikasikan jenis perbaikan kendaraan dengan akurat berdasarkan data teks laporan Surat Perintah Kerja (SPK) yang telah diolah sebelumnya.

Dalam penelitian ini, evaluasi dilakukan menggunakan dua pendekatan utama, yaitu *accuracy* dan *confusion matrix.* Kedua metrik ini dipilih karena mampu memberikan gambaran yang jelas mengenai efektivitas prediksi model terhadap klasifikasi multikelas yang digunakan dalam penelitian, yaitu Perbaikan Berkala (PB), Perbaikan Berkala Tidak Teratur (PBTT), dan *Adhoc.*

Akurasi digunakan untuk menunjukkan persentase prediksi yang benar dari total data uji, sedangkan *confusion matrix* digunakan untuk mengetahui distribusi prediksi benar dan salah pada tiap kelas target.

4.3.1 Evaluasi Model *Naive Bayes*

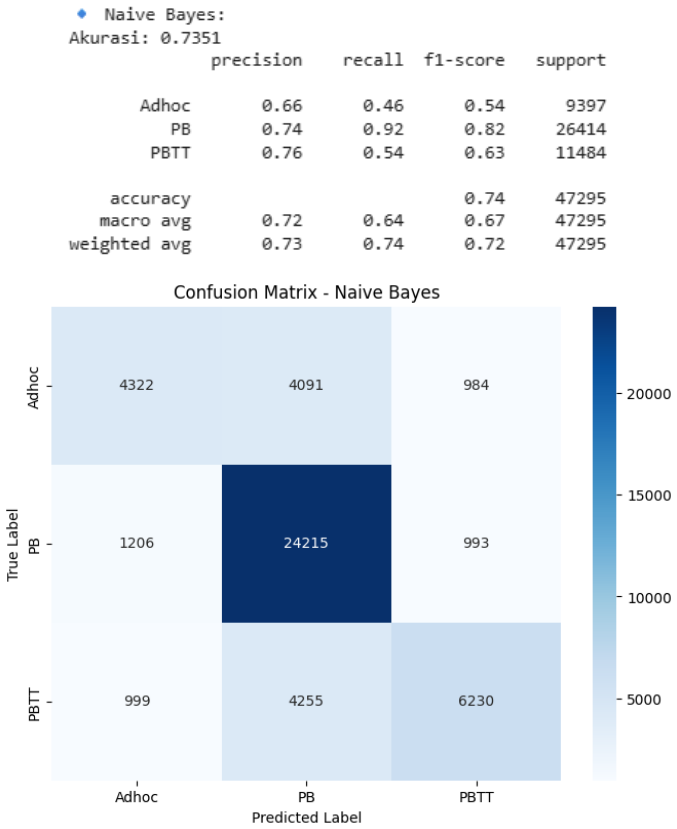
Model pertama yang diuji adalah *Multinomial Naive Bayes,* sebuah algoritma klasifikasi berbasis probabilistik yang umum digunakan dalam pemrosesan bahasa alami (NLP). Model ini dibangun dengan mengasumsikan bahwa setiap fitur dalam kata bersifat independen terhadap fitur lainnya.

Evaluasi model ini dilakukan dengan langkah – langkah berikut:

1. Pelatihan model menggunakan data hasil TF-IDF pada data latih.
2. Prediksi data uji menggunakan model *Naive Bayes* yang telah dilatih.

3. Menghitung akurasi prediksi, yaitu rasio antara jumlah prediksi yang benar terhadap total data uji.
4. Membentuk *confusion matrix* untuk melihat distribusi prediksi per kelas.

Berikut merupakan hasil *accuracy* dan *confusion matrix* dengan menggunakan model *naive bayes.*



Gambar 4. 8 Hasil Evaluasi *Naïve Bayes*

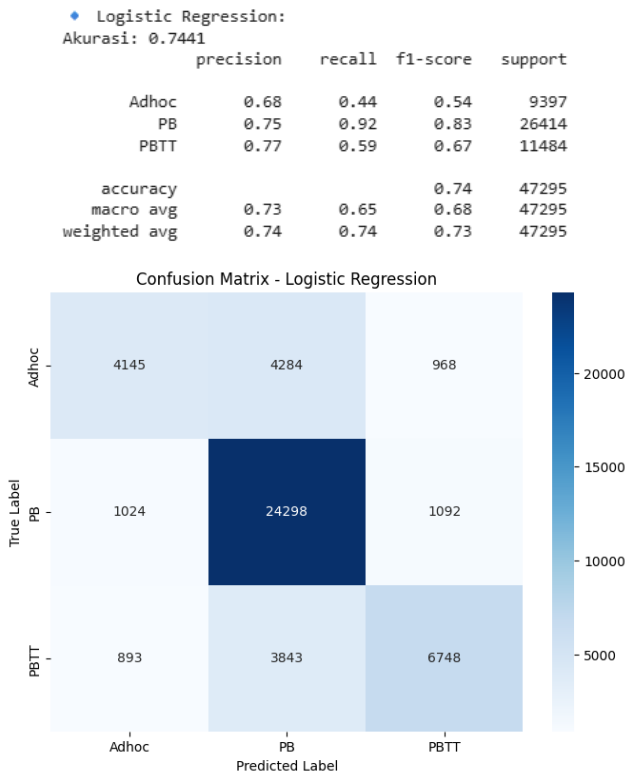
Model *naïve bayes* menunjukkan hasil yang cukup baik dalam melakukan klasifikasi jenis perbaikan kendaraan. Berdasarkan hasil evaluasi, model ini berhasil mencapai akurasi sebesar 0,7351, yang menunjukkan bahwa sekitar 73,51% dari data uji berhasil diklasifikasi dengan benar. Sedangkan hasil evaluasi *confusion matrix* menjelaskan bahwa model ini mampu mengklasifikasikan PB dengan cukup baik dengan prediksi benar sebesar 24.215. Namun, masih terjadi kesalahan klasifikasi pada kelas PBTT dan Adhoc yang cukup signifikan.

4.3.2 Evaluasi Model *Logistic Regression*

Model kedua yang dilakukan evaluasi yaitu *Logistic Regression,* model linier yang dapat digunakan untuk klasifikasi multikelas menggunakan pendekatan *One-vs-Rest*

(OvR). Model ini bekerja dengan membentuk batas keputusan yang memisahkan data antar kelas berdasarkan bobot fitur hasil dari TF-IDF.

Berikut merupakan hasil *accuracy* dan *confusion matrix* dengan menggunakan model *Logistic Regression.*



Gambar 4. 9 Hasil Evaluasi Model *Logistic Regression*

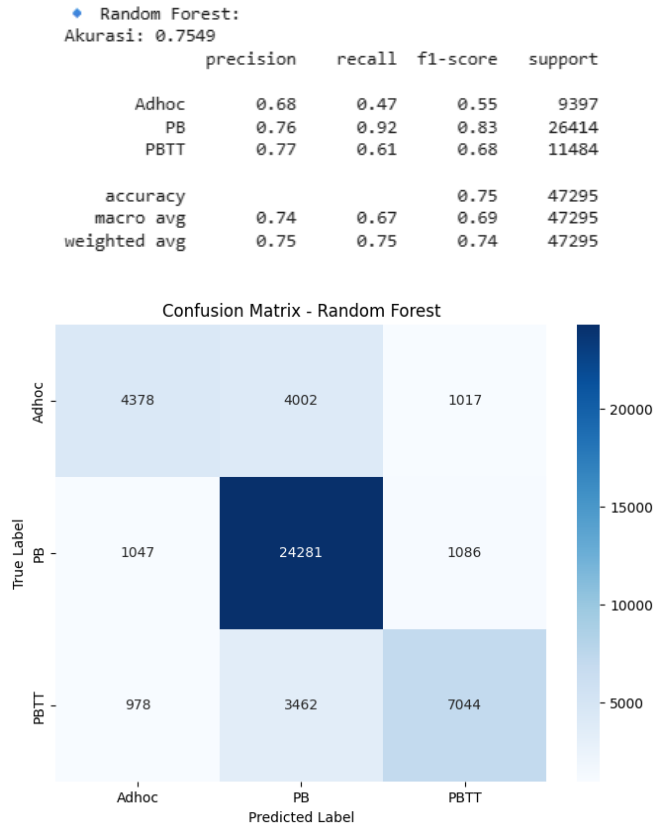
Model *Logistic Regression* menunjukkan performa yang baik dengan akurasi sebesar 0,7441 atau sekitar 74,41% data berhasil diklasifikasikan dengan benar. Sedangkan hasil evaluasi dengan *confusion matrix,* model paling akurat dalam mengklasifikasikan kelas PB, dengan 24,298 prediksi benar. Berdasarkan hasil evaluasi model *logistic regression* memberikan hasil yang lebih baik dari *naive bayes,* terutama dalam hal kemampuan mengenali kelas PB dan PBTT.

4.3.3 Evaluasi Model *Random Forest*

Model ketiga adalah *random forest,* yang merupakan model berbasis *ensemble learning.* Model ini menggabungkan sejumlah pohon keputusan (*decision tree*) untuk menghasilkan prediksi akhir berdasarkan voting mayoritas. Sehingga dengan

karakteristik yang kompleks pada data, *random forest* diharapkan mampu menangani variasi dan korelasi antar kata dalam data.

Berikut merupakan hasil *accuracy* dan *confusion matrix* dengan menggunakan model *random forest.*



Gambar 4. 10 Hasil Evaluasi Model *Random Forest*

Model *Random Forest* memberikan hasil terbaik dibandingkan dua model sebelumnya, dengan akurasi sebesar 0,7549 atau sekitar 75,49% data berhasil diklasifikasikan dengan benar. Sedangkan hasil evaluasi dengan *confusion* menunjukkan bahwa model ini paling akurat dalam mengklasifikasikan kelas PB dengan hasil sebesar 24.281 prediksi benar, serta menunjukkan peningkatan prediksi benar pada kelas PBTT dibandingkan model lain sekitar 7.044.

