TESIS

ANALISIS PERBANDINGAN ALGORITMA RANDOM FOREST DAN SVM PADA DATA SENSOR GERAK SMARTPHONE UNTUK MODEL KLASIFIKASI KEAMANAN BERKENDARA

Disusun oleh:

Nama : **Lisa Dinda Yunita**
NIM : **22.55.1215**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2024

TESIS

ANALISIS PERBANDINGAN ALGORITMA RANDOM FOREST DAN SVM PADA DATA SENSOR GERAK SMARTPHONE UNTUK MODEL KLASIFIKASI KEAMANAN BERKENDARA

COMPARATIVE ANALYSIS OF RANDOM FOREST AND SVM ALGORITHMS FOR DRIVING SAFETY CLASSIFICATION MODEL ON SMARTPHONE MOTION SENSOR DATA

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama	**: Lisa Dinda Yunita**
NIM	**: 22.55.1215**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2024

HALAMAN PENGESAHAN

ANALISIS PERBANDINGAN ALGORITMA RANDOM FOREST DAN SVM PADA DATA SENSOR GERAK SMARTPHONE UNTUK MODEL KLASIFIKASI KEAMANAN BERKENDARA

COMPARATIVE ANALYSIS OF RANDOM FOREST AND SVM ALGORITHMS FOR DRIVING SAFETY CLASSIFICATION MODEL ON SMARTPHONE MOTION SENSOR DATA

Dipersiapkan dan Disusun oleh

Lisa Dinda Yunita

22.55.1215

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Kamis, 1 Februari 2014

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 1 Februari 2024
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

ANALISIS PERBANDINGAN ALGORITMA RANDOM FOREST DAN SVM PADA DATA SENSOR GERAK SMARTPHONE UNTUK MODEL KLASIFIKASI KEAMANAN BERKENDARA

COMPARATIVE ANALYSIS OF RANDOM FOREST AND SVM ALGORITHMS FOR DRIVING SAFETY CLASSIFICATION MODEL ON SMARTPHONE MOTION SENSOR DATA

Dipersiapkan dan Disusun oleh

Lisa Dinda Yunita

22.55.1215

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Kamis, 1 Februari 2024

Pembimbing Utama	**Anggota Tim Penguji**
Prof. Dr. Ema Utami, S.Si., M.Kom	**Alva Hendi Muhammad, S.T., M.Eng., Ph.D.**
NIK. 190302037	**NIK. 190302493**
Pembimbing Pendamping	**Hanafi, S.Kom., M.Eng., Ph.D.**
	NIK. 190302024
Ainul Yaqin, M.Kom.	**Prof. Dr. Ema Utami, S.Si., M.Kom**
NIK. 190302255	**NIK. 190302037**

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 1 Februari 2024
Direktur Program Pascasarjana

Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Lisa Dinda Yunita
NIM : 22.55.1215
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Analisis Perbandingan Algoritma Random Forest dan SVM pada Data Sensor Gerak Smartphone untuk Model Klasifikasi Keamanan Berkendara

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom
Dosen Pembimbing Pendamping : Ainul Yaqin, M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 1 Februari 2024
Yang Menyatakan,

Lisa Dinda Yunita

HALAMAN PERSEMBAHAN

Alhamdulillahirobbil'alamiin, segala puji bagi Allah SWT yang telah mencurahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan TESIS ini dengan baik.

Karya ini saya persembahkan untuk:

1. Allah SWT, yang telah memberikan rahmat dan hidayah-Nya sehingga Tesis ini bisa tersusun dan selesai tanpa ada halangan apapun, terimakasih Ya Allah engkau telah memberikan kekuatan, kesabaran dan semangat yang luar biasa.
2. Suami tercinta, Ahmad Iwan Fadli, S.Kom.,M.Eng., kedua orang tua saya, Ibu Mujianti, S.Pd dan Alm. Bambang Trismanto, serta Keluarga yang telah memberikan dorongan, semangat, moral, materi, limpahan kasih sayang, dan do'a yang selalu menyertai setiap langkah ini.
3. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom dan Bapak Ainul Yaqin, M.Kom. yang telah memberikan bimbingan dalam Tesis ini.

HALAMAN MOTTO

Sebaik-baik manusia adalah yang bermanfaat bagi orang lain

Ilmu itu didapat dari lidah yang gemar bertanya dan akal yang senang berfikir

KATA PENGANTAR

Alhamdulliahirobbil'alamin, puji syukur kehadirat Allah SWT atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan Tesis ini.

Laporan ini disusun sebagai salah satu syarat kelulusan program Pascasarjana di Universitas AMIKOM Yogyakarta. Sejak persiapan sampai selesainya Tesis ini penulis menerima bantuan dan dukungan dari berbagai pihak yang penulis butuhkan guna terselesaikannya laporan ini. Untuk itu dalam kesempatan ini penulis mengucapkan terima kasih kepada :

1. Ibu Prof. Dr. Kusrini, M.Kom selaku Direktur Pascasarjana Magister Teknik Informatika Universitas AMIKOM Yogyakarta.
2. Bapak Prof. Dr. Ema Utami, S.Si., M.Kom. dan Bapak Ainul Yaqin, M.Kom. selaku dosen pembimbing yang telah memberikan bimbingan, waktu dan arahan dalam pembuatan Tesis ini.
3. Bapak Alva Hendi Muhammad, S.T., M.Eng., Ph.D dan Bapak Hanafi, S.Kom., M.Eng., Ph.D.. selaku dosen penguji dalam ujian Tesis ini.
4. Seluruh Dosen Pascasarjana Universitas AMIKOM Yogyakarta yang telah men-*sharing* ilmu selama perkuliahan
5. Semua pihak yang telah membantu dalam kelancaran penulisan Tesis ini baik langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, meskipun demikian penulis berharap semoga laporan ini bermanfaat bagi yang membacanya

dan penulis dengan senang hati menerima kritik dan saran yang membangun dari para pembaca.

Akhir kata penulis berharap semoga hasil karya ini dapat berguna serta bermanfaat bagi perkembangan Teknologi dan Informasi pada khususnya, serta sebagai kajian bagi mahasiswa Pascasarjana Universitas Amikom Yogyakarta lainnya dalam pengambilan Tesis.

Yogyakarta, 01 Februari 2024

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

INTISARI — Dalam era mobilitas modern, perhatian utama tertuju pada keselamatan berkendara. Menghadapi perilaku mengemudi yang tidak aman, seperti mengemudi dalam kondisi mabuk, kecepatan tinggi, dan menggunakan ponsel, merupakan tantangan signifikan yang memerlukan identifikasi pola secara efektif. Penelitian ini berfokus pada eksplorasi karakteristik berkendara melalui analisis data sensor gerak smartphone dari sistem transportasi online.

Data sensor gerak smartphone pengemudi direkam dan dianalisis dengan menerapkan metode klasifikasi menggunakan algoritma machine learning, seperti Random Forest dan Support Vector Machine. Hasil penelitian menunjukkan pendekatan ini efektif dalam memprediksi tingkat keselamatan berkendara, mempertimbangkan faktor krusial seperti kecepatan, akselerasi kendaraan, dan waktu tempuh. Lebih dari sekadar model klasifikasi, metode yang dikembangkan dalam penelitian ini merupakan langkah maju dalam memahami serta mengatasi perilaku mengemudi yang potensial membahayakan.

Analisis hasil menunjukkan bahwa model Random Forest memberikan kinerja lebih baik dengan akurasi sekitar 92.88% dibandingkan dengan model Support Vector Machine. Pembersihan outlier pada fitur waktu, kecepatan, dan akurasi GPS terbukti meningkatkan kinerja model, walaupun tantangan terkait data akurasi yang memiliki nilai ekstrim masih perlu ditangani secara lebih mendalam.

Penelitian ini mengindikasikan adanya peluang proaktif dalam meningkatkan keselamatan berkendara melalui alat identifikasi dan intervensi perilaku tidak aman. Dengan penerapan teknologi ini, diharapkan pengambilan keputusan terkait lalu lintas dapat menjadi lebih cerdas dan efisien. Hasil penelitian ini, tidak hanya sebuah model, melainkan juga kontribusi yang potensial besar terhadap perkembangan teknologi yang lebih aman dan efisien di sektor transportasi. Temuan ini memberikan landasan penting bagi pemerintah dan industri transportasi dalam menciptakan masa depan jalan raya yang lebih aman dan tertata rapi.

KATA KUNCI — Data Mining, CRISP-DM, Driving Behavior, Machine Learning, Classification.

ABSTRACT

ABSTRACT — Road safety is a paramount concern in the era of modern mobility. The significant challenge in addressing unsafe driving behaviors, such as driving under the influence, speeding, and mobile phone usage, lies in effectively identifying these patterns. To tackle this challenge, this research explores driving characteristics through leveraging motion sensor data from smartphones in advanced online transportation systems.

The motion sensor data from drivers' smartphones is recorded and analyzed using classification methods employing machine learning algorithms such as Random Forest and Support Vector Machine. The research findings indicate the effectiveness of this approach in predicting the level of road safety by considering crucial factors such as speed, vehicle acceleration, and travel time. The method developed in this study is not merely a classification model; it represents a significant stride in understanding and addressing potentially hazardous driving behaviors.

The research reveals that the Random Forest model outperforms the Support Vector Machine model, achieving an accuracy of approximately 92.88%. Outlier cleansing in features like time, speed, and GPS accuracy contributes to enhancing the model's performance. However, challenges persist concerning accuracy data with extreme values, requiring further attention.

This study opens the door to a more proactive approach to road safety by providing a robust tool for identifying and intervening in unsafe behaviors. With this technology, it is anticipated that decision-making in traffic-related matters can become more intelligent and efficient. The research findings are not just a model but also a potentially substantial contribution to the development of safer and more efficient transportation technology. We believe these findings will provide a crucial foundation for governments and the transportation industry to create a safer and more organized future for roadways.

KEYWORDS — Data Mining, CRISP-DM, Driving Behavior, Machine Learning, Classify

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Perilaku mengemudi merupakan faktor krusial yang mempengaruhi keselamatan, konsumsi bahan bakar, dan emisi gas dalam transportasi (WHO, 2020). Meskipun begitu, keselamatan lalu lintas menjadi sorotan utama di seluruh dunia. Menurut laporan World Health Organization (WHO) tahun 2020, lebih dari satu juta orang kehilangan nyawa akibat kecelakaan lalu lintas di seluruh dunia. Perilaku mengemudi yang tidak aman, seperti mengemudi dalam keadaan mabuk, kecepatan tinggi, dan penggunaan ponsel, seringkali dikaitkan dengan kehilangan konsentrasi dan menjadi risiko besar.

Dalam mengatasi tantangan ini, penelitian ini mengeksplorasi karakteristik berkendara dengan memanfaatkan data sensor gerak dari smartphone dalam sistem transportasi online. Metode klasifikasi menggunakan algoritma machine learning, seperti Random Forest dan Support Vector Machine, digunakan untuk mengkategorikan perilaku mengemudi sebagai normal atau agresif, sehingga dapat diketahui apakah pengemudi mengendarai secara aman atau berbahaya.

Penggunaan teknologi Intelligent Transportation Systems (ITS) dan evolusi dari Intelligent Vehicle Highway System (IVHS) menjadi ITS mencerminkan upaya berkelanjutan untuk mengatasi tantangan kecelakaan lalu lintas. ITS bertujuan meningkatkan informativitas, ketepatan waktu, kelancaran, dan keselamatan dalam sistem transportasi (Zhu., 2019). Dengan memanfaatkan data

sensor elektronik dan transmisi data, metode statistik di dalam ITS memberikan informasi penting, termasuk deteksi perilaku mengemudi dan deteksi mode kendaraan (Lu., 2018).

Machine learning (ML) adalah proses ekstraksi informasi pada dataset untuk mendapatkan sebuah pengetahuan menggunakan teknik tertentu dataset yang digunakan adalah data perjalanan khususnya data berkendara menggunakan mobil seperti data sensor accelerometer, sensor gyroscope dan data Global Positioning System (Pintye., 2020). Teknik Machine Learning yang digunakan adalah teknik klasifikasi dengan mengelompokkan data berdasarkan objek yang sudah diberi label secara manual sehingga didapatkan sebuah model klasifikasi yang mampu mendeteksi cara berkendara pada perjalanan yang akan datang (Chacko., 2023). Peneliti sebelumnya yang terkait juga melakukan penelitian tentang cara berkemudi, dari beberapa penelitian yang sudah ada peneliti kebanyakan menggunakan sensor gyroscope, sedangkan metode yang digunakan adalah metode klasifikasi dengan menerapkan beberapa algoritma SVM dan RF.

Penelitian sebelumnya telah mengukur kinerja klasifikasi algoritma Machine Learning Algorithms (MLA) dalam mendeteksi cara mengemudi dengan melakukan evaluasi terhadap SVM, RF, ANN, dan BN menggunakan data dari 4 fitur sensor smartphone, seperti akselerometer, akselerasi linier, magnetometer, dan gyroscope (Lu., 2018). Penelitian ini memberikan landasan bagi pemahaman performa algoritma klasifikasi pada konteks deteksi perilaku mengemudi. Selain itu, penelitian oleh (Nguyen., 2018) berfokus pada Human Activity Recognition (HAR), menggunakan data sensor accelerometer dan sensor suara pada smartphone.

Penelitian ini membandingkan performa classifier, termasuk Multi-Layer Perceptron, Decision Tree, dan SVM, serta mengatasi masalah imbalance dataset dengan menerapkan metode oversampling. Temuan ini memberikan perspektif tambahan tentang klasifikasi aktivitas manusia dan penanganan dataset yang tidak seimbang. Penelitian lain membahas klasifikasi mengemudi normal dan agresif dengan algoritme fuzzy menggunakan data sensor smartphone, menyoroti pentingnya sensor tambahan seperti gyroscope (Ben,. 2019). Selain itu, penelitian selanjutnya mengeksplorasi penggunaan sensor gyroscope untuk mendeteksi cara berkendara tidak aman, menunjukkan akurasi classifier yang baik dalam menilai berbagai kriteria gerakan berbahaya (Gorodnichev,.2019). Penelitian lain mengusulkan metode monitoring berbasis machine learning untuk mengidentifikasi mode transportasi dengan memproses data sensor smartphone, termasuk accelerometer, gyroscope, dan light sensors. Hasilnya menunjukkan bahwa metode classifier RF menghasilkan kinerja terbaik, memberikan wawasan tentang identifikasi mode transportasi yang lebih luas (Jahangiri,. 2018).

Pemahaman terhadap penelitian-penelitian sebelumnya ini penting dalam mengarahkan fokus penelitian terbaru, membentuk landasan teoretis, dan merinci kontribusi yang diharapkan dari penelitian yang sedang dilakukan. Penelitian ini menggarisbawahi pentingnya pembersihan outlier untuk meningkatkan kinerja model, yang terbukti dengan model Random Forest yang lebih unggul dibandingkan dengan Support Vector Machine. Meski demikian, penelitian ini juga menghadapi tantangan, seperti dataset yang tidak seimbang dan keterbatasan fitur. Dalam merespon dinamika penelitian pada analisis keamanan berkendara

menggunakan data sensor gerak smartphone, penelitian ini mempertimbangkan tantangan unik yang ditimbulkan oleh ukuran dataset yang besar. Dalam upaya untuk meningkatkan efisiensi komputasi dan memfokuskan perhatian pada informasi yang paling relevan, penelitian ini memilih pendekatan penghapusan data outlier sebagai strategi penanganan. Penghapusan data outlier dianggap sebagai solusi praktis yang dapat memungkinkan peningkatan signifikan dalam kecepatan eksekusi algoritma pembelajaran mesin. Dengan mengurangi ukuran dataset, penanganan outlier ini diharapkan memberikan ruang lingkup analisis yang lebih fokus, terutama terkait dengan pola pergerakan smartphone yang mewakili kondisi keamanan berkendara sehari-hari. Meskipun keuntungan efisiensi ini menjadi poin utama, penelitian ini juga mengakui perlunya evaluasi menyeluruh terhadap dampak penghapusan data terhadap akurasi dan kredibilitas model klasifikasi. Pilihan ini mengundang refleksi mendalam terhadap kriteria pemilihan outlier yang dihapus dan merangsang diskusi tentang peningkatan proses pengumpulan data untuk menghindari kemunculan outlier di masa depan. Dengan demikian, langkah ini diarahkan untuk memberikan kontribusi positif dalam mengoptimalkan analisis keamanan berkendara menggunakan data sensor gerak smartphone dalam skala besar. Secara keseluruhan, penelitian ini bertujuan untuk menjelajahi dan memberikan kontribusi dalam bidang keselamatan lalu lintas melalui aplikasi teknologi inovatif dan metodologi machine learning. Tujuannya adalah menciptakan pendekatan yang lebih proaktif terhadap deteksi perilaku mengemudi, membuka jalan untuk pengambilan keputusan cerdas dalam hal lalu lintas, dan menciptakan masa depan jalan raya yang lebih aman.

1.2. Rumusan Masalah

Latar belakang diatas menghasilkan rumusan masalah sebagai berikut:

a. Berapa nilai performa (akurasi, presisi, dan *recall*) yang dihasilkan oleh model klasifikasi *Random Forest* dan *Support Vector Machine*?
b. Fitur apa saja yang paling berpengaruh terhadap hasil performa klasifikasi *machine learning* untuk mendeteksi karakteristik berkendara

1.3. Batasan Masalah

Batasan pada penelitian yang dilakukan adalah sebagai berikut:

a. Menggunakan metodologi klasifikasi *Machine Learning*
b. Dataset penelitian adalah data yang telah disediakan oleh salah satu penyedia layanan transportasi daring tahun 2021.
c. Moda transportasi yang digunakan dalam dataset tersebut adalah Mobil, namun tidak ada penjelasan jenis dan tipe mobil yang digunakan.
d. Dataset adalah data perjalanan yang dilakukan di Indonesia, diambil secara acak sehingga tidak diketahui lokasi daerah/ kota pasti data perjalanan tersebut diambil.
e. Kondisi perjalanan seperti jalan berlubang, macet, dan perjalanan lancar tidak dapat diukur dalam dataset penelitian ini.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah:

a. Mengetahui Fitur apa saja yang paling berpengaruh terhadap deteksi karakteristik berkendara.

b. Mengetahui nilai performa yang dihasilkan tiap model klasifikasi dengan membandingkan algoritma Algoritma *Random Forest* dan *Support Vector Machine* berdasarkan evaluasi akurasi, presisi, *recall*.

c. Mengetahui karakteristik berkendara pengemudi di Indonesia.

1.5. Manfaat Penelitian

Manfaat penelitian ini adalah:

a. Dapat menjadi pedoman pengembangan penelitian dalam menganalisis karakteristik berkendara di Indonesia.

b. Berkontribusi secara ilmiah terhadap pemanfaatan pengolahan data smartphone untuk keselamatan berkendara yang merupakan salah sasaran *intelligent transport system* (ITS)

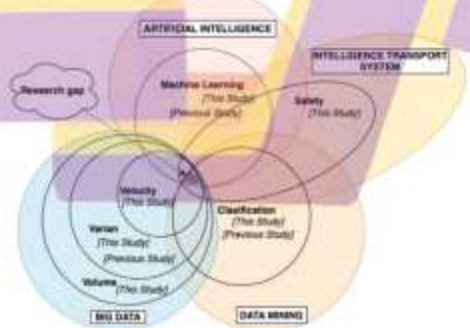
c. Hasil klasifikasi cara berkendara diharapkan dapat menjadi solusi terhadap keamanan berkendara sebagai *early warning* terjadinya kecelakaan lalu lintas.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian tentang data mining di berbagai ilmu cukup banyak dilakukan, salah satunya pada bidang keselamatan berkendara, beberapa metode data mining yang sering digunakan, yaitu klasifikasi dimana metode ini bertujuan untuk memprediksi suatu hasil dari sebagian variabel atau seluruh variabel untuk memprediksi sebuah kelas yang berisi dua nilai atau lebih. Penelitian ini memadukan beberapa teknologi untuk mendapat gap yang dapat diusung, beberapa teori yang diadaptasi diantaranya teori *Artificial intelligence*, *Big Data*, Data Mining dan *Intelligent transport system*. Pemetaan teori dalam penelitian ini adalah sebagai berikut:



Gambar 2. 1. Research Gap

Gambar 2.1 merupakan hasil pemetaaan dan pendalam teori, peneliti menemukan Gap di antara irisan teori yang saling berhubungan. Dalam melanjutkan penelitian ini, eksplorasi lebih lanjut pada bidang *Artificial Intelligence* (AI), khususnya *Machine Learning* dan *Safety*, dapat mengarah pada pengembangan model yang lebih adaptif dan kustomisasi algoritma untuk prediksi keamanan berkendara dengan tingkat akurasi yang lebih tinggi. Dalam konteks Big Data, pemahaman lebih mendalam tentang *Volume, Velocity,* dan *Variability* dapat memberikan wawasan tentang penanganan dan analisis data yang besar secara lebih efektif. Penelitian lebih lanjut pada *Classification* dalam Data Mining dapat memfokuskan pada identifikasi faktor-faktor yang paling berkontribusi terhadap karakteristik mengengemudi, dan bagaimana algoritma klasifikasi dapat mengekstraksi pola-pola ini. Integrasi AI, *Big Data*, dan *Data Mining* dalam *Intelligent Transport System* (ITS) dapat memberikan solusi cerdas yang lebih efektif dalam manajemen dan meningkatkan keselamatan lalu lintas. Sementara itu, identifikasi *Research Gap* dapat memperjelas area penelitian yang belum sepenuhnya dijelajahi, seperti dampak faktor-faktor tertentu pada kecelakaan lalu lintas atau pengembangan model yang dapat menangani dinamika perubahan dalam pola lalu lintas. Keseluruhan, pemahaman mendalam terhadap seluruh ekosistem teknologi ini akan memberikan kontribusi penting pada pengembangan solusi yang lebih canggih dan efektif untuk meningkatkan keselamatan berkendara. Selain pemetaan teori penelitian ini juga merujuk pada penelitian penelitian sebelumnya diantaranya beberapa penelitian di bawah ini.

Sebuah penelitian deteksi cara mengemudi dilakukan oleh (Júnior, 2017), dalam penelitian peneliti melakukan evaluasi kuantitatif kinerja Klasifikasi algoritma *Machine Learning Algorithms* diantaranya *Support Vector Machines* (SVM), *Random Forest* (RF), *Artificial Neural Networks* (ANN), dan *Bayesian Network* (BN) untuk mendeteksi cara mengemudi dengan menggunakan data dari 4 features sensor smartphone seperti (akselerometer, akselerasi linier, magnetometer, dan *gyroscope*) (Lu, 2018).

(Nguyen, 2018) Melakukan penelitian tentang *Human Activity Recognition* (HAR) atau dalam bahasa dapat disebut dengan aktivitas kegiatan manusia, dimana data yang digunakan adalah data sensor accelerometer, dan sensor suara pada *smartphone*. Penelitian ini melakukan klasifikasi aktivitas seseorang dengan kondisi, sedang duduk sedang berjalan atau sedang berlari. Peneliti membandingkan performa *classifier* yang digunakan dalam penelitian ini diantaranya, *Multi-Layer Perceptron, Decision Tree* dan SVM, Peneliti juga mengangkat topik *imbalance dataset,* dimana dari data yang dijelaskan memiliki perbandingan data sedang duduk (26.4%), sedang lari (1,9%), sedang jalan normal (45.91%), dari masalah tersebut peneliti mengusulkan metode *oversampling.* Hasil penelitian menyebutkan bahwa metode *oversampling* dapat mengatasi permasalahan *imbalance dataset,* dengan *classifier* terbaik dalam penelitian tersebut adalah *Multi-Layer Perceptron.* Perbandingan performa klasifikasi sekitar 15% untuk nilai FI *Multi-Layer Perceptron* dengan penerapan *oversampling*.

(Aljaafreh., 2022) Melakukan penelitian tentang aktivitas manusia, dimana data yang digunakan adalah data sensor *smartphone* mengemudi berdasarkan

sensor 2-axis accelerometer (x,y). Penelitian ini melakukan klasifikasi mengemudi secara normal dan agresif dengan algoritme *fuzzy*, hasil dari penelitian ini bahwa Sistem Klasifikasi *Fuzzy* mampu menyajikan data grafik perjalanan berdasarkan klasifikasi yang diharapkan. Dalam penelitian juga dinyatakan bahwa sensor accelerometer saja tidak cukup akurat dalam deteksi cara mengemudi, sehingga perlu dilakukan pengujian menggunakan sensor lain seperti Sensor *Gyroscope*.

(Yu., 2017) Mengusulkan suatu sistem dengan metode *machine learning* untuk mendeteksi dan mengidentifikasi tipe spesifik dari perilaku mengemudi yang tidak normal berdasarkan data 3-*axis gyroscope* dan 3-*axis accelerometer* pada sensor *smartphone*. Penelitian menggunakan 20 sampling perjalanan. pada penelitian ini yang diukur adalah cara mengemudi tidak aman berdasarkan 6 kriteria gerakan seperti berkelok-kelok (*Weaving*), Membanting stir (*Swerving*), Tergelincir (*Side Slipping*), Putar balik dengan radius sempit (*Fast* U-*turn*), Putar balik dengan radius lebar (*Turning with* a *wide radius*), dan Pengereman mendadak (*Sudden braking*). Metode klasifikasi yang digunakan adalah SVM dan *Neural Networks*. Berdasarkan hasil penelitian yang didapatkan Sensor *Gyroscope* memiliki tingkat akurasi *Classifier* paling baik diantara sensor lainya dalam melakukan deteksi cara berkendara.

(Jahangiri & Rakha, 2015) melakukan penelitian metode monitoring berbasis *machine learning* dengan metode klasifikasi multi class untuk mengidentifikasi mode transportasi (mobil, sepeda, bus, berjalan, dan berlari) metode klasifikasi yang digunakan adalah KNN, SVM dan RF, dengan mengolah data sensor *smartphone* (accelerometer, *gyroscope*, dan *light* sensors). Dari data

sensor yang disediakan dilakukan proses ekstraksi fitur dengan hasil akhir ditemukan 165 fitur. Berdasarkan hasil pengujian, Metode classifier RF menghasilkan kinerja terbaik. karakteristik classifier RF sangat diuntungkan dalam mengolah dataset dengan banyak fitur.

