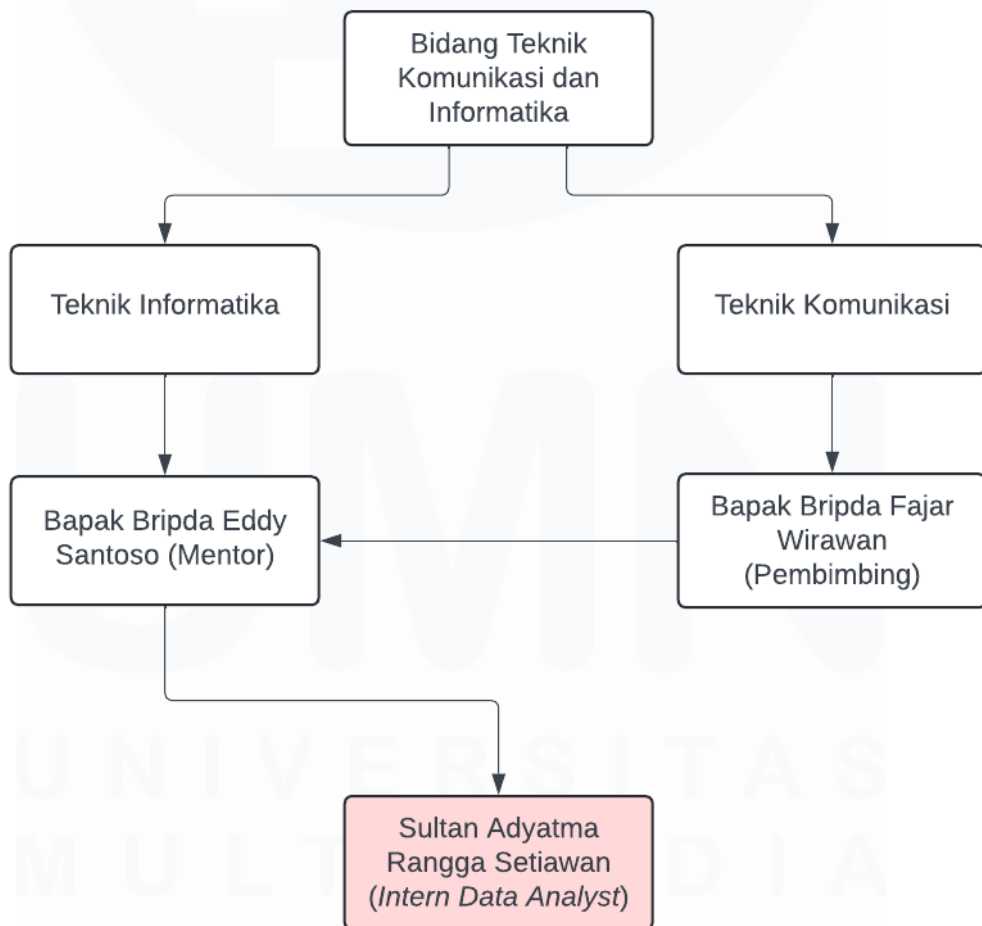


BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pelaksanaan kegiatan magang yang dilakukan di Polda Jawa Timur sebagai *internship data analyst* di bidang teknik informatika dan komunikasi kepolisian di bawah bimbingan Bapak Briptu Fajar Wirawan, dan mentor Bapak Briptu Eddy Santoso. Sebelum melakukan proyek, dilakukan survei oleh pembimbing terkait kecocokan proyek yang akan dikerjakan. Dan diberikan tanggung jawab untuk menyelesaikan proyek selama masa berkegiatan magang di Polda Jatim.



Gambar 3. 1 Bagan Struktur Kedudukan & Koordinasi Intern

Bapak Briptu Fajar Wirawan memberikan tugas untuk melakukan analisis data kecelakaan, sekaligus berkegiatan kerja lapang agar meningkatkan pengalaman dalam melakukan pekerjaan kantor juga proyek. Dalam menganalisis data kecelakaan digunakan pendekatan yang baru, dengan algoritma *random forest classifier*, proses *data pre-processing*, juga memberikan presentasi terkait data secara visual. Analisis data kemudian dipresentasikan untuk membantu Kepolisian Bidang Lalu Lintas untuk mengambil keputusan terkait tingginya kasus kecelakaan dan tingkat kefatalan di Surabaya.

Selain melakukan proyek, kegiatan magang diisi dengan kegiatan lapang kepolisian, baik dengan melakukan pendataan, patroli, juga kegiatan-kegiatan umum lainnya. Koordinasi dapat langsung dilakukan secara tatap muka, juga menggunakan Whatsapp. Bimbingan juga dilakukan oleh tim kerja bintanga lainnya dalam melakukan kegiatan kepolisian ketika tidak bersama pembimbing atau mentor.

3.2 Tugas dan Uraian Kerja Magang

Dalam melaksanakan kegiatan magang, ditugaskan untuk melaksanakan analisis data terhadap data kecelakaan lalu lintas kota Surabaya pada tahun 2018-2023. Pelaksanaan proyek ini berjalan selama rentang waktu magang, yaitu 17 Januari 2024 hingga 17 Mei 2024. Pada proses berjalannya magang, dilaksanakannya beberapa kegiatan kerja lapang yang diarahkan oleh pembimbing magang. Beberapa kegiatan juga menghabiskan waktu yang cukup lama sehingga diperlukannya mengambil jam kerja lembur. Namun, diberikan juga keringanan untuk tidak mengambil waktu kerja di esoknya atau cuti. Berikut merupakan rincian dari kegiatan kerja magang yang telah dilaksanakan:

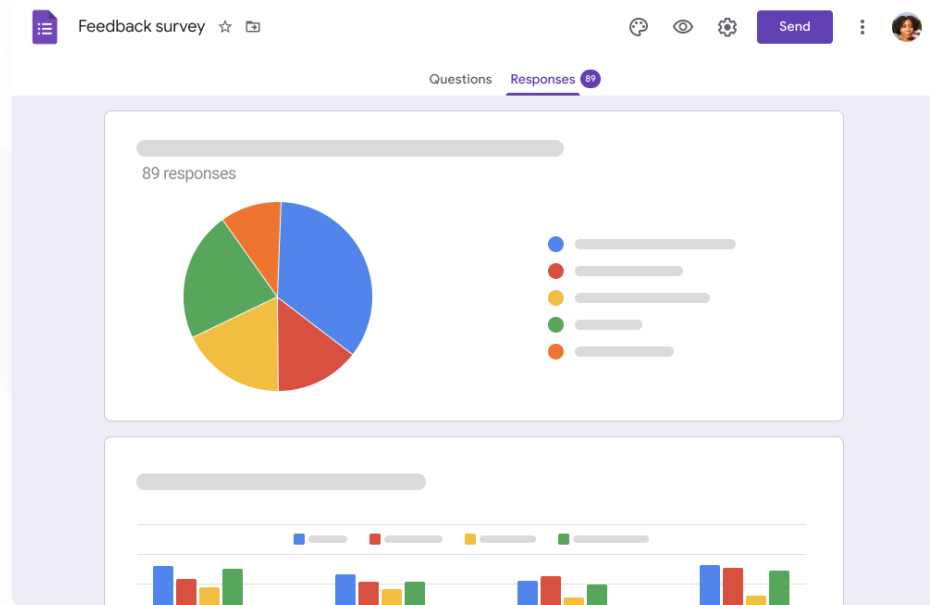
Tabel 3. 1 Rincian Kegiatan Magang

No	Kegiatan	Mulai	Selesai	Durasi
Orientasi Magang Polda Jatim				
1	Pengenalan Struktur Polda Jatim	17 Januari 2024	20 Januari 2024	4 Hari
2	Survei Proyek Analisis Data	22 Januari 2024	24 Januari 2024	3 Hari
3	Pengenalan Tools Internal Polda Jatim	25 Januari 2024	26 Januari 2024	2 Hari
4	Pengajuan Proposal dan Konfirmasi Proyek	29 Januari 2024	6 Februari 2024	10 Hari
Pekerjaan Lapangan				
1	Pendataan dan Pengecekan Data Kecelakaan Tahun 2024			
	- Februari	1 Februari 2024	7 Februari 2024	7 Hari
	- Maret	1 Maret 2024	5 Maret 2024	5 Hari
	- April	16 April 2024	19 April 2024	4 Hari
	- Mei	2 Mei 2024	8 Mei 2024	7 Hari
2	Kegiatan Patroli			
	- Malang	6 Maret 2024	8 Maret 2024	3 Hari
	- Probolinggo	20 Maret 2024	22 Maret 2024	3 Hari
3	Kegiatan Operasi Ketupat	1 April 2024	5 April 2024	6 Hari
Pelatihan Analisis Data				
1	Pengumpulan Data 2016-2020	12 Februari 2024	14 Februari 2024	3 Hari
2	Percobaan Data Menggunakan Algoritma <i>Random Forest Classifier</i>	19 Februari 2024	22 Februari 2024	4 Hari

No	Kegiatan	Mulai	Selesai	Durasi
3	Review dan revisi hasil pelatihan bersama tim kerja	26 Februari 2024	29 Februari 2024	4 Hari
Proyek Analisis Data Kecelakaan				
1	Pengumpulan Data Kecelakaan Lalu Lintas Tahun 2018-2023	25 Maret 2024	28 Maret 2024	4 Hari
2	<i>Data Preprocessing</i>	22 April 2024	24 April 2024	3 Hari
3	Pemodelan algoritma <i>Random forest classifier</i>	25 April 2024	26 April 2024	2 Hari
4	Review dan Revisi hasil proyek bersama tim kerja	14 Mei 2024	17 Mei 2024	4 Hari
Total Waktu				78 Hari

Selama proses magang, kegiatan pengecekan, pendataan, dan pengerjaan proyek analisis data menggunakan beberapa *tools*, berupa:

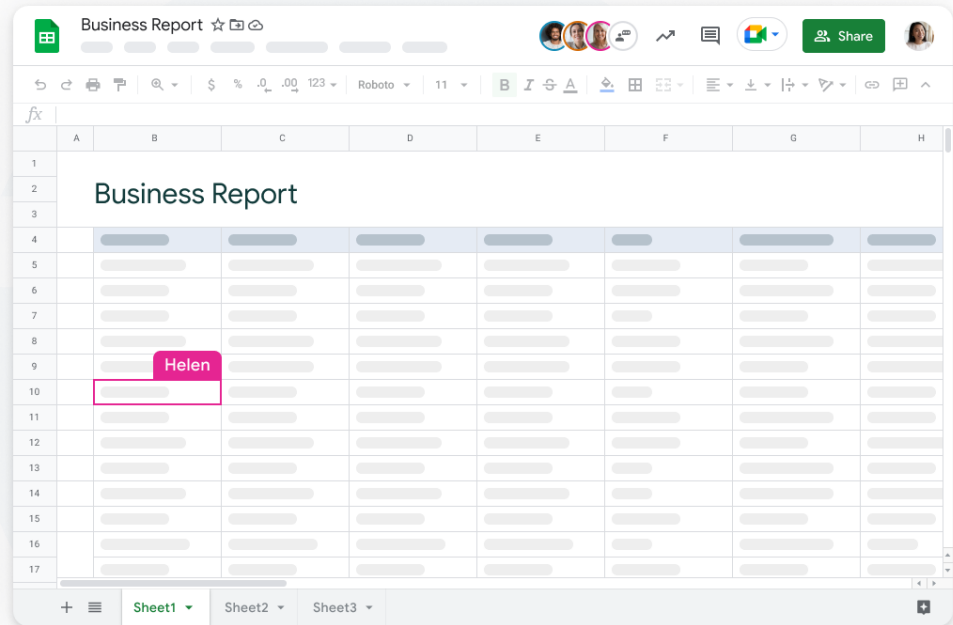
1. Google Form



Gambar 3. 2 Google Form

Google Form digunakan dalam proses melakukan *input* data, inputan ini dilakukan untuk melaksanakan kegiatan pendataan data kecelakaan, data hilang temu, data kriminalitas, juga data peminjaman barang. Yang nantinya akan masuk ke dalam data Google Docs. Google Form digunakan juga untuk memberi penilaian kinerja, lapor kerusakan, dan survei lainnya.