BAB V
PEMBAHASAN

5.1 Analisis Evaluasi Model

Output model yang dihasilkan dalam penelitian ini berupa klasifikasi laporan perbaikan kendaraan dengan menggunakan *machine learning* sebagai alat klasifikasi. Hasil klasifikasi tersebut mampu mengelompokkan laporan berdasarkan jenis perbaikan atau kategori masalah yang muncul di lapangan. Dengan demikian, laporan yang semula berbentuk teks bebas (*unstructured data*) dapat diubah menjadi informasi yang terstruktur dan lebih mudah dipahami oleh manajemen di kantor pusat.

Fungsi utama dari *output* model ini menyediakan informasi yang akurat, konsisten, dan terstandar. Hal ini menjawab langsung permasalahan ketidaksesuaian informasi yang sebelumnya sering terjadi karena perbedaan format laporan dan inkonsistensi isi laporan. Dengan adanya model klasifikasi ini:

a. Meningkatkan validitas evaluasi, di mana model dapat memfilter kesalahan pencatatan dan mendeteksi pola teks yang relevan dengan jenis perbaikan tertentu.
b. Mempercepat evaluasi, karena laporan otomatis diklasifikasikan, evaluasi kinerja tidak lagi membutuhkan proses manual yang lama.
c. Mendukung pengambilan Keputusan, di mana data hasil klasifikasi langsung digunakan untuk analisis lebih lanjut, misalnya tren kerusakan kendaraan, kualitas perawatan, atau efektivitas kantor cabang atau bengkel.

Evaluasi dilakukan terhadap data uji yang telah dipisahkan sebesar 20% dari total *dataset.* Data yang diolah belum pernah dilihat pada model sebelumnya, sehingga performa yang dihasilkan mencerminkan kemampuan generalisasi model terhadap data baru. Hasil dari pengolahan data menunjukkan bahwa ketiga algoritma memiliki performa yang berbeda satu sama lain meskipun menggunakan representasi teks yang sama. Hal ini menunjukkan bahwa algoritma memegang peran penting dalam menangani fitur – fitur hasil ekstraksi TF-IDF.

5.1.1 Akurasi Model

Pada penelitian ini, evaluasi akurasi terhadap tiga algoritma *machine learning* yakni *Naive Bayes, Logistic Regression,* dan *Random Forest,* telah dilakukan untuk mengklasifikasikan jenis perbaikan kendaraan berdasarkan data teks dari Surat Perintah

Kerja (SPK). Ketiga algoritma tersebut menunjukkan performa yang berbeda dalam hal akurasi dan kemampuan menangani ketidakseimbangan kelas. Berikut merupakan analisis akurasi untuk masing – masing algoritma yang digunakan.

a. *Naive Bayes*

   Model *Naive Bayes* berhasil mencapai akurasi sebesar 0,7351 atau sebesar 73,51% pada penelitian ini. Meskipun akurasi ini tergolong cukup baik untuk klasifikasi teks dengan tiga kelas, model ini menunjukkan kecenderungan sangat kuat untuk kelas PB yang mencapai 0,92, menandakan bahwa model ini sangat baik sering memprediksi PB dengan benar. Namun, dominasi performa pada kelas PB menyebabkan akurasi pada kelas minoritas menjadi lebih rendah. Pada kelas Adhoc, memiliki nilai *recall* hanya mencapai 0,46, yang berarti lebih dari setengah data Adhoc gagal dikenali secara akurat. Sedangkan pada kelas PBTT, meskipun memiliki nilai *precison* tinggi yaitu sebesar 0,76, dengan *recall* sebesar 0,54 hal ini menunjukkan bahwa kelemahan model dalam menangkap variasi teks untuk kelas yang tidak terlalu dominan. Selain itu, jika dilihat dari nilai *macro average f1-score* mendapatkan nilai sebesar 0,67 yang mengindikasikan bahwa performa model secara rata – rata antar kelas masih terbatas. Hal ini disebabkan oleh asumsi dasar model *Naive Bayes* yang menganggap independensi antar fitur, yang seharusnya dalam data teks tersebut, banyak istilah yang saling terkait secara kontekstual. Oleh karena itu, pada penelitian ini, meskipun model *Naive Bayes* bekerja dengan cepat dan efisien secara komputasi, akurasi yang dihasilkan belum cukup optimal untuk digunakan sebagai model utama dalam sistem klasifikasi jenis perbaikan kendaraan.

b. *Logistic Regression*

   Model *Logistic Regression* menunjukkan performa yang lebih baik dibandingkan *Naive Bayes,* dengan akurasi sebesar 0,7441 atau sebesar 74,41%. Model ini menunjukkan distribusi performa yang lebih seimbang antar kelas, dengan nilai *f1-score* yang relatif stabil. Nilai *recall* tertinggi masih terdapat pada kelas PB yaitu sebesar 0,92, sementara kelas PBTT memiliki nilai *recall* sebesar 0,59 dan Adhoc 0,44. Peningkatan terlihat pada kelas PBTT, di mana nilai *f1-score* meningkat menjadi 0,67, hal ini mengindikasikan model ini memiliki kemapuan yang lebih baik dalam mengenali pola pada data teks yang menggambarkan kelas PBTT.

Meskipun nilai *recall* pada kelas Adhoc belum meningkat secara signifikan dibanding algoritma *Naive Bayes,* nilai *precision* pada kelas Adhoc naik menjadi 0,68 yang berarti model ini lebih hati – hati dan cenderung benar saat memprediksi Adhoc. Secara keseluruhan nilai *macro average f1-score* sebesar 0,68 yang menunjukkan bahwa algoritma *Logistic Regression* mampu menjaga performa yang relatif merata antar ketiga kelas. Dengan demikian, algoritma *Logistic Regression* pada penelitian ini dapat dianggap sebagai model yang cocok digunakan sebagai *baseline* sistem klasifikasi, karena memiliki akurasi yang baik dan mampu menangkap hubungan linier antar fitur kata pada data teks Surat Perintah Kerja (SPK).

c. *Random Forest*

Pada model *Random Forest* menunjukkan hasil paling optimal dalam penelitian ini, dengan nilai akurasi tertinggi yaitu 0,7549 atau setara dengan 75,49%. Model ini unggul dalam hampir semua *matrix* evaluasi dan menunjukkan kemampuan paling baik dalam mengenali variasi kata dan struktur kalimat dalam deskripsi Surat Perintah Kerja (SPK). Nilai *f1-score* untuk ketiga kelas juga meningkat, dengan kelas PB meningkat menjadi 0,83, untuk kelas PBTT meningkat menjadi 0,68, dan kelas Adhoc meningkat menjadi 0,55. Selain itu, nilai *macro average f1-score* sebesar 0,69 yang menunjukkan bahwa model ini mampu menjaga keseimbangan performa di antara kelas – kelas yang tidak seimbang secara distribusi data. *Random Forest* bekerja dengan membangun beberapa pohon keputusan (*Decision Tree*) yang dilatih pada *subset* fitur dan data yang berbeda, yang memungkinkan model ini untuk mengenali pola – pola kompleks dan tidak linier dalam data. Kemampuan untuk melakukan voting mayoritas dari berbagai *tree* menjadikan *Random Forest* lebih tahan terhadap *overfitting* dan lebih adaptif terhadap karakteristik teks yang tidak terstruktur dari data SPK. Model ini juga memperlihatkan nilai *recall* pada kelas Adhoc tertinggi sebesar 0,47 dibandingkan dua model lainnya, yang menunjukkan bahwa meskipun data Adhoc tergolong sulit diprediksi, *Random Forest* masih mampu mengenali lebih baik. Oleh karena itu, pada penelitian ini, *Random Forest* dinilai sebagai algoritma paling cocok untuk membangun model klasifikasi jenis perbaikan kendaraan, karena mampu memberikan keseimbangan antara akurasi tinggi dan ketahanan terhadap variasi data yang kompleks.

5.1.2 Interpretasi Confusion Matrix

Confusion matrix merupakan salah satu metode evaluasi performa model klasifikasi dengan memperlihatkan seberapa banyak prediksi yang benar maupun salah terhadap masing – masing kelas. Pada penelitian ini, klasifikasi dilakukan terhadap tiga jenis perbaikan kendaraan yaitu PB, PBTT, dan Adhoc. Analisis *confusion matrix* memberikan wawasan yang lebih rinci terhadap kemampuan masing – masing algoritma dalam mengenali setiap kelas, yang tidak dapat dijelaskan hanya dengan mengguanakn akurasi.

a. *Naïve Bayes*

Pada *confusion matrix* untuk model *Naïve Bayes,* terdapat ketimpangan prediksi antar kelas yang cukup signifikan. Dari total 9.397 data yang benar – benar merupakan kelas Adhoc, hanya 4.322 data yang berhasil diprediksi dengan benar, sementara sisanya salah diklasifikasikan sebagai PB sebanyak 4.4091 dan PBTT sebanyak 984. Hal ini, mengindikasikan bahwa model sulit membedakan antara teks Adhoc dan PB, kemungkinan karena adanya istilah – istilah umum yang digunakan pada kedua kategori.

Untuk kelas PB, dari 26.414 data yang benar, 24.215 data berhasil diprediksi dengan tepat, dengan sebagian kecil salah diklasifikasikan sebagai Adhoc sebesar 1.206 dan PBTT sebesar 993. Hal ini memperkuat temuan bahwa PB adalah kelas yang paling mudah dikenali oleh model *Naive Bayes,* yang juga terdapat dari nilai *recall* yang tinggi yaitu sebesar 0,92. Hal ini disebabkan karena PB memiliki pola dan struktur teks yang relatif konsisten dan berulah.

Sementara pada kelas PBTT, dari 11.484 data aktual, hanya 6.230 yang berhasil diklasifikasikan dengan benar, sedangkan 4.255 diklasifikasikan sebagai PB dan 999 diklasifikasikan sebagai Adhoc. Tingginya angka kesalahan prediksi ke PB menunjukkan bahwa model *Naive Bayes* cenderung bias ke kelas mayoritas, terutama jika fitur kata yang digunakan dalam PBTT serupa dengan PB.