2.2. Keaslian Penelitian

Tabel 2. 1. Matriks literatur review dan posisi penelitian

Analisis Perbandingan Algoritma Random Forest dan SVM pada Data Sensor Gerak Smartphone untuk Model Klasifikasi Keamanan Berkendara

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Smartphone Inertial Measurement Unit Data Features for Analyzing Driver Driving Behavior	K. Kanwal F. Rustam R. Chaganti A. D. Jurcut I. Ashraf IEEE Sensors Journal, vol. 23, no. 11 2023	Studi ini bertujuan untuk mengeksplorasi penggunaan berbagai fitur dan kombinasi fitur, serta membandingkan kinerja model klasifikasi biner dan multi kelas. Selain itu, studi ini juga bertujuan untuk mengevaluasi kinerja model ML dan DL dalam prediksi perilaku pengemudi. Studi ini juga mengidentifikasi batasan dalam penggunaan dataset yang relatif kecil dan jumlah fitur yang terbatas, serta memberikan saran untuk pengembangan penelitian selanjutnya, seperti pengumpulan dataset yang lebih besar dan eksplorasi metode non-ML lainnya.	Beberapa model ML yang disebutkan adalah Random Forest (RF) dan Extreme Gradient Boosting Machines (XGBoost), yang mencapai akurasi 100% dalam klasifikasi multi kelas untuk jumlah data sebanyak 3084 data	Pada penelitian ini hanya menggunakan dua jenis sensor gerak, yaitu accelerometer dan gyroscope dari smartphone. Hal ini mungkin mengurangi akurasi dalam mengklasifikasikan perilaku mengemudi. Kelemahan penelitian ini adalah pengambilan data sensor yang hanya dilakukan pada satu jenis kendaraan dan dataset diambil dari Kaggle.	Dataset yang digunakan memiliki data kecepatan dan waktu. Rute perjalanan lebih dari satu perjalan. Metode klasifikasi yang digunakan membandingkan RF dan SVM

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
2	Performance Evaluation of Driving Behavior Identification Models through CAN-BUS Data	M. N. Azadani A. Boukerche IEEE Access 2020	Menganalisis dan mengidentifikasi perilaku berkendara melalui pemrosesan dan analisis data sensor yang dikumpulkan melalui teknologi CAN-BUS. mengimplementasikan berbagai model pembelajaran mesin dan pembelajaran mendalam yang didorong oleh data serta melaporkan hasil (cross-vulidation) menggunakan dataset berkendara naturalistik.	hasil penelitian menunjukkan model-model deep learning, seperti CNN dan DeepConvLSTM, memiliki kinerja yang baik. model machine learning klasik seperti Random Forest, Decision Tree, dan KNN menunjukkan kinerja yang **lebih rendah**.	kurangnya informasi tentang metode yang digunakan untuk pengambilan data dari mobil dan pemrosesan data tersebut. hanya menggunakan **satu dataset spesifik, yaitu dataset berkendara naturalistik**, yang dapat membatasi generalisasi temuan ke situasi berkendara yang berbeda. Tidak ada perbandingan langsung dengan pendekatan lain atau studi sebelumnya yang dapat memberikan pemahaman lebih mendalam tentang keunggulan relatif dari model yang diusulkan.	Dataset yang digunakan di ambil dari perjalanan menggunakan Mobil Rute perjalanan lebih dari satu perjalan. Metode klasifikasi yang digunakan membandingkan RF dan SVM
3	A Comparative Study of Aggressive Driving Behavior Recognition Algorithms Based on Vehicle	Y. Ma Z. Zhang S. Chen Y. Yu K. Tang IEEE Access 2018	Tujuan penelitian ini adalah mengembangkan metode pengenalan perilaku mengemudi agresif yang efektif dan dapat diandalkan, serta meningkatkan keselamatan jalan dan mengurangi kecelakaan dengan menggunakan data gerakan kendaraan yang tercatat melalui sensor smartphone.	Kesimpulan dari penelitian ini adalah bahwa metode pengenalan perilaku mengemudi agresif berdasarkan data gerakan kendaraan melalui sensor smartphone dapat berhasil. Algoritma seperti Gaussian mixture model (GMM), partial least squares	Perluas dataset untuk meningkatkan keakuratan dan generalisasi Lakukan evaluasi dan perbandingan lebih lanjut terhadap metode pengenalan Integrasi data dari sensor lain seperti GPS, radar kendaraan, atau pelacak mata. Sesuaikan ambang batas dan parameter algoritma untuk meningkatkan	Dataset yang digunakan memiliki data Gyroscope, Accelerometer, kecepatan dan waktu. Rute perjalanan lebih dari satu perjalan. Metode klasifikasi yang digunakan membandingkan

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	Motion Data			regression (PLSR), wavelet transformation, dan support vector regression (SVR) dapat mengenali perilaku mengemudi agresif dengan baik. Penggunaan dataset multi-sumber dan pemilihan ambang batas yang tepat dapat meningkatkan hasil pengenalan. Penelitian selanjutnya dapat memperluas pemahaman tentang perilaku mengemudi agresif dengan melibatkan data yang lebih beragam, seperti perangkat GPS dan radar kendaraan.	akurasi. Uji coba di lingkungan lapangan yang lebih luas. Dengan menerapkan saran-saran ini, penelitian ini dapat terus ditingkatkan dan memberikan kontribusi yang lebih besar dalam pengenalan dan pengurangan perilaku mengemudi agresif serta meningkatkan keselamatan jalan.	RF dan SVM
4	Driving Behaviour Analysis Using Machine and Deep Learning Methods for Continuous Streams of Vehicular Data	N. Peppes T. Alexakis E. Adamopoulou K. Demestichas Institute of Communication and Computer	Menyajikan platform terintegrasi yang menggunakan teknologi terkini dan memberikan wawasan tentang penggunaan metode tersebut dalam analisis perilaku pengemudi. metode yang digunakan adalah kombinasi antara metode ML Logistic Regression, SVMm Random Forest dan DL MLP RNN. Data yang digunakan adalah "vast streams of vehicular data"	Hasil penelitian menunjukkan bahwa metode klasifikasi ML SVM pada data sensor gerak smartphone dapat mengenali perilaku mengemudi dengan akurasi 100%.	Penelitian hanya mempertimbangkan perilaku mengemudi dalam tiga kategori (berhenti, melaju dan berbelok), yang mungkin kurang representatif dari perilaku mengemudi yang lebih kompleks. Kelemahan penelitian ini adalah pengambilan data sensor yang hanya dilakukan pada satu jenis	Dataset yang digunakan memiliki data Gyroscope, Accelerometer, kecepatan dan waktu. Rute perjalanan lebih dari satu perjalan. Metode klasifikasi yang digunakan membandingkan RF dan SVM

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
		Systems (ICCS) 2021			kendaraan dan satu rute perjalanan sehingga hasilnya tidak dapat digeneralisasi pada situasi lain.	
5	Feature Selection Based on Variance Distribution of Power Spectral Density for Driving Behavior Recognition	H. Nassuna O. S. Eyobu J. -H. Kim D. Lee, *IEEE Conference on Industrial Electronics and Applications (ICIEA)* *2019*	Bertujuan untuk mendeteksi dan mengenali perilaku mengemudi yang abnormal. Penelitian ini merupakan bagian dari upaya mencapai keselamatan dalam sistem transportasi cerdas (ITS). Penelitian ini mengusulkan pendekatan ekstraksi fitur dan menggunakan fitur-fitur yang diekstraksi tersebut untuk melatih model deep learning yang digunakan untuk mengenali perilaku mengemudi yang abnormal.	Eksperimen dilakukan menggunakan jaringan syaraf tiruan untuk menguji efisiensi pendekatan ekstraksi fitur yang diusulkan. Hasilnya menunjukkan akurasi sebesar 91,0% dapat dicapai dengan data accelerometer. Akurasi ditingkatkan menjadi 96,1% dengan menggabungkan data accelerometer dengan data gyroscope.	Penelitian ini menggunakan data dari accelerometer dan gyroscope untuk mengenali perilaku mengemudi. Namun, tidak dijelaskan secara detail tentang karakteristik data yang digunakan, seperti jumlah sampel, variasi kondisi jalan, atau variasi pengemudi. Memperluas variasi data dan melibatkan pengemudi dengan berbagai tingkat keahlian dan perilaku mengemudi dapat	Dataset yang digunakan memiliki data Gyroscope, Accelerometer, kecepatan dan waktu. Rute perjalanan lebih dari satu perjalan. Metode klasifikasi yang digunakan membandingkan RF dan SVM

2.3. Landasan Teori

2.3.1. Kecerdasan Buatan

Kecerdasan buatan adalah kecerdasan yang ditambahkan kepada suatu sistem yang bisa diatur dalam konteks ilmiah atau bisa disebut juga intelegensi artifisial. Rekayasa perangkat lunak telah lama menjadi domain yang menarik bagi penelitian kecerdasan buatan dan pembelajaran mesin. Metode-metode ini meningkatkan produktivitas pengolahan data, kualitas memproses data, dan mengurangi komputasi waktu yang begitu banyak dibutuhkan untuk memproses data (Khomh., 2018).

2.3.1.1. Definisi Kecerdasan Buatan

Definisi kecerdasan buatan sebagai suatu sistem yang memenuhi salah satu dari empat kategori sebagai berikut:

*a. **Thinking Humanly***

Mesin menjalankan proses berpikir seperti manusia dengan menirukan aktivitas manusia, seperti pengambilan keputusan, pemecahan masalah, dan pembelajaran. Pendekatan ini dikenal sebagai pendekatan kognitif. Manusia memiliki dua cara untuk berpikir, yaitu melalui introspeksi dan eksperimen. Introspeksi dilakukan dengan mempertanyakan dan membahas argumen untuk menemukan solusi ketika menghadapi suatu masalah.

*b. **Acting Humanly***

Sikap bersikap seperti manusia mencakup menirukan kebiasaan manusia dalam menyelesaikan masalah. Konsep ini diperkenalkan oleh

Alan Turing pada tahun 1950 melalui Tes Turing yang dirancang untuk menguji kemampuan manusia dalam mengidentifikasi mesin. Turing menguji tes ini dengan membuat manusia berkomunikasi dengan sebuah entitas mesin melalui *teletype*. Apabila dalam 5 menit manusia tidak dapat mengenali apakah entitas yang diinterogasi adalah manusia atau mesin, maka entitas tersebut lolos uji Tes Turing dan dapat dikatakan sebagai sistem cerdas. Namun demikian, entitas tersebut setidaknya harus memiliki kapabilitas mengenali suara, mengerti bahasa manusia, melakukan sintesis suara, representasi pengetahuan, menanggapi secara otomatis, pembelajaran mesin (*machine learning*), melakukan pertimbangan dan pengambilan keputusan.

c. ***Thinking Rationally***

Berpikir secara rasional merupakan studi yang memungkinkan komputer dapat membuat persepsi, memberikan tanggapan, dan menyikapi suatu permasalahan. Cara mencapai hal tersebut bagi sistem kecerdasan buatan adalah dengan cara memodelkan bagaimana manusia seharusnya berpikir dan menanggapi pada kondisi ideal.

d. ***Acting Rationally***

Bertingkah laku secara rasional) berarti bagaimana membangkitkan sikap yang rasional dalam hal proses komputasi. Bersikap rasional adalah bersikap untuk mencapai tujuan dengan tetap mempertimbangkan kondisi dan pemahaman diri. Agen cerdas sebagai sistem komputer dalam hal ini berperan melakukan pengambilan keputusan terbaik dengan tetap

mempertimbangkan situasi. contohnya seorang agen cerdas memainkan permainan catur, harapannya adalah bahwa agen cerdas tersebut akan membuat langkah-langkah terbaik untuk memenangkan pertandingan.

2.3.1.2. Penerapan Kecerdasan Buatan

Secara garis besar, kecerdasan buatan terbagi ke dalam dua paham pemikiran yaitu kecerdasan konvensional dan Kecerdasan *Komputasional Computational Intelligence* (CI). Kecerdasan konvensional kebanyakan melibatkan metode-metode yang sekarang diklasifikasikan sebagai pembelajaran mesin, yang ditandai dengan formalisme dan analisis statistik. Kecerdasan buatan memiliki berbagai penerapan sebagai berikut (Kuritsyn., 2018):

a. ***Fuzzy Logic* (FL)**

Mesin menggunakan teknik ini untuk menyesuaikan diri dengan situasi tertentu dan membuat keputusan yang tidak hanya didasarkan pada logika Boolean (Benar atau Salah). Teknik ini memungkinkan mesin untuk memberikan nilai di antara keduanya, seperti setengah benar dan setengah salah. Fuzzy Logic memiliki kelebihan yaitu fleksibilitas.

b. **Komputasi *Evolusioner***

Teknik ini mengadopsi proses evolusi alami di mana individu-individu terbaik akan bertahan hidup dan individu yang kurang unggul akan punah dalam populasi. *Algoritme* Genetik adalah contoh dari *evolutionary computing*. Dalam *algoritme* genetika, seleksi individu yang unggul dilakukan dengan menggunakan uji kecocokan (*fitness*). Operator genetik

dalam algoritme genetik dibagi menjadi dua jenis, yaitu mutasi dan rekombinasi.

c. ***Machine Learning***

Teknik Machine Learning memungkinkan sistem untuk belajar dari data dan menghasilkan hasil yang relevan daripada menggunakan pemrograman langsung. Pembelajaran mesin melibatkan penggunaan berbagai algoritma untuk belajar dari data dengan cara mendeskripsikan data dan memprediksi hasil secara iteratif (Zhu., 2019). Namun, hasil dari pembelajaran mesin dapat berubah tergantung pada kualitas data yang digunakan. Model Machine Learning diperoleh melalui pelatihan algoritma pembelajaran mesin menggunakan data. Setelah dilatih, model tersebut dapat menghasilkan output ketika diberi input. Contoh penggunaan Machine Learning adalah rekomendasi produk dalam e-commerce, di mana rekomendasi produk biasanya berdasarkan riwayat jelajah, riwayat pembelian, dan barang yang sebelumnya dilihat oleh pengguna.

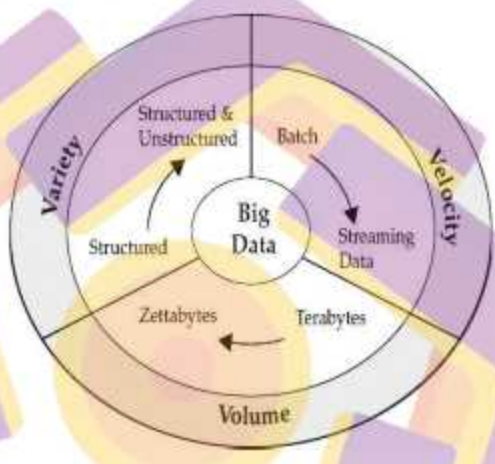
2.3.2. *Big Data*

Big data adalah istilah umum untuk segala kumpulan himpunan data dalam jumlah yang sangat besar dan komplek sehingga piranti pemrosesan data tradisional tidak mampu menganalisanya, akan tetapi semakin besar data yang dikumpulkan dapat menghasilkan informasi baru yang belum pernah diketahui sebelumnya. *Big data* adalah bidang yang sedang berkembang pesat yang melibatkan analisis dataset yang besar dan kompleks untuk mengekstrak wawasan dan value terhadap sebuah

informasi. *Big data* mencakup berbagai teknologi dan teknik, termasuk penyimpanan, pemrosesan, dan analisis data

2.3.2.1. Karakteristik Big Data

Big data memiliki 3 karakteristik, berikut adalah gambaran karakteristik big data secara umum (Bhat., 2021)



Gambar 2. 2.Karakteristik Big data.

Gambar 2.2 merupakan karakteristik dari *big data*, berikut adalah komponen yang ada dalam karakteristik big data;

a. Volume

Volume dimana sebuah *big data* merupakan sekumpulan data dengan *volume* yang sangat tinggi, dalam penelitian ini dataset yang digunakan memiliki jumlah data sekitar 16 juta baris data yang akan diolah dengan total keseluruhan data adalah kurang lebih 2 GB, data sebanyak ini sangatlah tidak efisien jika dianalisa menggunakan cara tradisional.

b. *Velocity*

Velocity atau kecepatan, dimana sebuah aliran data harus dapat menerima dan memproses data dengan kecepatan tinggi, serata mampu mengolah data secara *real time*. Pada tahapan penelitian ini metode *velocity* masih belum digunakan karna teknik *learning* masih menggunakan *batch data*, akan tetapi untuk tahapan implementasi data *streaming* akan langsung di analisa.

c. *Variety*

Variety atau variasi adalah banyaknya jenis data yang beredar berdasarkan bentuk dan jenisnya, dalam data tradisional umumnya data yang dikumpulkan berupa data terstruktur dan *fit*, akan tetapi pada teknologi *big data* data yang didapatkan umumnya tidak terstruktur dan berasal dari berbagai sumber data. pada penelitian ini varian data terlihat sangat jelas karena sensor yang digunakan adalah beberapa sensor perjalanan seperti, sensor *gyroscope*, sensor *accelerometer*, dan sensor GPS. ketiga sensor ini memiliki format penulisan data yang berbeda-beda satu sama lain.

2.3.2.2. Arsitektur Big data

Selain karakteristik dan jenis data yang berbeda teknologi *Big data* juga memiliki perbedaan pada desain arsitekturnya. Berikut adalah penjelasan arsitektur tradisional dan arsitektur *big data* (Seo., 2018).

a. Arsitektur Tradisional

Gambar 2. 3.Arsitektur Tradisional

Gambar 2.3 merupakan Arsitektur Tradisional. Arsitektur tradisional menunjukan dua sumber data yang menggunakan teknik integrasi ETL *(Extract Transform Load)* / *Change Data* Capture untuk mentransfer data ke dalam DBMS *Data Storage*, setelah itu data dapat dianalisis kemudian data dapat ditampilkan.

b. Arsitektur Big data

Gambar 2. 4. Arsitektur Big Data.

Gambar 2.4 adalah arsitektur *big data* fokus pada data *real time*, pada bagian pemrosesan digunakan teknik *Map Reduce*, teknik ini memungkinkan untuk melakukan proses *filter* data yang digunakan secara spesifik, setelah data ditemukan maka akan dianalisis langsung, data yang tidak terstruktur akan disimpan pada *data warehouse* (Santosa & Umam, 2018).

2.3.3. *Data Mining*

Data mining adalah bidang ilmu interdisipliner yang menggabungkan teknik-teknik pembelajaran mesin atau Machine Learning, pengenalan pola (Pattern Recognition), statistik, database, dan visualisasi untuk mengekstrak informasi atau pengetahuan penting dari suatu dataset berukuran besar dengan teknik tertentu. Tujuan utama dari data mining adalah untuk menemukan pola-pola yang tersembunyi di dalam data dan kemudian menerapkan pengetahuan ini untuk tujuan yang bermanfaat. Informasi yang dihasilkan dari proses data mining dapat digunakan untuk memprediksi hasil yang mungkin terjadi di masa depan atau membantu dalam pengambilan keputusan (Seo., 2018). Proses data mining melibatkan beberapa tahapan, seperti pemilihan data yang relevan, preprocessing data untuk menghilangkan data yang tidak penting atau mengisi nilai yang hilang, pemilihan algoritme yang tepat untuk menganalisis data, dan interpretasi hasil untuk mendapatkan pengetahuan yang berharga. Algoritme data mining yang umum digunakan termasuk klasifikasi, clustering, regresi, asosiasi, dan analisis tren. Contoh penerapan data mining yang sering kita jumpai adalah ketika suatu perusahaan ingin meningkatkan penjualan produknya. Dalam hal ini, data mining dapat digunakan untuk mengidentifikasi pola pembelian konsumen, menemukan asosiasi antara produk yang dibeli, dan memprediksi produk mana yang mungkin diminati oleh konsumen di masa depan. Dengan informasi ini, perusahaan dapat mengembangkan strategi pemasaran yang lebih efektif untuk meningkatkan penjualan produknya

2.3.3.1. Fungsi *data mining*

Data mining dapat membantu dalam pengambilan keputusan dengan memproses dan menganalisis data besar yang kompleks dan heterogen. Beberapa teknik data mining yang umum digunakan diantaranya klasifikasi, regresi, pengelompokan, dan asosiasi seperti pada gambar berikut.

Gambar 2. 5. Metode Data Mining

Gambar 2.5 merupakan *metode data mining* Beberapa fungsi yang dapat dilakukan menggunakan metode data mining seperti pada gambar 2.5, antara lain (Bachhety., 2020):

a. Klastering

Metode *Clustering* memiliki fungsi untuk mengelompokan objek dalam beberapa kelompok berdasarkan kemiripan antara beberapa objek, dimana dalam masing masing *Clustering* / kelompok objek saling tidak mirip, karakteristik *Clustering* tidak memerlukan data pelatihan yang sudah diberi label.

b. Klasifikasi

Metode klasifikasi memiliki fungsi melakukan pengelompokan objek berdasarkan kelompok yang sudah sudah ada, metode klasifikasi berbeda dengan *Clustering*, karakteristik metode klasifikasi membutuhkan data latih yang sudah diberi label atau kelas.

c. Regresi

Metode regresi memiliki fungsi hampir sama seperti metode Klasifikasi, yakni membutuhkan data latih yang telah diberi label. Perbedaan antara metode regresi dan klasifikasi berada pada outputnya, dimana *output* metode klasifikasi adalah nilai diskrit, sedangkan Metode regresi adalah nilai kontinyu.

d. Asosiasi

Metode Asosiasi memiliki fungsi melakukan asosiasi antara objek dalam suatu *dataset* dengan menghitung beberapa kali pada data yang mengandung dua item atau lebih yang saling berhubungan

2.3.3.2. Metode Pelatihan Data Mining

Secara garis besar metode pelatihan yang digunakan dalam teknik-teknik data mining dibedakan ke dalam tiga pendekatan (Santosa., 2018), yaitu;

a. Supervised Learning

i. Pembelajaran dengan *dataset* yang memiliki *targeted/label/class*

ii. Sebagian besar algoritme *data mining* (*estimation, prediction/forecasting, classification*) adalah *supervised learning*

iii. Algoritme melakukan proses *learning* berdasarkan nilai dari variabel *targeted* yang terasosiasi dengan nilai dari variabel predictor

b. Semi Supervised Learning

i. *Semi-supervised learning* adalah metode *data mining* yang menggunakan data dengan label dan tidak berlabel sekaligus dalam proses pembelajarannya

ii. Data yang memiliki *class* digunakan untuk membentuk model, data tanpa label digunakan untuk membuat batasan antar *class*

c. Unsupervised Learning

i. Algoritme *data mining* mencari pola dari semua *variabel* (*attribute*)

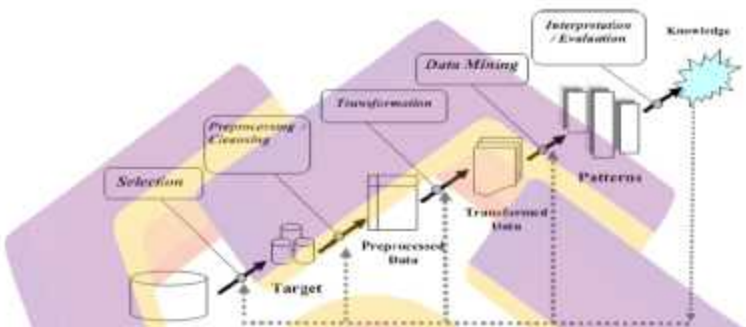
ii. Variable (*attribute*) yang menjadi target/label/class tidak ditentukan (tidak ada)

iii. Algoritme *clustering* adalah algoritme *unsupervised learning*

2.3.3.3. Proses Data mining

Penemuan pengetahuan melalui analisis data memiliki tantangan-tantangan utama yang dapat diatasi melalui pendekatan analisis data. Beberapa tantangan utama dalam analisis data, seperti data yang tidak terstruktur, keterbatasan perangkat keras dan perangkat lunak, serta kesulitan dalam memilih dan mengintegrasikan data dari berbagai sumber. Selain itu, penelitian ini juga membahas tentang pendekatan cerdas dalam analisis data (Al-Janabi & Samaher, 2021). Secara umum tahapan pada data mining terkait pada proses Knowledge Discovery and Data Mining (KDD). KDD adalah proses yang dibantu oleh

komputer untuk mengenali dan menganalisa sejumlah besar himpunan data dan mengekstrak informasi dan pengetahuan yang berguna, salah satu tahapan dalam keseluruhan proses KDD adalah data mining itu sendiri. Proses KDD secara garis besar dapat digambarkan sebagai berikut



Gambar 2. 6. Proses Knowledge Discovery and Data Mining.

Gambar 2.6 mengilustrasikan Proses *Knowledge Discovery* and *Data Mining* (KDD). Proses ini merupakan serangkaian langkah sistematis yang dilakukan untuk mengidentifikasi pola, informasi, atau pengetahuan yang berharga dari suatu dataset. Proses *Knowledge Discovery* and *Data Mining* memiliki peran penting dalam mengubah data mentah menjadi informasi yang berarti, memungkinkan pengguna untuk memahami pola tersembunyi, tren, atau pengetahuan yang dapat digunakan untuk mendukung pengambilan keputusan yang lebih baik.

2.3.4. *Intelligent Transport system*

ITS atau Sistem Transportasi Cerdas adalah gabungan beberapa teknologi termasuk penentuan posisi, komunikasi, sistem informasi, dan kontrol elektronik (Engelbrecht., 2015). Dalam hal teknologi pendukung ITS, GPS digunakan untuk

menentukan posisi, sementara GIS (Sistem Informasi Geografis) digunakan sebagai teknologi sistem informasi (Luo., 2011). Sistem navigasi ITS dapat dikelompokkan menjadi empat jenis: *Autonomous* ITS, *Fleet Management* ITS, *Advisory* ITS, dan *Inventory* ITS (Rinaldi., 2019). Dengan memanfaatkan teknologi informasi terkini, ITS menyatukan faktor manusia, jalan, dan kendaraan untuk menciptakan sistem transportasi yang lebih cerdas. Tujuan dari ITS adalah penerapan teknologi maju terhadap sarana transportasi agar lebih Aman, efisien, Berkemban lebih baik, dan ramah lingkungan (Seliverstov., 2019).

2.3.4.1. Ruang lingkup ITS

Sistem Autonomous ITS mencakup teknologi penentuan posisi dan peta elektronik yang terpasang pada kendaraan, yang bertujuan untuk meningkatkan kemampuan navigasi pengemudi. Sistem ini tidak berkomunikasi dengan sistem luar kendaraan kecuali jika menggunakan GPS untuk menentukan posisi, yang memerlukan antena untuk menerima sinyal GPS (Rinaldi., 2019). Telah dilakukan beberapa implementasi ITS.

a. Fleet Management

ITS berfungsi untuk mengelola kendaraan dari pusat pengontrol (*dispatch center*) melalui hubungan komunikasi. Dalam sistem ini kendaraan-kendaraan yang bersangkutan diperlengkapi dengan sistem penentuan posisi dan umumnya mereka tidak diperlengkapi dengan sistem peta elektronik. Kendaraan-kendaraan tersebut melaporkan posisinya ke pusat pengontrol sehingga pusat pengontrol mempunyai kemudahan untuk mengelola pergerakan kendaraan tersebut. Disamping memberikan

instruksi-instruksi mengenai pengarahan, pusat pengontrol juga bertanggung jawab memberikan informasi-informasi yang diperlukan oleh pengemudi kendaraan seperti informasi cuaca dan keadaan lalu lintas.

b. *Advisory* ITS

System ini menggabungkan aspek penentuan posisi dan sistem peta elektronik dari sistem *autonomous* ITS dengan aspek komunikasi dari arsitektur sistem *fleet management* ITS, Sistem *advisory* ITS adalah *autonomous* dalam artian bahwa sistem ini tidak di kontrol oleh suatu pusat pengontrol (*dispatch center*), tetapi pada saat yang sama sistem ini merupakan bagian dari armada kendaraan yang mendapat pelayanan dari pusat informasi lalu lintas. Pada beberapa sistem *advisory* ITS, kendaraan – kendaraan tertentu berdiri sendiri sebagai *traffic probes*, yang memberikan kendaraan-kendaraan lainnya (yang tidak terdefinisikan oleh pusat informasi lalu lintas) informasi-informasi terbaru tentang kondisi lalu lintas dan cuaca.

c. *Inventory* ITS.