2. Google Sheets



Gambar 3. 3 Google Sheets

Google Sheets digunakan untuk melakukan pengecekan data, data-data ini merupakan hasil *input* dari Google Form, juga pendataan manual lain, seperti pendataan pada bagian akuntansi yang mengelola keuangan di kepolisian. Pada masa magang, pendataan di Google Sheets digunakan untuk pengecekan juga sebagai data yang akan dianalisis menggunakan pada proyek data analisis.

pada Tahun 2016-2020 menggunakan algoritma SVM, KNN, ANN, Naive Bayes, dan Decision Tree. Di Kepolisian, Google Colab digunakan dalam melakukan analisis tingkat kecelakaan, dan kasus kriminalitas. Dan dalam proyek magang yang dilakukan, Google Colab juga digunakan untuk melakukan pelatihan analisis data, juga pengerjaan proyek analisis data kecelakaan di Surabaya pada tahun 2018-2023.

Selama proses magang, kegiatan pengecekan, pendataan, dan pengerjaan proyek analisis data menggunakan beberapa *tools*, berupa:

1. Kerja Lapang

Kerja lapang merupakan kegiatan untuk mengisi waktu kerja dengan mengikuti arahan dari pembimbing juga mentor. Kegiatan ini meliputi kegiatan yang diikuti oleh tim kerja, sehingga seluruh tim kerja ikut berpartisipasi dalam kegiatan. Beberapa kegiatan diluar *job desc intern data analyst*, namun masuk sebagai *job desc* kepolisian. Berikut adalah kegiatan kerja lapang yang dilakukan di Polda Jatim sebagai anggota tim di bidang TIK Kepolisian:

a. Pendataan dan pengecekan data bulanan

Pendataan dilakukan setiap bulannya, dengan mengelola data dari setiap cabang kepolisian resort di Jawa Timur, dikumpulkannya data-data kecelakaan pada Tahun 2024. Tidak hanya melakukan pendataan untuk kasus kecelakaan, dilakukan juga pendataan untuk kasus kriminal, hilang temu, dan pinjaman barang.

No. Laka	Alamat	Tanggal Kejadian	Tanggal Pelaporan	Tingkat Kecelakaan
1	Jl. Raya No. 123	2024-01-15	2024-01-16	Ringan
2	Jl. Sudirman No. 456	2024-01-20	2024-01-21	Berat
3	Jl. Diponegoro No. 789	2024-02-05	2024-02-06	Ringan
4	Jl. Veteran No. 101	2024-02-10	2024-02-11	Berat
5	Jl. Pemuda No. 202	2024-02-15	2024-02-16	Ringan
6	Jl. Sisinga No. 303	2024-02-20	2024-02-21	Berat
7	Jl. Kartika No. 404	2024-02-25	2024-02-26	Ringan
8	Jl. Hidayat No. 505	2024-03-01	2024-03-02	Berat
9	Jl. Kertajaya No. 606	2024-03-05	2024-03-06	Ringan
10	Jl. Kencana No. 707	2024-03-10	2024-03-11	Berat

Gambar 3. 6 Data Kecelakaan Tahun 2024

Pada gambar 3.6, yang merupakan data kecelakaan yang terjadi di Jawa Timur pada Tahun 2024, data ini mencakup lokasi Kepolisian Resort, Tanggal Kejadian, Tanggal Pelaporan, Tingkat Kecelakaan, Jumlah Meninggal Dunia, Jumlah Luka Berat, Jumlah Luka Ringan, Koordinat GPS Lintang dan Bujur, Titik Acuan/Referensi, Jarak ke Lokasi Kecelakaan, Arah Dari Titik Acuan ke Lokasi Kecelakaan, Informasi Khusus, Tipe Kecelakaan, Kondisi Cahaya, Cuaca, Kondisi Kecelakaan Dari Publik, Nomor Jalan, Nama Jalan, Titik Jalan, Fungsi Jalan, Kelas Jalan, Tipe Jalan, Bentuk Geometri, Batas Kecepatan, Kondisi Permukaan Jalan, Kemiringan Jalan, Status Jalan, Perkiraan Nilai Rugi Material Non Kendaraan, Total Nilai Rugi Material Kendaraan, dan Keterangan Kerugian. Dan pada awal tahun 2024 jumlah data telah memasuki angka 6000 data, yang membuktikan tingkat rawan kasus kecelakaan di daerah Jawa Timur. Pendataan ini dilakukan setiap awal bulan, dan diberikan akses terhadap data dengan menggunakan perangkat kantor. Beberapa data bersifat sensitif, seperti data kriminal dan data hilang temu dikarenakan memiliki data pribadi korban.

b. Patroli dengan tim kerja

Kegiatan kerja lapang ini mencakup satu tim kerja. Bersama dengan tim kerja yang ditunjuk melakukan patroli ke lokasi yang telah ditentukan. Kegiatan ini dilakukan bersama dengan sub bidang teknik komunikasi untuk melakukan perbaikan tower dan jaringan radio kepolisian, serta pendataan peminjaman alat-alat komunikasi kepada Kepolisian Resort yang telah ditentukan. Kegiatan patroli ini berlangsung lebih dari sehari, seluruh tim kerja akan diberikan fasilitas inap di tempat patroli. Hasil perbaikan akan dilaporkan kepada Kepala Sub Bidang dengan membawakan laporan terkait dengan kerusakan juga anggaran proyek patroli keluar kota. Dibawah bimbingan Bapak Briptu Fajar Wirawan yang merupakan anggota Teknik Komunikasi, selain melakukan proyek dari Teknik Informatika, beberapa anggota Teknik Informatika ditarik untuk membantu kegiatan patroli. Sebagai anggota TIK kepolisian, bersama tim kerja diarahkan untuk mengikuti kegiatan patroli ke 2 daerah:

- Probolinggo, pada hari pertama, melakukan pertemuan antara Bapak Kepala Sub Bidang Tekkom dengan Pak Kapolrestabes Probolinggo untuk pelaksanaan *maintenance* terhadap jaringan di Kepolisian Sektor Probolinggo. Proses *maintenance* tower dilakukan di hari kedua, dan proses dilaksanakan hingga malam hari. Lalu, pada hari ketiga dilakukan *meeting* dengan Kapolsek, untuk melaporkan kerusakan dan hasil perbaikan yang dilakukan tim TIK Polda Jatim. Kerusakan yang dialami mengharuskan untuk Kepolisian Probolinggo menggunakan perangkat baru dari tim TIK Polda Jatim, dan perangkat yang rusak akan diperbaiki di kantor utama TIK Polda Jatim di Ketintang, Surabaya.
- Malang, kegiatan patroli di Malang juga dalam rangka melakukan *maintenance*, namun kerusakan yang dialami

hanya pada perangkat radio dan *handy talky*. Perbaikan ini dilakukan langsung di kantor utama dan dikembalikan pada hari kelima. pada hari kedua sebelum kembali ke kantor utama, dilakukan pengecekan untuk memastikan kendala hanya berada pada perangkat kecil saja.

c. Operasi Ketupat Pre-lebaran

Di kepolisian terdapat operasi ketika akan diadakannya acara hari raya, salah satu kegiatan yang dilakukan yaitu Operasi Ketupat. Kegiatan ini dilakukan tim kerja untuk bergantian menjaga posko yang ditetapkan, salah satu lokasi poskonya yaitu kantor utama TIK Polda yaitu kantor Ketintang. Akan ditunjuk beberapa tim kerja yang mendapatkan *shift* berjaga selama 5 hari, dimulai dari tanggal 1 April 2024 hingga 5 April 2024. Pada masa berkegiatan, tim yang telah diberikan jadwal menginap selama melakukan operasi ini. Setelah operasi ketupat pre-lebaran, tim kerja dapat mengambil *shift* libur, dan berganti *shift* dengan tim kerja yang mendapatkan giliran untuk menjaga posko.



Gambar 3. 7 Dokumentasi Operasi Ketupat 2024

Pada gambar 3.7, kegiatan operasi ketupat dilakukan di posko yang ditentukan salah satunya Posko Asrama Polisi Ketintang. Kegiatan dilakukan bersama tim kerja Bapak Bripda Fery dan Bripda Taufiq.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

2. Pelatihan Analisis Data



Gambar 3. 8 Pelatihan Dengan Pembimbing dan Mentor

Sebelum melakukan proyek magang dalam menganalisis data kecelakaan lalu lintas, dilaksanakannya pelatihan analisis data dengan menggunakan data penelitian terdahulu yang telah menggunakan data terproses. Pada gambar 3.8, proses pelatihan dilakukan dengan melakukan konsultasi intensif untuk mendiskusikan serta menampilkan hasil pelatihan bersama Bapak Briptu

Fajar dan Briptu Eddy. Data diproses menggunakan Microsoft Excel secara manual. Proses *cleansing*, *formatting*, *converting*, dan *labelling data* dilakukan secara manual satu per satu.