Secara keseluruhan, meskipun model *Naive Bayes* memiliki efisiensi tinggi, *confusion matrix* ini menunjukkan bahwa model kurang mampu dalam melakukan klasifikasi secara seimbang antar kelas, terutama pada kelas minoritas seperti PBTT dan Adhoc.

b. *Logistic Regression*

Pada *confusion matrix* untuk model *Logistic Regression,* terlihat adanya peningkatan distribusi prediksi yang lebih seimbang dibandingkan model *Naïve Bayes.* Untuk kelas Adhoc, dari 9.397 data actual, 4.145 berhasil diklasifikasikan dengan benar, meskipun masih terdapat kesalahan prediksi terhadap kelas PB sebanyak 4.284 dan kelas PBTT sebanyak 968.

Untuk kelas PB, hasil prediksi lebih akurat dari 26.414 data aktual, sebanyak 24.298 berhasil diklasifikasikan secara tepat, hanya sebesar 1.024 termasuk kelas Adhoc dan sebesar 1.092 termasuk kelas PBTT. Hal ini menunjukkan bahwa PB adalah kelas paling dominan dan stabil dikenali oleh model ini.

Sedangkan pada kelas PBTT, dari 11.484 dari data aktual, terdapat 6.748 data berhasil diklasifikasikan dengan benar, dan sisanya tersebar pada kelas PB sebesar 3.843 dan kelas Adhoc sebesar 893. Peningkatan prediksi benar kelas PBTT dibandingkan model *Naive Bayes* menunjukkan bahwa model *Logistic Regression* lebih sensitif terhadap perbedaan pola antar kelas, meskipun belum optimal.

Berdasarkan hasil evaluasi *confusion matrix* ini menunjukkan bahwa model *Logistic Regression* mampu meningkatkan ketepatan prediksi untuk PBTT dan menjaga performa stabil pada kelas PB, namun masih memiliki kelemahan dalam membedakan Adhoc yang memiliki distribusi teks yang tidak terstruktur.

c. *Random Forest*

Confusion Matrix untuk model *Random Forest* menunjukkan hasil terbaik secara keseluruhan di antara algoritma lainnya. Untuk kelas Adhoc, dari 9.397 data aktual, terdapat 4.378 data berhasil diklasifikasikan dengan benar dan sisanya salah diklasifikasikan sebagai kelas PB yaitu sebesar 4.002 dan kelas PBTT sebesar 1.017. Hal ini menunjukkan bahwa *Random Forest* lebih handal dalam mengidentifikasi kategori Adhoc yang sulit diidentifikasi.

Pada kelas PB, dari 26.414 data aktual, terdapat 24.281 diprediksi dengan benar, hanya terdapat 1.047 dan 1.086 salah pada kelas Adhoc dan PBTT. Hal ini menunjukkan bahwa model *Random Forest* tetap mempertahankan performa tinggi pada kelas mayoritas, seperti dua model lainnya.

Perubahan yang signifikan terdapat pada kelas PBTT, dari 11.484 data aktual, sebanyak 7.044 berhasil diprediksi dengan tepat, dan hanya terdapat 3.462 salah diklasifikasikan pada kelas PB dan terdapat 978 diklasifikasikan pada kelas Adhoc.

Ini merupakan prediksi benar tertinggi untuk kelas PBTT di antara model lainnya. Hal ini menunjukkan bahwa model *Random Forest* lebih mampu mengenali pola kompleks dalam kombinasi kata – kata pada kelas PBTT.

Berdasarkan *confusion matrix* ini dapat disimpulkan bahwa model *Random Forest* menunjukkan performa klasifikasi terbaik dalam hal ketepatan dan keseimbangan prediksi antar kelas. Model ini mampu menangani variasi fitur yang tinggi dari hasil TF-IDF serta menangkap hubungan kompleks antar kata yang tidak mampu ditangani oleh model linear dan probabilistik.

5.2 Perbandingan Hasil Prediksi Model

Pada bagian ini dilakukan pembahasan mendalam mengenai perbandingan hasil prediksi ketiga algoritma yang telah dikembangkan, terhadap data testing pada klasifikasi jenis perbaikan kendaraan. Analisis dilakukan dengan cara mengukur jumlah prediksi benar serta tingkat akurasi pada data uji secara bertahap, mulai dari 10% hingga 100% dari keseluruhan data uji.

Tujuan evaluasi bertahap ini adalah untuk melihat konsistensi performa model pada *subset* data dengan ukuran yang berbeda, serta untuk membandingkan keunggulan relatif ketiga algoritma dalam menangani variasi ukuran data uji.

Berikut perbandingan hasil prediksi ketiga model tersebut:

a. *Naive Bayes*

Tabel 5. 1 Hasil Prediksi Model *Naive Bayes*

Persentase Data Uji	Jumlah Sampel	Prediksi Benar
10%	4729	3499
20%	9459	6933
30%	14188	10408
40%	18918	13881
50%	23647	17315
60%	28377	20810
70%	33106	24347
80%	37836	27781
90%	42565	31246
100%	47295	34767
Akurasi		0.7351

Berdasarkan Tabel 5.1 hasil evaluasi *Naive Bayes* menunjukkan akurasi keseluruhan sebesar 0,7351. Pada *subset* 10% data uji dengan sampel sebanyak 4729, model berhasil melakukan 3499 prediksi benar, menghasilkan akurasi sekitar 73 – 74%. Jumlah prediksi benar meningkat proposional dengan bertambahnya data uji, mencapai 34,767 prediksi benar pada 100% data uji dengan sampel sebesar 47,295.

Naive Bayes menhasilkan akurasi yang cukup stabil pada rentang 73% menunjukkan konsistensi performa meskipun ukuran *subset* data uji meningkat. Model ini memanfaatkan asumsi independensi antar fitur, yang membuatnya sederhana dan cepat, namun kurang mampu menangkap kompleks antar kata pada data teks.

Meskipun prediksi benar bertambah sesuai ukuran data uji, peningkatan tersebut linier dengan jumlah sampel tanpa lonjakan performa signifikan. Hal ini menandakan batas kemampuan model *Naive Bayes* pada *dataset* ini, di mana asumsi yang digunakan tidak selalu sesuai dengan distribusi data sebenarnya.

b. *Logistic Regression*

Tabel 5. 2 Hasil Prediksi Model *Logistic Regression*

Persentase Data Uji	**Jumlah Sampel**	**Prediksi Benar**
10%	4729	3523
20%	9459	7000
30%	14188	10539
40%	18918	14065
50%	23647	17554
60%	28377	21101
70%	33106	24639
80%	37836	28103
90%	42565	31629
100%	47295	35191
Akurasi		0.7441

Pada Tabel 5.2 hasil evaluasi untuk *Logistic Regression* menunjukkan akurasi 0,7441, sedikit lebih tinggi daripada *Naive Bayes.* Pada *subset* 10% data uji, model

melakukan 3523 prediksi benar dari 4729 sampel. Angka prediksi benar terus meningkat hingga mencapai 35,191 prediksi benar pada 100% data uji.

Logistic Regression memberikan performa yang lebih baik dibandingkan *Naive Bayes.* Hal ini dapat dijelaskan oleh kemampuannya dalam memodelkan hubungan linier antar fitur TF-IDF secara lebih fleksibel, tanpa asumsi independensi ketat.

Akurasi yang stabil pada 74% menunjukkan model ini mampu mempertahankan kualitas prediksi meskipun *subset* data uji bertambah besar. Ini menjadi keunggulan penting pada aplikasi klasifikasi teks berskala besar, di mana model diharapkan tetap konsisten saat konsisten saat dihadapkan pada data baru dalam jumalh besar.

c. *Random Forest*

Tabel 5. 3 Hasil Prediksi Model *Random Forest*

Persentase Data Uji	Jumlah Sampel	Prediksi Benar
10%	4729	3550
20%	9459	7069
30%	14188	10668
40%	18918	14255
50%	23647	17828
60%	28377	21409
70%	33106	25007
80%	37836	28529
90%	42565	32108
100%	47295	35703
Akurasi		0.7549

Hasil pada Tabel 5.3 menunjukkan bahwa *Random Forest* menghasilkan akurasi keseluruhan tertinggi yaitu 0,7549. Pada 10% data uji, model mencatat 3550 prediksi benar dari 4729 sampel. Jumlah prediksi benar meningkat konsisten hingga mencapai 35,703 pada 100% data uji.

Random Forest menunjukkan performa terbaik di antara ketiga model. Dengan pendekatan *ensemble* dari banyak pohon keputusan (*Decision Tree*), model ini mampu menangkap interaksi non-linier antar fitur TF-IDF yang kompleks.

Akurasi stabil dikisaran 75% pada semua *subset* data uji menegaskan kemampuannya untuk menggeneralisasi dengan baik pada data yang lebih besar. Hal ini membuat *Random Forest* unggul untuk klasifikasi teks dengan fitur TF-IDF, terutama dalam konteks kebutuhan prediksi yang akurat.

5.3 Perbandingan Kinerja Antar Model

Berikut adalah tabel perbandingan akurasi ketiga model sesuai dengan hasil dari penelitian ini:

Tabel 5. 4 Perbandingan Akurasi Model Klasifikasi

Model	Akurasi
Naive Bayes	0,7351
Logistic Regression	0,7441
Random Forest	0,7549

Berdasarkan Tabel 5.4 terlihat bahwa model *Random Forest* mencatatkan akurasi tertinggi sebesar 0,7549 atau sebesar 75,49%, mengungguli model *Logistic Regression* sebesar 0,7441 atau sebesar 74,41% dan model *Naive Bayes* sebesar 0,7351 atau sebesar 73,51%. Selisih akurasi relatif kecil namun konsistensi ini menjadi indikator bahwa model *Random Forest* memiliki kemampuan klasifikasi yang lebih baik dalam mengenali pola teks pada SPK dibandingkan dengan dua model lainnya.

Model *Naive Bayes,* meskipun sederhana dan efisien, menghasilkan akurasi paling rendah. Hal ini dapat disebabkan oleh asumsi dasar model *Naive Bayes* yang menganggap semua kata independen satu sama lain, padahal dalam data teks, istilah – istilah sering kali muncul dalam konteks tertentu yang saling berkaitan. Ketidakmampuan model ini dalam menangkap hubungan semantik antar kata menyebabkan penurunan akurasi, terutama pada jenis perbaikan yang memiliki deskripsi lebih bebas seperti PBTT atau Adhoc.