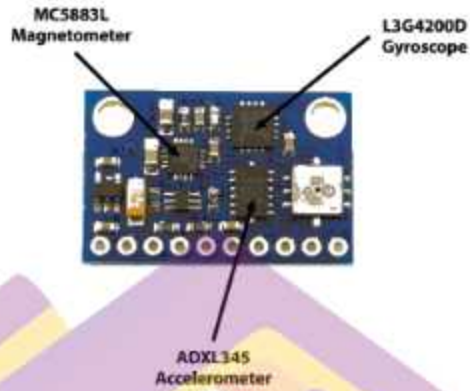
Sistem ini umumnya terdiri dari kendaraan yang berdiri sendiri dan dilengkapi dengan kamera video digital untuk mengumpulkan data terkait dengan jalan (termasuk koordinat dan waktu pengambilan data). Tujuannya adalah untuk keperluan inventarisasi jalan, pemeliharaan jalan, dan penyelidikan objek pengganggu lalu lintas. Kendaraan yang digunakan juga dilengkapi dengan perangkat penentu posisi, data logger, dan tampilan data dalam bentuk peta elektronik.

***d. Safe Driving* ITS**

Assistance for Safe Driving Adalah bentuk dari ITS yang sangat maju. Kendaraan dilengkapi dengan sejumlah sensor yang dapat mengarahkan pengemudi untuk berkendara dengan aman.

2.3.5. *Microelectromechanical Systems*

Microelectromechanical Systems (MEMS) adalah sebuah proses teknologi yang digunakan untuk membuat perangkat terintegrasi dalam ukuran kecil yang menggabungkan komponen mekanik dan listrik (Jia, 2018). Perangkat ini dibuat menggunakan sirkuit terpadu (IC) dengan beberapa teknik pemrosesan. Pada umumnya MEMS terdiri dari struktur mikro-mekanik, mikro-sensor, mikro-aktuator, dan mikro-elektronika, dimana semuanya terintegrasi ke dalam sebuah chip silikon. Perangkat MEMS ini memiliki kemampuan untuk merasakan, mengontrol, dan berjalan pada skala mikro, serta menghasilkan efek pada skala makro. sering pesatnya perkembangan teknologi, *smartphone* menjadi wadah bagi peneliti untuk penerapan teknologi mutakhir, salah satunya penerapan teknologi MEMS pada smartphone. MEMS menyederhanakan proses data analog menjadi data digital, dengan memanfaatkan media elektronik sehingga device MEMS dapat diimplementasikan pada *smartphone*. berikut salah satu perangkat MEMS yang diterapkan pada *smartphone*.



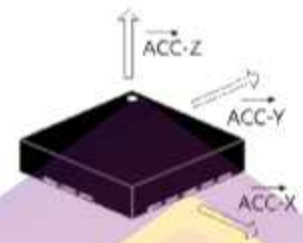
Gambar 2. 7. Perangkat MEMS Pada Smartphone.

Gambar 2.7 adalah bentu Perangkat MEMS Pada *Smartphone* Perangkat MEMS *smartphone* merupakan perangkat virtual yang menyediakan data masukan dari sekumpulan sensor-sensor fisik seperti accelerometer, *gyroscope*, dan *Global Positioning System*.

2.3.5.1. Accelerometer MEMS

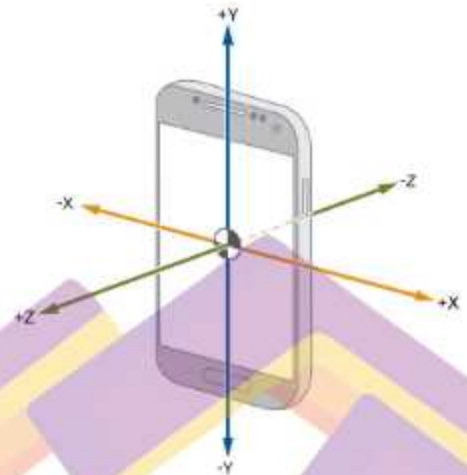
Accelerometer MEMS menggunakan sejenis pegas yang menerima gaya dan kemudian dikonversi menjadi perpindahan yang terukur. Berikutnya MEMS menggunakan struktur mikro dengan mengubah kapasitansi atau kristal mikroskopik yang akan menghasilkan tegangan bila tertekan. Dengan memanfaatkan percepatan statis, sensor Accelerometer dapat digunakan untuk mendeteksi kemiringan. Pengubahan mode landscape dan portrait pada layar smartphone juga memanfaatkan Accelerometer (Jia, 2018). Sedangkan pada percepatan linear, sensor pada aplikasinya dapat digunakan sebagai bagian dari peralatan navigasi. Saat ini terdapat 3 macam sensor Accelerometer, 1 sumbu, 2

sumbu (x-y), dan 3 Sumbu (x-y-z). Pada Gambar 2.8 merupakan pembagian 3 sumbu (x-y-z).



Gambar 2. 8. Sensor Accelerometer MEMS.

Gambar 2.8 adalah sensor accelerometer pada smartphone dapat mendeteksi orientasi perangkat melalui gerakan ke segala arah atau dengan menggoyangkan. Sensor ini mengukur percepatan perangkat pada tiga sumbu XYZ (kanan, kiri, atas, bawah, dan datar) sehingga memungkinkan aplikasi atau sistem untuk menentukan apakah smartphone sedang dalam orientasi berdiri (portrait) atau memanjang (*landscape*) dengan menggunakan data dari sensor ini.

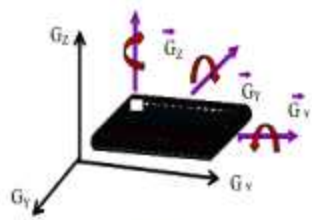


Gambar 2. 9. sip Kerja Accelerometer pada Smartphone

Gerakan sensor accelerometer di gambar 2.9. pada perangkat *Smartphone* yaitu dengan memiringkan (tilt) perangkat. Salah satu aplikasinya dalam perangkat *Smartphone* adalah ketika memiringkan perangkat saat display layar dalam keadaan portrait kemudian berubah menjadi *landscape*. Hasil pengukuran accelerometer menggunakan satuan SI (m/s2) dan pada perhitungannya dikurangi dengan percepatan gravitasi bersama dengan ketiga axis sensor.

2.3.5.2. Gyroscope MEMS

Gyroscope MEMS mengambil ide dari pendulum *Foucault* dan menggunakan elemen bergetar. *Chip* inilah yang dipakai pada *controller* dan *smartphone*. *Sensor Gyroscope* mengukur kecepatan sudut (rad/s) dalam 3 axis, yaitu *roll* (x), *pitch* (y), *yaw* (z).



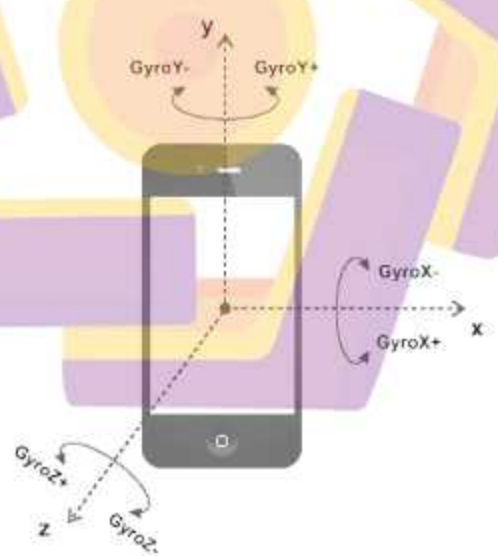
Gambar 2. 10. Sensor Gyroscope MEMS

3 axis *gyroscope* pada gambar 2.10. terdapat gimbal yang terpasang, dengan tiga gimbal, *gyroscope* dapat berputar di tiga sumbu namun rotor akan selalu tetap berputar di tiga sumbu namun rotor akan selalu tetap di posisinya

Prinsip rotor gyro adalah kekakuan dalam ruang atau inersia giroskopik. Hukum pertama *Newton* menyatakan jika gaya total suatu benda sama dengan nol. Rotor berputar dalam *gyroscope* mempertahankan sikap konstan dalam ruang selama tidak ada gaya yang mengubah gerakannya. Stabilitas ini meningkat jika rotor memiliki massa yang besar dan kecepatan. Karakteristik lain *gyroscope* adalah presesi. Presesi presesi. Presesi adalah gerakan memiringkan atau berputar terhadap sumbu *gyroscope* sebagai akibat gaya yang diterapkan. Ketika sebuah gaya diterapkan di tepi rotor *gyroscope*, maka rotor akan bergerak dalam arah yang stasioner, maka rotor akan bergerak dalam arah yang sama dengan gaya tersebut. Namun ketika rotor berputar, gaya yang sama mengakibatkan rotor bergerak berputar, gaya yang sama mengakibatkan rotor bergerak ke arah yang berbeda seolah-olah gaya diterapkan di olah gaya diterapkan di titik 90° di sekitar lingkaran dalam arah rotasi. Presesi titik 90° di sekitar lingkaran dalam arah rotasi. Perilaku *gyroscope* ditunjukkan dalam Persamaan (7.1)

$$\tau = \frac{dL}{dt} = \frac{d(I\omega)}{dt} = I\alpha \quad (1)$$

Sistem *gyroscope* pada smartphone berfungsi untuk mempertahankan atau mengukur orientasi perangkat, dengan menggunakan prinsip ketetapan momentum sudut, dan biasanya bekerja sama dengan accelerometer. *Gyroscope* terdiri dari sebuah roda berputar dengan piringan di dalamnya yang stabil. Alat ini umumnya digunakan pada robot, drone, dan alat canggih lainnya. Gyroscope pada smartphone memiliki sensor gyro yang bertugas untuk menentukan orientasi gerak dengan bertumpu pada roda atau cakram yang berotasi cepat pada sumbu. Sensor gyro juga berfungsi untuk mendeteksi gerakan perangkat sesuai dengan gravitasi atau mengukur gerakan pengguna.



Gambar 2. 11. Prinsip Kerja Sensor Gyroscope pada Smartphone.

Gambar 2.11 adalah Prinsip kerja *Sensor Gyroscope*, dimana untuk menggunakan gyro sensor, diperlukan kalibrasi menggunakan bandul sebelumnya untuk mendapatkan nilai faktor kalibrasi. Gyroscope dapat memberikan keluaran berupa kecepatan sudut dari tiga sumbu, yaitu sumbu x yang akan menjadi sudut phi (kanan dan kiri), sumbu y yang akan menjadi sudut theta (atas dan bawah), dan sumbu z yang akan menjadi sudut psi (depan dan belakang). Proses kalibrasi ini penting untuk memastikan akurasi pengukuran kecepatan sudut dari sensor.

2.3.5.3. Global Positioning System (GPS)

GPS atau Sistem Penentuan Posisi Global adalah sebuah sistem navigasi dan penentuan posisi yang dikelola oleh pemerintah Amerika Serikat. Fungsinya adalah memberikan informasi posisi, kecepatan tiga dimensi, dan waktu secara terus-menerus di seluruh dunia tanpa dipengaruhi oleh cuaca atau waktu tertentu, dan dapat digunakan oleh banyak orang secara bersamaan. Saat ini, GPS sangat populer di seluruh dunia dan telah banyak digunakan di Indonesia, terutama untuk aplikasi yang membutuhkan informasi tentang posisi.

1) ***Speedometer* GPS**

Speedometer GPS adalah perangkat pengukur kecepatan yang menggunakan perubahan data posisi koordinat bumi yang diperoleh dari satelit GPS yang diolah oleh prosesor menjadi informasi kecepatan. Kecepatan adalah besaran yang menunjukkan seberapa cepat benda berpindah. Satuan SI dari Kecepatan adalah m/s. Sedangkan jarak adalah angka yang menunjukkan seberapa jauh suatu benda berubah posisi. Satuan SI dari Jarak adalah meter (m). Adapun

waktu merupakan interval antara dua buah keadaan/kejadian, atau bisa merupakan lama berlangsungnya suatu kejadian. Satuan dari Waktu adalah sekon atau detik (Jahan, 2013). Tapi satuan dari kecepatan, jarak dan waktu bisa berubah tergantung pertanyaannya. beberapa parameter yang dapat diukur dalam *Speedometer* GPS adalah sebagai berikut

- Jarak

  Jarak yang ditempuh atau dalam matematika sering ditulis dengan s umumnya memiliki satuan Meter (m) dan Kilometer (km). Cara menghitung jarak tempuh adalah kecepatan dikali dengan waktu tempuh. Berikut ini persamaan jarak tempuh.

$$s = v \times t \quad (2)$$

- Waktu

  Waktu yang ditempuh atau dalam matematika sering ditulis dengan t umumnya memiliki satuan Detik (s). Cara Menghitung waktu tempuh adalah jarak tempuh dibagi dengan kecepatan. Berikut ini persamaan waktu tempuh.

$$t = s / v \quad (3)$$

- Kecepatan

  Kecepatan memiliki satuan Meter per sekon (m/s). Cara menghitung kecepatan adalah jarak tempuh dibagi dengan waktu tempuh dengan persamaan sebagai berikut.

$$v = s / t \quad (4)$$

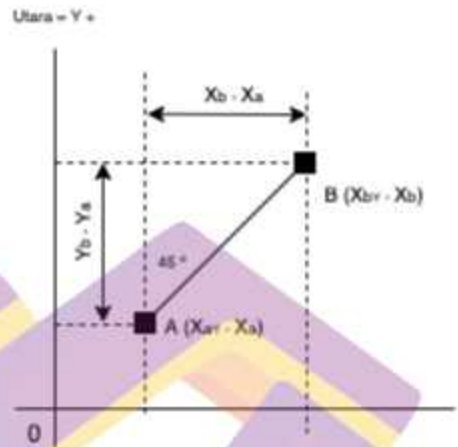
- Rata-Rata Kecepatan

$$v = \frac{S_{total}}{t_{total}} = \frac{V_1 x\ t_1 + V_2 x\ t_2 + \ldots + V_n x\ t_n}{t_1 + t_2 \ldots + t_n} \quad (5)$$

  Dari total seluruh kecepatan kita dapat menghasilkan rata - rata kecepatan, persamaan yang digunakan adalah sebagai berikut.

2) *Bearing* GPS

Bearing adalah sudut arah kendaraan berjalan menggunakan satuan derajat dengan utara sebagai titik 0 atau 360 derajat. Timur adalah 90 derajat, selatan 180 derajat, dan barat sebagai 270 derajat. Titik utara dapat sebagai utara magnetik (kompas), tapi bisa juga menggunakan utara absolut (kutub utara).



Gambar 2. 12. Pengukuran Sudut Bearing

Bearing memiliki sudut dari titik awal menuju titik tujuan atau akhir perjalan, atau posisi suatu objek dari pengamat. Jadi dalam suatu *bearing*, dapat terdiri dari beberapa *heading*. Pada gambar 2.12, *bearing* dari titik A menuju titik B adalah 45 derajat. *Bearing* disebut juga sebagai *azimuth*. *Bearing* dari A ke B dapat dihitung dengan menggunakan persamaan sebagai berikut

dihitung dengan menggunakan persamaan sebagai berikut

$$a_{AB} = tan^{-1} \frac{(Y_b - Y_a)}{(Y_b - Y_a)} \quad (6)$$

2.3.6. *Data pre-processing*

Sebelum menggunakan data dengan metode atau teknik data mining perlu dilakukan tahapan Data preprocessing pada penelitian ini kami menggunakan

beberapa metode dalam implementasinya, metode yang kami gunakan dalam penelitian ini adalah sebagai berikut

2.3.6.1. Principal Component Analysis (PCA)

PCA merupakan teknik yang digunakan untuk memproses data dengan cara mentransformasikan data secara linier ke dalam sistem koordinat baru dengan varians maksimum untuk menyederhanakan data tersebut. Dengan analisis komponen utama, dimensi suatu data dapat direduksi tanpa kehilangan karakteristik data secara signifikan. PCA bertujuan untuk menjelaskan struktur varians-kovarians melalui kombinasi linear dari variabel-variabel dan reduksi serta menginterpretasi data.

Analisis komponen utama sering kali dilakukan pada saat pengolahan data dalam kebanyakan penelitian yang datanya bersifat lebih besar besar/banyak/luas. PCA menjadi teknik yang paling awal diterapkan terhadap penelitian analisis Klasifikasi dimana komponen utama dipergunakan sebagai input untuk melakukan pengelompokan. PCA juga bermanfaat dalam regrouping variabel-variabel dengan melakukan penamaan ulang pada komponen utama yang terbentuk, dengan melihat karakteristik dominan variabel yang menyusunya

2.3.6.2. Ekstraksi Fitur

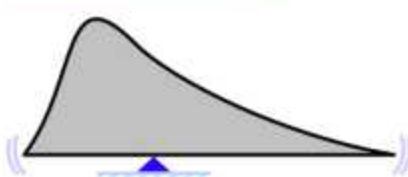
Ekstraksi Fitur merupakan fungsi yang dapat mengukur berbagai aspek dataset dan nilainya dapat menjadi ciri kumpulan data. Sementara ekstraksi fitur dapat didefinisikan sebagai sebuah proses dalam menganalisa dan kemudian mengubah dataset menjadi fitur database yang berarti untuk diproses menjadi langkah selanjutnya. Bagian ini mengenalkan jenis fitur yang diambil dari

kumpulan data log accelerometer, dapat mencerminkan perilaku atau pergerakan pengguna sesuai dengan keadaan dan kegiatan sehari hari . Tujuannya adalah untuk mengekstrak parameter yang mewakili informasi diskriminatif, misalkan setiap orang sering melakukan kegiatan duduk, berjalan, dan berbaring dengan komposisi dan keadaan yang berbeda beda. Setiap pengguna mungkin memiliki tempat kunjungan yang berbeda, bentuk tubuh yang berbeda-beda, lingkungan yang berbeda, aktivitas yang berbeda, dan lain-lain.

Beberapa fitur diambil dari sinyal *Accelerometer, Gyroscope* dan GPS, seperti *min, max, mean, standar deviasi*, jumlah besaran vektor, namun kurang informatif untuk fitur yang umum digunakan dalam eksperimen klasifikasi cara mengemudi yang ada. Persamaan *generic* seperti Mean, Median, dan Vektor dapat diterapkan pada sensor smartphone seperti yang terdapat pada *Accelerometer, Gyroscope*. dan GPS

1) *Mean*

Mean atau rata-rata hitung adalah nilai yang diperoleh dari jumlah sekelompok data dibagi dengan banyaknya data. kurva mean bertujuan untuk melihat rata - rata persebaran data sensor yang tersedia pada dataset



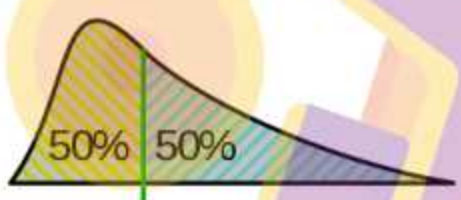
Gambar 2. 13. Ilustrasi Mean.

Untuk mengukur Mean / Rata-rata pada gambar 2.13, dalam dataset menggunakan persamaan berikut

$$\underline{X} = \frac{X_1 + X_1 + \ldots + X_n}{n} \quad (7)$$

2) *Median*

Median adalah nilai data yang terletak di tengah setelah data diurutkan. Dengan demikian, median membagi data menjadi dua bagian yang sama besar. kurva *Median* bertujuan untuk melihat nilai tengah pada persebaran data sensor yang tersedia pada dataset



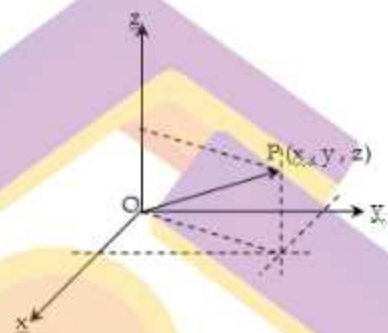
Gambar 2. 14. Ilustrasi Median.

Untuk mengukur *Median* / Nilai tengah pada gambar 2.14 dalam dataset menggunakan persamaan berikut

$$Me = X_{\frac{n+1}{2}} \quad (8)$$

3) Vektor Pada Ruang 3 Dimensi

Vektor di ruang 3 dimensi adalah vektor yang mempunyai 3 buah sumbu yaitu x , y , z yang saling tegak lurus dan perpotongan ketiga sumbu sebagai pangkal perhitungan.



Gambar 2. 15. Ilustrasi Vektor 3 Dimensi

Gambar 2.15 menggambarkan Vektor posisi titik P adalah vektor yaitu vektor yang berpangkal di titik O (0 , 0 , 0) dan berujung di titik P (x , y , z). *Modulus* / besar vektor posisi dapat dihitung dengan persamaan sebagai berikut.

$$|\overrightarrow{OP}| = \sqrt{x^2 + y^2 + z^2} \qquad (9)$$

2.3.6.3. Seleksi Fitur

Teknik ini merupakan tahapan pra pengolahan data mining yang sering digunakan. Tujuan seleksi fitur adalah untuk mengurangi jumlah fitur yang tidak relevan dalam penentuan suatu kelas. Fitur akan dianggap sebagai fitur yang

relevan apabila nilainya bervariasi secara sistematis dengan suatu kelas. Pengurangan jumlah fitur akan berdampak dalam kemudahan proses klasifikasi dan mempersingkat waktu eksekusi klasifikasi . Banyaknya fitur yang digunakan akan berdampak pada suatu flaw yang sering disebut sebagai curse of dimensionality, classifier tidak akan memberikan hasil yang optimal dan proses klasifikasi memakan resource yang besar. Selain itu, dengan berkurangnya jumlah fitur, maka proses visualisasi data akan lebih mudah.

Parameter yang digunakan dalam pemilihan subset fitur adalah relevansi fitur terhadap proses klasifikasi. Relevansi dapat dikategorikan menjadi tiga jenis, yaitu relevansi kuat, lemah, dan tidak relevan. Jika fitur dengan relevansi kuat tidak dipilih dalam klasifikasi, performa klasifikasi akan turun secara signifikan. Sementara itu, jika fitur dengan relevansi lemah dipilih, akurasi akan menurun (Hasanin & Khoshgoftaar, 2018). Fitur yang tidak relevan akan meningkatkan kebutuhan sumber daya tanpa memberikan peningkatan akurasi dalam proses klasifikasi. Ada empat langkah utama dalam proses seleksi fitur, yaitu pembangkitan subset, evaluasi subset, kriteria penghentian, dan validasi hasil.

2.3.7. Klasifikasi

Salah satu metode yang digunakan untuk mendeteksi cara mengemudi secara agresif atau normal adalah metode klasifikasi, metode ini merupakan proses pembuatan model atau target, dari model tersebut bertujuan untuk memprediksi target label kelas yang tidak diketahui. Model klasifikasi didapatkan dari proses training data set yang telah melewati tahapan pra proses data. kemudian model tersebut diterapkan pada data testing sehingga didapatkan akurasi untuk mengukur

berapa akurat sebuah model melakukan klasifikasi terhadap sebaran dataset secara keseluruhan. Proses klasifikasi terdiri dari empat komponen dataset yaitu :

a. *Label* merupakan variabel dependen bersifat kategoris dan merupakan label pada suatu instance.
b. *Predictor* merupakan variabel independen yang memiliki ciri ciri khusus sebagai atribut atau fitur yang akan digunakan untuk melakukan klasifikasi.
c. *Training Dataset* merupakan kumpulan data yang mempunyai nilai prediktor pada kelas dengan variabel *dependen*, dan digunakan dalam membuat model dan melatih model untuk mengenali kelas yang ada berdasarkan prediktor yang tersedia.
d. *Testing dataset* merupakan data baru yang tidak diketahui kelasnya kemudian diklasifikasikan dengan model (*Classifier*) yang dibuat sebelumnya kemudian dievaluasi kinerja modelnya.

2.3.7.1. Random Forest

Metode *Random Forest* (RF) adalah salah satu teknik ensemble yang bertujuan untuk meningkatkan akurasi metode klasifikasi dengan cara menggabungkan beberapa metode klasifikasi (Patil & Meena, 2017). RF sendiri awalnya menggunakan *decision* tree, yang merupakan teknik dasar dalam data mining. Pada *decision tree*, data input dimasukkan pada root dan diolah turun ke leaf untuk menentukan kelas data tersebut. RF sendiri terdiri dari kumpulan pengklasifikasi pohon terstruktur, di mana setiap pohon memberikan suara untuk kelas yang paling populer pada input x. Dengan kata lain, RF menggunakan kumpulan decision tree untuk mengklasifikasikan data ke dalam kelas tertentu.

Random Forest sendiri merupakan metode klasifikasi *supervised*, yang menghasilkan sebuah hutan dengan sejumlah pohon. Semakin banyak pohon pada hutan, semakin kuat hutan tersebut terlihat, dan semakin tinggi pula tingkat akurasinya. Dengan demikian, semakin banyak pohon yang digunakan, semakin tinggi juga tingkat akurasi yang dapat dicapai. Dalam *Random Forest*, informasi gain dan indeks gini digunakan untuk menentukan *root node* dan *rule*, seperti yang dilakukan pada metode klasifikasi lainnya. Namun, *Random Forest* membangun lebih dari satu pohon. Setiap pohon dibangun dengan menggunakan dataset yang diambil secara acak dari data training dengan atribut tertentu. Dengan kata lain, setiap pohon bergantung pada nilai dari sampel vektor yang independen dengan distribusi yang sama pada setiap pohon. Selama proses klasifikasi, setiap pohon memberikan suara untuk kelas yang paling populer. Ada tiga faktor yang memastikan keunikan dari setiap pohon, yaitu:

a. Setiap pohon dilatih dengan subset acak dari contoh data latih.
b. Selama pertumbuhan pohon, pembagian terbaik pada setiap node pohon ditemukan dengan cara mencari pada fitur acak.
c. Setiap pohon akan diproses hingga mencapai node terakhir.

Random Forest digunakan dalam praktik untuk klasifikasi dan regresi. Algoritma ini memiliki keunggulan dibandingkan dengan algoritma pendahulunya, yaitu *Classification* and *Regression Tree* (CART), karena meningkatkan kestabilan dan akurasi. Ini karena *tree* yang terbentuk merupakan hasil kombinasi dari banyak *tree* sehingga perubahan sedikit pada data tidak terlalu mempengaruhi tree secara keseluruhan. Dari segi pemrosesan komputasi (Mehanian, 2017). Random Forest

memiliki kelebihan karena dapat digunakan untuk melakukan regresi dan klasifikasi pada data kontinu dan kategori, tidak mudah mengalami *overfitting*, memiliki tingkat akurasi yang tinggi dan kokoh karena dibangun dari banyak *tree* keputusan, dapat digunakan secara langsung pada permasalahan kompleks atau besar, dan dapat diimplementasikan secara parallel tergantung pada satu atau dua parameter penyetelan.

Random Forest dapat menjadi pilihan yang baik untuk digunakan dalam analisis data sensor gerak *smartphone* untuk mengidentifikasi perilaku mengemudi yang berbahaya, karena memiliki beberapa kelebihan, di antaranya:

a. *Random Forest* dapat menghasilkan model yang lebih stabil dan akurat dibandingkan dengan *Decision Tree*, karena model *Random Forest* memadukan beberapa pohon keputusan (*Decision Tree*) secara acak (random), sehingga dapat mengurangi risiko *overfitting* pada model.
b. *Random Forest* dapat menangani data yang tidak seimbang (*imbalanced data*) dengan baik, yaitu ketika kelas yang akan diprediksi memiliki jumlah sampel yang tidak seimbang, misalnya jumlah sampel untuk kelas yang berbahaya jauh lebih sedikit dibandingkan dengan kelas yang tidak berbahaya.
c. *Random Forest* memiliki kemampuan untuk mengekstraksi fitur-fitur penting dari data, sehingga dapat membantu dalam mengidentifikasi faktor-faktor yang paling berpengaruh dalam perilaku mengemudi yang berbahaya.