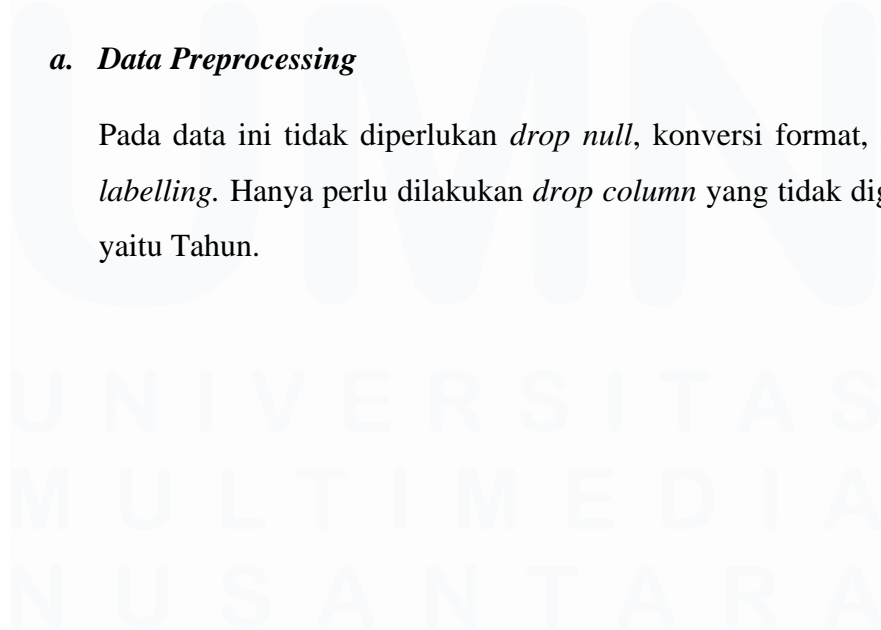
	A	B	C	D	E	F	G
1	Tahun	Hari Kejadian	Waktu Kejadian	Tingkat Kecelakaan	Tipe Kecelakaan	Kondisi Cahaya	Cuaca
2	2016	6	2	1		2	2
3	2016	6	1	1		13	1
4	2016	7	2	1		2	1
5	2016	1	1	1		2	1
6	2016	1	2	1		4	1
7	2016	1	3	2		3	1
8	2016	2	1	1		1	2
9	2016	2	4	1		5	2
10	2016	3	1	1		2	1
11	2016	7	3	3		6	1
12	2016	5	2	3		7	1
13	2016	5	3	2		8	1
14	2016	3	1	3		9	1
15	2016	6	4	1		10	1
16	2016	7	1	3		12	1
17	2016	7	1	1		11	1
18	2016	1	1	2		11	1
19	2016	1	1	1		1	1
20	2016	1	4	1		2	1
21	2016	2	1	3		14	1
22	2016	2	4	1		10	1
23	2016	3	1	1		4	1
24	2016	1	4	1		15	1
25	2016	4	1	1		8	2

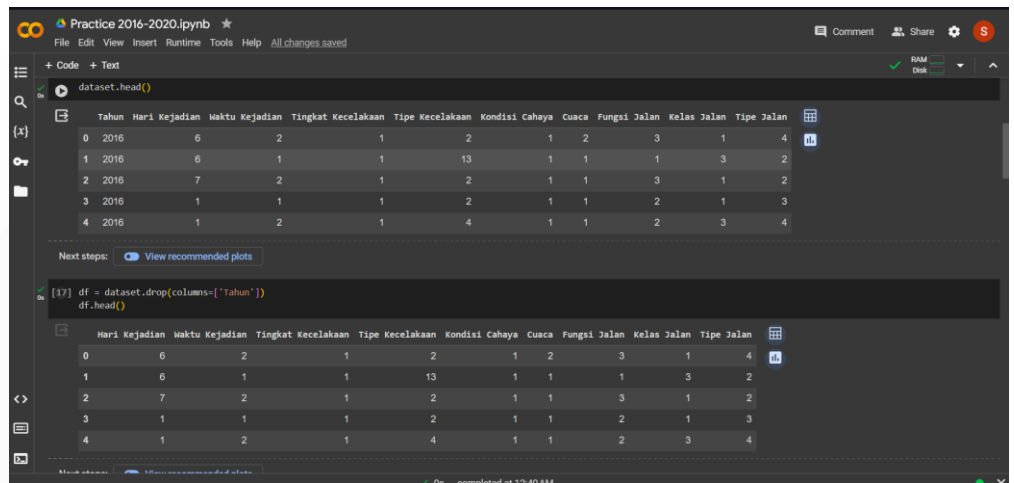
Gambar 3. 9 Data Terproses Pada Penelitian Sebelum

Gambar 3.9 merupakan gambar data yang digunakan dalam melakukan pelatihan analisis data. Dengan menggunakan data ini tidak diperlukannya *data pre-processing*, dikarenakan keseluruhan data telah menjadi numeric, hanya perlu melakukan proses *drop* kolom Tahun dan menggunakan SMOTE.

a. Data Preprocessing

Pada data ini tidak diperlukan *drop null*, konversi format, maupun *labelling*. Hanya perlu dilakukan *drop column* yang tidak digunakan yaitu Tahun.

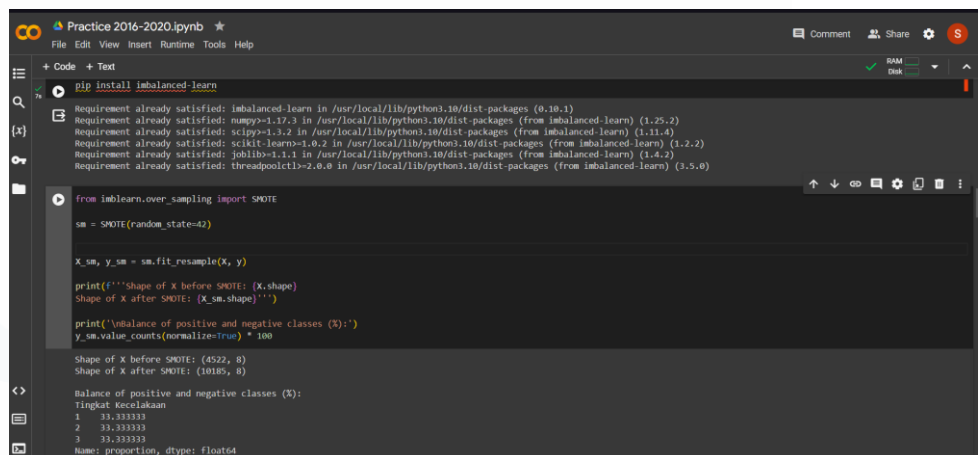




Gambar 3. 10 Melakukan Drop Kolom Year dan Konten Dataset

Pada gambar 3.10 ditampilkan proses penghapusan kolom tahun untuk mendapatkan seluruh data numerik serta faktor yang dibutuhkan. Setelah proses ini dilakukan, dapat mulai dibagikan menjadi data latih dan uji, dan nantinya akan dilakukan pemodelan menggunakan algoritma *random forest classifier*.

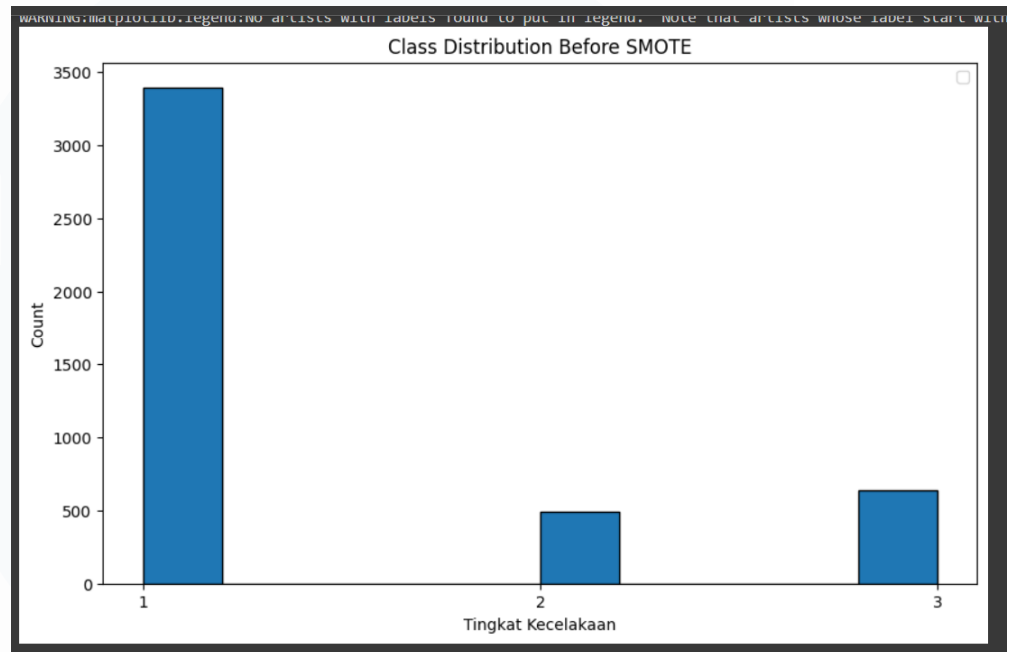
b. Penggunaan SMOTE



Gambar 3. 11 Penggunaan SMOTE Distribusi Data

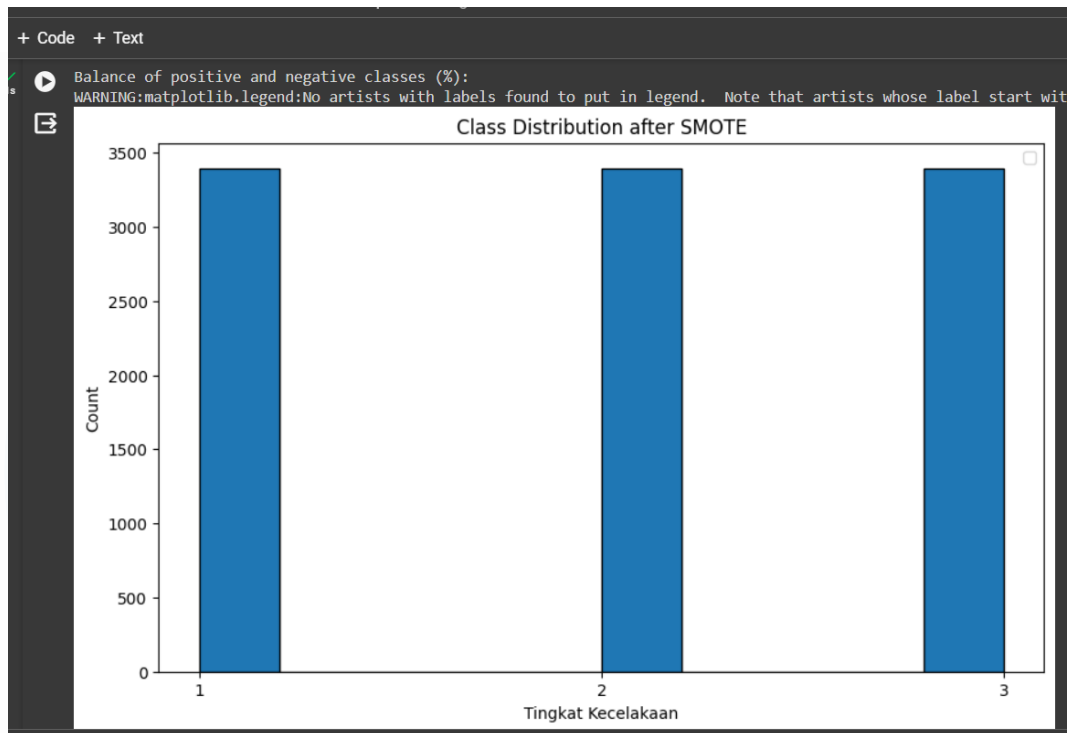
Pada gambar 3.11, dilakukannya distribusi agar data yang dilatih sama rata jumlah kategorinya. SMOTE ini melakukan *data balancing*

agar jumlah ketiga kategori yang akan menjadi klasifikasi utama setara.