Model *Logistic Regression* memberikan peningkatan akurasi dibandingkan model *Naive Bayes.* Model ini mampu mengenali relasi linier atau fitur hasil transformasi TF-IDF, namun masih memiliki keterbatasan dalam menangkap pola kompleks. Meskipun demikian, model *Logistic Regression* tetap menjadi pilihan yang baik jika dibutuhkan model yang lebih mudah dan efisien dalam hal pelatihan.

Sedangkan model *Random Forest* menjadi model terbaik dalam penelitian ini dari segi akurasi. Hal ini disebabkan oleh pendekatan *ensemble learning* yang dimilikinya, di mana model membangun pohon keputusan (*Decision Tree*) yang berbeda. Teknik ini membuat model

Random Forest lebih baik terhadap *overfitting,* lebih stabil dalam menghadapi data yang tidak seimbang, dan mampu mengenali pola – pola teks yang tidak linier atau kompleks.

5.4 Implementasi Model Machine Learning

Pada bagian ini menjelaskan implementasi model *machine learning* yang telah dibangun untuk memproses data baru. Tujuan implementasi ini adalah untuk menunjukkan kemampuan model dalam mengklasifikasi deskripsi kerusakan kendaraan ke dalam kategori jenis perbaikan kendaraan, yakni Perbaikan Berkala (PB), Perbaikan Berkala Tidak Teratur (PBTT), dan Adhoc. Tahapan implementasi meliputi pemrosesan data baru, prediksi menggunakan model yang telah dilatih, serta penyajian hasil prediksi.

5.4.1 Pemrosesan Data Baru

Pada tahap ini dilakukan pemrosesan awal terhadap data baru yang akan digunakan sebagai masukan bagi model machine learning. Data baru diperoleh dari dokumen Surat Perintah Kerja (SPK) yang berisi informasi terkait aktivitas perawatan dan perbaikan kendaraan. Dataset ini disusun dalam bentuk tabel dengan beberapa kolom, seperti Order Desc yang memuat detail pesanan atau identitas kendaraan, serta Item Text yang berisi uraian pekerjaan atau komponen yang dikerjakan.

Tabel 5. 5 *Dataset* Baru

No.	***Order Desc***	***Item Text***
1.	PB,KT-1912-KS,KM.79470,PB.80	G/T,FILTER AC,DENSO,1PC (Internal)
2.	PB,KT-8041-YJ,KM.104700,PB.105,G/T LINK	G/T,FILTER UDARA,NON GENUINE,1PC (Internal)
...	....	.....
149.	PA,KT-8378-YW,KM.77638,GT LAMPU KOTA	G/T,LAMPU KOTA,UNIVERSAL,1PC (Internal)
150.	PBTT,KT-8770-YW,KM.84946,G/T MAGNET CLUT	G/T,FILTER AC,DENSO,1PC (Internal)

Pada Tabel 5.5 merupakan data baru yang akan digunakan sebagai *input* untuk proses prediksi klasifikasi jenis perbaikan kendaraan. Data ini terdiri dari 150 baris yang

merepresentasikan berbagai jenis aktivitas perawatan atau perbaikan kendaraan. Data baru akan dimuat ke dalam model melalui beberapa langkah berikut.

Tabel 5. 6 *Code* Pemrosesan Data Baru

Code	**Fungsi**
`new_data = pd.read_excel("Data Baru.xlsx")` `print(f"Total data: {len(new_data)}")`	Memuat file *excel* berisi data baru ke dalam bentuk *Data Frame,* sekaligus menampilkan jumlah baris data yang berhasil dimuat
`new_data["Cleaned_Text"] = new_data["Item Text"].apply(preprocess_text)`	Melakukan *Preprocessing* pada kolom *Item Text* untuk membersikan data baru, sehingga menghasilkan kolom *Cleaned_Text* yang siap digunakan.
`new_texts_vector = vectorizer.transform(new_data["Cleaned_Text"])`	Mengubah teks bersih pada kolom *Cleaned_Text* menjadi bentuk vektor numerik menggunakan TF-IDF yang telah dilatih sebelumnya

Melalui tahapan ini, data baru yang semula berupa deskripsi teks mentah disiapkan dalam bentuk terstruktur dan bersih. Proses dimulai dengan pemanggilan *dataset* dari file *excel*, dilanjutkan dengan *preprocessing* untuk membersihkan teks, dan diakhiri dengan transfromasi ke bentuk numerik menggunakan TF-IDF. Tahapan ini memastikan bahwa data baru memiliki format dan representasi fitur yang konsisten dengan data pelatihan sebelumnya, sehingga siap untuk dianalisis lebih lanjut oleh model *machine learning* pada tahap prediksi.

5.4.2 Prediksi Model Yang Terpilih

Setelah data dalam bentuk vektor TF-IDF siap, dilakukan proses klasifikasi dengan menggunakan model *machine learning*. Dalam penelitian ini, prediksi data baru dilakukan dengan menggunakan model *Random Forest* yang telah dilatih sebelumnya. *Random Forest* dipilih karena pada tahapan evaluasi sebelumnya menunjukkan performa yang unggul dibandingkan dengan algoritma lainnya. Prediksi dilakukan menggunakan perintah *predict().*

Tabel 5. 7 *Code* Prediksi Model

Code	**Fungsi**
`predicted_classes = models['Random Forest'].predict(new_texts_vector)`	Menggunakan model *Random Forest* yang telah dilatih sebelumnya untuk memprediksi kelas atau kategori jenis perbaikan berdasarkan data baru

Dengan perintah tersebut, model *Random Forest* memproses data baru dalam bentuk numerik hasil transformasi TF-IDF untuk menghasilkan prediksi kategori perbaikan kendaraan. Tahap ini merupakan inti dari penerapan model *machine learning,* di mana sistem otomatis menentukan kelas yang sesuai untuk setiap deskripsi kerusakan kendaraan berdasarkan pola yang telah dipelajari.

5.4.3 Penyajian Hasil Prediksi

Setelah proses prediksi selesai dilakukan, langkah berikutnya adalah menyusun hasil prediksi ke dalam bentuk tabel yang mudah dibaca. Tahap ini dilakukan dengan membuat *DataFrame* baru yang berisi informasi asli dari data masukan beserta label hasil prediksi dari model *Random Forest*. Proses ini juga mencakup menampilkan hasil dalam bentuk tabel agar memudahkan pembacaan dan interpretasi oleh pengguna.

Tabel 5. 8 *Code* Penyajian Hasil Prediksi

Code	**Fungsi**
`result_df = pd.DataFrame({` `"Order Desc" : new_data["Order Desc"],` `"Item Text": new_data["Item Text"],` `"Random Forest Prediction": predicted_classes` `})`	Membuat *DataFrame* baru yang menyatukan kolom *Order Desc*, *Item Text*, dan hasil prediksi *Random Forest Prediction*, sehingga hasil klasifikasi dapat dilihat berdampingan dengan data asli.
`print(" Hasil Prediksi Data Baru:")` `display(result_df)`	Menampilkan isi *DataFrame* hasil prediksi

Dengan langkah ini, hasil prediksi disajikan secara terstruktur sehingga pengguna dapat dengan mudah membandingkan deskripsi kerusakan asli dengan

kategori jenis perbaikan yang dihasilkan oleh model. Penyajian hasil dalam bentuk tabel juga membantu proses evaluasi dan penerapan model *machine learning* pada data.

Tabel 5. 9 Data Hasil Prediksi

No	*Order Desc*	*Item Text*	*Prediction*
1.	PB,KT-1912-KS,KM.79470,PB.80	G/T,FILTER AC,DENSO,1PC (Internal)	PB
2.	PB,KT-8041-YJ,KM.104700,PB.105,G/T LINK	G/T,FILTER UDARA,NON GENUINE,1PC (Internal)	PB
...	....	.....	
149.	PA,KT-8378-YW,KM.77638,GT LAMPU KOTA	G/T,LAMPU KOTA,UNIVERSAL,1PC (Internal)	Adhoc
150.	PBTT,KT-8770-YW,KM.84946,G/T MAGNET CLUT	G/T,FILTER AC,DENSO,1PC (Internal)	PB

Tabel 5.9 menampilkan hasil akhir dari implementasi model *machine learning* Pada data baru yang sebelumnya telah melalui proses *preprocessing* dan transformasi ke bentuk numerik menggunakan TF-IDF. Kolom *Prediction* menunjukkan hasil klasifikasi otomatis yang dihasilkan oleh model *Random Forest.* Label prediksi terdiri dari kategori PB (Perbaikan Berkala), PBTT (Perbaikan Berkala Tidak Teratur), dan Adhoc, sesuai dengan jenis perbaikan kendaraan yang telah didefinisikan sebelumnya pada tahap pelatihan model.

Hasil implementasi model *machine learning* pada penelitian ini tidak hanya menunjukkan perbandingan akurasi antar algoritma, tetapi juga memberikan gambaran mengenai potensi penerapannya dalam sistem pencatatan perbaikan kendaraan di perusahaan. Saat ini, keterbatasan sistem yang hanya mampu mencatat satu kategori perbaikan per-laporan menjadi salah satu hambatan utama dalam memastikan perawatan kendaraan yang menyeluruh. Dengan pendekatan klasifikasi ini, perusahaan dapat mengidentifikasi lebih dari satu jenis perbaikan secara otomatis dari deskripsi kerusakan. Hal ini penting untuk menghindari kerusakan yang tidak terdeteksi dan

mengurangi risiko perawatan tidak lengkap. Penerapan model ini dapat mendukung proses pengambilan keputusan perawatan yagn lebih akurat dan efisien, serta meningkatkan kualitas layanan kepada pelanggan.