Namun, kelemahan dari *Random Forest* adalah kompleksitas model yang cukup tinggi, sehingga memerlukan waktu yang lebih lama untuk proses pelatihan

model dan pengolahan data yang cukup besar. Selain itu, interpretasi hasil dari model *Random Forest* dapat lebih sulit dibandingkan dengan *Decision Tree*. Oleh karena itu, perlu dilakukan evaluasi yang baik terhadap model *Random Forest* untuk memastikan bahwa model tersebut dapat menghasilkan prediksi yang akurat dan tergeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

2.3.7.2. Support Vector Machine (SVM)

Support Vector Machine (SVM) dapat menjadi pilihan yang tepat untuk digunakan dalam analisis data sensor gerak smartphone untuk mengidentifikasi perilaku mengemudi yang berbahaya, karena memiliki beberapa kelebihan, di antaranya:

a. SVM dapat menangani data yang kompleks dan multidimensi dengan baik, termasuk data sensor gerak smartphone yang memiliki banyak variabel dan dimensi.
b. SVM dapat menghasilkan model yang akurat dan stabil, terutama pada data yang tidak seimbang (*imbalanced* data), yaitu ketika jumlah sampel untuk kelas yang berbahaya jauh lebih sedikit dibandingkan dengan kelas yang tidak berbahaya.
c. SVM memiliki kemampuan untuk mengekstraksi fitur-fitur penting dari data, sehingga dapat membantu dalam mengidentifikasi faktor-faktor yang paling berpengaruh dalam perilaku mengemudi yang berbahaya.

Namun, kelemahan dari SVM adalah model yang dihasilkan relatif sulit diinterpretasikan, sehingga sulit untuk mengetahui faktor-faktor apa yang mempengaruhi perilaku mengemudi yang berbahaya. Selain itu, SVM memiliki

kendala dalam pengolahan data yang sangat besar atau kompleks, karena memerlukan waktu yang lama untuk pelatihan model dan pengolahan data yang cukup besar. Oleh karena itu, perlu dilakukan evaluasi yang baik terhadap model SVM untuk memastikan bahwa model tersebut dapat menghasilkan prediksi yang akurat dan tergeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

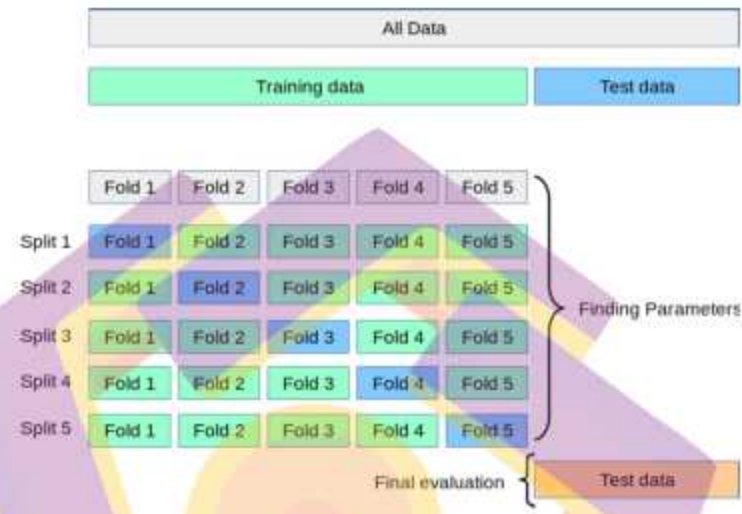
2.3.8. Validasi

2.3.8.1. Cross Validation

Metode *Cross Validation* merupakan salah satu metode untuk mengevaluasi algoritma learning dengan membagi data menjadi dua segmen yaitu. Segmen pertama digunakan untuk pembelajaran (*learning*) pelatihan (*training*) model, segmen kedua digunakan untuk validasi model. Ciri khas dari *Cross Validation* adalah set training dan validasi harus disilangkan (*cross-over*) dalam putaran berturut-turut sehingga setiap titik data memiliki peluang untuk divalidasi . Dasar dari *Cross Validation* adalah *K-Fold Cross Validation* (Wong & Yang, 2017).

Dalam *Cross Validation*, langkah pertama kali adalah data akan dipartisi ke dalam segmen atau *fold* yang sama atau identik (nyaris sama). Berikutnya adalah iterasi ke k dari *training* dan validasi dilakukan sedemikian rupa sehingga dalam setiap iterasi *fold* data yang berbeda dimunculkan (*held-out*) untuk validasi, sementara sisa fold k-1 digunakan untuk training. Pada gambar 2.18 menggambarkan contoh dengan k − 3. Bagian yang gelap menggambarkan data untuk *training* sedangkan bagian terang menggambarkan data untuk validasi

(*testing*). Dalam data mining maupun *machine learning* 10-*fold cross-validation* (k=10) merupakan yang paling umum atau sering digunakan.



Gambar 2. 16. Skema Cross Validation

2.3.8.2. Confusion Matrix

Confusion matrix digunakan untuk melihat seberapa baik atau seberapa besar performa berdasarkan parameter pengujian yang dihasilkan dari model klasifikasi yang sudah dibuat untuk memprediksi atau mengklasifikasi kelas dari data testing. Rincian hasil klasifikasi berupa prediksi kelas ditampilkan di atas dan kelas yang aktual di bawah kiri.

Tabel 2. 2. Confusion Matrix.

	Actually Positive	***Actually Negative***
Predicted Positive	*True Positive (TP)*	*False Positive (FP)*
Predicted Negative	*False Negative (FN)*	*True Negative (TN)*

TP : Jumlah kelas positif yang diklasifikasi sebagai positif

FP : Jumlah kelas negatif yang diklasifikasi sebagai positif

TN : Jumlah kelas negatif yang diklasifikasi sebagai negatif

FN : Jumlah kelas negatif yang diklasifikasi sebagai negatif

2.3.8.3. Index Pengukuran

Berdasarkan *Confusion Matrix*, maka dapat diperoleh beberapa variabel pengukuran yang dapat digunakan untuk mengukur dan mengevaluasi kinerja klasifikasi:

1) Akurasi, menunjukkan sejauh mana hasil skrining sesuai dengan kenyataannya, atau, proporsi subjek yang diidentifikasi dengan benar sesuai dengan standar yang terbaik yang telah disepakati bersama (*gold standard*), Akurasi menjawab pertanyaan "Berapa persen perjalan yang benar diprediksi aman dan berbahaya dari keseluruhan perjalanan".

$$Accuracy \qquad \frac{TP+TN}{TP+FP+TN+FN} x100\% \qquad (10)$$

2) *Precision* Merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. *Precision* menjawab pertanyaan "Berapa persen perjalanan yang benar aman dari keseluruhan perjalanan yang diprediksi aman dan sebaliknya. Untuk menghitung nilai *Precision*, dapat digunakan persamaan berikut:

$$Precision \qquad \frac{TP}{TP+FP} x\ 100\% \qquad (11)$$

3) *Sensitivity*, atau *True Positive Rate* (TPR) adalah proporsi subjek positif dan teridentifikasi oleh sistem dimana semua subjek merupakan subjek positif. Sensitivity menjawab pertanyaan "Berapa persen perjalanan yang diprediksi aman dibandingkan keseluruhan perjalanan yang sebenarnya aman".Untuk menghitung nilai *Sensitivity*, dapat digunakan persamaan berikut:

$$Sensitivity \quad \frac{TP}{TP + F} \tag{12}$$

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Pada tahap ini dilakukan kajian pustaka mengenai langkah-langkah dalam metode klasifikasi yang telah digunakan pada penelitian-penelitian sebelumnya terutama pada kasus klasifikasi cara berkendara. Dalam tahapan studi pustaka ditemukan bahwa pada penelitian sebelumnya yang relevan, telah dilakukan skenario eksperimen membandingkan performa algoritme machine learning pada dataset perjalanan berdasarkan fitur dataset yang mirip pada penelitian ini, seperti akselerometer, *gyroscope* and GPS. karakteristik dataset yang digunakan mirip dengan dataset penelitian ini, sehingga pada penelitian ini akan digunakan model eksperimen pada penelitian sebelumnya, dan akan dibandingkan dengan model eksperimen pada penelitian ini.

3.2. Metode Pengumpulan Data

Dalam penelitian mengenai analisis pengolahan data sensor gerak smartphone untuk mengetahui perilaku mengemudi, metode eksperimen akan digunakan untuk mengeksplorasi hubungan sebab-akibat antara variabel-variabel yang diamati. Metode eksperimen akan memungkinkan untuk memanipulasi variabel independen dataset yang digunakan, waktu penggunaan aplikasi. Data yang dihasilkan dari sensor smartphone akan dikirim ke server aplikasi. kemudian diperlukan *query* khusus untuk mengekstrak data fitur dan data label dari server aplikasi tersebut menjadi bentuk CSV. Data yang di berikan hanyalah data sampel

yang diberikan oleh wali data dalam bentuk CSV tersebut, peneliti tidak memiliki keluluasan lebih untuk melakukan melakukan pengumpulan data diluar dari dataset yang telah disediakan.

3.3. Metode Analisis Data

Dalam rangka mengeksplorasi dan menganalisis data perjalanan secara menyeluruh, penelitian ini mengadopsi pendekatan *Cross-Industry Standard Process for Data Mining* (CRISP-DM). Metode ini memberikan struktur dan kerangka kerja yang sistematis untuk menangani tahapan analisis data mulai dari pemahaman bisnis, pemahaman data, *preprocessing data*, pemodelan, evaluasi, hingga implementasi. Pendekatan CRISP-DM menekankan pada iterasi dan siklus berulang dalam proses penambangan data, memungkinkan peneliti untuk menggali wawasan yang mendalam dari dataset perjalanan. Dengan merinci langkah-langkah mulai dari definisi tujuan bisnis hingga penerapan model dan evaluasi, penelitian ini memanfaatkan metodologi yang terbukti dan diterima secara luas untuk memastikan keakuratan dan interpretabilitas analisis data. Pendekatan ini diharapkan dapat memberikan landasan metodologis yang kokoh untuk penelitian ini, memungkinkan eksplorasi yang efektif terhadap pola dan tren dalam data perjalanan. Penelitian ini memanfaatkan alat dan bahan tertentu untuk mendukung implementasi metodologi CRISP-DM. Alat dan bahan utama yang digunakan dalam tahapan pemrosesan dan analisis data adalah sebagai berikut;

3.3.1. Alat

3.3.1.1. Perangkat keras

Perangkat keras yang digunakan dalam penelitian ini adalah laptop dengan spesifikasi *prosesor Intel® Core™* i5 3317U, CPU @ 1.70Ghz, RAM 8GB, dan SSD 256GB laptop digunakan untuk pemrosesan seluruh tahapan pengujian.

3.3.1.2. Perangkat Lunak

Perangkat lunak yang digunakan untuk penelitian ini yaitu:

a. *Google Colab 1.0.0*, yaitu aplikasi yang digunakan untuk menjalankan dokumen berformat ipynb dengan bahasa pemrograman Python melalui web browser *Google Chrome Version 114.0.5735.198*.
b. Bahasa pemrograman *Python 3.10.12*.
c. *Library* yang digunakan diantaranya
    1. *NumPy 1.22.4*: Library untuk komputasi numerik menggunakan array multidimensi.
    2. *Pandas 1.5.3*: Library untuk manipulasi dan analisis data.
    3. *Matplotlib 3.7.1*: Library untuk visualisasi data dalam bentuk grafik dan plot.
    4. Scikit-learn 1.2.2: Library untuk pembelajaran mesin dan analisis data yang mencakup berbagai algoritma dan utilitas.

3.3.2. Bahan

Bahan yang digunakan pada penelitian ini adalah dataset berupa data sensor *smartphone* pengemudi, data direkam berdasarkan perjalanan dengan jarak tempuh bervariasi. Cara mengemudi diambil secara acak dengan rincian sebagai berikut.

a. Penyedia *dataset* adalah salah satu perusahaan teknologi yang beroperasi di Indonesia untuk melayani transportasi umum melalui jasa pemesanan transportasi *online*
b. Mode kendaraan yang digunakan adalah mobil.

c. *Dataset* merupakan data sensor *Gyroscope*, sensor *Accelerometer*, dan Sensor GPS.
d. Tersedia Data Fitur dengan jumlah sekitar 16 juta baris data.
e. Tersedia Data Label dengan jumlah 20.018 baris data, data ini berisi Booking ID perjalan yang telah diberi label.
f. Total ukuran dataset adalah 1.89 GB

Informasi utama yang tersedia pada dataset yang disediakan adalah sebagai berikut:

Tabel 3. 1. Tabel Dataset.

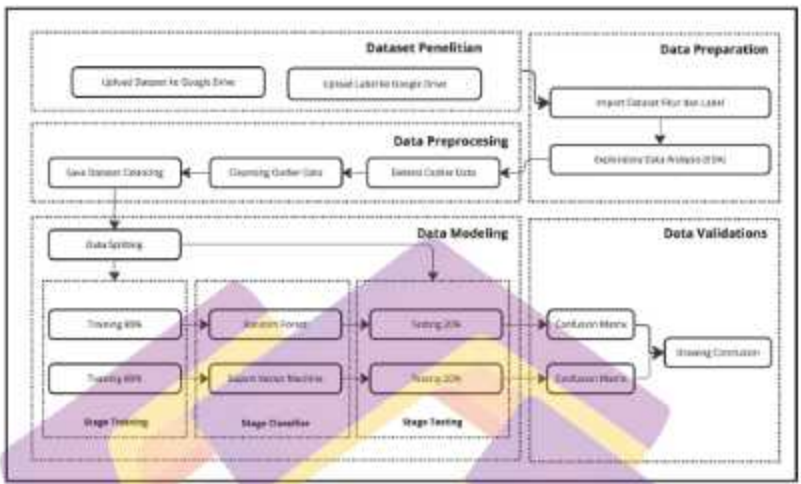
No	Kolom	Tipe Data	Deskripsi	Sensor
1	*second*	*Float (14)*	Catatan waktu perekaman data (s)	Default
2	*acceleration_x*	*Float (14)*	Pengukuran *acceleration* pada sumbu x	Accelerometer
3	*acceleration_y*	*Float (14)*	Pengukuran *acceleration* pada sumbu y	Accelerometer
4	*acceleration_z*	*Float (14)*	Pengukuran *acceleration* pada sumbu z	Accelerometer
5	*gyro_x*	*Float (14)*	Pengukuran *gyroscope* pada sumbu x	Gyroscope
6	*gyro_y*	*Float (14)*	Pengukuran *gyroscope* pada sumbu y	Gyroscope
7	*gyro_z*	*Float (14)*	Pengukuran *gyroscope* pada sumbu z	Gyroscope
8	*Accuracy*	*Float (14)*	Pengukuran Akurasi berdasarkan GPS	GPS

Tabel 3. 1. Tabel Dataset. (Lanjutan)

9	*Bearing*	*Float (14)*	Arah kompas dari posisi awal ke posisi yang akan dituju berdasarkan GPS dengan satuan ukur derajat	GPS
10	*Speed*	*Float (14)*	Pengukuran kecepatan GPS (m/s)	GPS
11	*label*	*Int (1)*	Class kategori cara berkendara yang dilakukan secara manual pada setiap perjalanan dengan asumsi • *Class* '0' perjalanan aman • *Class* '1' perjalanan berbahaya	-

3.4. Alur Penelitian

Alur penelitian ini secara umum dibagi menjadi beberapa tahap yang terdiri atas *Data Source*, *Data Cleaning*, *Data Preprocessing*, data *Modelling* hingga tahap pengujian dengan melakukan pengukuran hasil *confusion matrix* hasil klasifikasi terhadap dataset, dan diakhiri dengan perbandingan untuk uji kerja dari model tersebut agar didapatkan model paling unggul untuk bekerja pada dataset penelitian.



Gambar 3. 1. Alur Penelitian

Alur penelitian pada gambar 3.1 yang akan dilakukan, dalam diagram alir digambarkan bahwa, dalam penelitian ini akan dilakukan skenario eksperimen, yaitu eksperimen menggunakan metode machine learning dengan menguji pada model klasifikasi *Random Forest* dan *Support Vector Machine* kemudian dilakukan perbandingan performa masing-masing model. Langkah - langkah yang dilakukan dalam penelitian ini terbagi menjadi beberapa tahap sebagai berikut.

3.4.1. Data Understanding

Tahap ini mencakup proses pengumpulan data dan analisis deskriptif terhadap data penelitian. Dataset penelitian yang digunakan merupakan data data perjalanan dengan jarak tempuh bervariasi dan cara mengemudi diambil secara acak. Dataset tersebut didapat dari data transaksi taksi daring yang telah disediakan oleh salah satu perusahaan teknologi yang beroperasi di Indonesia untuk melayani angkutan umum melalui jasa pemesanan transportasi *online*. Dalam penelitian ini,

dataset menggunakan data sensor *smartphone*, beberapa sensor yang akan menjadi fokus fitur dalam penelitian ini seperti Sensor accelerometer, sensor *gyroscope* dan kecepatan. Perekaman data dalam satu perjalanan dimulai dari titik jemput hingga berakhir pada titik antar sesuai jarak, rute, dan waktu tempuh masing-masing perjalanan yang dikelompokan dalam satu Booking.ID kemudian dikirim ke Server Aplikasi. Setiap perjalanan diambil berdasarkan waktu nyata, sehingga ukuran data bervariasi. Detail tahapan yang dilakukan dalam fase *Data Collection* diantaranya;

a. *Database Administrator* akan mengekspor data dari server dengan *parameter ID* data sensor dan waktu dan disimpan menjadi *dataset*.
b. Tim aplikasi memiliki dataset yang sudah diberi label dan dataset tersebut diberikan ke peneliti untuk dijadikan sebagai data training.

Pada tahapan data collection peneliti hanya sebatas menerima dataset tidak terlibat langsung dalam tahapan di atas. Data yang diberikan ke peneliti ada dua jenis data yaitu data Fitur dan data Label, kedua data ini dikirimkan oleh pihak aplikasi yang sudah di upload ke google drive sehingga peneliti dapat mengakses data tersebut.

3.4.2. Data *Preparation*

Data yang telah dikumpulkan pada langkah sebelumnya. diupload ke dalam *google drive* untuk mempermudah pemanggilan atau *import* di *tools google kolaps*. *Dataset* Mentah terdiri dari 2 file data terpisah yaitu *file* data fitur dan *file* data label. Data fitur berisi data sensor *Gyroscope*, sensor *Accelerometer*, dan data GPS, sedangkan data label berisi *Booking* perjalanan yang telah diberi label perjalanan

aman dan label perjalanan berbahaya, namun tidak semua data fitur yang disediakan memiliki label.

Data yang tidak memiliki label atau memiliki nilai *null* dapat mempengaruhi klasifikasi. Sebagian besar algoritma klasifikasi membutuhkan label yang jelas untuk melatih model. Data yang tidak memiliki label atau memiliki nilai null tidak memberikan informasi yang cukup bagi model untuk belajar dan membuat prediksi yang akurat (Q. Liu & M. Hauswirth,2020), oleh karena itu peneliti akan melakukan drop data yang tidak memiliki label.

Secara umum, data mentah yang diolah dalam penelitian ini memiliki jumlah data sekitar 16 juta dari 20.018 perjalanan, jumlah data setiap perjalanan bervariasi berdasarkan jarak tempuh yang dilalui. Setiap melakukan tahapan *preprocessing* data, *dataset* baru disimpan berdasarkan proses yang telah dilakukan. Selanjutnya dibersihkan terlebih dahulu agar lebih optimal, seperti melakukan penghapusan data yang tidak memiliki label.

3.4.3. Data Preprocessing

Tujuan utama dari tahapan ini adalah melakukan pengolahan data lebih lanjut dikarenakan *dataset* yang digunakan sangatlah besar. *Dataset* yang diolah dalam penelitian ini memiliki jumlah data sekitar 16 juta dari 20.018 perjalanan. Tahap demi tahap pada dataset mentah, data yang diolah adalah data sensor *smartphone* pengemudi dari masing-masing perjalanan dari titik jemput dan berakhir pada titik antar, *dataset* yang disediakan adalah data fitur dan data label. Berikut adalah langkah Langkah *preprocessing* data tersebut.

3.4.3.1. Proses Ekstraksi Fitur

Sebelum melalui tahapan klasifikasi, setiap perjalanan yang terdeteksi dilakukan ekstraksi fitur. Fitur-fitur inilah yang nantinya akan digunakan oleh mesin classifier untuk membedakan dua jenis objek kelas yang berbeda. Ekstraksi fitur akan dilakukan pada data pergerakan sensor dan telah melalui pemetaan kedalam dua jenis direktori aman dan berbahaya. Seperti yang telah disebutkan sebelumnya, pemetaan jenis gerakan kedalam dua jenis direktori ini akan mempermudah proses pelabelan dalam tahap ekstraksi fitur, sehingga, setiap data yang telah melalui proses ekstraksi fitur akan dapat dengan mudah diketahui label perjalanan yang sesuai.

3.4.3.2. **Proses Seleksi Fitur**

Pada tahapan ini dilakukan penerapan metode seleksi fitur untuk mengetahui fitur apa. Tujuan utama dari seleksi fitur dalam dataset untuk meningkatkan performa model dan meningkatkan interpretasi hasil. Dalam analisis data atau pembuatan model, penggunaan fitur atau variabel yang tidak relevan atau tidak signifikan dapat mempengaruhi kualitas model atau analisis tersebut. Dengan melakukan seleksi fitur, variabel yang tidak relevan atau tidak signifikan dapat dihilangkan, sehingga dapat mengurangi dimensi data dan meningkatkan akurasi model atau hasil analisis. Selain itu, seleksi fitur juga dapat membantu mempercepat waktu komputasi karena data yang lebih sedikit yang perlu diolah. Selain meningkatkan performa model dan interpretasi hasil, seleksi fitur juga dapat membantu dalam mencegah *overfitting*, yaitu kondisi di mana model terlalu mempelajari data pelatihan dan tidak mampu memprediksi data baru dengan

akurasi yang baik. Dengan menghilangkan variabel yang tidak relevan atau tidak signifikan, model dapat menjadi lebih sederhana dan tidak terlalu kompleks, sehingga dapat mengurangi kemungkinan *overfitting*.

3.4.3.3. Simpan Dataset Cleansing

Setelah proses *cleaning* dan *preprocessing* selesai dilakukan, selanjutnya dilakukan menyimpan data hasil *cleansing* tersebut agar dapat digunakan dalam tahap selanjutnya, file dataset di simpan menjadi bentuk File *Comma-Separated Values* (CSV).

3.4.4. Data Modelling

Pada tahapan data modeling aktivitas yang akan dilakukan adalah membuat model dari data yang sudah diproses sebelumnya. Tahapan yang dilakukan diantaranya;

a. *Split dataset*, pada tahapan ini *dataset* yang sudah dibersihkan akan di *split*/ di bagi menjadi 2 bagian 80% dan 20%.
b. *Stage Training*, pada tahapan ini data sebanyak 80% akan di gunakan untuk *training*.
c. *Stage Classifications*, pada tahap ini data training yang sudah siapkan akan diproses menggunakan *classifier* dengan data label. tahapan training akan dilakukan pada masing-masing *classifier* (RF dan SVM).
d. *Stage testing*, pada tahapan ini data model training akan di testing menggunakan data 20% untuk mendapatkan nilai pengujian.

3.4.5. Model Validasi

Pada tahapan ini, akan dilakukan proses perbandingan hasil Dari hasil uji model, hasil pengujian akan didapatkan masing-masing tingkat akurasi, presisi, *recall* dari tiap jenis data latih. untuk mendapatkan nilai parameter performa di atas digunakan *Confusion matrix* untuk menunjukkan jumlah prediksi benar atau salah yang dilakukan oleh model untuk setiap kelas. Matriks ini membantu dalam mengevaluasi tingkat kesalahan model, seperti *false positive* dan *false negative*, serta memperoleh informasi tentang kekuatan dan kelemahan model dalam mengklasifikasikan data. Dengan melakukan analisis model klasifikasi yang komprehensif, peneliti dapat memahami kekuatan dan kelemahan model yang telah dibangun, mengidentifikasi area yang perlu ditingkatkan, dan mengambil langkah-langkah untuk meningkatkan kinerjanya. Analisis ini juga membantu dalam memahami karakteristik data, mengoptimalkan ambang batas klasifikasi, dan memvalidasi model sebelum diterapkan pada data dunia nyata. Setelah semua skenario selesai dilaksanakan, akan dilakukan analisis dan penarikan kesimpulan penelitian.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

Pada bagian ini dijelaskan metode pengolahan data dengan metode CRISM-DM *Business Understanding*, *Data Understanding*, *Data Preparation*, *Data Modeling*, *Data Evaluation*, dan *Data Deployment*

4.1. Business Understanding

Dalam pengembangan teknologi, data pergerakan yang dihasilkan dari sensor smartphone dapat membantu untuk menganalisa pergerakan dari smartphone itu sendiri yang secara tidak langsung dapat diasumsikan sebagai pergerakan pengemudi. Masalah yang ingin diselesaikan adalah mengurangi angka kecelakaan dengan melakukan monitoring perilaku mengemudi. Data sensor smartphone dan beberapa karakteristiknya diharapkan dapat menjadi objek penelitian untuk mengidentifikasi perilaku mengemudi, sehingga harapan dari proses analisis jenis perjalanan dapat ditemukan sebuah model klasifikasi perjalanan aman dan perjalanan berbahaya.

4.2. Data Understanding

Dataset perjalanan dengan informasi *gyroscope*, accelerometer, kecepatan, dan waktu terdengar menarik dan kaya akan informasi yang dapat dieksplorasi. Dataset tersebut memiliki beberapa kolom yang mencakup informasi seperti seperti pada tabel 4.1 berikut:

Tabel 4. 1. Deskripsi Fitur Dataset.

No	Kolom	Tipe Data	Fungsi	Deskripsi
1	*second*	*Float (14)*	Catatan waktu perekaman data (s)	Default Sistem
2	*acceleration_x*	*Float (14)*	Pengukuran *acceleration* pada sumbu x	Accelerometer memiliki satuan nilai meter per detik kuadrat (m/s^2) dengan nilai pada setiap sumbu antara -16 m/s^2 hingga +16 m/s^2.
3	*acceleration_y*	*Float (14)*	Pengukuran *acceleration* pada sumbu y	
4	*acceleration_z*	*Float (14)*	Pengukuran *acceleration* pada sumbu z	
5	*gyro_x*	*Float (14)*	Pengukuran *gyroscope* pada sumbu x	Gyroscope Memiliki satuan nilai radian per detik (rad/s) dengan nilai pada setiap sumbu antara -200 °/s hingga +200 °/s
6	*gyro_y*	*Float (14)*	Pengukuran *gyroscope* pada sumbu y	
7	*gyro_z*	*Float (14)*	Pengukuran *gyroscope* pada sumbu z	
8	*Accuracy*	*Float (14)*	Pengukuran Akurasi berdasarkan GPS	GPS
9	*Bearing*	*Float (14)*	Arah kompas dari posisi awal ke posisi yang akan dituju berdasarkan GPS dengan satuan ukur derajat	GPS
10	*Speed*	*Float (14)*	Pengukuran kecepatan GPS (m/s)	GPS
11	*label*	*Int (1)*	Class kategori cara berkendara yang dilakukan	-

Tabel 4. 1. Deskripsi Fitur Dataset. (Lanjutan)

			secara manual pada setiap perjalanan dengan asumsi • *Class* '0' perjalanan aman • *Class* '1' perjalanan berbahaya	

Berikut adalah penjelasan singkat untuk dataset yang digunakan pada tabel 4.1 :

1. Data Sensor *Gyroscope*: Data *gyroscope* dengan tiga sumbu (gyro x, y, dan z).
2. Data Sensor Accelerometer: Data accelerometer dengan dua sumbu (acceleration x,y dan x).
3. Data GPS: Data GPS dengan beberapa fitur seperti Kecepatan berupa pengukuran perjalanan dalam satuan meter per detik, Waktu merupakan pengukuran timestamp dalam satuan detik dan Accuracy untuk Pengukuran Akurasi berdasarkan GPS.
4. Data Label: Data label menyatakan perjalanan aman (0) atau berbahaya (1) Class kategori cara berkendara yang dilakukan secara manual oleh pemilik data pada setiap perjalanan dengan asumsi *Class* '0' perjalanan aman *Class* '1' perjalanan berbahaya.