Gambar 3. 12 Kelas Data Sebelum Distribusi SMOTE

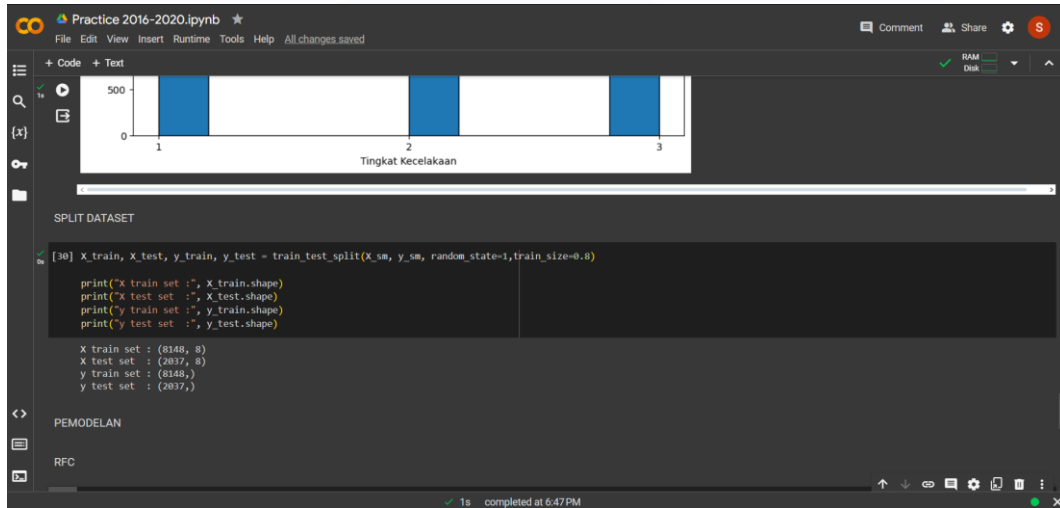
Dapat dilihat pada gambar 3.12, tingkat kecelakaan pada data dibagi menjadi 3, yaitu ringan, sedang, dan berat. Tanpa dilakukannya SMOTE, algoritma akan dengan mudah membaca klasifikasi tingkat kecelakaan ringan. Sedangkan, klasifikasi tingkat kecelakaan sedang dan berat memiliki jumlah dibawah 1000.



Gambar 3. 13 Kelas Data Setelah Distribusi SMOTE

Pada gambar 3.13, merupakan hasil distribusi yang dilakukan oleh SMOTE. Dengan menggunakan SMOTE, analisis data dapat menangani ketidakseimbangan sebuah data, juga mengatasi kasus *overfitting*. Kinerja dari model yang digunakan akan meningkat, dan SMOTE tidak sama sekali mengubah kelas mayoritas, yang dalam kasus ini yaitu tingkat kecelakaan ringan.

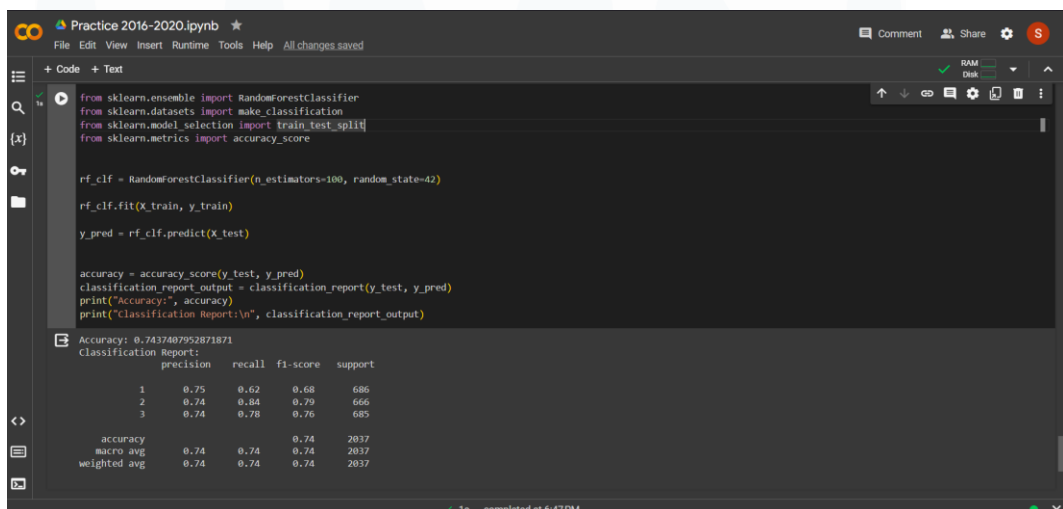
c. Pembagian data latih dan data uji



Gambar 3. 14 Pembagian Data Uji Dan Latih

Setelah menggunakan SMOTE untuk mendistribusikan kelas tingkat kecelakaan, dilakukannya *splitting data* atau pembagian data menjadi data latih dan data uji. Pada gambar 3.14, pembagian antara data latih dan data uji berbanding 8:2. Dengan jumlah data latih adalah 8000 data, dan jumlah data uji yaitu 2000 data.

d. Pemodelan menggunakan algoritma *random forest classifier*

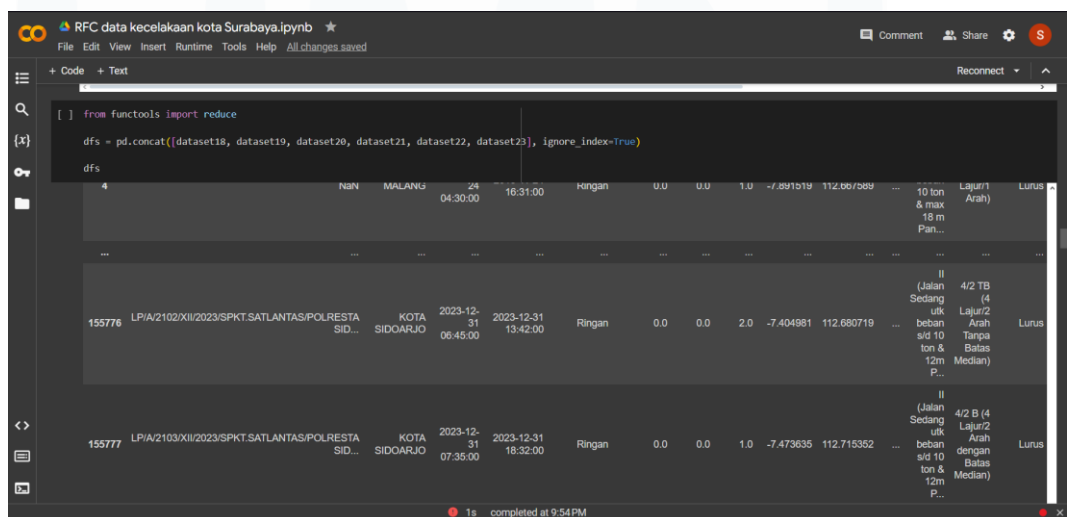


Gambar 3. 15 Hasil Pemodelan Algoritma Random Forest Classifier

Dapat dilihat pada gambar 3.15, pemodelan dilakukan dengan menggunakan algoritma *random forest classifier*, algoritma dipilih sebagai perbandingan dengan algoritma yang telah digunakan pada analisis data penelitian sebelum. Pada gambar 3.14, Hasil dari algoritma *random forest classifier* ditampilkan menggunakan *classification report*. Dengan rata-rata *precision*, *recall*, dan *f1-score* di angka 0.74, dan menghasilkan akurasi sebesar 0.744.

3. Proyek Analisis Data Kecelakaan Lalu Lintas Surabaya Tahun 2018-2023

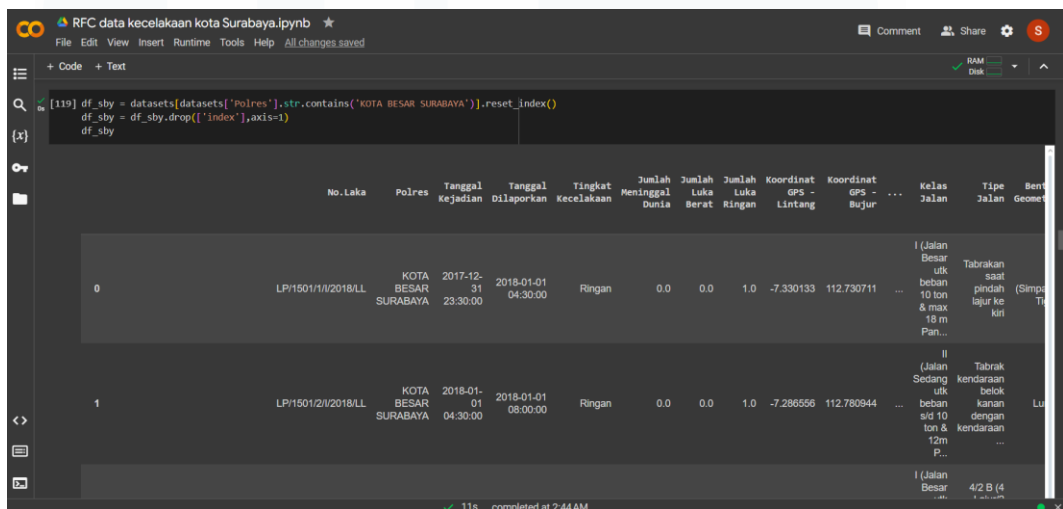
Proyek analisis data dilakukan menggunakan data kecelakaan lalu lintas di Surabaya, dengan mengajukan proposal bersama dengan bimbingan dalam mengumpulkan kelengkapan data kecelakaan tahun 2018-2023. Penggunaan data ini hanya dilakukan untuk kegiatan analisis data di bawah naungan Bapak Briptu Fajar Wirawan dalam memonitor hasil kerja proyek analisis data selama kegiatan magang. Pengumpulan data dilakukan secara bertahap, dimulai dari pengajuan proposal, konfirmasi kegiatan, serta penerimaan data kecelakaan lalu lintas. Analisis data dilakukan sebagai proyek utama kegiatan magang yang dilakukan di Polda Jawa Timur.