BAB VI
PENUTUP

6.1 Kesimpulan

Dari hasil pengolahan data dan analisis yang dilakukan terdapat beberapa kesimpulan dari penelitian ini sebagai berikut:

1. Berdasarkan hasil penelitian, dapat disimpulkan bahwa ketidaksesuaian informasi antara data lapangan dengan laporan yang diterima memiliki pengaruh yang cukup signifikan terhadap keakuratan evaluasi kinerja perusahaan. Infromasi yang tidak konsisten dan tidak terstruktur menyebabkan data yang diterima tidak dapat merepresentasikan kondisi sebenarnya dilapangan. Perbedaan format pencatatan di tiap unit operasional menyebabkan data sulit diintegrasikan sehingga meninbulkan inkonsistensi dalam evaluasi, serta terkadang terjadi kesalahan input atau distorsi informasi dari pihak lapangan mengurangi validitas dara yang diterima, sehingga hasil evaluasi berpotensi bias dan tidak objektif.

   Akibat dari permasalahan tersebut, evaluasi kinerja di kantor pusat menjadi kurang akurat dan tidak sepenuhnya dapat dijadikan dasar dalam pengambilan keputusan strategis. Hal ini berdampak pada terhambatnya efektivitas manajemen dalam menyusun kebijakan, meningkatkan risiko kesalahan dalam alokasi sumber daya, serta menurunnya efisiensi operasional perusahaan. Dengan demikian, ketidaksesuaian informasi terbukti menjadi salah satu faktor krusial yang menurunkan akurasi evaluasi kinerja, sehingga perlu adanya solusi berupa sistem pengolahan data yang lebih terstandar, terintegrasi, dan mampu meminimalkan distorsi informasi dari lapangan.
2. Penggunaan TF-IDF (*Term Frequency-Inverse Document Frequency*) terbukti efektif dalam merepresentasikan data teks dari laporan Surat Perintah Kerja (SPK) menjadi fitur numerik yang dapat diproses oleh algoritma *machine learning.* Dengan pembobotan TF-IDF, kata – kata yang relevan terhadap jenis perbaikan kendaraan mendapatkan bobot yang sesuai konteks, membantu model mempelajari pola yang membedakan kategori antar *class.*

   model *machine learning* menghasilkan akurasi yang cukup baik. Berdasarkan hasil evaluasi menggunakan *matrix* akurasi diperoleh hasil sebagai berikut:

a. *Naive Bayes* menunjukkan akurasi sebesar 0,7351. Model ini efektif dalam mengenali kelas mayoritas (PB) dengan nilai *recall* sebesar 0,92, tetapi memiliki kelemahan dalam mengenali kelas minoritas seperti PBTT dan Adhoc. Model ini terlalu bergantung pada asumsi independensi fitur, sehingga performanya menurun pada data teks yang kompleks.
b. *Logistic Regressionn* memiliki nilai akurasi sebesar 0,7441. Model ini memberikan peningkatan performa dibandingkan *Naive Bayes,* terutama dalam klasifikasi kelas PBTT dengan nilai *f1-score* meningkat menjadi 0,67. *Logistic Regression* dapat menangkap hubungan linier antara fitur dan label, namun masih memiliki keterbatasan keterbatasan dalam menangkap hubungan non-linier yang sering muncul dalam teks.
c. *Random Forest* memberikan hasil terbaik dengan akurasi sebesar 0,7549. Model ini mampu menangani kompleksitas tinggi dari data SPK, mengenali pola tidak linier, serta menghasilkan distribusi prediksi yang paling seimbang. Kemampuannya dalam menggabungkan banyak pohon keputusan (*Decission Tree*) menjadikan model ini lebih tahan terhadap *overfitting* dan lebih adaptif terhadap data yang tidak seimbang.

Dengan demikian, dapat disimpulkan bahwa dari ketiga algoritma yang diuji, *Random Forest* merupakan model *Machine Learning* yang paling cocok untuk klasifikasi jenis perbaikan pada penelitian ini. Model ini tidak hanya unggul secara metrik evaluasi, tetapi juga menunjukkan kestabilan dan keandalan dalam mengenali seluruh ketegori jenis perbaikan, menjadikan solusi yang layak untuk diimplementasikan dalam sistem klasifikasi jenis perbaikan kendaraan.

6.2 Saran

Berdasarkan hasil penelitian serta kesimpulan yang diperoleh, penulis menyampaikan beberapa saran yang ditujukan bagi pihak perusahaan maupun untuk penelitian selanjutnya.

1. PT Adi Sarana Armada Tbk dapat mengimplementasi model yang telah dilakukan uji untuk mengklasifikasi jenis perbaikan kendaraan, sebagai bahan evaluasi maupun pengawasan armada serta sebagai dasar pengambilan keputusan.
2. Penelitian lanjutan dapat melakukan perbandingan antara metode klasifikasi yang digunakan saat ini dengan pendekatan lain seperti *deep learning* agar diperoleh hasil yang lebih optimal.

3. Tahapan *Preprocessing* seperti *stopword removal, steeming* dan lainnya sebaiknya disesuaikan dengan kebutuhan dan konteks masing – masing data, agar hasil klasifikasi menjadi lebih akurat dan relevan.

DAFTAR PUSTAKA

Aditya, B. R., No, J. T., Buah, T., & Bandung, B. (2015). Penggunaan Web Crawler Untuk Menghimpun Tweets dengan Metode Pre-Processing Text Mining. *Jurnal Infotel*, *7*(2). http://situs.com

Afida, D., Udayanti, E. D., & Kartikadharma, E. (2021). Aplikasi Text Mining untuk Klasterisasi Aduan Masyarakat Kota Semarang Menggunakan Algoritma K-means. *TRANSFORMTIKA*, *18*(2), 215–224.

Agusvina, N., & Santoso, N. (2018). *Pengelompokan Artikel Berbahasa Indonesia Dengan Menggunakan Reduksi Fitur Information Gain Thresholding Dan K-Means* (Vol. 2, Issue 10). http://j-ptiik.ub.ac.id

Alessa, A., & Faezipour, M. (2019). Preliminary flu outbreak prediction using twitter posts classification and linear regression with historical centers for disease control and prevention reports: Prediction framework study. *JMIR Public Health and Surveillance*, *5*(2). https://doi.org/10.2196/12383

ALG Global. (2024). *How can policymakers encourage the use of public transport in urban environments?* https://alg-global.com/blog/land/how-can-policymakers-encourage-use-public-transport-urban-environments

Arifin, N., Enri, U., & Sulistiyowati, N. (n.d.). *STRING (Satuan Tulisan Riset dan Inovasi Teknologi) PENERAPAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) DENGAN TF-IDF N-GRAM UNTUK TEXT CLASSIFICATION*.

Arsad, A., Hamid, M., & Santosa, M. (2024). PENERAPAN TEKS MINING DAN COSINE SIMILARITY UNTUK MENENTUKAN KESAMAAN DOKUMEN SKRIPSI. In *IJIS Indonesian Journal on Information System*.

Arteaga, C., Paz, A., & Park, J. W. (2020). Injury severity on traffic crashes: A text mining with an interpretable machine-learning approach. *Safety Science*, *132*. https://doi.org/10.1016/j.ssci.2020.104988

Chamira, S. (2022). Implementasi Metode Text Mining Frequency-Invers Document Frequency (Tf-Idf) Untuk Monitoring Diskusi Online. In *Journal of Informatics, Electrical and Electronics Engineering* (Vol. 1, Issue 3). https://djournals.com/jieeeJIEEE,

Chowdhury, S., & Alzarrad, A. (2023). Applications of Text Mining in the Transportation Infrastructure Sector: A Review. In *Information (Switzerland)* (Vol. 14, Issue 4). MDPI. https://doi.org/10.3390/info14040201

Dokter Mobil. (2025, May 17). *7 masalah umum gangguan sistem kelistrikan mobil*. https://www.doktermobil.com/sistem-kelistrikan-mobil-2/

Farhan, Triase, & Harahap, A. M. (2023a). Penggunaan Algoritma Naive Bayes Dalam Text Mining Untuk Klasifikasi Pasal UU ITE. *Juli*, 314–322. https://ojs.trigunadharma.ac.id/index.php/jsk/index

Farhan, Triase, & Harahap, A. M. (2023b). Penggunaan Algoritma Naive Bayes Dalam Text Mining Untuk Klasifikasi Pasal UU ITE. *Juli*, 314–322. https://ojs.trigunadharma.ac.id/index.php/jsk/index

Finda, S. M., & Utomo, D. W. (2024). Klasifikasi Stunting Balita menggunakan Metode Ensemble Learning dan Random Forest. *Jl. Imam Bonjol No*, *15*(02). https://doi.org/10.35970/infotekmesin.v15i2.2326

Goldberg, D. E., & Holland, J. H. (1988). *Genetic Algorithms and Machine Learning*.

Gupta, S., Saluja, K., Goyal, A., Vajpayee, A., & Tiwari, V. (2022). Comparing the performance of machine learning algorithms using estimated accuracy. *Measurement: Sensors*, *24*. https://doi.org/10.1016/j.measen.2022.100432

Hakim, I., Asdi, A., Dicky, M., Lubis, S., Harahap, M. N., Ridwan, L., & Bara, B. (2025). *Klasifikasi Kualitas Produk Mesin Pertanian Berdasarkan Evaluasi Kinerja Algoritma Random Forest* (Vol. 6, Issue 1).