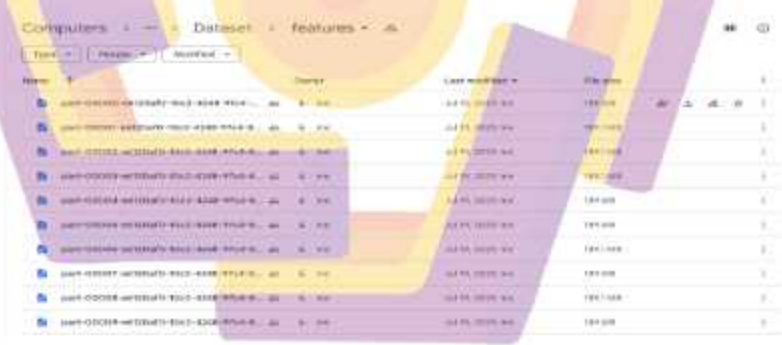
4.3. Data Preparation

4.3.1. *Import Dataset*

Untuk melakukan analisis data menggunakan *Google Colab*, langkah pertama yang diperlukan adalah mengimport dataset yang akan digunakan. *Google Colab* adalah lingkungan pengembangan berbasis web yang memungkinkan pengguna untuk menjalankan kode *Python* secara interaktif. Proses *impor* dataset dapat dilakukan dengan beberapa langkah sederhana:

4.3.1.1. Upload Dataset

Dalam penelitian ini dataset yang diberikan oleh pihak aplikasi berupa file CSV kemudian data tersebut di upload ke *google drive* untuk mempermudah penyimpanan data dan sekaligus mempermudah koneksi pengolahan dataset menggunakan *Google Colab*. Berikut adalah hasil upload dataset ke *google drive*



Gambar 4. 1. Dataset Penelitian

Gambar 4.1 merupakan dataset penelitian yang akan digunakan, dataset berjumlah 10 file CSV dengan ukuran masing-masing 189 MB dengan total kisaran 1.9 GB. Masing - masing dataset memiliki fitur yang sama akan tetapi

dengan list data yang berbeda yang nantinya ke 10 dataset ini akan digabungkan untuk mempermudah proses pengolahan data.

4.3.1.2. Import Library

Pada *Google Colab*, pengguna memiliki akses ke berbagai *library Python* yang dapat digunakan dalam pengembangan proyek *machine learning*, analisis data, dan pengolahan data. *Library-library* ini memungkinkan pengguna untuk memanfaatkan berbagai fungsionalitas dan alat analisis yang kuat berikut adalah *script python* dengan pemanggilan *library* yang digunakan untuk mengolah dataset.

```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
sb.set()
%matplotlib inline
plt.rcParams['agg.path.chunksize'] = 10000
```

Kode di atas adalah bagian dari setup awal untuk melakukan analisis data dengan menggunakan pustaka-pustaka yang diperlukan. Berikut adalah penjelasan dari setiap baris kode tersebut:

1. *import numpy as np*: Mengimpor pustaka NumPy dengan menggunakan alias np. NumPy adalah pustaka Python yang digunakan untuk operasi numerik, terutama array multidimensi dan fungsi matematika.
2. *import pandas as pd*: Mengimpor pustaka Pandas dengan menggunakan alias pd. Pandas adalah pustaka *Python* yang digunakan untuk manipulasi dan analisis data, khususnya dalam bentuk struktur data seperti *Dataframe*.

3. *import seaborn* as *sb*: Mengimpor pustaka *Seaborn* dengan menggunakan alias sb. *Seaborn* adalah pustaka *Python* yang digunakan untuk visualisasi data yang berdasarkan pustaka *Matplotlib*, tetapi dengan tampilan yang lebih menarik dan lebih mudah digunakan.
4. *import matplotlib.pyplot as plt*: Mengimpor pustaka *Matplotlib* dengan menggunakan alias plt. *Matplotlib* adalah pustaka *Python* yang paling populer untuk *plotting* data dan visualisasi.
5. *sb.set()*: Baris ini mengatur tampilan default untuk *plot Seaborn*. Ini akan mengubah tampilan default *Seaborn* untuk menghasilkan plot yang lebih bagus dan lebih menarik secara visual.
6. *%matplotlib inline*: adalah "*magic command*" khusus untuk lingkungan *Jupyter Notebook*. Dengan menggunakan *%matplotlib inline*, *output plot* akan ditampilkan secara langsung di dalam notebook setelah kode dijalankan.
7. plt.rcParams['agg.path.chunksize'] = 10000: Baris ini mengatur ukuran chunk untuk proses rendering dalam *Matplotlib*. Ini dapat membantu menghindari beberapa masalah yang muncul saat plotting data yang sangat besar.

4.3.1.3. *Read Dataset*

Selanjutnya ketika dataset sudah siap akan dilakukan pembacaan dataset, pada tahapan ini kami akan melakukan pembacaan dataset yang sudah di upload ke *google drive* kemudian akan melakukan penggabungan dataset menjadi 1 agar mudah untuk diolah pada *google colab*. Berikut adalah script yang digunakan.

```
from pathlib import Path
import glob
from google.colab import drive
drive.mount('/content/gdrive')
path = r'gdrive/Othercomputers/My MacBook Air/MTI
AMIKOM/TESIS/Ekperimen/Dataset features'
all_files = glob.glob(path + "/*.csv")
li = []
for filename in all_files:
    df = pd.read_csv(filename, index_col=None,
header=0)
    li.append(df)
data_all_files = pd.concat(li, axis=0,
ignore_index=True)
data_all_files.to_csv('dataset.csv', index=False)
```

Skrip diatas menggabungkan data dari beberapa file CSV yang berada dalam satu direktori tertentu. Direktori tersebut diakses melalui *Google Drive*, dan kemudian menggunakan modul *glob*, semua *file* dengan ekstensi .csv di dalam direktori tersebut diambil. Setiap file CSV dibaca menggunakan *pd.read_csv*() dari Pandas dan dimasukkan ke dalam *list li*. Akhirnya, semua *Data Frame* di dalam *list* li digabungkan menjadi satu *Dataframe* besar menggunakan *pd.concat*(). *Dataframe* hasil gabungan ini disimpan dalam variabel data. Dengan skrip di atas, setelah berhasil menggabungkan data dari beberapa file CSV menjadi satu *Data Frame* kemudian dilakukan proses

penyimpanan dataset menjadi file CSV baru dengan nama *dataset.csv* yang siap digunakan untuk analisis lebih lanjut atau pemrosesan data lainnya. Setelah setup awal ini, dapat melanjutkan dengan membaca data, melakukan EDA, dan melakukan visualisasi data sesuai dengan kebutuhan analisis.

4.3.2. Exploratory Data Analysis (EDA)

Dalam *Exploratory Data Analysis* (EDA) dapat melakukan berbagai analisis untuk memahami dataset perjalanan secara mendalam dan menemukan pola menarik. Berikut adalah beberapa hal yang bisa dilakukan dalam EDA:

4.3.2.1. Statistik Ringkasan

Statistik ringkasan, atau juga dikenal sebagai ringkasan deskriptif, adalah teknik statistik yang digunakan untuk menyajikan dan menggambarkan karakteristik dasar dari suatu set data. Statistik ringkasan memberikan informasi tentang sebaran, pusat, dan bentuk distribusi data. Berikut adalah beberapa statistik ringkasan yang umum digunakan: Hitungan statistik ringkasan seperti mean, median, standar deviasi, minimum, dan maksimum untuk setiap kolom. Tujuan dari statistik ringkasan untuk membantu memahami distribusi data dan melihat apakah ada nilai ekstrim atau outlier. Berikut adalah *script* untuk melakukan tahapan ini

```
import pandas as pd
# Membaca dataset perjalanan
data = pd.read_csv('dataset.csv')
# Statistik ringkasan untuk kolom-kolom numerik
summary_stats = data.describe()
# Menampilkan statistik ringkasan
print(summary_stats)
```

Dari hasil diatas dapat dijelaskan deskripsi data secara general. Berikut adalah hasil analisis deskripsi dataset penelitian yang digunakan berdasarkan data sensor yang tersedia.

A. Analisis deskripsi data sensor Accelerometer

Berikut adalah analisis dari hasil deskripsi data untuk fitur akselerasi (*acceleration*) dalam sumbu x, y, dan z:

Tabel 4. 2. Deskripsi data sensor Accelerometer

Feature	*acceleration_x*	*acceleration_y*	*acceleration_z*
count	16154418	16154418	16154418
mean	0.06931	4.46436	0.89273
std	1.42370	8.13282	3.25289
min	-78.41969	-72.99412	-78.44842
25%	-0.50782	-2.11399	-0.93377
50%	0.06160	9.08121	0.77409
75%	0.63538	9.70969	2.74907
max	66.87346	75.05589	78.05576

Tabel 4.2 merupakan Hasil analisis data sensor Accelerometer, analisis menunjukkan bahwa nilai rata-rata dari ketiga fitur accelerometer (acceleration_x, *acceleration_y*, dan *acceleration_z*) mendekati nol, yang berarti nilai accelerometer cenderung seimbang antara percepatan positif dan negatif

selama perjalanan. Namun, nilai standar deviasi yang bervariasi menunjukkan bahwa data accelerometer memiliki tingkat variabilitas yang berbeda-beda pada ketiga sumbu. Selain itu, ditemukan beberapa nilai ekstrim yang tidak wajar, seperti nilai maksimum dan minimum yang mencapai -78 dan 78 pada fitur *acceleration_x dan acceleration_z*. Hal ini menunjukkan adanya kemungkinan adanya data outlier atau kesalahan dalam pengukuran, sehingga perlu dilakukan handling data outlier untuk memastikan keandalan analisis. Summary ini memberikan gambaran awal tentang distribusi dan karakteristik data *accelerometer* pada perjalanan, namun perlu dilakukan analisis lebih lanjut dan preprocessing data untuk memastikan kevalidan dan kegunaan hasil analisis tersebut.

B. Analisis deskripsi data sensor *Gyroscope*

Berikut adalah analisis dari hasil deskripsi data untuk fitur gyro (*Gyroscope*) dalam sumbu x, y, dan z:

Tabel 4. 3. Deskripsi data sensor Gyroscope

Features	*gyro_x*	*gyro_y*	*gyro_z*
count	16154418	16154418	16154418
mean	-0.00171	0.00027	-0.00025
std	0.14450	0.33988	0.14801
min	-48.45575	-74.88861	-53.55445
25%	-0.02678	-0.02995	-0.01876
50%	-0.00064	0.00026	-0.00003
75%	0.02330	0.03142	0.01823
max	39.83975	80.31496	66.30078

Dari hasil deskripsi statistik di atas, pada tabel 4.3 dapat dilihat bahwa nilai rata-rata dari ketiga fitur *gyroscope* (gyro_x, gyro_y, dan gyro_z) sangat mendekati nol, yang menunjukkan perangkat relatif stabil atau mengalami sedikit rotasi dalam sumbu-sumbu tersebut. Standar deviasi yang relatif kecil menunjukkan bahwa nilai-nilai *gyroscope* cenderung berkumpul di sekitar rata-ratanya namun, terdapat beberapa nilai yang ekstrem di masing-masing fitur *gyroscope*, seperti nilai maksimum dan minimum yang jauh dari rata-rata dan kuartil-kuartil. Hal ini menunjukkan adanya beberapa pergerakan ekstrem yang diamati dalam sumbu-sumbu tersebut. Kehadiran nilai-nilai ekstrem ini perlu diperhatikan dalam analisis lebih lanjut dan dapat mengindikasikan aktivitas atau peristiwa yang menarik untuk diselidiki lebih lanjut.

C. Analisis Deskripsi Data Sensor GPS

Berikut adalah analisis dari hasil deskripsi data untuk fitur GPS dalam *Accuracy*, *Bearing*, *Speed*, dan *Second*:

Tabel 4. 4. Deskripsi data sensor GPS

Features	*Accuracy*	*Bearing*	*Speed*	*second*
count	16154418	16154418	16154418	16154418
mean	11.60744	168.97712	9.00663	3799.90486
std	86.86924	107.29621	8.10629	1435847.7
min	0.75000	0.00000	-2.00000	0.00000
25%	3.90000	78.00000	1.02000	241.00000
50%	4.25500	168.96212	7.53000	520.00000
75%	8.00000	263.00000	15.48000	863.00000
max	6070.10100	359.99948	148.01863	1.495.796.757

Tabel 4.4 menunjukkan hasil analisis nilai deskripsi data sensor GPS menunjukkan bahwa terdapat beberapa data yang memiliki nilai yang ekstrem dan mungkin tidak wajar untuk fitur GPS seperti akurasi, arah (*bearing*), kecepatan (*speed*), dan waktu (*second*). Nilai-nilai yang ekstrim seperti ini perlu diperiksa lebih lanjut untuk memastikan keabsahan data dan kemungkinan adanya data outlier atau kesalahan dalam pengambilan data. Sebagai contoh, pada fitur "*Accuracy*", nilai maksimum mencapai 6070,101. Nilai ini mungkin tidak wajar karena umumnya akurasi GPS pada perangkat smartphone biasanya tidak melebihi beberapa puluh meter. Demikian pula, pada fitur "*Second*", terdapat nilai minimum -2 dan maksimum sebesar 1.495.796.757,00. Nilai ini tidak mungkin terjadi dalam data waktu perjalanan, sehingga perlu dilakukan pemeriksaan lebih lanjut. Penting untuk melakukan validasi lebih lanjut terhadap data-data tersebut dan memastikan apakah nilai-nilai tersebut merupakan data outlier yang perlu dihapus atau ada alasan yang sah untuk nilai-nilai ekstrem tersebut. Jika diperlukan, data outlier dapat diatasi melalui teknik-teknik seperti penghapusan outlier, substitusi nilai, atau transformasi data.

Berdasarkan hasil analisis deskripsi dataset yang dimiliki, di temukan data di beberapa fitur yang tidak normal diantaranya fitur *acceleration_x*, acceleration_z, *Accuracy*, dan *Second* sehingga perlu dilakukan analisa distribusi data untuk melihat apakah benar data yang tersedia tidak normal.

4.3.2.2. Analisis Distribusi Data

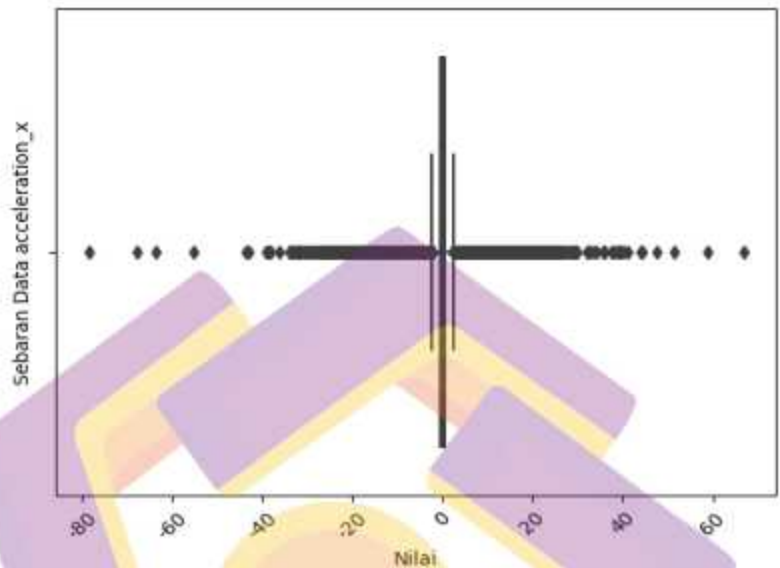
Berdasarkan dari hasil Statistik Ringkasan terdapat distribusi data yang tidak biasa atau *Outlier* pada fitur *Accuracy* dan *second* oleh sehingga perlu

dilihat lebih mendalam terkait visualisasi datanya untuk melihat distribusi dari kedua fitur ini. Berikut beberapa yang dapat dilakukan dalam Visualisasi Distribusi.

A. *Box Plot*

Box Plot dapat mengetahui apakah ada kelompok data yang memiliki nilai yang sangat rendah atau tinggi, yang mungkin mengindikasikan masalah atau keanehan dalam dataset berikut adalah script yang digunakan.

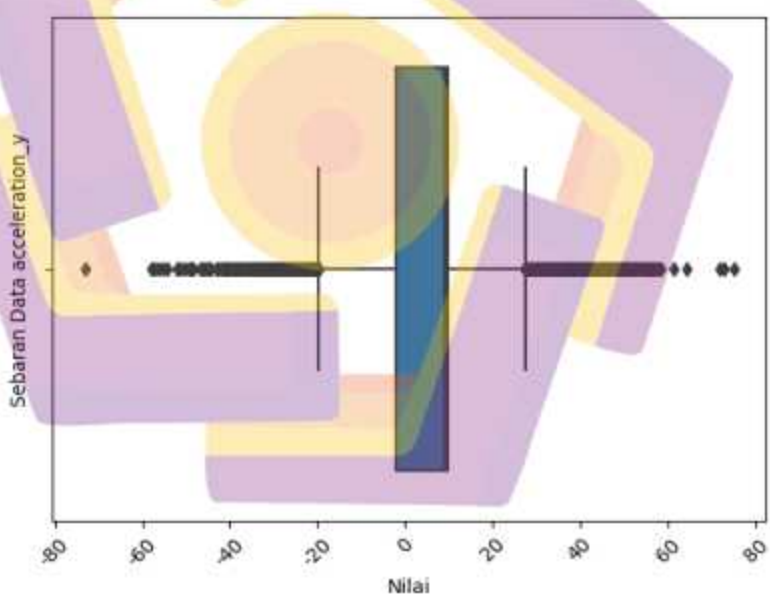
```
# Visualisasi distribusi kolom 'second'
sns.boxplot(x=data['fitur'])
plt.title('Box Plot Distribusi Data')
plt.xlabel('Nilai')
plt.ylabel('Sebaran Data')
# Mengatur label sumbu x dengan format '{:.0f}'
plt.xticks(rotation=45)
plt.gca().xaxis.set_major_formatter('{:.0f}'.format)
plt.show()
```



Gambar 4. 2. Distribusi data Acceleration sumbu x

Gambar 4.2 merupakan hasil Boxplot distribusi data *acceleration_x*, dilihat dari distribusi datanya dominan data tersebar di *range* nilai antara -5 hingga 4, mayoritas data tersebar di nilai 0. Sebagian data tersebar di *range* nilai antara -5 hingga -40 untuk nilai negatif dan *range* nilai antara 5 hingga 30 untuk nilai positif, kemudian sebagian kecil tersebar di di *range* nilai antara -41 hingga -80 untuk nilai negatif dan *range* nilai antara 31 hingga 65 untuk nilai positifnya. Berdasarkan hasil Boxplot distribusi data *acceleration_x*, Analisis menggambarkan hasil Boxplot distribusi data *acceleration_x* dengan sangat baik mengidentifikasi beberapa aspek penting dari distribusi data ini diantaranya seperti dominan pada Nilai 0, mayoritas data berada di sekitar nilai 0, yang terlihat dari kotak (box) yang berpusat di sekitar nilai tersebut. Ini menunjukkan

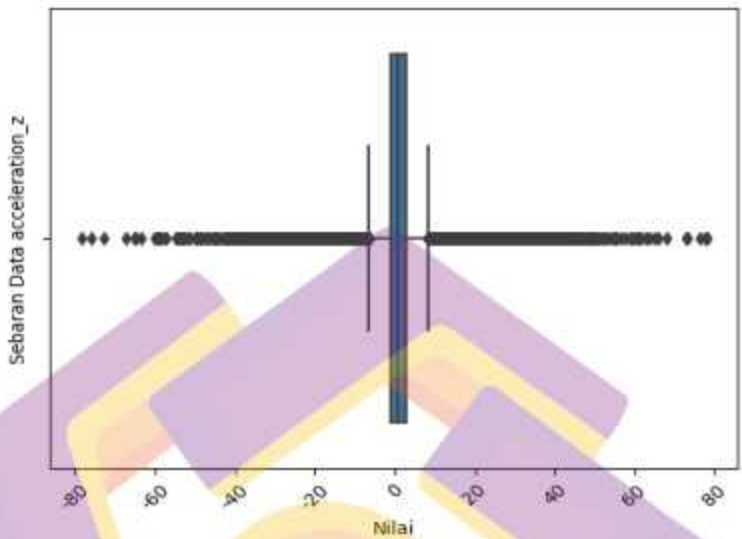
bahwa data cenderung berkumpul di sekitar nilai 0. Rentang Nilai Positif dan Negatif, hasil pengamatan bahwa ada sebagian data yang tersebar di rentang nilai negatif (kurang dari 0) dari sekitar -5 hingga -80, serta di rentang nilai positif (lebih dari 0) dari sekitar 5 hingga 65. Hal ini mengindikasikan variasi dalam data, dengan beberapa data memiliki nilai yang lebih ekstrem. *Outliers*, hasil tidak secara khusus menyebutkan *outlier* dalam deskripsi, tetapi dari hasil Box Plot, selanjutnya peneliti dapat mengidentifikasi adanya outlier berdasarkan titik-titik di luar janggut (*whisker*). *Outlier-outlier* ini mungkin memiliki nilai yang sangat ekstrem, di luar rentang yang dianggap normal.



Gambar 4. 3. Distribusi data Acceleration sumbu y

Gambar 4.3 merupakan hasil *Boxplot* distribusi data *acceleration_y*, dilihat dari distribusi datanya dominan data tersebar di *range* nilai antara -2

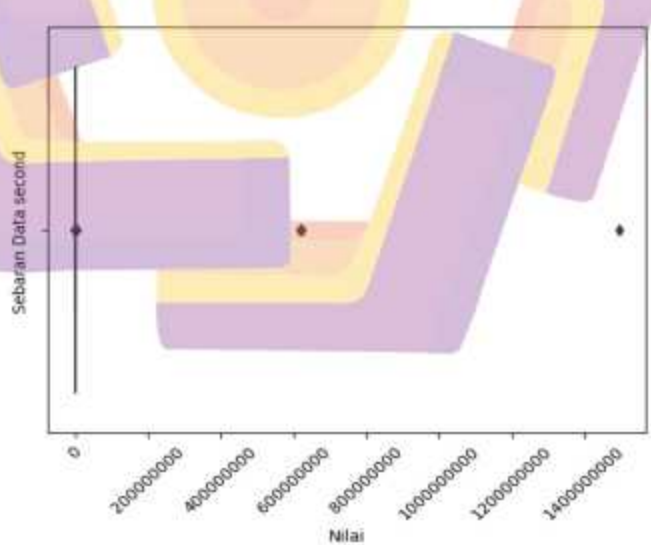
hingga 10, mayoritas data tersebar di nilai 3. Sebagian data tersebar di *range* nilai antara -21 hingga -60 nilai terbanyakanya di -20 untuk nilai negatif dan *range* nilai antara 28 hingga 65 yang mayoritas datanya tersebar di nilai 27 untuk nilai positif, kemudian sebagian kecil tersebar di di *range* nilai antara -3 hingga -19 untuk nilai negatif dan *range* nilai antara 31 hingga 65 untuk nilai positifnya. Berdasarkan analisis dari hasil *Boxplot* distribusi data acceleration_y, dapat dikatakan bahwa data memiliki variasi yang cukup signifikan. Meskipun rentang nilai dominan data terletak antara -2 hingga 10 dengan median sekitar 3, terdapat beberapa nilai outlier di luar rentang ini, terutama pada nilai -20. Selain itu, terdapat dua puncak konsentrasi data yang signifikan di sekitar nilai 27 dan 3. Namun, perlu diingat bahwa variasi yang besar pada rentang nilai -21 hingga -60 dan 28 hingga 65 mengindikasikan keberadaan data yang jauh dari rentang dominan. Ini mungkin menunjukkan adanya variasi ekstrim yang mempengaruhi sebagian data. Oleh karena itu, data *acceleration_y* perlu diteliti lebih lanjut untuk memahami faktor-faktor yang menyebabkan variasi tersebut. Kualitas data dapat dinyatakan baik jika distribusi data berada dalam rentang yang konsisten dan tidak terlalu beragam, serta tidak ada nilai ekstrim yang signifikan. Dalam hal ini, variasi yang cukup luas dan keberadaan *outlier* mungkin mempengaruhi kualitas data. Evaluasi lebih lanjut, termasuk identifikasi penyebab variasi dan *outlier*, akan membantu memutuskan apakah data tersebut memiliki kualitas yang baik untuk tujuan analisis tertentu.



Gambar 4. 4. Distribusi data Acceleration sumbu z

Gambar 4.4 merupakan hasil *Boxplot* distribusi data *acceleration_z*, dilihat dari distribusi datanya dominan data tersebar di *range* nilai antara -2 hingga 4, mayoritas data tersebar di nilai 1. Sebagian data tersebar di *range* nilai antara -3 hingga -55 nilai terbanyakanya di -5 untuk nilai negatif dan *range* nilai antara 5 hingga 68 yang mayoritas datanya tersebar di nilai 6 untuk nilai positif, kemudian sebagian kecil tersebar di di *range* nilai antara -36 hingga -80 untuk nilai negatif dan *range* nilai antara 69 hingga 80 untuk nilai positifnya. Dari analisis distribusi data acceleration_z melalui Boxplot, tergambar bahwa sebagian besar data berkumpul dalam rentang nilai -2 hingga 4, dengan puncak terbanyak pada angka 1. Namun, nilai-nilai ekstrem juga terlihat. Di sisi negatif, data tersebar dalam rentang -3 hingga -55, dengan puncak terbanyak pada -5. Pada sisi positif, data memiliki sebaran antara 5 hingga 68, dengan puncak utama

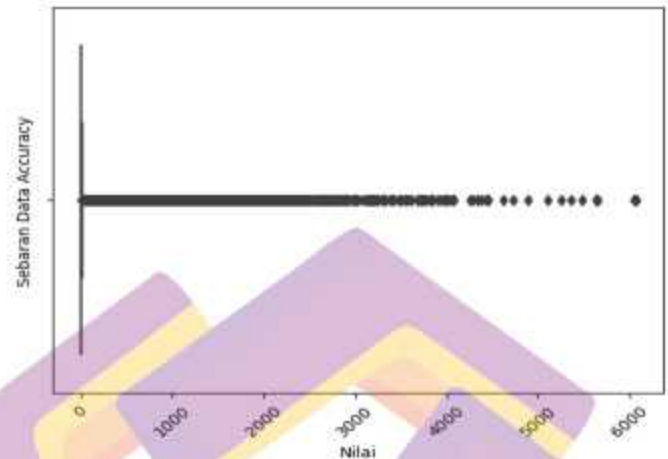
pada angka 6. Tersedia juga sejumlah data ekstrem, mencakup nilai rendah di rentang -36 hingga -80, dan nilai tinggi di rentang 69 hingga 80. Analisis ini mengindikasikan variasi data yang signifikan dan potensi adanya nilai-nilai ekstrim yang bisa mempengaruhi interpretasi dan model analisis lebih lanjut. Dari analisis data *acceleration_z*, dapat disimpulkan bahwa distribusi nilai-nilai ini tidak mengikuti pola distribusi normal. Terlihat bahwa data memiliki beberapa puncak atau modus yang mencerminkan variasi dalam rentang nilai. Selain itu, nilai-nilai ekstrem yang tersebar di kedua sisi distribusi menunjukkan adanya potensi anomali atau outlier dalam data. Oleh karena itu, dalam pengolahan data *acceleration_z*, langkah-langkah untuk mengidentifikasi dan mengatasi *outlier* serta memahami variasi pola data menjadi kunci penting untuk menghasilkan hasil analisis yang akurat dan informasi yang berarti.