Gambar 3. 16 Menggabungkan Data Kecelakaan Lalu Lintas Jawa Timur Tahun 2018 hingga 2023

Pada gambar 3.16, dataset menggunakan kumpulan dari 6 data yaitu data kecelakaan lalu lintas Jawa Timur tahun 2018 hingga 2023. Data digabungkan menggunakan fungsi *concat*, dan menghasilkan satu *big data* dengan jumlah data mencapai 155.777 data kecelakaan selama 6 tahun di daerah Jawa Timur. Data dikumpulkan dari seluruh Kepolisian Resort dan dilaporkan ke Kepolisian Daerah Jawa Timur selama masa laporan.

a. Data Preprocessing



Gambar 3. 17 Data Kecelakaan Daerah Surabaya

Selanjutnya dilakukan *data preprocessing*, pada gambar 3.17 dilakukannya pemilihan lokasi data yang akan dianalisis. Pada laporan proyek magang ini, lokasi yang dituju yaitu Kota Surabaya. Dengan mengambil data hanya di Polres Kota Besar Surabaya sebagai objek penelitian data analisis. Setelah memilih lokasi data, data yang awalnya berjumlah 150 ribu menjadi 7000 data.

```

RFC data kecelakaan kota Surabaya.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
df_sby['Tanggal Kejadian'] = pd.to_datetime(df_sby['Tanggal Kejadian'])

def categorize_time(hour):
    if 6 <= hour < 12:
        return 'Pagi'
    elif 12 <= hour < 18:
        return 'Siang'
    elif 18 <= hour < 21:
        return 'Sore'
    else:
        return 'Malam'

day_mapping = {
    'Monday': 'Senin',
    'Tuesday': 'Selasa',
    'Wednesday': 'Rabu',
    'Thursday': 'Kamis',
    'Friday': 'Jumat',
    'Saturday': 'Sabtu',
    'Sunday': 'Minggu'
}

df_sby['Hari Kejadian'] = df_sby['Tanggal Kejadian'].dt.day_name().map(day_mapping)
df_sby['waktu kejadian'] = df_sby['Tanggal Kejadian'].dt.hour
df_sby['waktu kejadian'] = df_sby['waktu kejadian'].apply(categorize_time)
df_sby['Tahun'] = df_sby['Tanggal Kejadian'].dt.year
df_sby = df_sby[df_sby['Tahun'] != 2017]
df_sby
3s completed at 3:37 AM

```

Gambar 3. 18 Pengubahan Format Waktu

Pada gambar 3.18, dilakukan pengubahan format waktu dari *datetime* dibagi menjadi Tahun, Hari, dan Waktu Kejadian. Pengubahan format waktu dilakukan agar dapat mengklasifikasi waktu rawan kecelakaan. Pada proses ini, format *datetime* diambil poin intinya, seperti waktu paling rawan, hari paling rawan, dan tahun sebagai visualisasi laporan tingkat kecelakaan tahunan di Polda Jawa Timur.

```

df = df_sby[['Tahun', 'Hari Kejadian', 'waktu kejadian', 'Tipe Kecelakaan', 'Kondisi Cahaya', 'Cuaca', 'Fungsi Jalan', 'Kelas Jalan', 'Tipe Jalan', 'Tingkat Kecelakaan']]
df

```

	Tahun	Hari Kejadian	waktu kejadian	Tipe Kecelakaan	Kondisi Cahaya	Cuaca	Fungsi Jalan	Kelas Jalan	Tipe Jalan	Tingkat Kecelakaan
1	2018	Senin	Malam	Tabrak kendaraan belakang kanan dengan kendaraan...	Terang / Jelas	Cerah	Arteri	II (Jalan Sedang utk beban s/d 10 ton & 12m P...	Tabrak kendaraan belakang kanan dengan kendaraan...	Ringan
2	2018	Selasa	Pagi	Di ruas jalan, pejalan kaki menyeberang dari k...	Terang / Jelas	Cerah	Kolektor	I (Jalan Besar utk beban 10 ton & max 18 m Pan...	4/2 B (4 Lajur/2 Arah dengan Batas Median)	Ringan
4	2018	Selasa	Malam	Tabrakan saat gerakan putar balik	Terang / Jelas	Berawan / Mendung	Arteri	I (Jalan Besar utk beban 10 ton & max 18 m Pan...	Tabrakan saat gerakan putar balik	Ringan
5	2018	Selasa	Sore	Tabrakan kendaraan belakang kiri dengan kendaraan...	Terang / Jelas	Cerah	Arteri	I (Jalan Besar utk beban 10 ton & max 18 m Pan...	Tabrakan kendaraan belakang kiri dengan kendaraan...	Ringan
6	2018	Selasa	Pagi	Tabrakan depan - belakang	Terang / Jelas	Cerah	Kolektor	I (Jalan Besar utk beban 10 ton & max 18 m Pan...	2/1 (2 Lajur/1 Arah)	Ringan
...
7005	2023	Jumat	Malam	Tabrakan depan - depan	Terang / Jelas	Cerah	Lokal / Lingkungan	II (Jalan Sedang utk beban s/d 10 ton & 12m P...	2/2 TB (2 Lajur/2 Arah Tanpa Batas Median)	Berat
7006	2023	Sabtu	Pagi	Di simpang, tabrakan dengan kendaraan B yang d...	Terang / Jelas	Cerah	Arteri	II (Jalan Sedang utk beban s/d 10 ton & 12m P...	6/2 B (6 Lajur/2 Arah dengan Batas Median)	Ringan
7007	2023	Minggu	Malam	Tabrakan kendaraan belakang kanan dengan kendaraa...	Terang / Jelas	Cerah	Arteri	I (Jalan Besar utk beban 10 ton & max 18 m Pan...	2/2 TB (2 Lajur/2 Arah Tanpa Batas Median)	Ringan
7008	2023	Minggu	Malam	Tabrakan kendaraan belakang dengan kendaraan...	Terang / Jelas	Cerah	Kolektor	II (Jalan Sedang utk beban s/d 10 ton & 12m P...	2/2 TB (2 Lajur/2 Arah Tanpa Batas Median)	Berat

Gambar 3. 19 Pemilihan Seluruh Variabel Penting

Pada gambar 3.19, dilakukan pemilihan variabel atau faktor yang digunakan untuk klasifikasi. Variabel yang digunakan disamakan

dengan variabel pada saat pelatihan analisis data. Menggunakan variabel independen berupa Tahun, Hari Kejadian, Waktu Kejadian, Tipe Kecelakaan, Kondisi Cahaya, Cuaca, Fungsi Jalan, Tipe Jalan, dan Kelas Jalan untuk melakukan klasifikasi terhadap variabel dependen yaitu Tingkat Kecelakaan.

```

[125] print(df.isnull().sum())
print("\nNaN Total", df.isnull().sum().sum())

Tahun      0
Hari Kejadian  0
Waktu Kejadian  0
Tipe Kecelakaan  0
Kondisi Cahaya  10
Cuaca      1
Fungsi Jalan  12
Kelas Jalan  15
Tipe Jalan  2
Tingkat Kecelakaan  17
dtype: int64

NaN Total      57

[126] df = df.dropna()
print("\nNaN Total", df.isnull().sum().sum())
print("Dataset Shape:", df.shape)

NaN Total : 0
Dataset Shape: (6939, 10)

[127] sns.heatmap(df.isnull(), char=False, cmap="BuPu_r")
plt.title("Missing Values")
plt.show()

```

Gambar 3. 20 Drop Null

Dengan melakukan proses *drop null* pada gambar 3.20, terdapat total 57 *null* pada data yang akan dibuang. Proses ini disebut dengan *dropping null*, menggunakan fungsi *dropna* atau *drop not a number* pada *data frame* yang digunakan. Sisa data setelah dilakukannya *drop null* berjumlah 6939 data.

```

le = LabelEncoder()

df["Tipe Kecelakaan"] = le.fit_transform(df["Tipe Kecelakaan"])
df["Kondisi Cahaya"] = le.fit_transform(df["Kondisi Cahaya"])
df["cuaca"] = le.fit_transform(df["cuaca"])
df["Fungsi Jalan"] = le.fit_transform(df["Fungsi Jalan"])
df["Kelas Jalan"] = le.fit_transform(df["Kelas Jalan"])
df["Tipe Jalan"] = le.fit_transform(df["Tipe Jalan"])

df.head()

```

	Tahun	Hari Kejadian	Waktu Kejadian	Tipe Kecelakaan	Kondisi Cahaya	Cuaca	Fungsi Jalan	Kelas Jalan	Tipe Jalan	Tingkat Kecelakaan
1	2018	Senin	Malam	23	2	1	0	1	20	Ringan
2	2018	Selasa	Pagi	3	2	1	1	0	5	Ringan
4	2018	Selasa	Malam	44	2	0	0	0	35	Ringan
5	2018	Selasa	Sore	39	2	1	0	0	31	Ringan
6	2018	Selasa	Pagi	33	2	1	1	0	1	Ringan

```

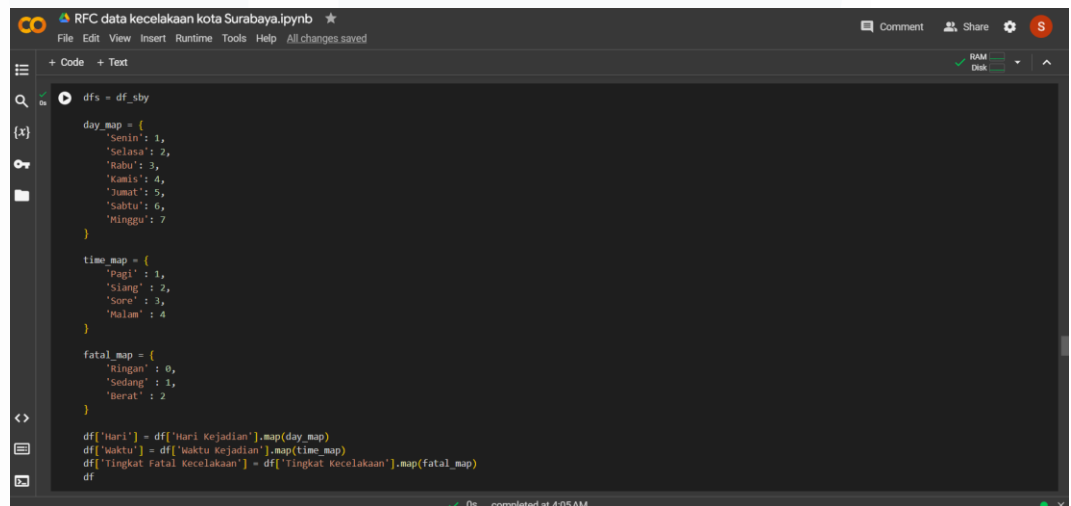
[169] dfs = df_sby

day_map = {
    "Senin": 1,
    "Selasa": 2,