Hidayah, N., & Dodiman. (2024). Implementasi Algoritma Multinomial Naïve Bayes, TF-IDF dan Confusion Matrix dalam Pengklasifikasian Saran Monitoring dan Evaluasi Mahasiswa Terhadap Dosen Teknik Informatika Universitas Dayanu Ikhsanuddin. *Jurnal Akademik Pendidikan Matematika*, *10*(1). https://doi.org/10.55340/japm.v10i1.1491

Khodadadi, A., Ghandiparsi, S., & Chuah, C.-N. (2022). A Natural Language Processing and deep learning based model for automated vehicle diagnostics using free-text customer service reports. *Machine Learning with Applications*, *8*, 100424. https://doi.org/10.1016/j.mlwa.2022.100424

Kordbagheri, A., Kordbagheri, M., Tayim, N., Fakhrou, A., & Davoudi, M. (2025). Using advanced machine learning algorithms to predict academic major completion: A cross-sectional study. *Computers in Biology and Medicine*, *184*. https://doi.org/10.1016/j.compbiomed.2024.109372

Li, D. (2024). An interactive teaching evaluation system for preschool education in universities based on machine learning algorithm. *Computers in Human Behavior*, *157*. https://doi.org/10.1016/j.chb.2024.108211

Li, F. (2020). In-Vehicle Speech Text Classification based on Multiple Machine Learning Algorithms. *Machine Learning Theory and Practice*, *1*(1). https://doi.org/10.38007/ml.2020.010102

Li, J., & Wu, C. (2023). Deep Learning and Text Mining: Classifying and Extracting Key Information from Construction Accident Narratives. *Applied Sciences*, *13*(19), 10599. https://doi.org/10.3390/app131910599

Lowin, M. (2024). A Text-Based Predictive Maintenance Approach for Facility Management Requests Utilizing Association Rule Mining and Large Language Models. *Machine Learning and Knowledge Extraction*, *6*(1), 233–258. https://doi.org/10.3390/make6010013

LPS Konstruksi. (2024). *Pengadaan jasa sewa kendaraan: Solusi tepat untuk efisiensi bisnis Anda.* https://lspkonstruksi.com/blog/pengadaan-jasa-sewa-kendaraan-solusi-tepat-untuk-efisiensi-bisnis-anda

Mayasari, L., & Indarti, D. (2022). KLASIFIKASI TOPIK TWEET MENGENAI COVID MENGGUNAKAN METODE MULTINOMIAL NAÏVE BAYES DENGAN PEMBOBOTAN TF-IDF. *Jurnal Ilmiah Informatika Komputer*, *27*(1), 43–53. https://doi.org/10.35760/ik.2022.v27i1.6184

Medashe, M., Oluwaseyi, M. M., & Sunday, A. M. (2020). *Specifications and Analysis of Digitized Diagnostics of Automobiles: A Case Study of on Board Diagnostic (OBD II)*. https://doi.org/10.13140/RG.2.2.17131.49442

Mena, A. F. L., Berrezueta, M. F. G., Alexandra, J. M. D., & Juan, P. P. A. (2024). Vehicle preventive maintenance: a comprehensive analysis of its impact on society, economic, and environmental factors in General Villamil Playas City. *South Florida Journal of Development*, *5*(1), 65–75. https://doi.org/10.46932/sfjdv5n1-006

Nasution, M. H., & Sofyan, Muh. I. (2020). *Penerapan Metode Decision TreeUntuk Memprediksi Prestasi Siswa Kelas XIIDilihat dari Nilai Akhir Semester di SMK Negeri 1 SelongTahun Pelajaran 2017/2018*. https://e-journal.hamzanwadi.ac.id/index.php/infotek/article/view/1825/pdf_28

Nie, Y., Xian, C., Luo, J., Zhang, J., Wang, Y., & Li, C. (2025). Bagging machine learning algorithms for rapid identification, classification, evaluation and upscaling in unconventional reservoir. *Geoenergy Science and Engineering*.

Pratama, E. E., & Lestika Atmi, R. (2020). *A Text Mining Implementation Based on Twitter Data to Analyse Information Regarding Corona Virus in Indonesia*. https://ejournal.upi.edu/index.php/JCS

Putri, E. K., & Setiadi, T. (2014). *PENERAPAN TEXT MINING PADA SISTEM KLASIFIKASI EMAIL SPAM MENGGUNAKAN NAIVE BAYES*.

Ren, W., Zhang, H., & Chen, M. (2022). A Method of Domain Dictionary Construction for Electric Vehicles Disassembly. *Entropy*, *24*(3). https://doi.org/10.3390/e24030363

Riandari, F., & Simangunsong, A. (2019). PENERAPAN ALGORITMA C4.5 UNTUK MENGUKUR TINGKAT KEPUASAN MAHASISWA. *Terakreditasi DIKTI*, *3*(2), 1–7.

Ridwansyah, T. (2022). KLIK: Kajian Ilmiah Informatika dan Komputer Implementasi Text Mining Terhadap Analisis Sentimen Masyarakat Dunia Di Twitter Terhadap Kota Medan Menggunakan K-Fold Cross Validation Dan Naïve Bayes Classifier. *Media Online*, *2*(5), 178–185. https://djournals.com/klik

Safii, M. (2018). *IMPLEMENTASI DATA MINING DENGAN METODE POHON KEPUTUSAN ALGORITMA ID3 UNTUK MENENTUKAN STATUS MAHASISWA*.

Sari, H., Leonarde Ginting, G., & Zebua, T. (2021). *Penerapan Algoritma Text Mining dan TF-IDF Untuk Pengelompokan Topik Skripsi Pada Aplikasi Repository STMIK Budi Darma*. *2*(7), 414–432. https://ejurnal.seminar-id.com/index.php/tin

Sathishkumar, S., & Ibrahim, M. M. (2025). Evaluation of machine learning algorithms on hydrogen boosted homogeneous charge compression ignition engine operation for performance and emission prediction. *Process Safety and Environmental Protection*, *195*. https://doi.org/10.1016/j.psep.2025.01.010

Susetyoko, R., Yuwono, W., Purwantini, E., & Ramadijanti, N. (2022). *Perbandingan Metode Random Forest, Regresi Logistik, Naïve Bayes, dan Multilayer Perceptron Pada Klasifikasi Uang Kuliah Tunggal (UKT)*. *7*(1).

Syifah, N. (2024a, February 15). *Servis berkala mobil perlu gak, sih? Ini jenis, manfaat dan waktu. Bengkelly*. https://bengkelly.co.id/servis-berkala-mobil-perlu-gak-sih-ini-jenis-manfaat-dan-waktu/

Syifah, N. (2024b, May 13). *Oli mobil bocor? Ini penyebab, dampak dan solusinya. Fleet Maintenance*. https://fleetmaintenance.co.id/oli-mobil-bocor-ini-penyebab-dampak-dan-solusinya/

Timmisetty, D. R., Prudhvi, A., Krishna, C. D., & Prasanna, S. H. L. (2024). *Text Mining for Effective Classification of Car Problem Severity Levels*. https://aisel.aisnet.org/mwais2024/34

Torres, Y., Nadeau, S., & Landau, K. (2021). Classification and quantification of human error in manufacturing: A case study in complex manual assembly. *Applied Sciences (Switzerland)*, *11*(2), 1–23. https://doi.org/10.3390/app11020749

Vernanda, D., Nurizati, Z., & Hidayat, A. (2024). *KLASIFIKASI NANAS LAYAK JUAL MENGGUNAKAN METODE NAÏVE BAYES* (Vol. 10).

Wati, R., Ernawati, S., & Rachmi, H. (2023). Pembobotan TF-IDF Menggunakan Naïve Bayes pada Sentimen Masyarakat Mengenai Isu Kenaikan BIPIH. *Jurnal Manajemen Informatika (JAMIKA)*, *13*(1), 84–93. https://doi.org/10.34010/jamika.v13i1.9424

Wulandari, D. A., Rohmat Saedudin, R., & Andreswari, R. (n.d.). *ANALISIS SENTIMEN MEDIA SOSIAL TWITTER TERHADAP REAKSI MASYARAKAT PADA RUU CIPTA KERJA MENGGUNAKAN METODE KLASIFIKASI ALGORITMA NAIVE BAYES ANALYSIS OF TWITTER SOCIAL MEDIA SENTIMENT ON THE PUBLIC'S REACTION TO THE DRAFTS OF JOB CREATION LAW USING THE CLASSIFICATION METHOD NAIVE BAYES ALGORITHM*.

Xiao, L., Li, Q., Ma, Q., Shen, J., Yang, Y., & Li, D. (2024). Text classification algorithm of tourist attractions subcategories with modified TF-IDF and Word2Vec. *PloS One*, *19*(10), e0305095. https://doi.org/10.1371/journal.pone.0305095

Yang, J., Li, Y., Liu, Q., Li, L., Feng, A., Wang, T., Zheng, S., Xu, A., & Lyu, J. (2020). Brief introduction of medical database and data mining technology in big data era. In *Journal of Evidence-Based Medicine* (Vol. 13, Issue 1, pp. 57–69). Blackwell Publishing. https://doi.org/10.1111/jebm.12373

Yu, C. T., Lam, K., & Salton, G. (1982). *Term Weighting in Information Retrieval Using the Term Precision Model*.

Zhao, D., Gong, M., Zhao, D., Liu, W., & Chen, W. (2024). Active suspension and steering system control of emergency rescue vehicle based on sliding mode dual robust coordination control. *Advances in Mechanical Engineering*, *16*(6). https://doi.org/10.1177/16878132241259720

LAMPIRAN

A- *Import Library* dan Pemanggilan Data

```
import pandas as pd
import numpy as np
import re
import string
import nltk
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
from sklearn.preprocessing import LabelEncoder
from collections import Counter
```

```
# Install dan import stemming
!pip install Sastrawi
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

```
Requirement already satisfied: Sastrawi in /usr/local/lib/python3.11/dist-packages (1.0.1)
```

```
# Inisialisasi
nltk.download('stopwords')
stop_words = set(stopwords.words('indonesian'))
internal_words = {"internal", "eksternal", "no", "po", "pa", "pb", "ad", "km", "pc", "ts", "b",
    "ucv", "ucu", "ucy", "gt", "g/t", "cp", "oh",
    "karawaci", "madiun", "jogja", "badri", "dwiki",
    "vendor", "astra", "world", "stooring", "other",
    "part", "adhoc", "electrical", "engine", "undrcarriage",
    "awo", "aw"}
factory = StemmerFactory()
stemmer = factory.create_stemmer()
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
# Load Dataset
df = pd.read_excel("SPK Jan - Mar 2024.xlsx")
print(f"Total data: {len(df)}")
df
```

```
Total data: 250368
```

	Order Desc	Item Text	Jenis Perbaikan
0	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	NO PO : 4102186285 (Internal)	Adhoc
1	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	Adhoc
2	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	Adhoc
3	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	V-Belt	Adhoc
4	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	V-Belt P/S	Adhoc
...	...	...	...
250363	PB,BE-9665-YB,KM.125586,PB.125	Lampu box	PB
250364	PBTT,BD-1316-EC,KM.30361,G/T WIPER	G/T WIPER AUS (Internal)	PBTT
250365	PBTT,BD-1316-EC,KM.30361,G/T WIPER	PO 4102251025 (Internal)	PBTT
250366	PBTT,BD-1316-EC,KM.30361,G/T WIPER	Wiper	PBTT
250367	PBTT,BD-1316-EC,KM.30361,G/T WIPER	Wiper	PBTT

250368 rows × 3 columns

B- *Pre-processing Data*