Gambar 4. 5. Distribusi data Second GPS

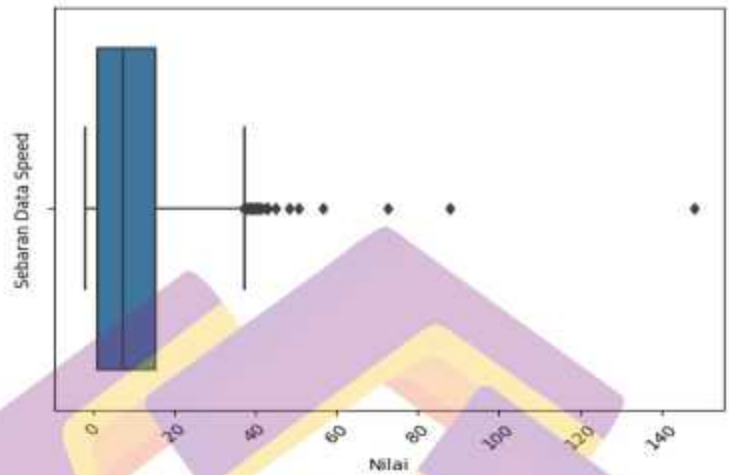
Gambar 4.5 merupakan hasil *Boxplot* distribusi data Waktu pada atribut sensor GPS, dilihat dari distribusi datanya dominan data tersebar di *range* nilai antara 0 hingga 8200 detik, ada empat data perjalanan memiliki nilai 0 hingga 61.9316.568 untuk booking ID 858993459333, 1460288880770, 1434519076976, 1108101562401 dan satu perjalanan dengan rentang waktu waktu 0 hingga 1.495.796.757 untuk *booking ID* 1503238553722. Beberapa data yang kami tandai booking IDnya merupakan perjalan dengan waktu yang tidak wajar. Analisis data waktu pada atribut sensor GPS menghasilkan beberapa temuan kunci. Distribusi data waktu menunjukkan mayoritas perjalanan memiliki rentang waktu yang relatif singkat, dominan antara 0 hingga 8200 detik.

Namun, temuan yang mencolok adalah adanya data perjalanan dengan nilai waktu yang mencurigakan. Empat perjalanan memiliki rentang waktu yang mencapai 61.9316.568 detik, sedangkan satu perjalanan memiliki rentang waktu yang sangat ekstrem, mencapai 1.495.796.757 detik. Data dengan nilai waktu 0 juga muncul, yang mengindikasikan adanya ketidakwajaran atau kesalahan dalam pengukuran atau pengumpulan data. Tindakan yang diambil untuk menandai data dengan nilai waktu yang tidak wajar adalah langkah yang tepat, karena data semacam ini dapat merusak validitas dan hasil analisis lebih lanjut. Validasi dan pemeriksaan mendalam terhadap data waktu yang mencurigakan akan memberikan pemahaman yang lebih baik tentang sifat sebenarnya dari perjalanan dan pola sensor GPS yang terkait.



Gambar 4. 6. Distribusi data Accuracy GPS

Gambar 4.6 merupakan hasil Boxplot distribusi data *Accuracy* pada atribut sensor GPS, dilihat dari distribusi datanya dominan data tersebar di *range* nilai antara 0 - 8 berdasarkan nilai pengukuran deskripsi dataset 0 - 75 % data. Kemudian 25% sisanya sebaran data 9 - nilai maksimal 6070. Dengan melihat persebaran data tersebut, dapat disimpulkan bahwa distribusi data *Accuracy* tidak mengikuti distribusi normal. Adanya sejumlah data yang memiliki nilai yang jauh di atas rentang mayoritas dapat dianggap sebagai *outlier* atau nilai ekstrim. Hal ini menunjukkan bahwa data *Accuracy* memiliki variasi yang cukup besar dan tidak terdistribusi secara merata. Oleh karena itu, data *Accuracy* pada atribut sensor GPS tidak dapat dianggap sebagai data yang normal. Sebaiknya dilakukan analisis lebih lanjut untuk memahami faktor-faktor yang menyebabkan nilai-nilai ekstrim tersebut dan apakah ada pengaruh signifikan terhadap analisis yang akan dilakukan.



Gambar 4. 7. Distribusi data Speed GPS

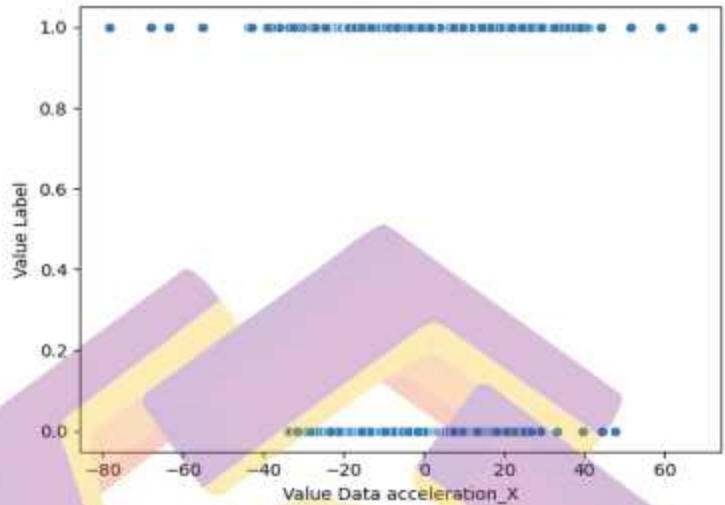
Gambar 4.7 merupakan hasil Boxplot distribusi data *Speed* pada atribut sensor GPS, dilihat dari distribusi datanya dominan data tersebar di *range* nilai antara 1 - 15 berdasarkan nilai pengukuran deskripsi dataset 0 - 75 % data. Kemudian kecepatan tertingginya adalah 148, akan tetapi dalam data kecepatan di temukan nilai minimum -2. Ketika melihat rentang nilai dari 0 hingga 75% data, yaitu 1 hingga 15, dan kemudian melihat nilai maksimal 148, maka hal ini menunjukkan adanya sejumlah data yang berada di luar rentang mayoritas. Selain itu, adanya nilai minimum yang negatif (-2) juga menunjukkan variasi data yang mencakup nilai yang tidak umum dalam konteks kecepatan. Dengan demikian, data *Speed* pada atribut sensor GPS tidak dapat dianggap sebagai data yang normal. Adanya nilai ekstrim baik pada sisi maksimal maupun minimal menunjukkan bahwa distribusi data Speed memiliki variasi yang signifikan dan tidak mengikuti pola distribusi normal. Sebaiknya dilakukan analisis lebih lanjut

untuk memahami faktor-faktor yang menyebabkan nilai-nilai ekstrim tersebut dan bagaimana dampaknya terhadap analisis yang akan dilakukan.

B. *Scatter Plot*

Scatter Plot adalah jenis visualisasi yang digunakan untuk menampilkan hubungan antara dua variabel atau fitur dalam bentuk titik-titik pada bidang koordinat, di sini kami ingin melihat hubungan antara fitur *second* dan *Accuracy* terhadap *label*. *Scatter Plot* sangat berguna untuk melihat pola, korelasi, dan sebaran data, terutama ketika ingin mencari tahu apakah ada hubungan atau tren tertentu antara kedua variabel tersebut berikut script yang digunakan.

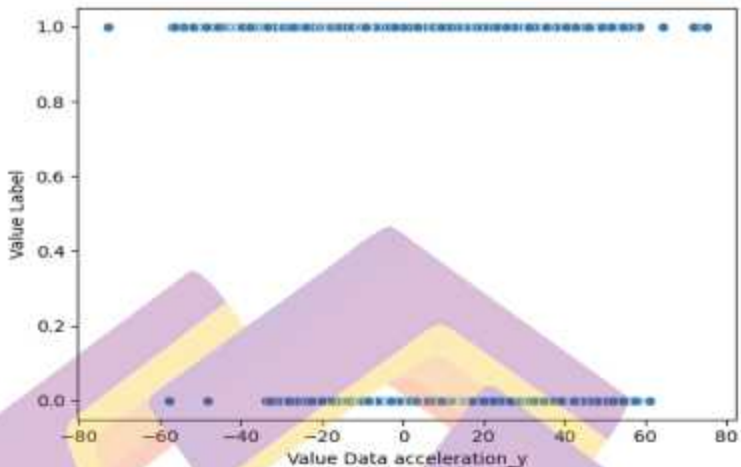
```
import matplotlib.pyplot as plt
import seaborn as sns
# Visualisasi hubungan antara kolom 'fitur' dan
'label'
sns.scatterplot(x='fitur', y='label', data=data)
plt.title('Scatter Plot Hubungan fitur dan Label')
plt.xlabel('Value Data fitur')
plt.ylabel('Value Label')
# Mengatur label sumbu x dengan format '{:.0f}'
plt.xticks(rotation=45)
plt.gca().xaxis.set_major_formatter('{:.0f}'.format)
plt.show()
```



Gambar 4. 8. Scatter Plot Acceleration_x

Gambar 4.8 adalah hasil *Scatter Plot* yang menggambarkan distribusi data *acceleration_X* pada atribut sensor *Accelerometer*. Dilihat dari distribusi data tersebut, pada label 1 terdapat persebaran nilai yang lebih luas, yaitu dari nilai 0 hingga -80 untuk nilai negatif dan dari 0 hingga 60 untuk nilai positif. Sedangkan pada label 0, data cenderung berkumpul di sekitar nilai 0 hingga -35 untuk nilai negatif, dan dari 0 hingga 50 untuk nilai positif.

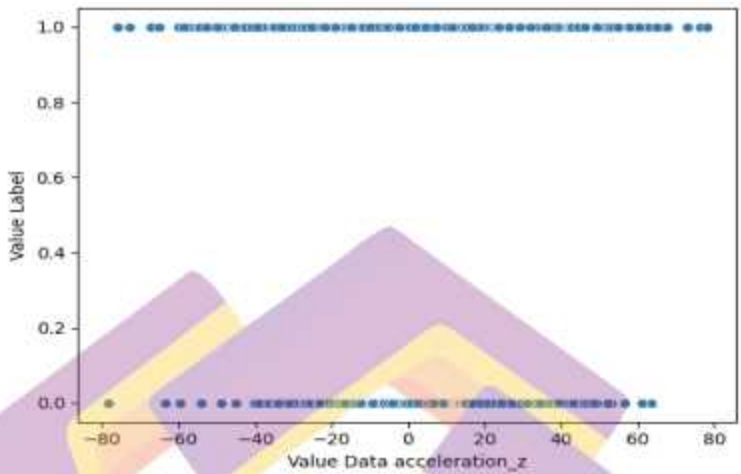
Dari hasil *Scatter Plot* di atas, terlihat bahwa distribusi nilai data pada kedua label, yaitu label 0 dan label 1, masih cukup serupa dalam rentang nilai yang sama. Meskipun terdapat beberapa data yang tersebar cukup luas pada label 1, namun secara umum data tersebut masih terlihat normal. Namun, perlu diperhatikan bahwa sebaran data pada rentang -50 hingga -80 kemungkinan merupakan outlier.



Gambar 4. 9. Scatter Plot Acceleration_y

Gambar 4.9 menunjukkan hasil *Scatter Plot* yang menggambarkan distribusi data *acceleration_y* pada atribut sensor *Accelerometer*. Dari distribusi data tersebut, terlihat bahwa pada label 1 terdapat persebaran nilai yang lebih luas dan seimbang, yaitu dari nilai 0 hingga -80 untuk nilai negatif dan dari 0 hingga 80 untuk nilai positif. Sedangkan pada label 0, data cenderung berkumpul di sekitar nilai 0 hingga 60 untuk nilai positif, dan dari 0 hingga -40 untuk nilai negatif dengan kecenderungan data label 0 tersebar di nilai positif. Terlihat bahwa distribusi nilai data pada kedua label, yaitu label 0 dan label 1, masih cukup serupa dalam rentang nilai yang sama.

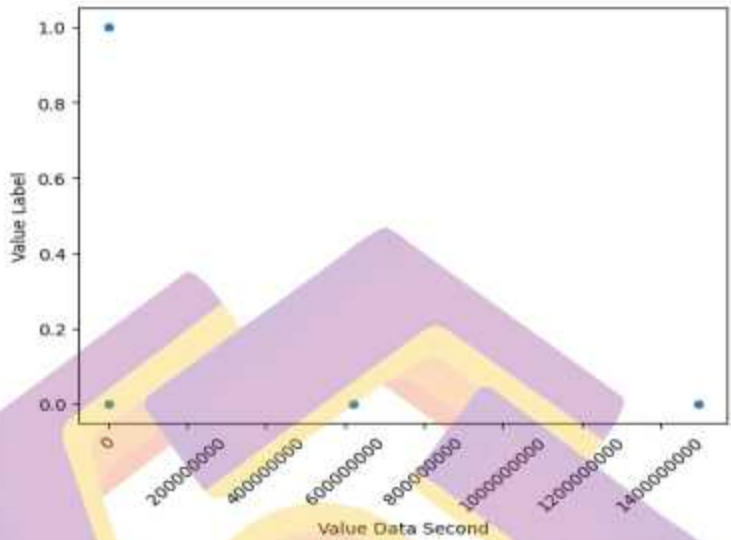
Meskipun terdapat beberapa data yang tersebar cukup luas pada label 0, namun secara umum data tersebut masih terlihat normal. Namun, perlu diperhatikan bahwa sebaran data label 0 pada rentang -40 hingga -60 dan pada label 1 pada rentan -60 hingga -80 kemungkinan merupakan outlier



Gambar 4. 10. Scatter Plot Acceleration_z

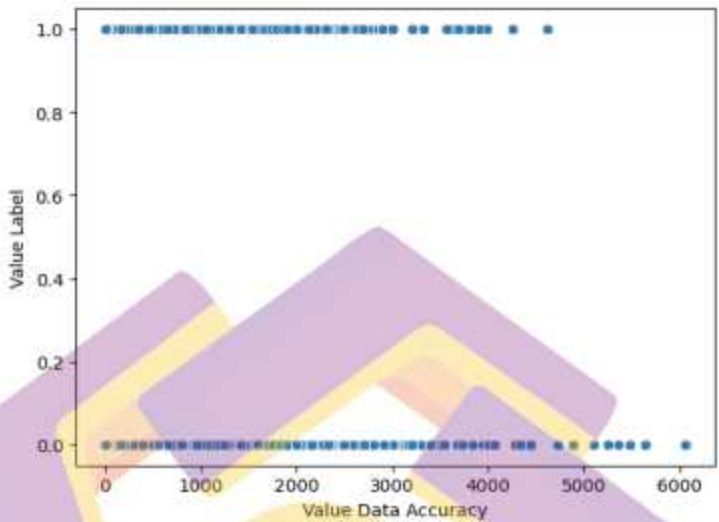
Gambar 4.10 adalah hasil *Scatter Plot* yang menggambarkan distribusi data *acceleration_z* pada atribut sensor *Accelerometer*. Dilihat dari distribusi data tersebut, pada label 1 terdapat persebaran nilai yang lebih luas, yaitu dari nilai 0 hingga -80 untuk nilai negatif dan dari 0 hingga 80 untuk nilai positif. Sedangkan pada label 0, data cenderung berkumpul di sekitar nilai 0 hingga -60 untuk nilai positif, dan dari 0 hingga -40 untuk nilai negatif.

Dari hasil *Scatter Plot* di atas, terlihat bahwa distribusi nilai data pada kedua label, yaitu label 0 dan label 1, masih cukup serupa dalam rentang nilai yang sama. Meskipun terdapat beberapa data yang tersebar cukup luas pada label 1, namun secara umum data tersebut masih terlihat normal. Namun, perlu diperhatikan bahwa sebaran data label 0 pada rentang -40 hingga -80 kemungkinan merupakan outlier.



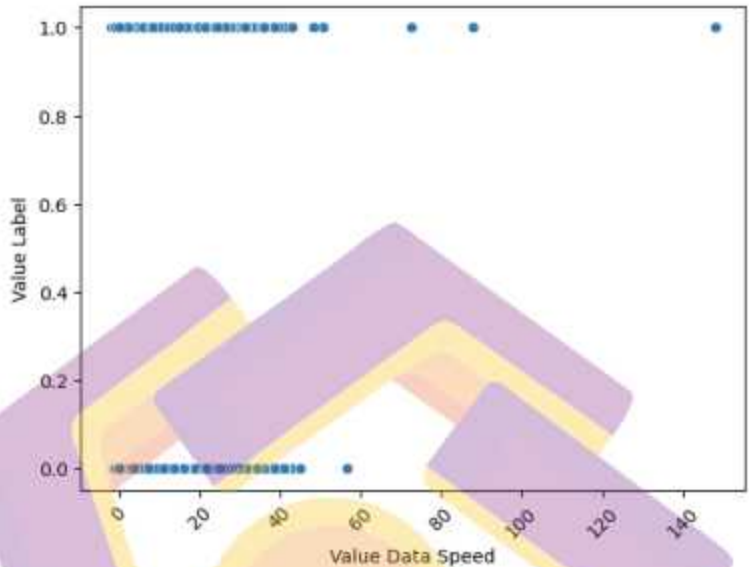
Gambar 4. 11. Scatter Plot Second

Gambar 4.11 menunjukkan hasil Scatter Plot yang menggambarkan distribusi data Second pada atribut sensor GPS. Dari distribusi data tersebut, terlihat bahwa pada label 0 terdapat persebaran nilai yang lebih luas, yaitu dari nilai 0 hingga 1.400.000.000. Sedangkan pada label 1, data cenderung berkumpul di sekitar nilai 0 hingga 100. Terlihat bahwa distribusi nilai data pada kedua label, yaitu label 0 dan label 1, tidak normal pada label 0 pada nilai 600.000.000 dan 1.400.000.000 sehingga perlu dilakukan preprocess data lebih lanjut.



Gambar 4. 12. Scatter Plot Accuracy

Gambar 4.12 menunjukkan hasil *Scatter Plot* yang menggambarkan distribusi data *Accuracy* pada atribut sensor GPS. Dari distribusi data tersebut, terlihat bahwa pada label 0 terdapat persebaran nilai yang lebih luas, yaitu dari nilai 0 hingga 6000. Sedangkan pada label 1, data cenderung berkumpul di sekitar nilai 0 hingga 4000. Terlihat bahwa distribusi nilai data pada kedua label, yaitu label 0 dan label 1 cenderung tidak normal karena terdapat beberapa data yang tersebar cukup luas pada label 0, perlu diperhatikan bahwa sebaran data label 0 pada rentang 4000 hingga 6000 kemungkinan merupakan outlier



Gambar 4. 13. Scatter Plot Speed

Gambar 4.13 menunjukkan hasil *Scatter Plot* yang menggambarkan distribusi data *Speed* pada atribut sensor GPS. Dari distribusi data tersebut, terlihat bahwa pada label 1 terdapat persebaran nilai yang lebih luas, yaitu dari nilai 0 hingga 140. Sedangkan pada label 0, data cenderung berkumpul di sekitar nilai 0 hingga 40. Terlihat bahwa distribusi nilai data pada kedua label, yaitu label 0 dan label 1, tidak pada rentan data yang sama terdapat beberapa data yang tersebar cukup luas pada label 1, perlu diperhatikan bahwa sebaran data label 1 pada rentang 60 hingga 140 kemungkinan merupakan outlier

Dari dua visualisasi data yang telah dilakukan pada fitur second dan accuracy ditemukan hasil persebaran data yang tidak normal pada fitur *second*, dapat kita lihat hasil menunjukkan lompatan value data yang aneh dan tidak bisa,

sedangkan pada fitur Accuracy sebaran data masih terlihat normal. Dari hasil analisis sebaran data yang dilakukan perlu dilakukan normalisasi pada fitur *second*.

4.3.2.3. Korelasi

Korelasi antara fitur-fitur dalam konteks analisis data merujuk pada sejauh mana dua atau lebih fitur atau variabel dalam dataset saling berhubungan. Ini membantu Anda memahami bagaimana perubahan dalam satu fitur berkaitan dengan perubahan dalam fitur lainnya. Korelasi dapat menjadi positif, negatif, atau netral (tidak ada hubungan). Berikut adalah script yang digunakan untuk matrik korelasi

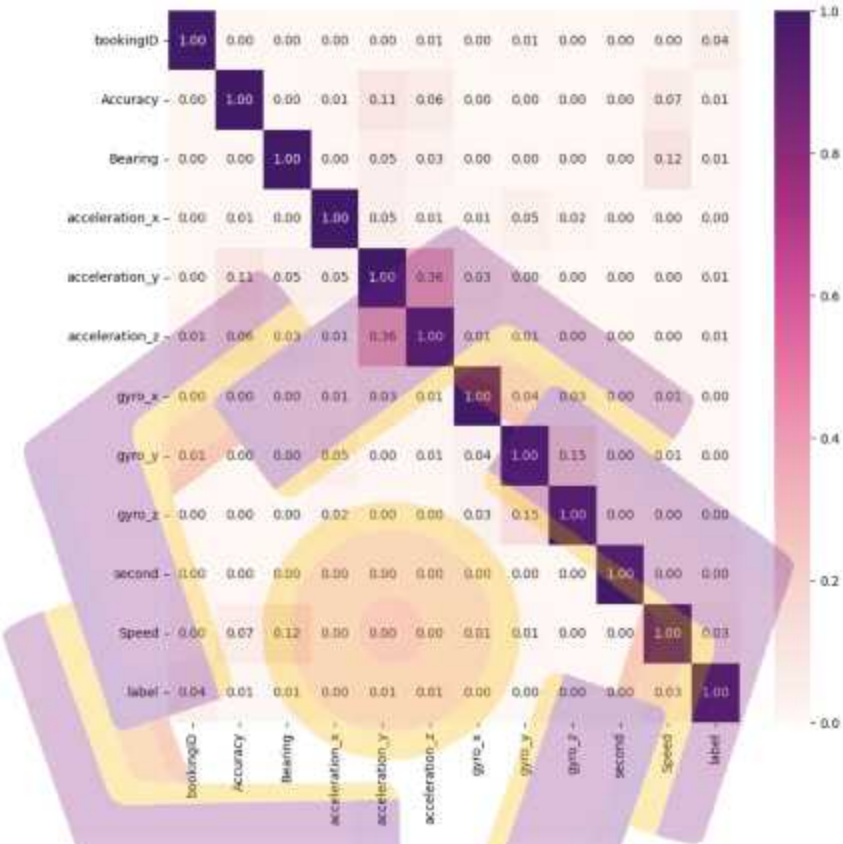
```
import seaborn as sb f, axes = plt.subplots(1, 1,
figsize=(10, 10))
sb_heatmap =
sb.heatmap(data_drop_outliers_secon_speed_accuacy.cor
r().abs(), vmin = 0, vmax = 1, annot = True, fmt =
".2f", cmap="RdPu", cbar=True)
```

Dengan menggunakan script diatas dapat diketahui Jenis-jenis korelasi yang umum digunakan meliputi:

1. Korelasi Positif: Ini terjadi ketika nilai dua fitur bergerak searah. Artinya, ketika nilai satu fitur naik, nilai fitur lainnya juga cenderung naik. Korelasi positif biasanya berkisar antara 0 hingga 1, dengan 1 menunjukkan hubungan yang sangat kuat. Dalam dataset penelitian ini terdapat korelasi Positif untuk fitur *Acceleration_y* - *Acceleration_z* memiliki nilai 0,36, *Accuracy* - *Acceleration_y* memiliki nilai 0,11,

gyro_y - gyro_z dengan nilai kemudian *speed - Bearing* memiliki nilai 0,12.

2. Korelasi Negatif: Ini terjadi ketika nilai dua fitur bergerak berlawanan arah. Artinya, ketika nilai satu fitur naik, nilai fitur lainnya cenderung turun, dan sebaliknya. Korelasi negatif biasanya berkisar antara -1 hingga 0, dengan -1 menunjukkan hubungan yang sangat kuat.
3. Korelasi Nol: Ini berarti tidak ada hubungan yang jelas antara dua fitur. Perubahan dalam satu fitur tidak mempengaruhi perubahan dalam fitur lainnya, Sebagai contoh dalam dataset ini *Bearing - Accuracy, Accuracy - gyro_x* dan lainnya dapat dilihat pada gambar dibawah ini.



Gambar 4. 14. Matrik Korelasi

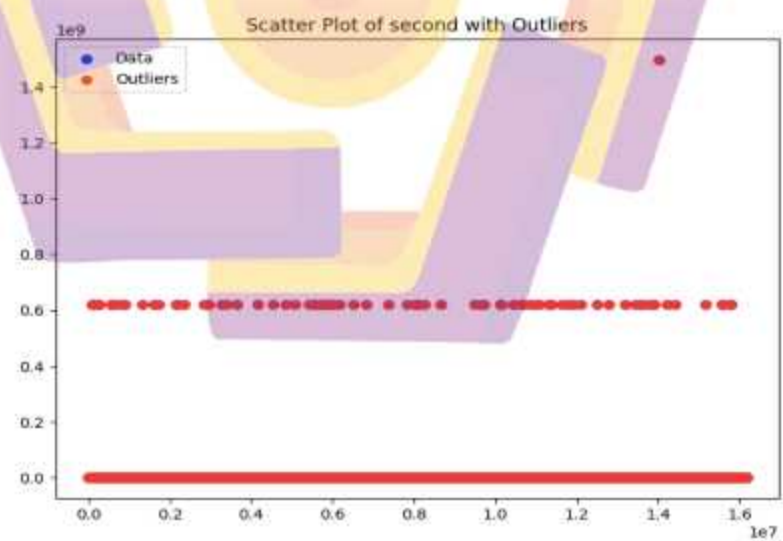
Gambar 4.14 adalah Matrik Korelasi EDA yang membantu memahami data dengan lebih baik, menemukan pola menarik, dan membantu mempersiapkan data untuk tugas-tugas analisis lebih lanjut seperti klasifikasi atau regresi. Selain itu, EDA juga membantu mengidentifikasi asumsi yang mungkin perlu diuji atau pertimbangkan selama proses pemodelan data.