```

Gambar 3. 21 Proses Labelling Variabel

Gambar 3.21 merupakan proses *Labelling* menggunakan fungsi *Label Encoder*, yang mengubah kategori *multiclass* menjadi angka. *Label encoder* akan mencari *unique value* atau variabel-variabel yang terdapat pada suatu kelas, yang nantinya akan dikonversi ketika menemukan variabel yang berbeda secara bertahap. Proses ini dilakukan agar memenuhi syarat untuk melaksanakan pemodelan menggunakan algoritma, yaitu diharuskan keseluruhan variabel berbentuk angka atau *numeric*.



```
dfs = df_sby

day_map = {
    'Senin': 1,
    'Selasa': 2,
    'Rabu': 3,
    'Kamis': 4,
    'Jumat': 5,
    'Sabtu': 6,
    'Minggu': 7
}

time_map = {
    'Pagi': 1,
    'Siang': 2,
    'Sore': 3,
    'Malam': 4
}

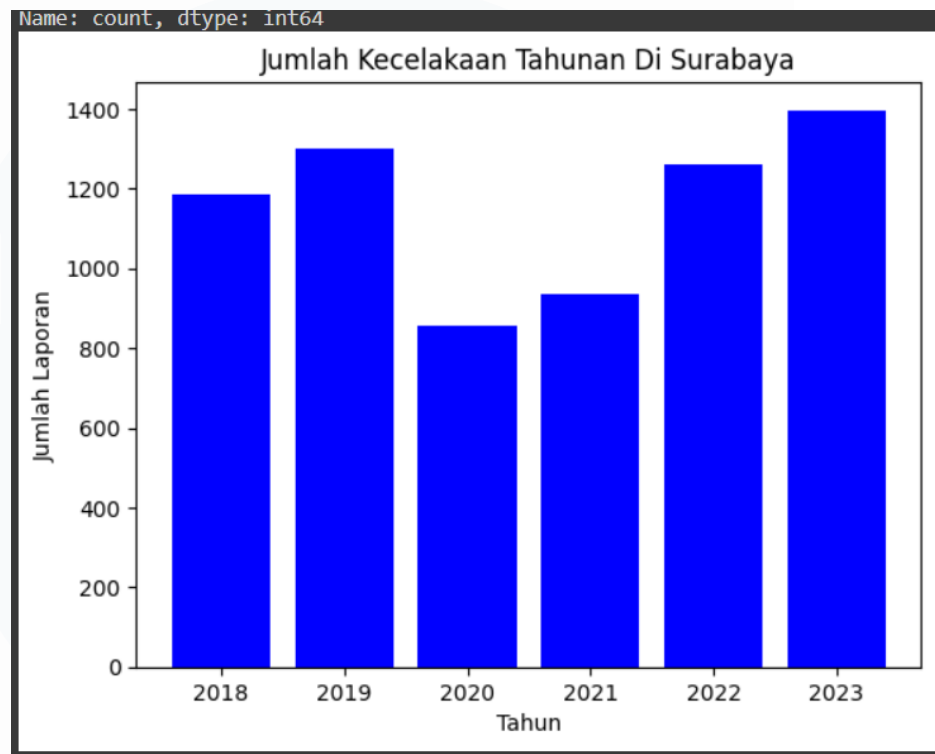
fatal_map = {
    'Ringan': 0,
    'Sedang': 1,
    'Berat': 2
}

df['Hari'] = df['Hari Kejadian'].map(day_map)
df['waktu'] = df['waktu Kejadian'].map(time_map)
df['Tingkat Fatal Kecelakaan'] = df['Tingkat Kecelakaan'].map(fatal_map)
df
```

Gambar 3. 22 Proses Labelling Variabel Berurut

Pada *data frame* terdapat variabel yang berurut, seperti Hari Kejadian, Waktu Kejadian, dan variabel dependen yaitu Tingkat Kecelakaan. Proses *labelling* variabel ini secara manual agar angka yang dihasilkan berurutan. Pada gambar 3.22, terlihat contoh dari *labelling* secara manual dengan mengubah *value* Pagi menjadi 1, Siang menjadi 2, Sore menjadi 3, dan Malam menjadi 4. Begitu juga dengan Hari Senin hingga Minggu dan Tingkat Kecelakaan Ringan hingga Berat.

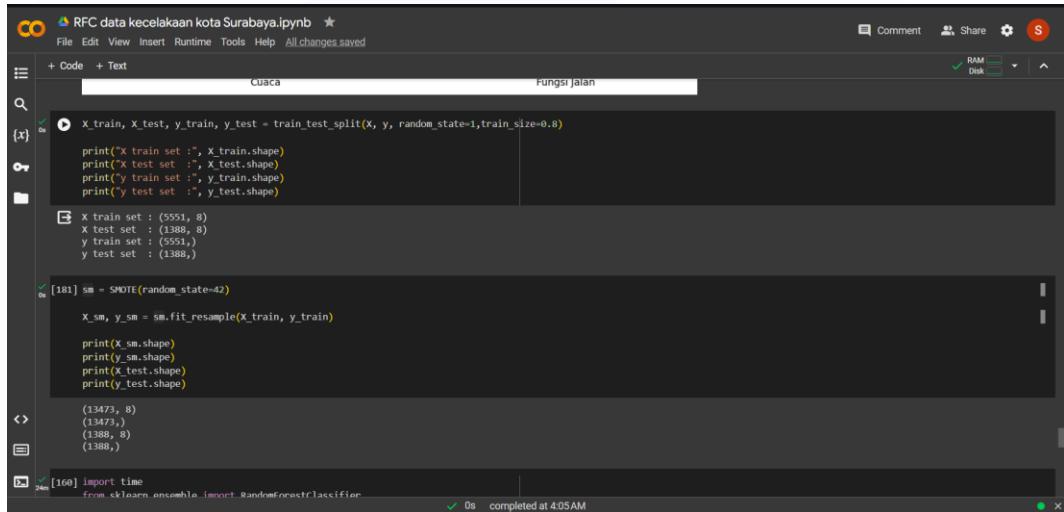
b. Visualisasi



Gambar 3. 23 Visualisasi Angka Kecelakaan Tahunan Di Surabaya

Pada gambar 3.23 menjelaskan jumlah laporan kecelakaan yang masuk di Polres Kota Besar Surabaya. Dimana rata-rata kecelakaan lalu lintas yang terjadi setiap tahunnya mencapai angka 1150. Penurunan angka kecelakaan pada tahun 2020 merupakan berita baik, namun kasus pada tahun selanjutnya tidak dapat dikurangi melainkan meningkat cukup signifikan di 3 tahun kemudian. Pada tahun 2023 jumlah tertinggi angka kecelakaan selama 6 tahun ini, mencapai angka 1397 angka kasus kecelakaan lalu lintas. Ditotalkan secara keseluruhan, data yang tercatat di kepolisian mencapai 6900 angka kasus kecelakaan yang terjadi selama 6 tahun di Surabaya.

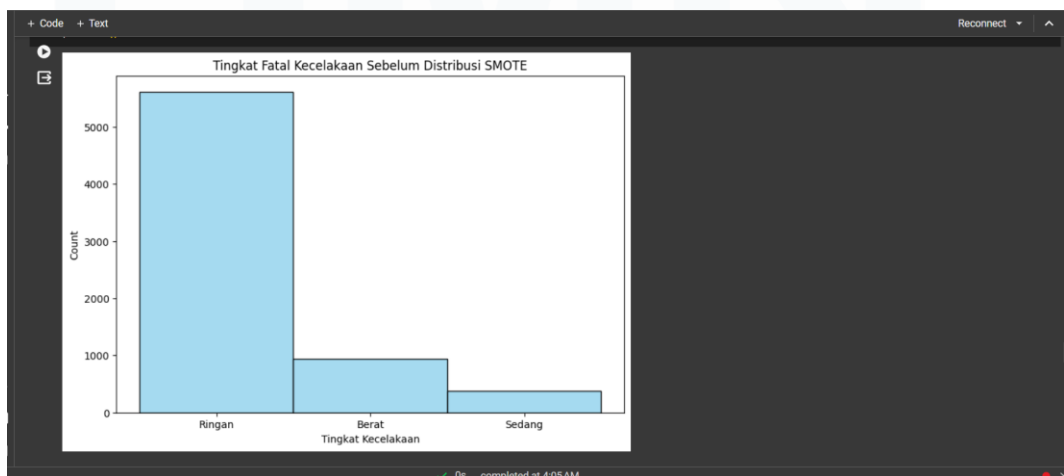
c. Penggunaan SMOTE



```
File Edit View Insert Runtime Tools Help All changes saved
RFC data kecelakaan kota Surabaya.ipynb
+ Code + Text
Cuaca Fungsi Jalan
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1, train_size=0.8)
print("X train set :", X_train.shape)
print("X test set :", X_test.shape)
print("y train set :", y_train.shape)
print("y test set :", y_test.shape)
X train set : (5551, 8)
X test set : (1388, 8)
y train set : (5551,)
y test set : (1388,)
[181] sm = SMOTE(random_state=42)
X_sm, y_sm = sm.fit_resample(X_train, y_train)
print(X_sm.shape)
print(y_sm.shape)
print(X_test.shape)
print(y_test.shape)
(13473, 8)
(13473,)
(1388, 8)
(1388,)
[160] import time
from sklearn.ensemble import RandomForestClassifier
0s completed at 4:05AM
```

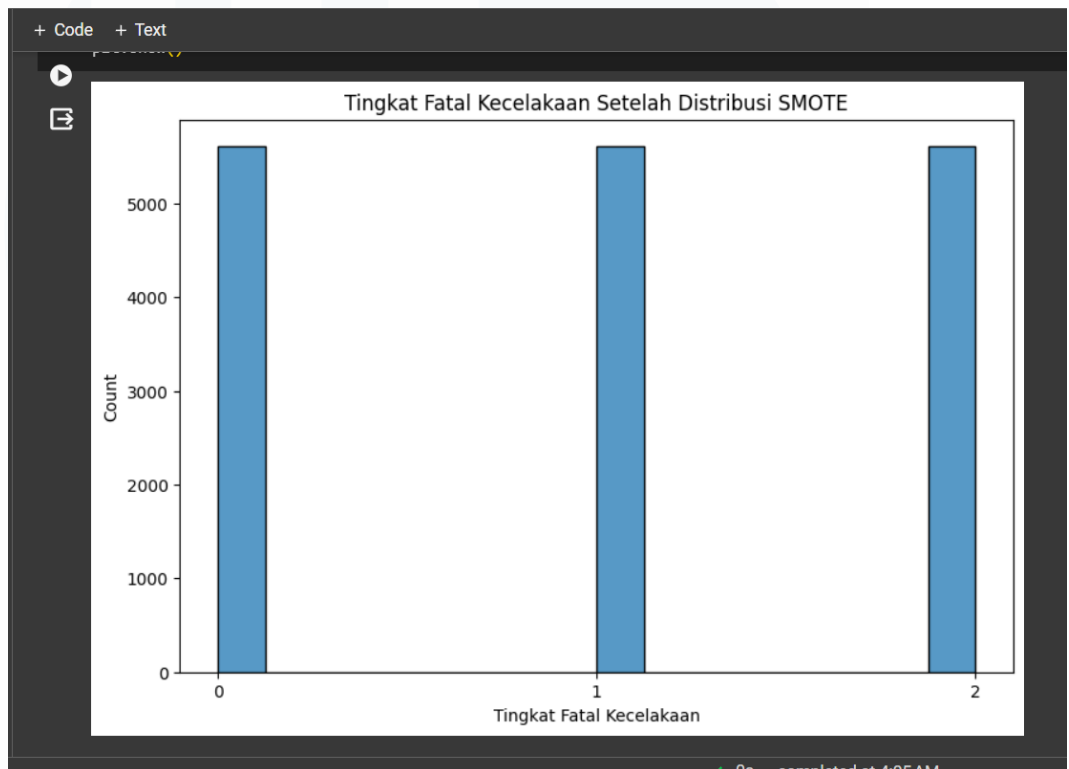
Gambar 3. 24 Distribusi Data Latih Menggunakan SMOTE