```
df = df.drop_duplicates()
df = df.dropna(subset=["Item Text"]).reset_index(drop=True)
```

```
def preprocess_text(text):
    text = str(text).lower()
    text = re.sub(r'\d+', '', text)
    text = text.translate(str.maketrans('', '', string.punctuation))
    text = re.sub(r'\s+', ' ', text).strip()
    words = [word for word in text.split() if word not in stop_words and word not in internal_words]
    stemmed = [stemmer.stem(word) for word in words]
    return " ".join(stemmed)

df["Cleaned_Text"] = df["Item Text"].apply(preprocess_text)
df[["Item Text", "Cleaned_Text"]].head()
```

	Item Text	Cleaned_Text
0	NO PO : 4102186285 (Internal)	
1	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	pabucukmgt vbelt
2	V-Belt	vbelt
3	V-Belt P/S	vbelt ps
4	KUNJUNGAN ASTRA WORLD (Internal)	kunjung

```
print(f"Total data after preprocessing: {len(df)}")
df
```

```
Total data after preprocessing: 236474
```

	Order Desc	Item Text	Jenis Perbaikan	Cleaned_Text
0	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	NO PO : 4102186285 (Internal)	Adhoc	
1	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	Adhoc	pabucukmgt vbelt
2	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	V-Belt	Adhoc	vbelt
3	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	V-Belt P/S	Adhoc	vbelt ps
4	PA,B-9215-UXA,KM.47940,STOORING	KUNJUNGAN ASTRA WORLD (Internal)	Adhoc	kunjung
...	...	...	...	...
236469	PB,BE-9665-YB,KM.125586,PB.125	Lampu	PB	lampu
236470	PB,BE-9665-YB,KM.125586,PB.125	Lampu box	PB	lampu box
236471	PBTT,BD-1316-EC,KM.30361,G/T WIPER	G/T WIPER AUS (Internal)	PBTT	wiper aus
236472	PBTT,BD-1316-EC,KM.30361,G/T WIPER	PO 4102251025 (Internal)	PBTT	
236473	PBTT,BD-1316-EC,KM.30361,G/T WIPER	Wiper	PBTT	wiper

236474 rows × 4 columns

```
# Terapkan preprocessing
df["Cleaned_Text"] = df["Item Text"].apply(preprocess_text)

print("\n🔹 Contoh Data Setelah Preprocessing:")
print(df[["Item Text", "Cleaned_Text"]].head())
```

```
🔹 Contoh Data Setelah Preprocessing:
                                   Item Text      Cleaned_Text
0             NO PO : 4102186285 (Internal)
1  PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE  pabucukmgt vbelt
2                                    V-Belt             vbelt
3                                 V-Belt P/S          vbelt ps
4          KUNJUNGAN ASTRA WORLD (Internal)           kunjung
```

C- Eksplorasi Data & Visualisasi

```
df.head()
```

	Order Desc	Item Text	Jenis Perbaikan	Cleaned_Text
0	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	NO PO : 4102186285 (Internal)	Adhoc	
1	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	Adhoc	pabucukmgt vbelt
2	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	V-Belt	Adhoc	vbelt
3	PA,B-9304-UCU,KM.65562,G/T V-BELT ENGINE	V-Belt P/S	Adhoc	vbelt ps
4	PA,B-9215-UXA,KM.47940,STOORING	KUNJUNGAN ASTRA WORLD (Internal)	Adhoc	kunjung

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 236474 entries, 0 to 236473
Data columns (total 4 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   Order Desc       236474 non-null  object
 1   Item Text        236474 non-null  object
 2   Jenis Perbaikan  236474 non-null  object
 3   Cleaned_Text     236474 non-null  object
dtypes: object(4)
memory usage: 7.2+ MB
```

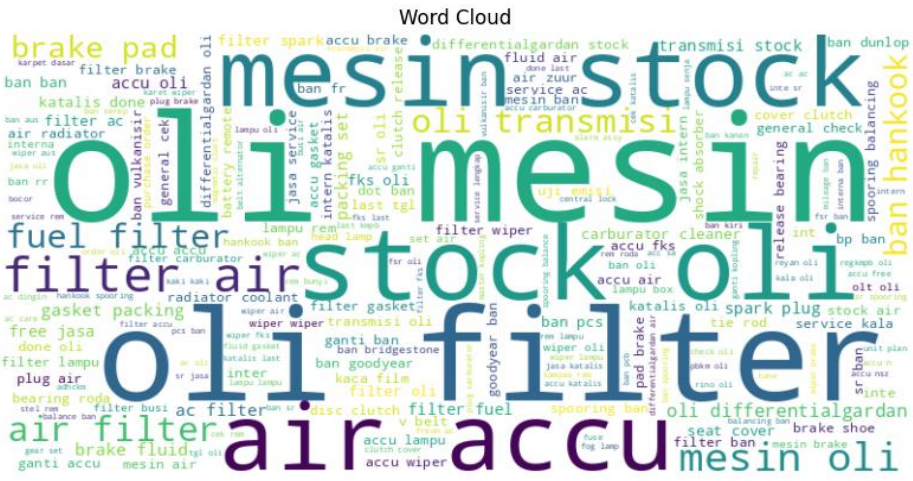
```
print("\nMissing values per column:")
print(df.isnull().sum())

print("\nNumber of unique values per column:")
print(df.nunique())
```

```
Missing values per column:
Order Desc         0
Item Text          0
Jenis Perbaikan    0
Cleaned_Text       0
dtype: int64

Number of unique values per column:
Order Desc         49868
Item Text          70871
Jenis Perbaikan        3
Cleaned_Text       29137
dtype: int64
```

```
# Visualisasi Wordcloud
wordcloud = WordCloud(width=800, height=400, background_color="white").generate(" ".join(df["Cleaned_Text"]))
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title("Word Cloud")
plt.show()
```



D- TF-IDF dan Split Data

```
# TF-IDF
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(df["Cleaned_Text"])
y = df["Jenis Perbaikan"]
```

```
# Split Data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

E- *Training Model Machine Learning*

```
# Model Machine Learning
results = {}
models = {
    "Naive Bayes": MultinomialNB(),
    "Logistic Regression": LogisticRegression(max_iter=1000, random_state=42),
    "Random Forest": RandomForestClassifier(n_estimators=100, random_state=42)
}

y_preds = {}

for model_name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    y_preds[model_name] = y_pred
    results[model_name] = {
        "accuracy": accuracy_score(y_test, y_pred),
        "classification_report": classification_report(y_test, y_pred)
    }
```

F-1 Evaluasi Model *Machine Learning*

```
# Hasil Akurasi
for model, res in results.items():
    print(f"\n 🔹 {model}:")
    print(f"Akurasi: {res['accuracy']:.4f}")
    print(res["classification_report"])
```

```
 🔹 Naive Bayes:
Akurasi: 0.7351
              precision    recall  f1-score   support

       Adhoc       0.66      0.46      0.54      9397
          PB       0.74      0.92      0.82     26414
        PBTT       0.76      0.54      0.63     11484

    accuracy                           0.74     47295
   macro avg       0.72      0.64      0.67     47295
weighted avg       0.73      0.74      0.72     47295


 🔹 Logistic Regression:
Akurasi: 0.7441
              precision    recall  f1-score   support

       Adhoc       0.68      0.44      0.54      9397
          PB       0.75      0.92      0.83     26414
        PBTT       0.77      0.59      0.67     11484

    accuracy                           0.74     47295
   macro avg       0.73      0.65      0.68     47295
weighted avg       0.74      0.74      0.73     47295


 🔹 Random Forest:
Akurasi: 0.7549
              precision    recall  f1-score   support

       Adhoc       0.68      0.47      0.55      9397
          PB       0.76      0.92      0.83     26414
        PBTT       0.77      0.61      0.68     11484

    accuracy                           0.75     47295
   macro avg       0.74      0.67      0.69     47295
weighted avg       0.75      0.75      0.74     47295
```

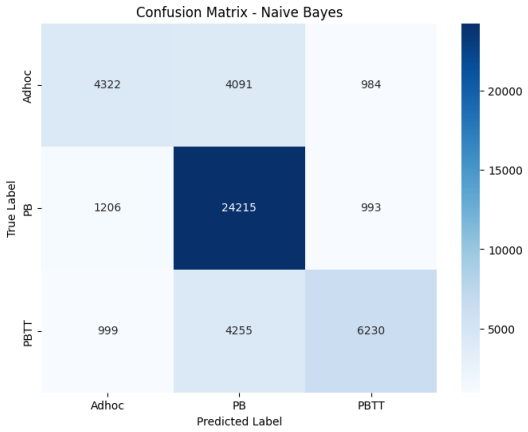
F-2 Evaluasi Model *Machine Learning*

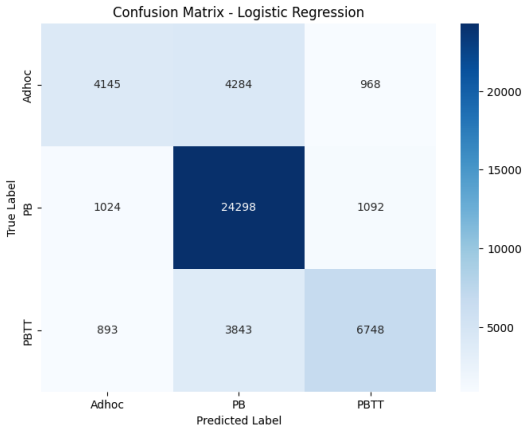
```
# Confusion Matrix
def plot_confusion_matrix(y_true, y_pred, title):
    cm = confusion_matrix(y_true, y_pred)