4.4. Data Preprocessing

4.4.1. Deteksi Outlier

Berikut adalah strategi dan teknik untuk mengatasi nilai-nilai *outlier* dalam pemodelan data. *Outlier*, atau nilai yang jauh dari pola umum dalam suatu dataset, dapat memiliki dampak signifikan terhadap performa model. Oleh karena itu, pemahaman dan penanganan outlier menjadi kritis dalam proses pembangunan model. Bab ini akan membahas berbagai metode dan pendekatan yang dapat digunakan untuk mengidentifikasi, menangani, dan, jika perlu, menghilangkan *outlier* agar hasil model lebih konsisten dan dapat diandalkan. Berikut adalah tahapan pengecekan data *outlier* yang sudah di sepesifikan pada fitur *second, Speed*, dan *Accuracy:*

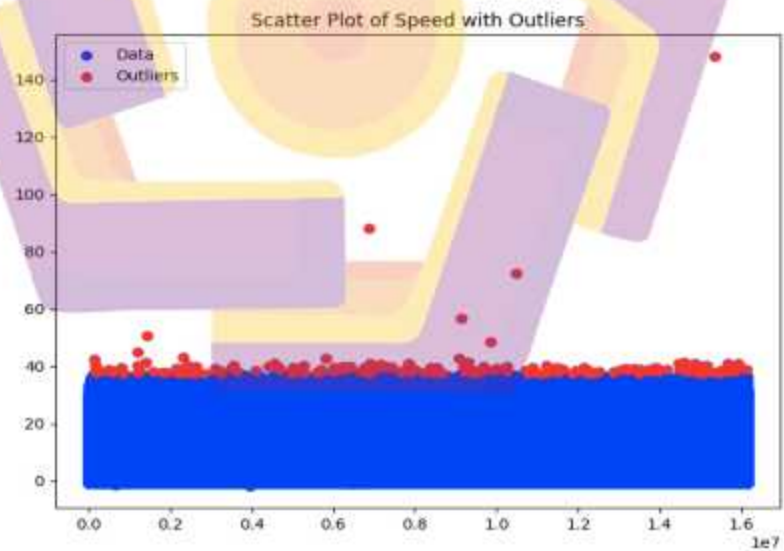
A. Deteksi *Outlier Second*



Gambar 4. 15. Deteksi Outlier Fitur Second

Berdasarkan hasil deteksi outlier pada fitur *Second* pada gambar 4.15, ditemukan bahwa distribusi *outlier* sangat dominan tersebar dalam dataset. Keberadaan outlier ini dapat mengakibatkan pergeseran yang signifikan dalam representasi data, mengurangi kualitas dan kejelasan informasi yang terkandung dalam set tersebut. Sebagai respons terhadap temuan ini, langkah-langkah Eksplorasi Data (EDA) telah diimplementasikan untuk memahami karakteristik dan pola yang mendasari penyebaran *outlier*. Oleh karena itu, dilakukan sebuah langkah *preprocessing* dengan melakukan filter pada data fitur *Second* dengan rentang waktu 0 hingga 2000. Langkah ini diambil dengan pertimbangan untuk menghilangkan dampak outlier yang signifikan.

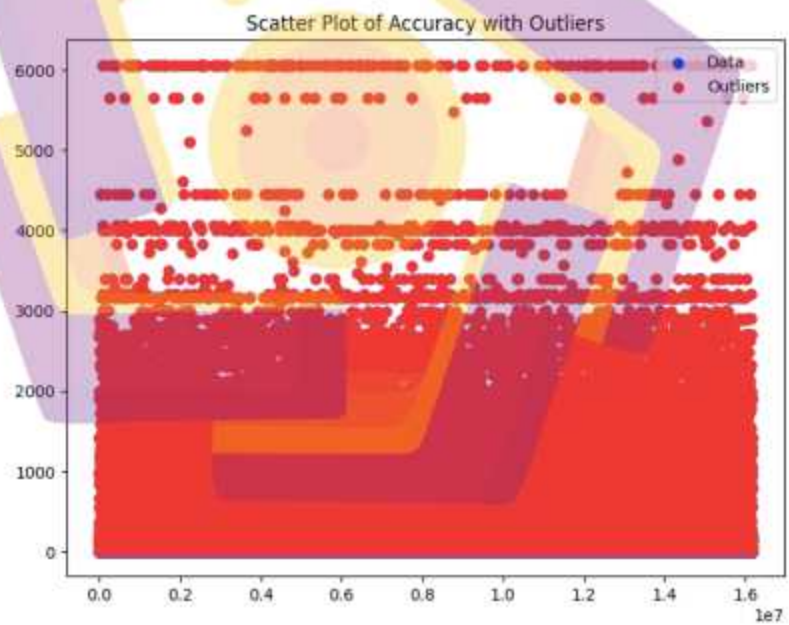
B. Deteksi Outlier Speed



Gambar 4. 16. Deteksi Outlier Fitur Speed

Dari hasil pengamatan pada proses deteksi *outlier* pada fitur *Speed* pada gambar 4.16, terlihat adanya nilai-nilai *outlier* yang terkonsentrasi pada rentang data 40 hingga 140. Melalui analisis *Exploratory Data Analysis* (EDA), ditemukan bahwa terdapat nilai minimal pada data *Speed* yang mencapai -2, yang juga dianggap sebagai *outlier*. Temuan ini menunjukkan ketidaksesuaian data terhadap pola umum dan dapat merusak validitas analisis statistik yang dilakukan. Oleh karena itu, langkah-langkah preprocessing lebih lanjut diperlukan untuk mengatasi dampak dari *outlier* tersebut.

C. Deteksi *Outlier Accuracy*



Gambar 4. 17. Deteksi Outlier Fitur Accuracy

Gambar 4.17 merupakan hasil analisis data sensor GPS. Analisis awal mengindikasikan potensi adanya data *outlier* pada fitur "*Accuracy*". Distribusi nilai pada fitur ini menunjukkan adanya nilai ekstrim, terutama dengan nilai maksimum yang mencapai 6070,101. Sebagian besar nilai cenderung berada dalam kategori *outlier*, mengacu pada rentang nilai yang tidak sesuai dengan karakteristik umum akurasi GPS pada perangkat *smartphone*. Meskipun demikian, dalam konteks ini, perlu dilakukan pemeriksaan lebih lanjut untuk memahami apakah nilai-nilai ekstrem tersebut dapat secara sah dianggap sebagai *outlier* atau mungkin disebabkan oleh faktor lain, seperti kesalahan pengukuran atau kelalaian dalam pengumpulan data. Validasi mendalam menjadi suatu keharusan untuk mengidentifikasi akar penyebab dari nilai-nilai ekstrem ini. Langkah-langkah preprocessing data yang tepat kemudian dapat diimplementasikan untuk memastikan integritas data sebelum dilakukan analisis lebih lanjut. Keseluruhan proses ini memastikan bahwa hasil analisis yang dihasilkan memberikan representasi yang akurat dan dapat diandalkan dari perjalanan yang direkam dalam dataset.

4.4.2. Cleansing Outlier

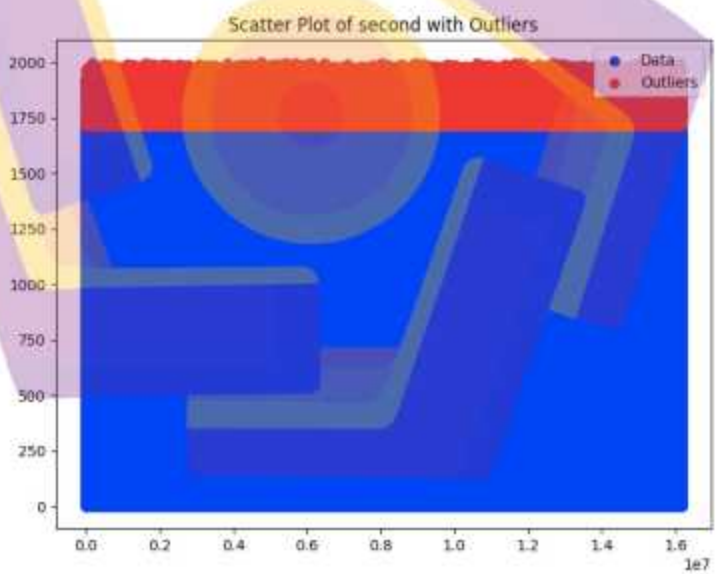
Proses pembersihan outlier (cleansing outlier) pada ketiga fitur, yaitu Second, Speed dan Accuracy menjadi langkah kritis dalam memastikan integritas data sebelum dilakukan analisis lebih lanjut. Dalam konteks fitur "Second," rentang nilai waktu yang mencakup nilai negatif dan ekstrem yang tidak mungkin dalam konteks waktu perjalanan menandakan adanya outlier.

A. Cleansing Outlier Second

Sebagai tindak lanjut, data *outlier* dapat diatasi dengan menghapus data di luar rentang yang masuk akal, dalam hal ini, data dengan nilai waktu yang kurang dari 0 dan lebih dari 2000. Berikut adalah *script* yang digunakan untuk *cleansing Outlier Second*

```
df = data
suspicious_booking_ids = df[df['second'] >
2000]['bookingID'].unique()
data_drop_outliers_secon =
df[~df['bookingID'].isin(suspicious_booking_ids)]
```

Berkut adalah hasil *cleansing* yang telah dilakukan



Gambar 4. 18. Cleansing Outlier Fitur Second

Gambar 4.18 merupakan hasil *Cleansing* dengan mengoperasikan suatu *DataFrame* menggunakan *Python*, dan nampaknya hasil tersebut ditujukan

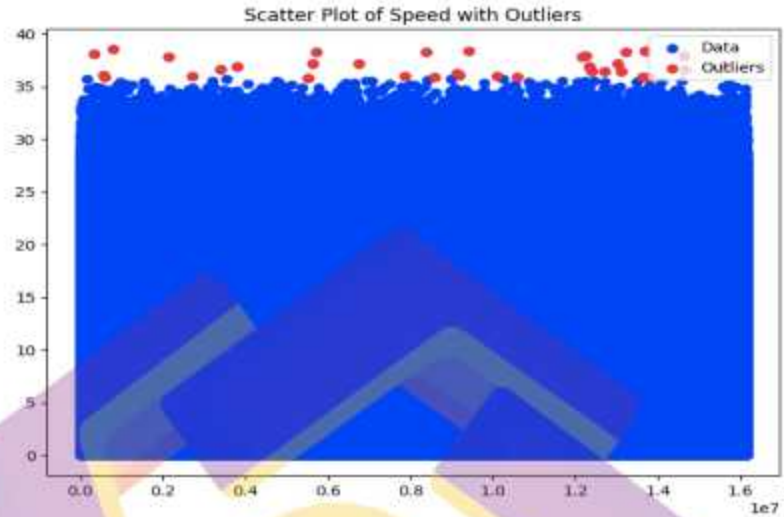
untuk membersihkan data yang memiliki nilai di kolom *'second'* lebih besar dari 2000 pada *DataFrame* yang disimpan dalam variabel data. Hasilnya adalah *DataFrame* baru yang tidak lagi memiliki data yang memiliki *second* lebih besar dari 2000 berdasarkan *booking ID*. Artinya, data yang dianggap sebagai *outliers* berdasarkan kriteria tersebut telah dihapus dari *DataFrame* baru dan terlihat dominasi data berwarna biru akan tetapi ada beberapa data berlabel merah (*outliers*).

B. Cleansing Outlier Speed

Pada fitur *Speed*, hasil analisis menunjukkan adanya outlier dalam rentang nilai 40 hingga 140. Langkah pembersihan *outlier* kemudian dilakukan dengan menghapus data yang masuk ke dalam rentang tersebut. Proses ini bertujuan untuk meningkatkan kualitas data dan memastikan bahwa nilai-nilai ekstrim yang mungkin tidak mencerminkan perjalanan yang sebenarnya tidak mempengaruhi analisis secara tidak proporsional. Berikut adalah script yang digunakan untuk *cleansing Outlier Second*

```
df = data_drop_outliers_secon
suspicious_booking_ids = df[df['Speed'] <
0]['bookingID'].unique()
data_drop_outliers_secon_speed =
df[~df['bookingID'].isin(suspicious_booking_ids)]
```

Berkut adalah hasil cleansing yang telah dilakukan



Gambar 4. 19. Cleansing Outlier Fitur Speed

Gambar 4.19 merupakan hasil *Cleansing* dengan mengoperasikan suatu *DataFrame* menggunakan *Python*, dan nampaknya hasil tersebut ditujukan untuk membersihkan data yang memiliki nilai di kolom *speed* < 0 pada *DataFrame* yang disimpan dalam variabel data. Hasilnya adalah *DataFrame* baru yang tidak lagi memiliki data yang memiliki *speed* < 0 berdasarkan *booking ID*. Artinya, data yang dianggap sebagai outliers berdasarkan kriteria tersebut telah dihapus dari DataFrame baru dan terlihat dominasi data berwarna biru.

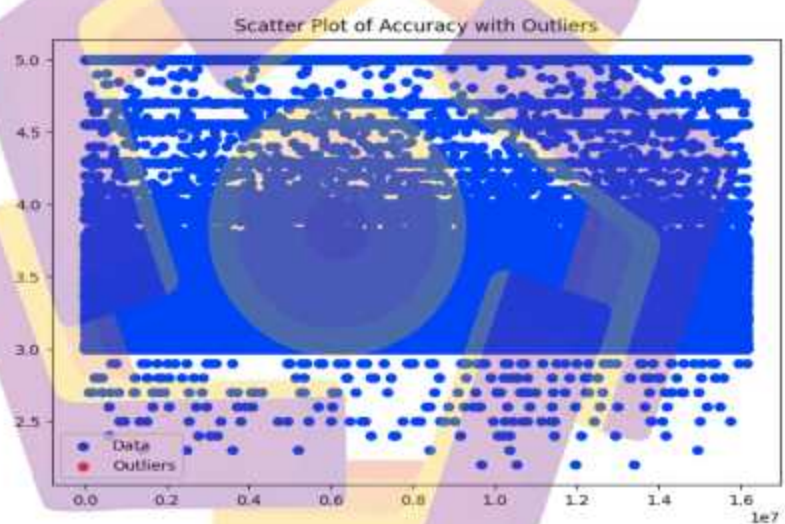
C. *Cleansing Outlier Accuracy*

Untuk fitur "*Accuracy*," nilai ekstrim yang melebihi batas umum akurasi GPS pada perangkat *smartphone* perlu mendapatkan perhatian khusus. Dengan mengidentifikasi dan menghapus data dengan nilai akurasi yang tidak masuk akal, seperti yang dicontohkan oleh nilai maksimum yang mencapai 6070,101,

Berdasarkan hasil pengamatan EDA pemilihan batas atas 8 akan tetapkan berdasarkan populasi terbanyak yaitu 75% data. Berikut adalah *script* yang digunakan untuk *cleansing Outlier Accuracy*

```
df = data_drop_outliers_secon_speed
suspicious_booking_ids = df[df['Accuracy'] >
8]['bookingID'].unique()
data_drop_outliers_secon_speed_accuacy =
df[~df['bookingID'].isin(suspicious_booking_ids)]
```

Berkut adalah hasil *cleansing* yang telah dilakukan



Gambar 4. 20. Cleansing Outlier Fitur Accuracy

Gambar 4.20 merupakan hasil *Cleansing* dengan mengoperasikan suatu *DataFrame* menggunakan *Python*, dan nampaknya hasil tersebut ditujukan untuk membersihkan data yang memiliki nilai di kolom *Accuracy* > 8 pada *DataFrame* yang disimpan dalam variabel data. Hasilnya adalah *DataFrame* baru yang tidak lagi memiliki data yang memiliki *speed* >8 berdasarkan *booking*

ID. Artinya, data yang dianggap sebagai *outliers* berdasarkan kriteria tersebut telah dihapus dari DataFrame baru dan terlihat dominasi data berwarna biru. Proses cleansing outlier ini merupakan langkah awal yang esensial dalam tahapan pra-analisis data, memastikan bahwa dataset yang digunakan bersih dari anomali yang dapat mengarah pada interpretasi yang salah.

4.5. Data Modeling

Dalam tahap klasifikasi, setelah proses *preprocessing* data *outlier* dilakukan, peneliti dapat menjalankan berbagai tugas untuk memprediksi label kategori atau kelas dari dataset perjalanan. Sebagai contoh, peneliti dapat fokus pada prediksi apakah suatu perjalanan dapat dianggap aman (label 0) atau berbahaya (label 1). Langkah pertama dalam tahap ini adalah memastikan bahwa data yang digunakan telah melewati proses cleansing outlier untuk memastikan kualitas dan integritasnya.

4.5.1. Pembagian Dataset

Selanjutnya, peneliti membagi dataset menjadi dua subset utama: data pelatihan (*training* data) dan data pengujian (*testing* data). Data training yang digunakan adalah semua fitur yang tersedia pada dataset yang tertuang pada tabel 4.1. Peneliti menggunakan semua fitur tersebut karena ingin mengetahui sensor apa saja yang berpengaruh terhadap karakteristik berkendara.

Data pelatihan digunakan untuk melatih model klasifikasi, sementara data pengujian digunakan untuk menguji kinerja model yang telah dilatih. Pemilihan model klasifikasi, seperti *Random Forest* atau *Support Vector*

Machine, dapat didasarkan pada karakteristik data dan tujuan analisis yang diinginkan. Berikut adalah *script* untuk Pembagian *Dataset*;

```
from sklearn.model_selection import train_test_split
X =
data_drop_outliers_secon_speed_accuacy.drop(['label',
'bookingID'], axis=1)
y =
data_drop_outliers_secon_speed_accuacy['label'].astyp
e(int)
# Split dataset into training and testing sets
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
random_state=42)
```

Script di atas menggunakan *library scikit-learn* untuk membagi dataset menjadi dua subset: data pelatihan (*training* data) dan data pengujian (*testing* data) dalam konteks pembangunan model klasifikasi. Fitur yang digunakan untuk pelatihan dan pengujian diambil dari dataset yang telah melalui proses pre-processing, termasuk penghapusan outlier pada fitur *second*, *speed*, dan *accuracy*. Data pelatihan, yang terdiri dari 80% dari dataset, digunakan untuk melatih model, sementara data pengujian (20% dari dataset) digunakan untuk menguji seberapa baik model tersebut dapat melakukan prediksi. Variabel X menyimpan fitur-fitur yang digunakan untuk prediksi, sedangkan variabel y menyimpan label atau kategori yang akan diprediksi, dalam hal ini label 0 (aman) atau 1 (berbahaya). Proses pemisahan dataset ini dilakukan secara acak dengan menggunakan seed (*random_state*) 42 untuk memastikan reprodusibilitas.

4.5.2. Training Model

Pembahasan mengenai "*Training Model*" memasuki tahap kritis dalam proses pengembangan model klasifikasi. Pada tahapan ini, model akan dihasilkan dan disesuaikan dengan data latih untuk dapat memahami pola dan karakteristik dari dataset. Proses training ini melibatkan penggunaan algoritma *machine learning* untuk menyesuaikan parameter-model agar dapat membuat prediksi yang optimal. Selain itu, pemilihan matrik evaluasi yang tepat juga menjadi perhatian utama dalam mengukur performa model yang telah dilatih. Dengan demikian, tahap training model menjadi pondasi penting dalam upaya menghasilkan model klasifikasi yang handal dan mampu memberikan prediksi yang akurat terhadap data baru. Dalam tahap *Training Model*, dua algoritma machine learning yang digunakan adalah *Random Forest* (RF) dan *Support Vector Machine* (SVM).

A. *Modeling Random Forest* (RF)

Pada *Random Forest*, model akan dilatih dengan menerapkan sekumpulan pohon keputusan, di mana setiap pohon melakukan prediksi, dan hasilnya diambil melalui voting. Model RF akan menyesuaikan diri dengan data latih untuk memahami pola dan variasi dalam dataset secara keseluruhan, Berikut adalah *script* yang digunakan

```
from sklearn.ensemble import RandomForestClassifier
# Initialize Random Forest classifier
rf_classifier =
RandomForestClassifier(n_estimators=100,
random_state=42)
# Train the classifier on the training data
```

```
rf_classifier.fit(X_train, y_train)
# Make predictions on the testing data
y_pred = rf_classifier.predict(X_test)
```

Script diatas adalah implementasi penggunaan algoritma *Random Forest* pada tahap Training Model. Pertama, kita menginisialisasi kelas *Random Forest Classifier* dengan menggunakan `*RandomForestClassifier*` dari pustaka *scikit-learn*. Dalam kasus ini, kita menentukan jumlah pohon keputusan (*n_estimators*) sebanyak 100, dan menentukan nilai *seed* untuk kekonsistenan hasil (*random_state*=42). Setelah inisialisasi, model Random Forest dilatih menggunakan data latih dengan memanggil metode `*fit*()`. Dalam konteks ini, variabel `*X_train*` adalah data atribut atau fitur, dan `*y_train*` adalah label yang sesuai. Model akan mempelajari pola dan hubungan dalam data latih untuk dapat melakukan prediksi. Setelah proses training selesai, model yang sudah terlatih kemudian digunakan untuk membuat prediksi terhadap data uji (`X_test`). Prediksi ini diperoleh dengan memanggil metode `*predict*()`. Hasil prediksi disimpan dalam variabel `*y_pred*` dan dapat digunakan untuk evaluasi kinerja model pada tahap selanjutnya. Proses ini merupakan langkah kritis dalam penggunaan algoritma *Random Forest* untuk klasifikasi data.

B. Modeling *Support Vector Machine (SVM)*

Sementara itu, *Support Vector Machine* (SVM) akan melakukan training dengan mencari hyperplane terbaik yang dapat memisahkan antara kelas yang berbeda. SVM berusaha untuk menemukan batas keputusan yang optimal, yang memaksimalkan margin antara kelas. Dengan kata lain, SVM mencari garis

terbaik yang memisahkan dua kelas sehingga jaraknya (margin) ke datapoint terdekat dari masing-masing kelas adalah maksimum. Berikut adalah *script* yang digunakan

```
from sklearn.svm import SVC
# Initialize SVM classifier
svm_classifier = SVC(kernel)
# Train the classifier on the training data
svm_classifier.fit(X_train, y_train)
# Make predictions on the testing data
y_pred_svm = svm_classifier.predict(X_test)
```

Script diatas adalah implementasi penggunaan algoritma *Support Vector Machine* (SVM) pada tahap *Training Model*. Berikut adalah penjelasan. Pertama, peneliti menggunakan modul SVM (SVC) dari pustaka *scikit-learn* untuk menginisialisasi kelas *Support Vector Machine Classifier*. Dalam hal ini, peneliti menggunakan kernel linear dengan menentukan kernel='*linear*' untuk mendefinisikan fungsi kernel yang digunakan oleh SVM. Selain itu, kita juga menentukan nilai seed untuk kekonsistenan hasil (*random_state*=42). Selanjutnya, model SVM dilatih dengan menggunakan data latih. Proses training ini dilakukan dengan memanggil metode *fit*(), dimana *X_train* menyatakan data atribut atau fitur, dan *y_train* adalah label yang sesuai. Model akan mencari *hyperplane* terbaik yang dapat memisahkan data ke dalam kelas yang berbeda.

Setelah model terlatih, selanjutnya digunakan untuk membuat prediksi terhadap data uji (*X_test*). Metode *predict*() digunakan untuk mendapatkan hasil prediksi, yang disimpan dalam variabel *y_pred_svm*. Prediksi ini dapat

selanjutnya dievaluasi untuk mengukur kinerja model SVM pada dataset yang diberikan.

4.5.3. Evaluasi Model

Setelah melalui proses training, kedua model akan siap untuk melakukan prediksi terhadap data uji. Prediksi ini dilakukan dengan memasukkan data uji ke dalam model yang sudah terlatih, dan model akan mengeluarkan prediksi berdasarkan pola yang telah dipelajari selama tahap training. Hasil prediksi ini kemudian dapat dievaluasi menggunakan berbagai metrik performa, seperti akurasi, presisi dan recall untuk memahami sejauh mana model mampu memberikan prediksi yang akurat dan dapat diandalkan. Berikut script yang digunakan

```
from sklearn.model_selection import
from sklearn.metrics import accuracy_score,
classification_report
# Evaluate the model
report = classification_report(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)
```

Script diatas adalah implementasi evaluasi model setelah melakukan prediksi model pada tahap *Training* Model. Berikut adalah penjelasanya. Pertama, peneliti menggunakan modul *scikit-learn* untuk melakukan evaluasi model. Dari modul *model_selection*, peneliti mengimpor fungsi yang diperlukan, dan dari modul metrics seperti *accuracy_score*,

classification_report, *precision_score*, *recall_score*, dan *confusion_matrix*. Selanjutnya, digunakan metode *classification_report* untuk menghasilkan laporan klasifikasi yang melibatkan beberapa metrik seperti akurasi, *precision* dan *recall*. Selain itu, kita menghitung nilai *precision*, *recall*, dan *confusion matrix* menggunakan fungsi yang telah diimpor. Precision mengukur seberapa baik model dalam memprediksi positif, recall mengukur seberapa baik model dalam menangkap *instance positif*, dan *confusion matrix* memberikan gambaran lebih rinci tentang performa model dalam memprediksi *true positive*, *true negative*, *false positive*, dan *false negative*.

Tahapan klasifikasi membantu peneliti untuk membangun model untuk memprediksi label kategori dari dataset perjalanan. Dengan model yang tepat, dapat memperoleh prediksi yang akurat tentang apakah perjalanan aman atau berbahaya berdasarkan fitur-fitur yang ada. Selain itu, Peneliti juga dapat mengeksplorasi berbagai algoritma dan teknik untuk meningkatkan performa model klasifikasi.

4.6. Model Validasi

Evaluasi model menggunakan berbagai metrik dan teknik validasi. Pertama, *Confusion Matrix* digunakan untuk memberikan wawasan rinci tentang kinerja model, memisahkan prediksi yang benar dan salah untuk setiap kelas. Selanjutnya, *Model Classification Report* memberikan gambaran komprehensif tentang akurasi, *presisi* dan *recall* dari *model*. Analisis perbandingan antara dua model, yaitu *Random Forest* (RF) dan *Support Vector Machine* (SVM), memperlihatkan perbandingan kinerja keduanya dalam hal akurasi, presisi, dan

recall. Terakhir, *Cross-Validation Scores* memberikan pemahaman tentang seberapa baik model dapat umumkan pada data baru. Gabungan dari semua informasi ini memberikan pandangan menyeluruh tentang kekuatan dan kelemahan model, memandu dalam pengambilan keputusan dan pengembangan lebih lanjut berikut adalah penjelasan tahapan Model Validasi.

4.6.1. Confusion Matrix

Confusion Matrix digunakan untuk memberikan wawasan rinci tentang kinerja model, memisahkan prediksi yang benar dan salah untuk setiap kelas, Berikut adalah hasil *Confusion Matrix* dari model klasifikasi;

4.6.1.1. *Random Forest (RF)*

Tabel 4. 5. Confusion Matrix Random Forest (RF)

	Prediksi	
Aktual	**Aman**	**Berbahaya**
Aman	81431	2953
Berbahaya	4710	18574

Tabel 4.5 adalah Confusion Matrix hasil dari model Random Forest memberikan pemahaman mendalam tentang kinerja model dalam mengklasifikasikan kondisi keamanan berkendara. True Positives (TP) sebanyak 81,431 menunjukkan bahwa model mampu mengidentifikasi dengan benar perjalanan yang sebenarnya aman. Ini mengindikasikan tingkat keberhasilan dalam memberikan peringatan yang tepat pada situasi yang memang tidak berpotensi berbahaya. Namun, perlu diperhatikan bahwa False Positives (FP)

sebanyak 2,953 menunjukkan adanya situasi di mana model memberikan prediksi kondisi aman, padahal sebenarnya berpotensi berbahaya. Hal ini mungkin terjadi karena kompleksitas variasi kondisi di jalan yang sulit diprediksi oleh model, dan perlu dilakukan analisis lebih lanjut untuk memahami penyebab kesalahan ini.

True Negatives (TN) sebanyak 18,574 mencerminkan kemampuan model dalam mengenali dengan benar perjalanan yang sebenarnya berbahaya. Keberhasilan ini memberikan keyakinan bahwa model dapat memberikan peringatan yang tepat dan berguna dalam situasi potensial yang membahayakan. Namun, penting untuk diperhatikan bahwa False Negatives (FN) sebanyak 4,710 menunjukkan adanya prediksi yang salah bahwa perjalanan berbahaya, padahal sebenarnya aman. Hal ini menimbulkan pertanyaan tentang situasi di mana model kurang responsif terhadap kondisi yang seharusnya memicu peringatan, dan analisis mendalam diperlukan untuk meningkatkan sensitivitas model terhadap skenario-skenario ini.