Pada gambar 3.24, SMOTE digunakan untuk mendistribusikan data latih. maka dari itu penggunaan SMOTE dilakukan setelah dilaksanakan pembagian data latih dan data uji. Hal ini dikarenakan pengujian algoritma diharuskan menggunakan data *real* atau data yang belum diolah. Pada masa latihan, digunakannya SMOTE terhadap kedua data memberikan peningkatan terhadap akurasi algoritma, dikarenakan kesamaan antara data latih dan data uji signifikan. Pada proyek ini dilakukan mengikuti prosedur SMOTE yang digunakan hanya kepada data latih saja.



Gambar 3. 25 Data Latih Sebelum Distribusi SMOTE

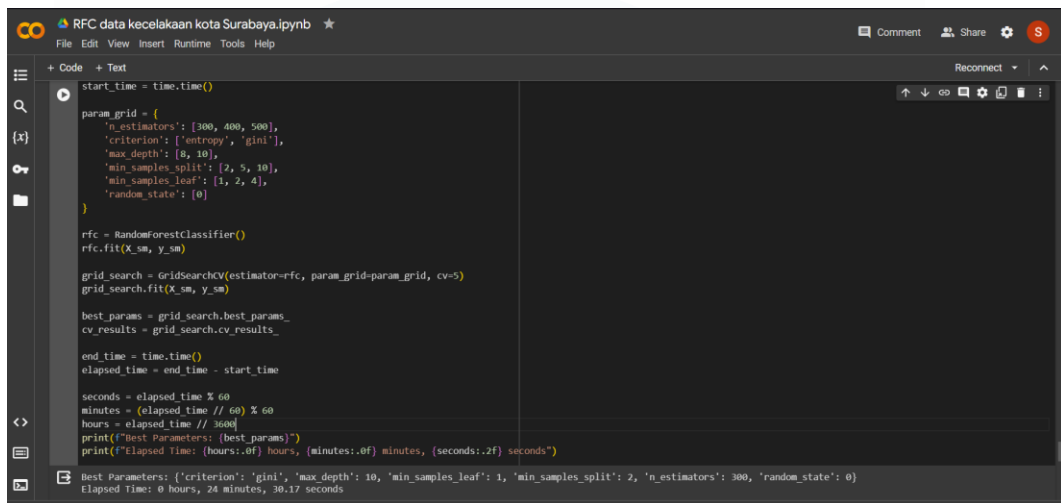
Kelas data pada Gambar 3.25 ditampilkan sangat bias terhadap tingkat kecelakaan ringan. Maka dari itu, digunakan SMOTE agar pelatihan model algoritma *random forest classifier* tidak bersifat bias.



Gambar 3. 26 Data Latih Setelah Distribusi SMOTE

Setelah melakukan pendistribusian SMOTE, data latih untuk model yang dikembangkan akanimbang kepada ketiga kelas tingkat kecelakaan seperti pada gambar 3.26. Pada gambar 3.24, hasil dari pembagian data latih dan uji yang memiliki rasio 8:2, yaitu 5.500 banding 1.300. Dan setelah didistribusikan dengan SMOTE data latih memiliki jumlah data 13.400 data.

d. Pengaplikasian *tuning parameter*



```
start_time = time.time()

param_grid = {
    'n_estimators': [300, 400, 500],
    'criterion': ['entropy', 'gini'],
    'max_depth': [8, 10],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'random_state': [0]
}

rfc = RandomForestClassifier()
rfc.fit(X_sm, y_sm)

grid_search = GridSearchCV(estimator=rfc, param_grid=param_grid, cv=5)
grid_search.fit(X_sm, y_sm)

best_params = grid_search.best_params_
cv_results = grid_search.cv_results_

end_time = time.time()
elapsed_time = end_time - start_time

seconds = elapsed_time % 60
minutes = (elapsed_time // 60) % 60
hours = elapsed_time // 3600
print("Best Parameters: (best_params)")
print("Elapsed Time: (hours:.0f) hours, (minutes:.0f) minutes, (seconds:.2f) seconds")

Best Parameters: {'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300, 'random_state': 0}
Elapsed Time: 0 hours, 24 minutes, 30.17 seconds
```

Gambar 3. 27 *Tuning Parameter Random Forest Regressor*

Pada gambar 3.27 proses *tuning parameter* diaplikasikan untuk mencari *hyper parameter* pada algoritma *random forest classifier*.

Kombinasi parameter yang digunakan yaitu:

'n_estimators': [300, 400, 500],

'criterion': ['entropy', 'gini'],

'max_depth': [8, 10],

'min_samples_split': [2, 5, 10],

'min_samples_leaf': [1, 2, 4],

'random_state': [0]

Kombinasi akan dicoba satu per satu oleh pembelajaran mesin untuk mencari akurasi tertinggi dari seluruh kombinasi parameter yang ditentukan. Proses pembelajaran mesin memakan waktu selama 25 menit dan menghasilkan kombinasi parameter terbaik.

```

RFC data kecelakaan kota Surabaya.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Best Parameters: {'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300, 'random_state': 0}
Elapsed time: 0 hours, 24 minutes, 30.17 seconds

scores = cv_results['mean_test_score']
results_df = pd.DataFrame(cv_results['params'])
results_df['Score'] = scores

results_df = results_df.sort_values(by='Score', ascending=False).reset_index(drop=True)

results_df['Rank'] = results_df.index + 1
para_frame = results_df[['Rank'] + list(results_df.columns[:-1])]
para_frame[:5]

```

Rank	criterion	max_depth	min_samples_leaf	min_samples_split	n_estimators	random_state	Score
0	1	gini	10	1	2	300	0.690199
1	2	gini	10	1	2	400	0.690050
2	3	gini	10	1	2	500	0.689234
3	4	entropy	10	1	2	500	0.687749
4	5	entropy	10	1	2	400	0.687155

```

rfc_best = RandomForestClassifier(**best_params)
rfc_best.fit(X_train, y_train)

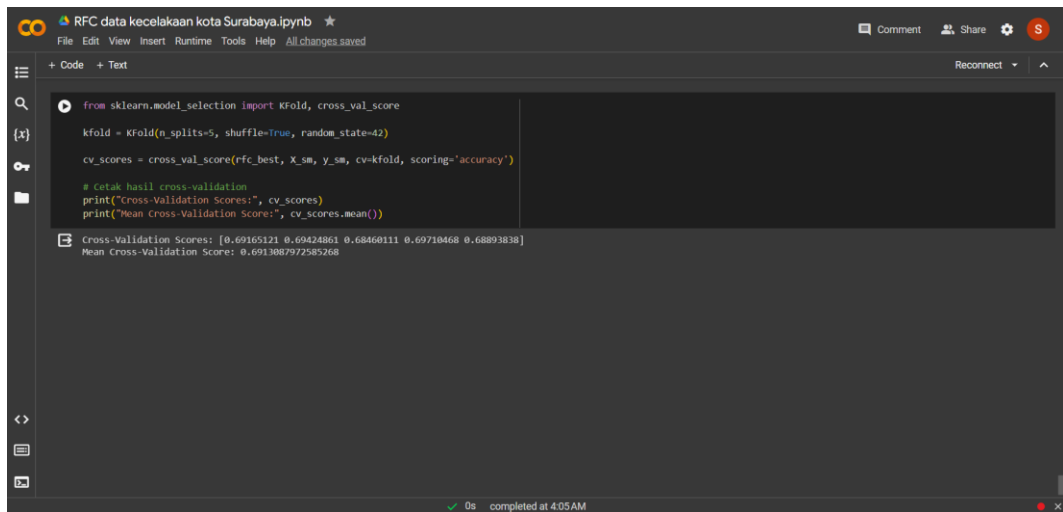
```

0s completed at 4:05 AM

Gambar 3. 28 Peringkat Parameter Terbaik oleh *Tuning Parameter*

Kemudian dilihat dari gambar 3.28, hasil dari *tuning parameter* dibuatkannya ke dalam tabel agar menampilkan 5 kombinasi parameter terbaik untuk pemodelan algoritma *random forest classifier* menggunakan data latih. Hasil kombinasi parameter tertinggi mendapatkan akurasi sebesar 0.69 dengan parameter berupa:

- 'criterion': 'gini',
- 'max_depth': 10,
- 'min_samples_leaf': 1,
- 'min_samples_split': 2,
- 'n_estimators': 300,
- 'random_state': 0



```
from sklearn.model_selection import KFold, cross_val_score
kfold = KFold(n_splits=5, shuffle=True, random_state=42)
cv_scores = cross_val_score(rfc_best, X_sm, y_sm, cv=kfold, scoring="accuracy")
# cetak hasil cross-validation
print("Cross-Validation Scores:", cv_scores)
print("Mean Cross-Validation Score:", cv_scores.mean())
```

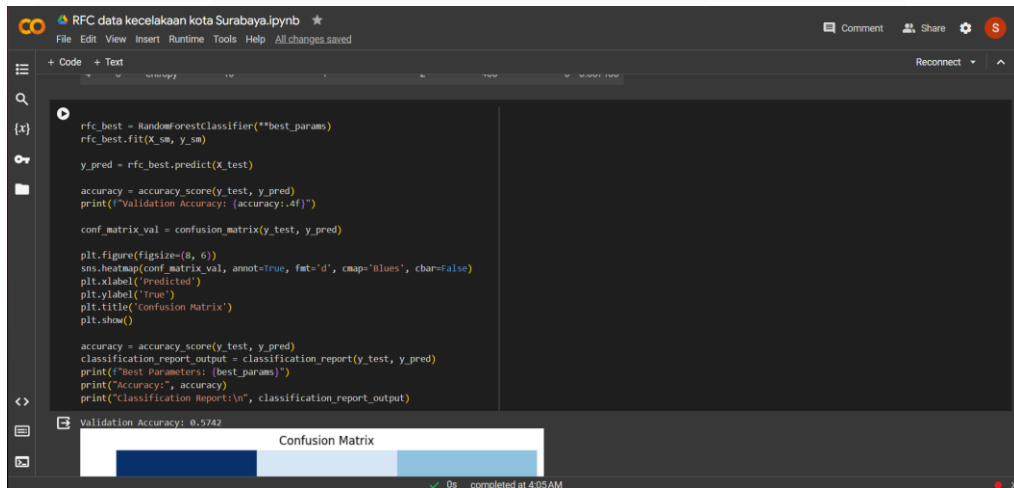
Cross-Validation Scores: [0.69165121 0.69424861 0.68460111 0.69710468 0.68893838]
Mean Cross-Validation Score: 0.6913087972585268

0s completed at 4:05AM

Gambar 3. 29 *Cross Validation* Hasil Pemodelan Data Latih

Pada gambar 3.29, *cross validation* digunakan untuk melakukan pengecekan akurasi secara berkala untuk membandingkan hasil akurasi parameter yang telah ditentukan melalui *tuning parameter*. Setelah dilakukan *cross validation* secara berkala, hasil akurasi mendekati akurasi dari pengukuran parameter pertama. Menghasilkan skor rata-rata 0.691. *KFold cross validation* melakukan pembagian terhadap dataset menjadi beberapa *fold* atau lipatan yang akan diukur akurasinya secara berkala.

e. Hasil Pemodelan Algoritma *Random Forest Classifier* Terhadap Data Uji



```
rfc_best = RandomForestClassifier(**best_params)
rfc_best.fit(X_sm, y_sm)

y_pred = rfc_best.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print("Validation Accuracy: {accuracy:.4f}")

conf_matrix_val = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_val, annot=True, fmt='d', cmap='Blues', char=False)
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("Confusion Matrix")
plt.show()

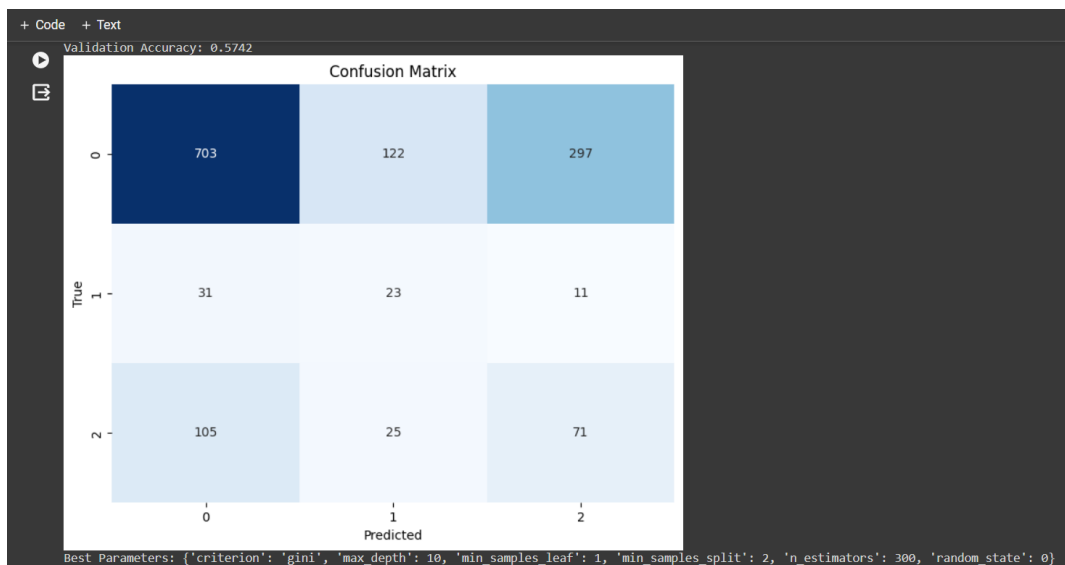
accuracy = accuracy_score(y_test, y_pred)
classification_report_output = classification_report(y_test, y_pred)
print("Best Parameters: (best_params)")
print("Accuracy:", accuracy)
print("Classification Report:\n", classification_report_output)
```

Validation Accuracy: 0.5742

Confusion Matrix

Gambar 3. 30 Pengujian Model Algoritma *Random Forest Classifier*

Gambar 3.30 menampilkan kerja *code* yang menerapkan pemodelan algoritma *random forest classifier* dengan mengambil parameter dari *tuning parameter* yang telah dilakukan. pada gambar 3.x ,parameter yang telah digunakan pada data latih menampilkan akurasi berupa 0.69. Namun, pada hasil data uji yang merupakan data asli belum terkelola, hasil dari model *random forest classifier* turun mencapai angka 0.57. Hal ini dikarenakan perbedaan sifat antara data latih dan data uji, yang dimana data latih yang digunakan sudah dilakukan proses distribusi data agar data bersifat *balance*. Sedangkan pada tahapan pengujian data, data uji diharuskan untuk tidak dikelola atau didistribusikan agar sifat data tetap sama.



Gambar 3. 31 *Confusion Matrix* Terhadap Ketiga Tingkat Kecelakaan

Dapat dilihat pada gambar 3.31 *confusion matrix*, data latih memiliki sifat bias yang cukup tinggi, angka 0 atau tingkat kecelakaan ringan mendominasi data, yang menyebabkan turunnya tingkat akurasi. Penurunan ini merupakan hal yang normal, dikarenakan tingkat kecelakaan ringan memiliki perbandingan yang signifikan terhadap kelas tingkat kecelakaan sedang dan tingkat kecelakaan berat.

```

BEST PARAMETERS: { 'CRITERION': 'gini', 'max_depth': 10,
Accuracy: 0.5742074927953891
Classification Report:

```

	precision	recall	f1-score	support
0	0.84	0.63	0.72	1122
1	0.14	0.35	0.20	65
2	0.19	0.35	0.24	201
accuracy			0.57	1388
macro avg	0.39	0.44	0.39	1388
weighted avg	0.71	0.57	0.62	1388

Gambar 3. 32 Hasil Laporan Klasifikasi Tingkat Rawan Kecelakaan

Rincian dari pengujian model ditampilkan menggunakan *classification report*. Pada gambar 3.32, terdapat hasil angka akurasi

model algoritma *random forest classification* yaitu 0.574. Dan dapat dilihat juga pada data uji jumlah tingkat kecelakaan ringan lebih besar 3 banding 1 terhadap tingkat kecelakaan sedang dan berat. Dan hasil klasifikasi dirata-ratakan mendapat *precision* sebesar 0.39, *recall* 0.44, dan *f1-score* 0.39.

3.3 Kendala yang Ditemukan

Beberapa kendala yang dialami selama melakukan magang di Polda Jawa Timur sebagai *intern data analyst*, berupa:

1. Proses *data pre-processing* secara manual

Proses *data pre-processing* dilakukan secara manual, dengan melakukan konversi label satu persatu. Juga proses penghapusan parameter yang tidak dibutuhkan dilakukan dengan mengubah data excel menjadi data baru. Hal ini akan menghambat kinerja analisis data dan dapat terjadi kesalahan dalam mengubah label.

2. Tidak menggunakan *tuning parameter* pada setiap algoritma

Analisis data dilakukan tidak menggunakan proses *tuning parameter*, menyebabkan algoritma yang digunakan harus diproses satu per satu jenis parameternya. Dengan ini, keakuratan algoritma yang digunakan masih belum pasti, walaupun tingkat akurasi tinggi dapat saja sebuah algoritma mengalami *overfitting* dikarenakan tidak setaranya akurasi data latih dan uji.

3. *Package SMOTE* tidak kompatibel

Ada package yang tidak kompatibel pada saat melakukan analisis data, *SMOTE* yang berfungsi sebagai data balancing, agar jumlah data latih seimbang pada parameter tingkat kecelakaan. Dikarenakan versi *SMOTE* terdahulu tidak dapat digunakan terhadap versi Python saat ini. Oleh karena itu, dilakukan research terhadap package *SMOTE* bersama tim kerja untuk

mengubah penggunaan distribusi data lainnya atau menetapkan penggunaan SMOTE.

4. Kompleksitas data terkait dengan bias kelas tingkat kecelakaan

Akurasi yang didapat dari hasil pemodelan algoritma terhadap data uji berupa 0.57. Hasil ini masih memiliki keraguan dikarenakan klasifikasi terhadap suatu kasus yang darurat berada pada angka yang kecil. Hal ini dikarenakan kompleksitas dari data yang cukup signifikan terkait dengan biasanya kelas tingkat kecelakaan. Jumlah kecelakaan ringan memiliki angka signifikan jauh dengan tingkat kecelakaan sedang dan berat.

3.4 Solusi atas Kendala yang Ditemukan

Dengan kendala ini, beberapa Solusi yang dapat dilakukan untuk kelancaran kegiatan magang di Polda Jawa Timur:

1. Memperbaharui proses *data pre-processing* agar lebih efektif dan efisien

Mengubah proses *data-preprocessing* dengan langsung melakukan *merge* terhadap data yang dibutuhkan. Data yang digunakan berjumlah 5 data, yaitu data kecelakaan tahun 2018, 2019, 2020, 2021, 2022, dan 2023. Menggunakan seluruh proses pada *code* Python untuk melakukan *cleansing*, *formatting*, *converting*, dan *labelling*. Proses ini dapat meningkatkan keefektifan dan keefisienan pengerjaan analisis, serta keakuratan dalam memproses sebuah data.

2. Mengaplikasikan *tuning parameter* pada proyek analisis data

Mengaplikasikan *tuning parameter* dalam menggunakan algoritma *Random forest classifier*. Dengan ini, analisis data yang dilakukan dapat mencari parameter terbaik untuk algoritma *random forest classifier*. Juga meningkatkan akurasi dan mengurangi potensi terjadinya *overfitting*.

3. *Research* penggunaan SMOTE *data balancing* pada versi Python terbaru

Dalam proses melakukan *research* penggunaan SMOTE untuk versi Python saat ini, tim kerja mencari solusi dan menemukan penggunaan SMOTE agar kompatibel dengan python saat ini. Mulai dari mencari versi Python yang lama hingga mencari prosedur distribusi data yang baru. Hingga menemukan penggunaan SMOTE pada versi terbaru Python dan berhasil untuk menggunakan distribusi data SMOTE pada proyek analisis data kecelakaan lalu lintas.

4. Menganalisis data kecelakaan di seluruh daerah Jawa Timur

Dengan angka data yang mencapai lebih dari ratusan ribu, akan sangat membantu proses pemodelan algoritma yang akan dirancang. Agar dapat mengklasifikasi dengan baik serta memberikan tingkat akurasi yang signifikan tinggi dibanding proyek analisis data di Surabaya saja.