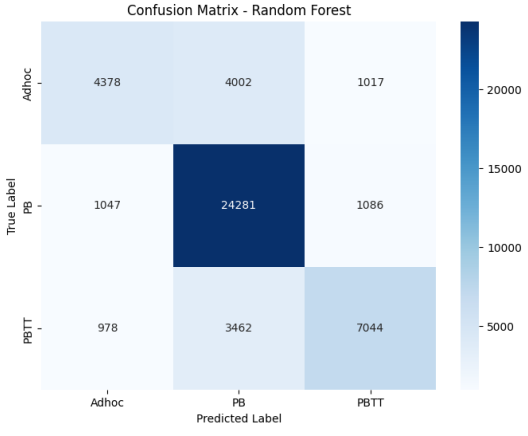
    # Dapatkan unique labels langsung dari y_true dan sort
    labels = sorted(y_true.unique())

    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
                xticklabels=labels, yticklabels=labels)
    plt.title(f"Confusion Matrix - {title}")
    plt.xlabel("Predicted Label")
    plt.ylabel("True Label")
    plt.show()

for model_name, y_pred in y_preds.items():
    plot_confusion_matrix(y_test, y_pred, model_name)
```







G-1 Implementasi Model

```
# Load Dataset Baru
new_data = pd.read_excel("Data Baru.xlsx")
print(f"Total data: {len(new_data)}")
new_data
```

```
Total data: 150
```

	Order Desc	Item Text
0	PB,KT-1912-KS,KM.79470,PB.80	G/T,FILTER AC,DENSO,1PC (Internal)
1	PB,KT-8041-YJ,KM.104700,PB.105,G/T LINK	G/T,FILTER UDARA,NON GENUINE,1PC (Intern
2	PB,KT-8041-YJ,KM.104700,PB.105,G/T LINK	G/T,FUEL FILTER,NON GENUINE,2PC (Interna
3	PBTT,KT-8336-YW,KM.47623,PM CHECK	G/T,LAMPU KOTA,UNIVERSAL,1PC (Internal)
4	PA,KT-8378-YW,KM.69522,REP ELK ROTARI	G/T,LAMPU KOTA,UNIVERSAL,1PC (Internal)
...	...	...
145	PA,KT-8652-YO,KM.99854,G/T, LAMPU KABUT	G/T,LAMPU KABUT,OSRAM 12V 55W,1PC (Inter
146	PA,KT-8652-YO,KM.99854,G/T, LAMPU KABUT	G/T,LAMPU KOTA,UNIVERSAL,1PC (Internal)
147	PA,KT-8335-YW,KM.92494,GT MOTOR WASHER	G/T,MOTOR WASHER,UNIVERSAL,1PC (Internal
148	PA,KT-8378-YW,KM.77638,GT LAMPU KOTA	G/T,LAMPU KOTA,UNIVERSAL,1PC (Internal)
149	PBTT,KT-8770-YW,KM.84946,G/T MAGNET CLUT	G/T,FILTER AC,DENSO,1PC (Internal)

150 rows × 2 columns

```
# Apply Preprocessing
new_data["Cleaned_Text"] = new_data["Item Text"].apply(preprocess_text)

# Transform ke TF-IDF
new_texts_vector = vectorizer.transform(new_data["Cleaned_Text"])

# Prediksi dengan Random Forest
predicted_classes = models['Random Forest'].predict(new_texts_vector)

# DataFrame hasil prediksi
result_df = pd.DataFrame({
    "Order Desc" : new_data["Order Desc"],
    "Item Text": new_data["Item Text"],
    "Random Forest Prediction": predicted_classes
})

# Hasil Prediksi
print(" Hasil Prediksi Data Baru:")
display(result_df)
```

```
Hasil Prediksi Data Baru:
```

	Order Desc	Item Text	Random Forest Prediction
0	PB,KT-1912-KS,KM.79470,PB.80	G/T,FILTER AC,DENSO,1PC (Internal)	PB
1	PB,KT-8041-YJ,KM.104700,PB.105,G/T LINK	G/T,FILTER UDARA,NON GENUINE,1PC (Intern	PB
2	PB,KT-8041-YJ,KM.104700,PB.105,G/T LINK	G/T,FUEL FILTER,NON GENUINE,2PC (Interna	PB
3	PBTT,KT-8336-YW,KM.47623,PM CHECK	G/T,LAMPU KOTA,UNIVERSAL,1PC (Internal)	Adhoc
4	PA,KT-8378-YW,KM.69522,REP ELK ROTARI	G/T,LAMPU KOTA,UNIVERSAL,1PC (Internal)	Adhoc
...	...	...	...
145	PA,KT-8652-YO,KM.99854,G/T, LAMPU KABUT	G/T,LAMPU KABUT,OSRAM 12V 55W,1PC (Inter	Adhoc
146	PA,KT-8652-YO,KM.99854,G/T, LAMPU KABUT	G/T,LAMPU KOTA,UNIVERSAL,1PC (Internal)	Adhoc
147	PA,KT-8335-YW,KM.92494,GT MOTOR WASHER	G/T,MOTOR WASHER,UNIVERSAL,1PC (Internal	PB
148	PA,KT-8378-YW,KM.77638,GT LAMPU KOTA	G/T,LAMPU KOTA,UNIVERSAL,1PC (Internal)	Adhoc
149	PBTT,KT-8770-YW,KM.84946,G/T MAGNET CLUT	G/T,FILTER AC,DENSO,1PC (Internal)	PB

150 rows × 3 columns

```
#Simpan ke Excel
result_df.to_excel("Hasil_Prediksi_Data_Baru.xlsx", index=False)
```

H-1 *Output Rule Model*

```python
# Rule Naive Bayes
from sklearn.naive_bayes import MultinomialNB
nb_model = MultinomialNB()
nb_model.fit(X_train, y_train)

import pandas as pd
import numpy as np

feature_names = vectorizer.get_feature_names_out()
data = {}

for idx, class_label in enumerate(nb_model.classes_):
    data[f'log P(word|class={class_label})'] = nb_model.feature_log_prob_[idx]

df_nb = pd.DataFrame(data, index=feature_names)
df_nb_sorted = df_nb.sort_values(by=df_nb.columns[0], ascending=False).head(20)
display(df_nb_sorted)
```

	log P(word|class=Adhoc)	log P(word|class=PB)	log P(word|class=PBTT)
lampu	-3.839186	-4.368510	-5.364514
sr	-3.857273	-4.261426	-3.668664
ac	-4.346747	-4.434435	-4.513644
ban	-4.374490	-5.139787	-2.302653
wiper	-4.389978	-4.139675	-4.027953
rem	-4.502643	-4.991691	-5.344390
brake	-4.545534	-4.242429	-4.895548
spooring	-4.622463	-5.732867	-3.736368
cover	-4.712512	-7.679201	-4.865391
filter	-4.780954	-2.460724	-5.128746
seat	-4.842441	-8.852820	-5.656987
adhckm	-4.869981	-11.164576	-8.229346
kaca	-4.884305	-7.999632	-5.842303
roda	-4.885504	-6.778697	-6.384774
accu	-4.912906	-3.228428	-3.035545
film	-5.009565	-9.199163	-5.834122
bearing	-5.009762	-6.771067	-5.314027
box	-5.083693	-7.128786	-7.666318
service	-5.111992	-5.019798	-5.432202
fluid	-5.124434	-4.945132	-6.338025

```python
## Rule Logistic Regression
# Ambil fitur (kata) dan koefisien model
feature_names = vectorizer.get_feature_names_out()
coefs = lr_model.coef_
classes = lr_model.classes_

# Buat DataFrame: untuk setiap kelas
top_n = 10
top_words_per_class = {}

for i, class_label in enumerate(classes):
    class_coefs = coefs[i]
    top_indices = np.argsort(class_coefs)[::-1][:top_n]

    top_words_per_class[f"Top words for class={class_label}"] = feature_names[top_indices]
    top_words_per_class[f"Coef (toward class={class_label})"] = class_coefs[top_indices]

# Gabungkan ke dalam DataFrame hasil
df_rules_lr_multiclass = pd.DataFrame(top_words_per_class)

# Tampilkan
print("Logistic Regression Rules (Multiclass):")
display(df_rules_lr_multiclass)
```

H-2 *Output Rule Model*

Logistic Regression Rules (Multiclass):

	Top words for class=Adhoc	Coef (toward class=Adhoc)	Top words for class=PB	Coef (toward class=PB)	Top words for class=PBTT	Coef (toward class=PBTT)
0	adhockmgt	3.541478	kelipat	4.241172	gtban	4.018952
1	rekondisi	3.108699	kala	4.012534	vulkanisir	3.377421
2	adhckm	2.746982	gtoli	3.740264	regkmgtban	3.253173
3	adhckmcp	2.630053	olt	3.212201	dot	3.244501
4	adhckmalarm	2.591778	free	3.144031	kmpc	3.226955
5	pabucukmgt	2.496356	gtfilter	3.048955	release	3.178312
6	pabuzxkmgt	2.300719	pbkm	3.012962	zuur	3.153667
7	cylinder	2.289763	regkmpb	2.881191	regkmban	3.136428
8	piston	2.276282	lengkap	2.871031	ori	3.112311
9	metal	2.187742	air	2.831442	checkbersihkan	3.054244

```python
import pandas as pd
import numpy as np

# Ambil fitur (kata-kata)
feature_names = vectorizer.get_feature_names_out()

# Ambil importance dari model
rf_importances = rf_model.feature_importances_

# Buat DataFrame
df_rf_importances = pd.DataFrame({
    "Feature": feature_names,
    "Importance": rf_importances
})

# Urutkan berdasarkan importance tertinggi
df_rf_sorted = df_rf_importances.sort_values(by="Importance", ascending=False).head(20)

# Tampilkan
print("Random Forest Top Features (Rules):")
display(df_rf_sorted)
```

✅ Random Forest Top Features (Rules):

	Feature	Importance
1146	ban	0.084696
6660	oli	0.076520
56	accu	0.038540
3145	filter	0.035531
715	air	0.032090
5972	mesin	0.030029
4025	hankook	0.018533
11552	stock	0.016035
3175	fks	0.012537
11393	spooring	0.011637
3573	goodyear	0.008754
9516	pcs	0.008722
3381	ganti	0.007661
4768	katalis	0.007082
2264	cover	0.006966
3253	free	0.006457
259	adhckm	0.005974
10203	regkmpb	0.005500
2176	clutch	0.005482
36	ac	0.005409