Melalui pemahaman elemen-elemen dalam Confusion Matrix, dapat disimpulkan bahwa kinerja model memiliki aspek positif namun juga beberapa tantangan. Evaluasi dan perbaikan lebih lanjut pada model diperlukan untuk meminimalkan kesalahan yang dapat mempengaruhi keputusan pengemudi dan mengoptimalkan kontribusi model terhadap keselamatan berkendara secara keseluruhan. Hal ini membuka pintu untuk penelitian lebih lanjut terkait penyesuaian dan penyempurnaan model agar dapat memberikan kontribusi yang lebih besar dalam konteks keamanan berkendara.

4.6.1.2. *Support Vector Machine (SVM)*

Tabel 4. 6. Confusion Matrix Support Vector Machine (SVM)

	Prediksi	
Aktual	**Aman**	**Berbahaya**
Aman	1241572	0
Berbahaya	400179	0

Berdasarkan hasil evaluasi model menggunakan *Confusion Matrix* Tabel 4.5, dapat disimpulkan bahwa model *Random Forest* (RF) secara keseluruhan memberikan performa yang lebih baik daripada model *Support Vector Machine* (SVM) Tabel 4.6 dalam konteks dataset. RF berhasil mengidentifikasi perjalanan yang sebenarnya aman dengan baik, menunjukkan keseimbangan yang baik antara *True Positives* dan False *Positives*. Meskipun terdapat beberapa prediksi yang salah, kinerja RF lebih handal daripada SVM. Di sisi lain, model SVM mengalami kesulitan signifikan dalam mengklasifikasikan perjalanan berbahaya, dalam matriks tersebut, tidak ada prediksi yang benar untuk kelas "Berbahaya" (*True Negative* dan *False Negative* sama-sama 0), sementara sebagian besar prediksi justru masuk ke dalam kelas "Aman". Terdapat beberapa justifikasi yang mungkin menjelaskan fenomena ini. Faktor utama yang mempengaruhi adanya ketidakseimbangan kelas dalam data pelatihan, di mana mayoritas sampel termasuk ke dalam kelas "Aman". SVM tidak dapat memprediksi perjalanan Berbahaya dikarenakan Dataset yang digunakan

Imbalance dan dataset sangatlah banyak, sehingga komputasi SVM tidak berjalan dengan baik. berdasarkan karakteristiknya SVM memang tidak terlalu baik dalam menghandle data dengan jumlah yang besar karena komputasinya akan tinggi. Model SVM cenderung memilih mayoritas kelas untuk memaksimalkan akurasi, sehingga hasilnya cenderung memprediksi kelas "Aman" secara eksklusif. Oleh karena itu, untuk dataset ini, RF lebih direkomendasikan sebagai model klasifikasi yang lebih efektif.

4.6.2. Model Classification Report

Pada bagian ini, akan diuraikan hasil dari evaluasi kinerja model Klasifikasi berdasarkan *Classification* Report. *Classification Report* memberikan wawasan mendalam tentang kemampuan model dalam melakukan klasifikasi pada setiap kelas, *termasuk* presisi, *recall*. Dalam konteks evaluasi model klasifikasi, terdapat dua kelas yang diamati, yakni *Class* 0 dan *Class* 1. Presisi mencerminkan sejauh mana model mengklasifikasikan data ke dalam kelas tertentu dengan benar, sementara recall menunjukkan sejauh mana model dapat mengidentifikasi semua instance dari suatu kelas. F1-score, sebagai harmonic mean dari presisi dan *recall*, memberikan gambaran keseluruhan tentang kinerja model. Tentunya, evaluasi tidak hanya melibatkan aspek kelas individu, tetapi juga mencakup nilai rata-rata dari keseluruhan kelas (*macro avg* dan *weighted avg*). Keseluruhan, hasil akhir dari evaluasi model adalah accuracy, yang mencerminkan sejauh mana model dapat mengklasifikasikan data secara benar. Beriku lanjut hasil evaluasi yang didapatkan dari

Classification Report untuk mendapatkan pemahaman yang lebih komprehensif tentang kemampuan model klasifikasi

4.6.2.1. *Random Forest (RF)*

Hasil dari model klasifikasi *Random Forest* yang peneliti terapkan adalah sebagai berikut:

Tabel 4. 7. Model Classification Report Random Forest (RF)

Classification Report Random Forest				
	Precision	**Recall**	**F1-score**	**Support**
Class 0	0.95	0.97	0.96	84384
Class 1	0.86	0.80	0.83	23284
accuracy			0.93	107668
macro avg	0.9	0.88	0.89	107668
weighted avg	0.93	0.93	0.93	107668

Tabel 4.7 merupakan hasil Classification Report dari model Random Forest menunjukkan performa yang sangat baik dalam menganalisis keamanan berkendara dengan menggunakan data sensor gerak smartphone. Dengan tingkat akurasi mencapai 93%, model ini mampu memberikan prediksi yang konsisten dan akurat terhadap kondisi berkendara. Pada kelas yang menunjukkan kondisi aman (Label 0), model memiliki presisi sebesar 95%, menandakan bahwa sekitar 95% dari prediksi yang dikategorikan sebagai kondisi aman benar-benar merupakan kondisi aman. Tingkat recall yang tinggi (97%) pada kelas ini juga mengindikasikan kemampuan model untuk mendeteksi sebagian besar kondisi aman. Sebaliknya, pada kelas yang menunjukkan kondisi berpotensi berbahaya

(Label 1), model memiliki presisi sebesar 86%, menunjukkan sekitar 86% dari prediksi yang dikategorikan sebagai berpotensi berbahaya adalah benar-benar berpotensi berbahaya. Meskipun recall pada kelas ini sedikit lebih rendah (80%), F1-score yang mencapai 83% menunjukkan keseimbangan yang baik antara presisi dan recall. Rata-rata dari masing-masing metrik (Macro Avg) menunjukkan kinerja yang baik secara keseluruhan pada semua kelas, dengan nilai sebesar 0.90, 0.88, dan 0.89. Begitu pula, Weighted Avg memberikan penekanan lebih besar pada kelas mayoritas, dan hasilnya menunjukkan kinerja yang baik secara umum dengan nilai 0.93, 0.93, dan 0.93. Keseluruhan, model Random Forest ini dapat dianggap sebagai alat yang handal dalam mendukung upaya mitigasi risiko keamanan berkendara melalui penerapan teknologi inovatif dan pendekatan machine learning.

4.6.2.2.*Support Vector Machine (SVM)*

Tabel 4. 8. Model Classification Report Support Vector Machine (SVM)

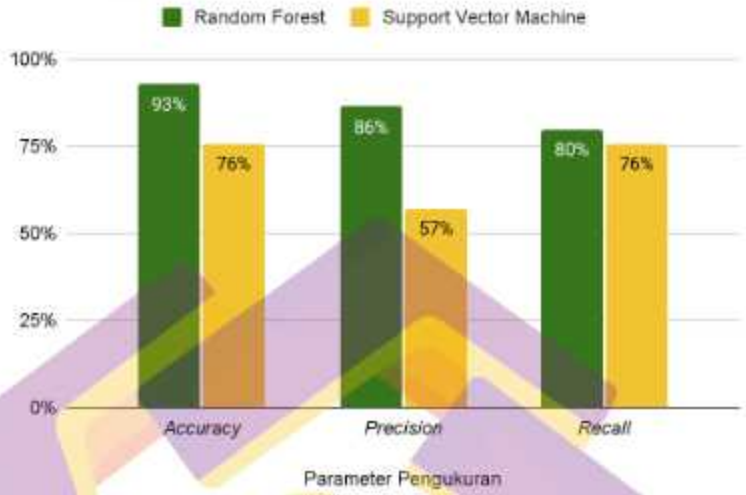
Classification Report Support Vector Machine				
	Precision	***Recall***	***F1-score***	***Support***
Class 0	0.76	1.00	0.86	1241572
Class 1	0.00	0.00	0.00	400179
accuracy			0.76	1641751
macro avg	0.38	0.50	0.43	1641751
weighted avg	0.57	0.76	0.65	1641751

Dalam perbandingan antara dua model klasifikasi, yaitu *Random Forest* (RF) pada tabel 4.6 dan *Support Vector Machine* (SVM) pada tabel 4.7, *Random*

Forest menonjol sebagai pilihan yang lebih unggul untuk dataset perjalanan ini. RF mencapai tingkat akurasi sebesar 93%, menunjukkan kemampuannya dalam memprediksi perjalanan dengan tingkat ketepatan yang tinggi. Model ini efektif mengidentifikasi perjalanan aman dan berbahaya, memberikan hasil yang konsisten meskipun adanya ketidakseimbangan kelas. Di sisi lain, SVM menunjukkan performa yang kurang memuaskan dengan akurasi 76%. Meskipun sangat baik dalam mengenali perjalanan yang sebenarnya aman, model ini menghadapi kesulitan dalam mengklasifikasikan perjalanan berbahaya. Kinerja SVM dipengaruhi oleh ketidakseimbangan kelas yang signifikan dalam dataset. Secara keseluruhan, *Random Forest* adalah pilihan yang lebih handal dan seimbang untuk tugas klasifikasi perjalanan ini. Dengan akurasi tinggi dan kemampuan yang baik dalam mengenali kelas minoritas, RF menawarkan solusi klasifikasi yang dapat diandalkan untuk dataset ini.

4.6.3. Perbandingan Hasil Klasifikasi

Berikut adalah pembahasan Hasil Kualifikasi perbandingan kinerja dua model klasifikasi yang diimplementasikan, yaitu *Random Forest* (RF) dan *Support Vector Machine* (SVM). Analisis dilakukan melalui beberapa matrik evaluasi, termasuk akurasi, *precision* dan *recall*, untuk memahami sejauh mana kedua model dapat memprediksi label perjalanan (Aman/Berbahaya). Evaluasi tersebut memberikan gambaran yang jelas tentang keunggulan model RF dibandingkan dengan SVM dalam tugas klasifikasi perjalanan pada dataset yang digunakan. Berikut adalah hasil Perbandingan Hasil Klasifikasi;



Gambar 4. 21. Perbandingan Hasil Klasifikasi

Gambar 4.21 adalah hasil perbandingan model *Random* Forest (RF) dan *Support Vector Machine* (SVM) Secara keseluruhan hasil perbandingan hasil klasifikasi pada gambar 4.21, Random Forest menunjukkan kinerja yang sangat baik dalam hal akurasi dan precision dibandingkan dengan *Support Vector Machine*. Dalam penelitian ini *Random Forest* muncul sebagai pilihan yang lebih baik karena mampu memberikan keseimbangan yang baik antara presisi dan recall pada kedua kelas, sementara SVM mengalami kendala signifikan dalam mendeteksi kelas berbahaya dan cenderung memberikan *false positives*. Keunggulan Random Forest dalam menangani database yang kompleks dan tidak seimbang dapat menjadi alasan utama mengapa model ini lebih dipilih untuk tugas klasifikasi keamanan berkendara ini.

4.6.4. Cross-Validation

Berdasarkan hasil evaluasi model, terutama dengan perbandingan antara Random Forest (RF) dan Support Vector Machine (SVM), terlihat bahwa Random Forest memberikan performa yang lebih baik dalam hal akurasi, precision, dan recall. Oleh karena itu, Perlu dilakukan evaluasi model Random Forest lebih lanjut menggunakan metode cross-validation. Cross-validation adalah pendekatan yang kuat untuk mengevaluasi kinerja model. Dengan melakukan cross-validation, model dievaluasi pada beberapa subset data, sehingga memberikan gambaran yang lebih konsisten tentang seberapa baik model dapat digeneralisasi ke data yang belum pernah dilihat sebelumnya. Dengan melibatkan cross-validation dalam proses pengujian, dapat memastikan bahwa model Random Forest tidak hanya berkinerja baik pada dataset tertentu, tetapi juga dapat digeneralisasikan dengan baik ke data baru

Tabel 4. 9. Cross-Validation Result

No	Cross-Validation	Akurasi
1	Subset Data kfold - ke 1	92.80%
2	Subset Data kfold - ke 2	92.92%
3	Subset Data kfold - ke 3	92.81%
4	Subset Data kfold - ke 4	92.87%
5	Subset Data kfold - ke 5	92.93%
Nilai Akurasi Terendah		**92.80%**
Nilai Akurasi Tertinggi		**92.93%**
Rerata Nilai Akurasi		**92.86%**

Berdasarkan hasil cross-validation pada tabel 4.9 untuk model Random Forest, terlihat bahwa akurasi pada setiap subset data k-*fold* stabil dan tinggi, dengan nilai konstan sebesar 93%. Nilai akurasi terendah, tertinggi, dan rata-rata semuanya sama, menunjukkan konsistensi yang baik dalam kinerja model di seluruh subset data berikut adalah hasil Analisisnya

1. Stabilitas Model: Hasil *cross-validation* menunjukkan bahwa model *Random Forest* memberikan hasil akurasi yang stabil di seluruh subset data *k-fold*. Ini menandakan bahwa model memiliki kemampuan yang baik untuk menggeneralisasi ke berbagai subset data.
2. Kinerja Tinggi: Dengan nilai akurasi sebesar 92.93%, model *Random Forest* telah memberikan kinerja yang sangat baik dalam mengklasifikasikan data pada setiap iterasi *cross-validation*. Hal ini dapat diartikan bahwa model mampu dengan baik mengidentifikasi pola dan hubungan dalam data.

3. Tidak Ada Varian yang Signifikan: Konsistensi nilai akurasi yang tinggi antara subset data menunjukkan bahwa model tidak terlalu dipengaruhi oleh variasi dalam pemilihan subset data. Ini mengindikasikan robustness model terhadap variasi dalam distribusi data.
4. Tidak Ada *Overfitting*: Jika model memiliki performa yang sangat tinggi pada subset pelatihan tetapi tidak konsisten di subset pengujian, itu dapat menjadi tanda overfitting. Namun, dalam kasus ini, karena kinerja yang baik pada seluruh subset, kemungkinan *overfitting* menjadi lebih rendah.

Secara keseluruhan, hasil *cross-validation* memberikan dukungan yang kuat untuk keandalan dan konsistensi model *Random Forest* dalam melakukan klasifikasi pada dataset yang digunakan. Model ini dapat dianggap sebagai pilihan yang baik untuk pemodelan masalah klasifikasi untuk penelitian ini.

4.6.5. Fitur Penting

Fitur *importance* atau signifikansi fitur adalah nilai yang menunjukkan sejauh mana suatu fitur memberikan kontribusi atau pengaruh terhadap kinerja model. Dalam konteks pohon keputusan, seperti yang digunakan dalam algoritma *Random Forest*, fitur *importance* diukur berdasarkan seberapa sering suatu fitur digunakan untuk membuat keputusan di seluruh pohon (*ensemble*). Semakin sering suatu fitur digunakan, semakin penting fitur tersebut dianggap.

Fitur *importance* biasanya diukur dengan metrik Gini *Importance* atau *Mean Decrease in Impurity* (MDI). *Gini Importance* mengukur penurunan impuritas (*Gini impurity*) di setiap node yang disebabkan oleh suatu fitur. Semakin besar penurunan impuritas, semakin penting fitur tersebut dianggap.

Fitur importance memberikan informasi tentang fitur mana yang memiliki kontribusi besar dalam membedakan kelas atau membuat prediksi. Semakin tinggi nilai fitur importance, semakin besar kontribusinya terhadap prediksi model.

Berikut Hasil dari pengukuran *Fitur importance*

Tabel 4. 10. Fitur importance Result

No	*Feature*	*Importance Score*
1	*second*	0.006926
2	*Speed*	0.001976
3	*Accuracy*	0.001325
4	*acceleration_x*	0.000857
5	*acceleration_z*	0.000749
6	*acceleration_y*	0.000674
7	*Bearing*	0.000486
8	*gyro_y*	0.000480
9	*gyro_z*	0.000229
10	*gyro_x*	0.000149

Pada tabel 4.10 nilai *importance* pada fitur-fitur model klasifikasi mengungkapkan bahwa waktu perjalanan (*second*) menjadi fitur paling berpengaruh dengan nilai importance tertinggi, menunjukkan bahwa variabel waktu perjalanan memberikan kontribusi paling signifikan dalam melakukan prediksi kategori perjalanan. Diikuti oleh kecepatan (*Speed*) yang menjadi faktor kedua dalam pengaruhnya terhadap model, meskipun dengan nilai importance yang lebih rendah dibandingkan waktu perjalanan. Akurasi (*Accuracy*) juga memberikan kontribusi yang cukup berarti, menandakan bahwa informasi

mengenai akurasi dalam data GPS dapat memengaruhi hasil klasifikasi. Sementara itu, fitur accelerometer *(acceleration_x, acceleration_z, acceleration_y), gyroscope (gyro_y, gyro_z, gyro_x)*, dan *bearing* memainkan peran sekunder dengan nilai *importance* yang lebih rendah. Meskipun memberikan kontribusi, pemahaman ini memberikan arahan pada fokus peningkatan pada fitur-fitur yang paling krusial untuk meningkatkan performa model klasifikasi. Selanjutnya, analisis ini dapat menjadi dasar untuk pengambilan keputusan terkait perbaikan model atau penyempurnaan fitur-fitur tertentu.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan rumusan masalah, penjabaran penelitian dan pembahasan mengenai pengolahan data sensor gerak smartphone untuk klasifikasi karakteristik mengemudi dapat ditarik beberapa konklusi di antaranya yaitu.

1. Hasil modeling menunjukkan bahwa *Random Forest* memiliki kinerja yang sangat baik dengan akurasi sebesar 93%, dibandingkan dengan *Support Vector Machine* yang memiliki akurasi 76%.
2. Sensor GPS memiliki pengaruh paling signifikan, terutama pada fitur *second* dan *speed*, berbeda dengan penelitian sebelumnya yang menyoroti sensor *Gyroscope* sehingga Kecepatan diidentifikasi sebagai faktor kunci yang berpengaruh pada keamanan lalu lintas di Indonesia, memberikan wawasan penting terkait faktor-faktor yang mempengaruhi keselamatan berkendara.
3. Melalui *Exploratory Data Analysis* (EDA) pada sensor *Gyroscope*, Accelerometer, dan GPS, terdeteksi adanya data yang tidak wajar pada sensor GPS, khususnya pada fitur *Accuracy, speed*, dan *second.*
4. Dalam menangani *outlier,* metode yang dipilih adalah drop data, karena metode normalisasi tidak memadai mengingat jumlah data yang besar.

5.2. Saran

Berdasarkan hasil penelitian ini, beberapa saran untuk penelitian lebih lanjut dapat dipertimbangkan:

1. Eksplorasi Model Lain: Selain *Random Forest*, perlu untuk mengeksplorasi model klasifikasi lainnya lain yang mungkin memiliki performa yang baik pada dataset ini. Perbandingan performa antara beberapa model dapat memberikan wawasan yang lebih mendalam.
2. Penanganan *Imbalance Class*: Jika terdapat ketidakseimbangan dalam jumlah sampel antara kelas aman dan berbahaya, perlu dipertimbangkan teknik-teknik penanganan ketidakseimbangan kelas, seperti *oversampling*, undersampling, atau menggunakan metode khusus untuk menangani ketidakseimbangan tersebut.
3. Analisis Lebih Lanjut tentang *Outlier*: Melakukan analisis lebih lanjut tentang outlier pada fitur-fitur tertentu dan mempertimbangkan strategi penanganan yang lebih cermat untuk outlier-outlier tersebut.

DAFTAR PUSTAKA

PUSTAKA BUKU

Bachhety, Shivam, Ramneek Singhal, and Rachna Jain. "Intelligent Data Analysis with Data Mining." *Intelligent Data Analysis*, 2020, 63–83. https://doi.org/10.1002/9781119544487.ch4.

K. T. Nguyen, F. Portet, and C. Garbay, *"Dealing with Imbalanced data sets for Human Activity Recognition using Mobile Phone sensors,"* 3rd Int. Work. Smart Sens. Syst., 2018.

B. Santosa and A. Umam, *"Data Mining dan Big Data Analytics," Data Mining dan Big Data Analytics.* pp. 31–32, 2018.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Khomh, Foutse, Bram Adams, Jinghui Cheng, Marios Fokaefs, and Giuliano Antoniol. *"Software Engineering for Machine-Learning Applications: The Road Ahead."* IEEE Software 35, no. 5 (2018): 81–84. https://doi.org/10.1109/ms.2018.3571224.

Bhat, Showkat Ahmad, and Nen-Fu Huang. "Big Data and Ai Revolution in Precision Agriculture: Survey and Challenges." *IEEE Access* 9 (2021): 110209–22. https://doi.org/10.1109/access.2021.3102227.

Al-Janabi and Samaher. "Overcoming the Main Challenges of Knowledge Discovery through Tendency to Intelligent Data Analysis." *2021 International Conference on Data Analytics for Business and Industry (ICDABI)*, 2021. https://doi.org/10.1109/icdabi53623.2021.9655916.

Pintye, I. (2020). Machine learning methods in smartphone-based activity recognition. 2020 IEEE 14th International Symposium on Applied Computational Intelligence and Informatics (SACI). https://doi.org/10.1109/saci49304.2020.9118784.

Ben Ahmed, K., Goel, B., Bharti, P., Chellappan, S., & Bouhorma, M. (2019). Leveraging smartphone sensors to detect distracted driving activities. IEEE Transactions on Intelligent Transportation Systems, 20(9), 3303–3312. https://doi.org/10.1109/tits.2018.2873972

Chacko, R., Binoj, R. V., & Ameenudeen, P. E. (2023). Performance improvement of a MEMS gyroscope using filtering and Machine Learning

Methods. 2023 International Conference on Control, Communication and Computing (ICCC). https://doi.org/10.1109/iccc57789.2023.10165457.

Gorodnichev, M. G., Polyantseva, K. A., Moseva, M. S., & Yashina, M. V. (2019). On some methods for correcting accelerometer and gyroscope data using Machine Learning Algorithms. 2019 International Conference "Quality Management, Transport and Information Security, Information Technologies" doi.org/10.1109/itqmis.2019.8928397

C. Ma, X. Dai, J. Zhu, N. Liu, H. Sun, and M. Liu, "DrivingSense: Dangerous Driving Behavior Identification Based on Smartphone Autocalibration," Mobile Information Systems, vol. 2017. 2017, doi: 10.1155/2017/9075653.

D. N. Lu, D. N. Nguyen, T. H. Nguyen, and H. N. Nguyen, "Vehicle mode and driving activity detection based on analyzing sensor data of smartphones," Sensors (Switzerland), vol. 18, no. 4, pp. 1–25, 2018, doi: 10.3390/s18041036.

Kanwal, K., Rustam, F., Chaganti, R., Jurcut, A. D., & Ashraf, I. (2023). Smartphone inertial measurement unit data features for analyzing driver driving behavior. IEEE Sensors Journal, 23(11), 11308–11323. https://doi.org/10.1109/jsen.2023.3256000

L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big Data Analytics in Intelligent Transportation Systems: A Survey," IEEE Trans. Intell. Transp. Syst., vol. 20, no. 1, pp. 383–398, 2019, doi: 10.1109/TITS.2018.2815678.

P. Patil, N. Yaligar, and S. Meena, "Comparison of Performance of Classifiers - SVM, RF and ANN in Potato Blight Disease Detection Using Leaf Images," 2017 IEEE Int. Conf. Comput. Intell. Comput. Res. ICCIC 2017, pp. 1–5, 2018, doi: 10.1109/ICCIC.2017.8524301.

Ferreira, J., Carvalho, E., Ferreira, B. V., de Souza, C., Suhara, Y., Pentland, A., & Pessin, G. (2017). Driver behavior profiling: An investigation with different smartphone sensors and machine learning. PLOS ONE, 12(4). https://doi.org/10.1371/journal.pone.0174959.

J. Yu, Z. Chen, Y. Zhu, Y. Jennifer Chen, L. Kong, and M. Li, "Fine-Grained Abnormal Driving Behaviors Detection and Identification with Smartphones," IEEE Trans. Mob. Comput., vol. 16, no. 8, pp. 2198–2212, 2017, doi: 10.1109/TMC.2016.2618873.

A. Kuritsyn, M. Kharlamov, S. Prokhorov, and D. Shcherbinin, "Application of Artificial Intelligence Systems in the Process of Crew Training," Proc. - 2018 Int. Conf. Artif. Intell. Appl. Innov. IC-AIAI 2018, pp. 55–59, 2019, doi: 10.1109/IC-AIAI.2018.8674440.

C. K. Seo, J. H. Kim, and S. Y. Kwon, "A study on modeling using big data and deep learning method for failure diagnosis of systems," Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018. pp. 4747–4751, 2019, doi: 10.1109/BigData.2018.8622644.

H. Erol, B. M. Tyoden, and R. Erol, "Classification Performances of Data Mining Clustering Algorithms for Remotely Sensed Multispectral Image Data," 2018 IEEE Int. Conf. Innov. Intell. Syst. Appl. INISTA 2018, pp. 1–4, 2018, doi: 10.1109/INISTA.2018.8466320.

M. Rinaldi, E. Picarelli, G. Laskaris, A. D'Ariano, and F. Viti, "Mixed hybrid and electric bus dynamic fleet management in urban networks: A model predictive control approach," MT-ITS 2019 - 6th Int. Conf. Model. Technol. Intell. Transp. Syst., pp. 1–8, 2019, doi: 10.1109/MTITS.2019.8883387.

Y. A. Seliverstov, S. A. Seliverstov, V. I. Komashinskiy, A. A. Tarantsev, N. V. Shatalova, and V. A. Grigoriev, "Intelligent systems preventing road traffic accidents in megalopolises in order to evaluate," Proc. 2017 20th IEEE Int. Conf. Soft Comput. Meas. SCM 2017, pp. 489–492, 2017, doi: 10.1109/SCM.2017.7970626.

Jia, C., Zhao, L., Jiang, W., Liu, X., Yu, M., Huang, M., Xia, Y., Zhao, Y., & Zhao, Y. (2018). Impact experiment analysis of MEMS ultra-high G piezoresistive shock accelerometer. 2018 IEEE Micro Electro Mechanical Systems (MEMS). https://doi.org/10.1109/memsys.2018.8346718

T. Hasanin and T. M. Khoshgoftaar, "The effects of random undersampling with simulated class imbalance for big data," Proc. - 2018 IEEE 19th Int. Conf. Inf. Reuse Integr. Data Sci. IRI 2018, pp. 70–79, 2018, doi: 10.1109/IRI.2018.00018.

Mehanian, C., Jaiswal, M., Delahunt, C., Thompson, C., Horning, M., Hu, L., McGuire, S., Ostbye, T., Mehanian, M., Wilson, B., Champlin, C., Long, E., Proux, S., Gamboa, D., Chiodini, P., Carter, J., Dhorda, M., Isaboke, D., Ogutu, B., … Bell, D. (2017). Computer-automated malaria diagnosis and quantitation using convolutional neural networks. 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). https://doi.org/10.1109/iccvw.2017.22.

T. T. Wong and N. Y. Yang, "Dependency Analysis of Accuracy Estimates in k-Fold Cross Validation," IEEE Trans. Knowl. Data Eng., vol. 29, no. 11, pp. 2417–2427, 2017, doi: 10.1109/TKDE.2017.2740926.

Q. Liu and M. Hauswirth, "A Provenance Meta Learning Framework for Missing Data Handling Methods Selection," 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2020, pp. 0349-0358, doi: 10.1109/UEMCON51285.2020.9298089.

PUSTAKA ELEKTRONIK

World Health Organization (WHO), "World Health Organization (WHO), "Global status report on road safety 2022.". Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries