BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Tabel 2.1 Penelitian Terdahulu

No.	Jurnal	Judul	Penulis	Metode	Hasil
1.	International Journal of Innovative Technology and Exploring Engineering (IJITEE) vol. 09. No. 7 pp. 1346-1350	Traffic Accidents Severity Prediction using Support Vector Machine Models	Zeinab Farhat, Ali Karouni, Bassam Daya, Pierre Chauvet, Nizar Hamadeh[5]	SVM	Menggunakan SVM dengan kernel RBF pada data kecelakaan di Lebanon, menghasilkan akurasi 86% untuk klasifikasi luka dan kematian. Relevan dengan penelitian ini yang juga menerapkan SVM dalam klasifikasi tingkat kecelakaan.
2.	Jurnal EKSPONENSIAL vol. 14 No.2	Klasifikasi Tingkat Keparahan Korban Kecelakaan Lalu Lintas di Kota Samarinda Menggunakan Algoritma K-Nearest Neighbor dan Naive Bayes	Salsabila N, Goejantoro R, Syaripuddin[4]	KNN, Naïve Bayes	Menerapkan model KNN dan Naïve Bayes pada data kecelakaan Polres Kota Samarinda tahun 2020 dan 2021. Memiliki kemiripan *scope* dengan penelitian ini untuk mengambil data kecelakaan dari kepolisian. Total data yang dikelola sebanyak 291 data, dan menghasilkan akurasi model KNN di 75.9% dan model Naïve Bayes di 79.3%.
3.	Evolusi: Jurnal Sains dan Manajemen, Vol 12 No. 2 pp. 42-49	Optimalisasi Algoritma Random Forest Menggunakan SMOTE untuk Prediksi Pembatalan Tamu Hotel	Candra Agustina, Eka Rahmawati[6]	Random Forest, SMOTE	Menerapkan teknik SMOTE yang similar dengan penelitian ini. Model random forest yang dihasilkan mengalami peningkatan akurasi dari 88% menjadi 90% akurasi setelah menggunakan SMOTE.

4.	Jurnal Nasional Teknologi Informasi dan Aplikasinya vol. 03. No. 1 pp. 940-948	Klasifikasi Tingkat Keparahan Kecelakaan Lalu Lintas Menggunakan Random Forest Classifier	Purbhawa G, Wibawa G[7]	Random Forest Classifier, Oversampling	Menerapkan teknik *oversampling* juga untuk data *open source* dari Kaggle. Data yang digunakan similar yaitu data kecelakaan dengan total data mencapai 12.300 data. Menggunakan model random forest untuk mengklasifikasi tingkat kecelakaan juga, teknik *oversampling* mampu meningkatkan akurasi dari 85% menjadi 92%.
5.	Conference: 15th International Conference on Developments in eSystems Engineering (DeSE) pp. 480-485	Data Augmentation Using Generative Adversarial Networks to Reduce Data Imbalance with Application in Car Damage	Mahyoub M, Natalia F, Sudirman S, Liatsis P[8]	VGG 19, Augmentation, DCGAN	Menerapkan teknik *sampling* atau *augmentation* terhadap data gambar kendaraan. Penelitian ini juga melakukan perbandingan terhadap teknik *augmentation* standar dan DCGAN untuk optimasi model VGG19 terhadap gambar kerusakan kendaraan. Teknik *augmentation* berhasil meningkatkan akurasi validasi menggunakan model VGG19, yang semula mendapatkan akurasi 44.1%, dengan menggunakan teknik standar hasil meningkat menjadi 84.6%, dan menggunakan model DCGAN, akurasi dapat meningkat mencapai 85.3%
6.	UNP Journal of Statistics and Data Science Vol. 2 No.1 pp. 8-15	Classification the Characteristics of Traffic Accident Victims in Pariaman Using the Chi-square Automatic	Manja Danova Putri, Dina Fitria, Zilrahmi[9]	Chi-squared Automatic Interaction Detection	Penelitian mencakup klasifikasi tingkat kecelakaan menjadi ringan dan meninggal. Menggunakan data Kota Pariaman tahun 2022, dengan total 338 data. Dengan menggunakan model CHAID untuk data

		Interaction Detection Algorithm			wilayah Kota Pariaman berhasil mendapatkan akurasi 91%
7.	International Journal of Informatics and Communication Technology Vol. 13 No. 1 pp. 42-49	Traffic accident classification using IndoBERT	Naufal M, Girsang A[10]	IndoBERT, SVM, RandomForest	Penelitian klasifikasi terhadap data kecelakaan lalu lintas, menggunakan data dari *open source* Kaggle yang memiliki 4.245 data. Dikategorikan menjadi kecelakaan berdasarkan kendaraan korban, motor, mobil, bis, dan lainnya, Menghasilkan nilai akurasi tinggi pada algoritma IndoBERT yang mencapai 94%. Percobaan menggunakan model SVM mendapatkan akurasi sebesar 88% dan RandomForest mendapatkan 86%.
8.	Scientific Journal of Informatics Vol. 11 No. 2 pp. 401-412	The Impact of Balanced Data Techniques on Classification Model Performance	Pardede J, Prasetia Pamungkas D[11]	KNN, Naïve Bayes, Decision Tree, SMOTE, B-SMOTE, SMOTE-ENN	Penelitian membandingkan rasio SMOTE *oversampling* ke dalam 10 tahapan. Proses rasio dimulai dari *oversampling* sebesar 10% sampel utama hingga 100% sampel utama. Proses ini mengubah jumlah kategori yang lebih sedikit menjadi persetase rasio kategori yang lebih banyak. Pada percobaan tipe SMOTE, yaitu B-SMOTE dan SMOTE-ENN. Menghasilkan akurasi tertinggi menggunakan SMOTE ENN yaitu 98% dengan rasio *oversampling* 100% menggunakan model KNN. Dilanjutkan dengan kedua model lainnya menggunakan Decision

					Tree yaitu 90% dan 91% dengan menggunakan rasio 100% juga.
9.	Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) Vol. 11 No. 5 pp. 1033-1041	Penerapan SMOTE Untuk Mengatasi Imbalance Class Dalam Klasifikasi Kepribadian MBTI Menggunakan Naive Bayes Classifier	Persada Pulungan M, Purnomo A, Kurniasih A[12]	SMOTE, Naïve Bayes, Logistic Regression	Penelitian menggunakan SMOTE untuk melakukan optimalisasi *sample* dengan menggunakan beberapa model algoritma Naïve Bayes. Hasil peningkatan model multinomial dan Logistic Regression, dari 71% dan 79% menjadi 75% dan 80%. Sedangkan model lainnya mengalami penurunan atau menetap. Seperti model Bernoulli turun signifikan dari 68% menjadi 58%.
10.	International Journal of Transportation Science and Technology	Classification of traffic accidents' factors using TrafficRiskClassifier	Sun W, Abdullah L, Khalid F, Sulaiman P [13]	TrafficRisk Classifier	Penelitian menggunakan algoritma TrafficRiskClassifier dalam mengklasifikasi data kecelakaan. Data dikumpulkan dari data visual, video, dan tekstual yang dikonversi ke dalam bentuk tekstual yang akan diklasifikasi menjadi 3 tingkat kecelakaan, Ringan, Serius, dan Fatal. Penelitian ini melakukan data latih terhadap setiap variabel, mengindikasikan keadaan pengendara, jalan dan musim. Hasil akurasi yang dilakukan klasifikasi ketiganya mencapai rata-rata 82 % sampai 95%. Proses dilanjutkan dengan mengklasifikasi tingkat

					kecelakaan berdasarkan faktor utama yang sangat mempengaruhi, dan menghasilkan nilai akurasi sebesar 98% pada kecelakaan fatal, 86% serius, dan 85% ringan.
11.	PLoS ONE Vol. 16 No. 1 pp. 1-29	Classification of road traffic injury collision characteristics using text mining analysis: Implications for road injury prevention	Giummarra M, Beck B, Gabbe B[14]	Text Mining, Chi Square Test	Penelitian ini melakukan analisis data menggunakan chi square untuk menghitung p-value terhadap teks data kecelakaan. Mengklasifikasi respon terhadap post-kecelakaan, baik laporan kecelakaan antar dua atau lebih kendaraan, kendaraan dengan pejalan kaki, kehilangan kendali, juga bersangkutan dengan hewan liar.
12.	International Journal of Technology and Education Research Vol. 2 No. 2 pp. 62-77	Multinomial Logistic Regression Model to Analysis Traffic Accident on Indonesia's Regional Data	Tripena A, Apriliana Y, Prabowo A, Sugandha A[15]	Multinomial Logistic Regression	Penelitian menggunakan anlisis metode statistic multinomial logistic regression terhadap data kecelakaan Kota Cilacap tahun 2021, yang memiliki jumlah data sebanyak 867 data. Menghasilkan final output berupa p-value dari seluruh variabel prediksi, seperti tipe kecelakaan, Lokasi, situasi, dan jumlah korban

Berdasarkan tabel 2.1, banyak penelitian yang mengangkat tema kecelakaan lalu lintas, dimana tema ini merupakan kasus nyata yang menjadi permasalahan umum tidak hanya mencakup wilayah bahkan secara Nasional. Beberapa model klasik hingga kompleks modern telah diangkat namun belum ada praktik yang

mengkombinasikan model dengan variasi *sampling.* Beberapa penelitian terkait mengangkat tema ini dengan analisis probabilita statistia dan klasifikasi data kecelakaan. Terdapat model yang mampu menganalisis data gambar, tekstual, dan tabular. Menggunakan beragam model seperti Random Forest, KNN, Naïve Bayes, SVM, IndoBERT, dan TrafficRiskClassifier. Pengembangan model juga sering disandingkan dengan teknik *sampling* atau *augmentation* untuk mengoptimasi model terhadap kompleksitas sebuah dataset. Hasil dari beberapa penelitian terhadap klasifikasi kecelaakan menggunakan model KNN sebesar 76% dan Naïve bayes 79% tanpa *oversampling.* Ada juga model klasik seperti SVM yang mampu melakukan klasifikasi tingkat kecelakaan dan mendapatkan hasil akurasi 86%. Beberapa penelitian juga membuktikan peningkatan angka akurasi menggunakan teknik *sampling* atau *augmentation* dengan hasil peningkatan 2-5%, bahkan akurasi validasi yang meningkat hingga 41.2%.

Maka dari itu, penelitian ini akan mencoba membandingkan menggunakan model klasik yaitu algoritma KNN, SVM, Decision Tree, dan Naïve Bayes dengan menggunakan metode SMOTE untuk kasus klasifikasi data kecelakaan lalu lintas di wilayah Jawa Timur tahun 2024. Decision Tree merupakan model baru yang ingin diangkat pada penelitian ini untuk bereksperimen terhadap dataset kecelakaan lalu lintas Polda Jatim. Data yang digunakan memiliki 28.000 data, diperoleh dari Kepolisian Daerah Jawa Timur. Membawakan model algoritma baru yang diharapkan dapat mengklasifikasi Tingkat kecelakaan lebih baik dari algoritma sebelumnya. Penelitian ini juga ingin menggunakan teknik *oversampling* (SMOTE, ROS, dan RUS) untuk membandingkan performa dan kecocokan model untuk mengklasifikasi tingkat kecelakaan.

2.2 Teori Penelitian

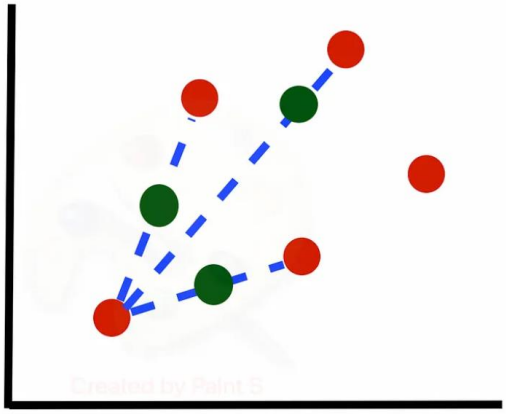
2.2.1 Laka Lantas

Laka lantas merupakan singkatan dari kecelakaan lalu lintas di instansi kepolisian. Pada markas kepolisian terdapat unit satuan kerja berfokuskan di bidang lalu lintas, yaitu satuan kerja lantas. Satuan kerja

lantas menyelenggarakan tugas dalam membina fungsi lalu lintas, baik melakukan penyidakan dan penegakkan hukum lalu lintas untuk memelihara kesejahteraan berkendara[16].

2.2.2 Metode *oversampling* SMOTE

SMOTE (*Synthetic Minority Oversampling Technique*) merupakan metode turunan dari metode *oversampling*. Proses SMOTE dilakukan dengan mensintetis kelas terkecil kemudian menduplikasi agar jumlah kelas seimbang. Proses ini dapat membantu meningkatkan akurasi dari prediksi dan akan mengatasi ketidakseimbangan kelas dari suatu data[6].



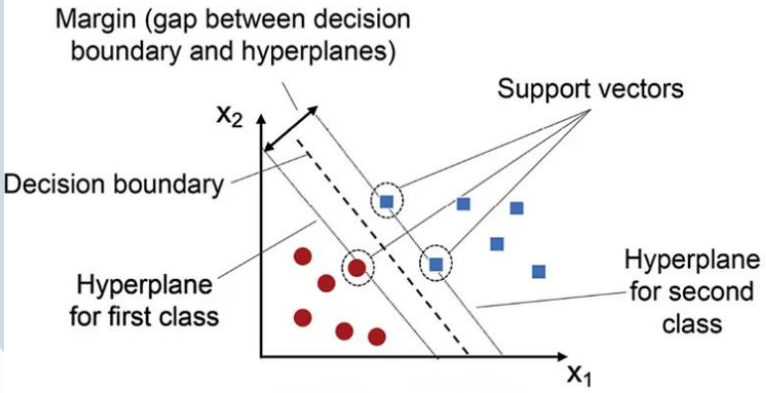
Gambar 2.1 Proses *Sampling* SMOTE[17]

Pada gambar 2.1, proses *oversampling* dilakukan pada nilai data yang kecil. Proses ini dilakukan dengan cara menghubungkan nilai k yang berdekatan, lalu membentuk nilai baru dengan sampel data yang baru. Hal ini dilakukan untuk menyeimbangkan data dengan ketidakseimbangan nilai yang tinggi[17]. Contoh pada data tingkat kecelakaan, yaitu jumlah data tingkat kecelakaan ringan lebih banyak dibandingkan tingkat rendah dan berat.

2.3 Framework dan Algoritma Penelitian

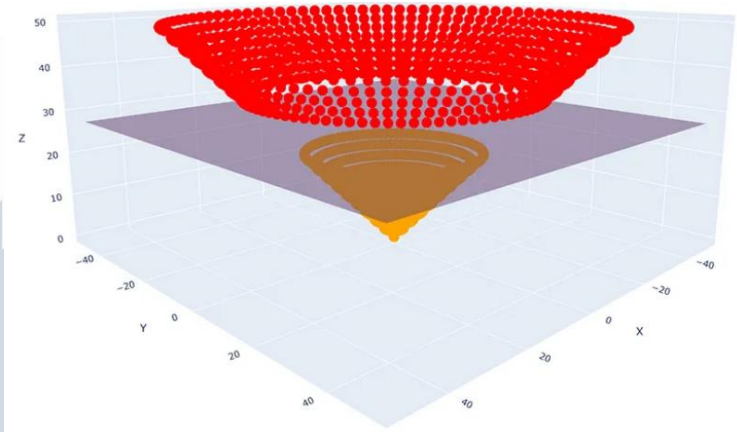
2.3.1 SVM

SVM (*Support Machine Vector*) merupakan sebuah algoritma *machine learning* yang digunakan untuk melakukan klasifikasi dan regresi. Seringkali digunakan dalam melakukan pengenalan pola, analisis gambar, juga pemrosesan *natural language*. Terdapat 2 cara kerja algoritma SVM klasifikasi, ada klasifikasi linear dan non-linear.



Gambar 2.2 Proses SVM Linear[18]

Pada gambar 2.2, proses SVM Linear yaitu dengan membentuk Margin atau garis yang membatasi dan memberikan celah terhadap nilai kategori. Pada gambar 2.2, dibagi menjadi 2 kubu, merah dan biru, Nilai yang menyentuh garis *hyperplane* disebut sebagai *support vector*. *Support vector* dari kubu merah menyentuh *hyperplane* pertama, maka yang diwakili memasuki kubu merah dikategorikan merah. Sebaliknya dengan kubu biru yang perwakilannya menyentuh *hyperplane* kedua[18].

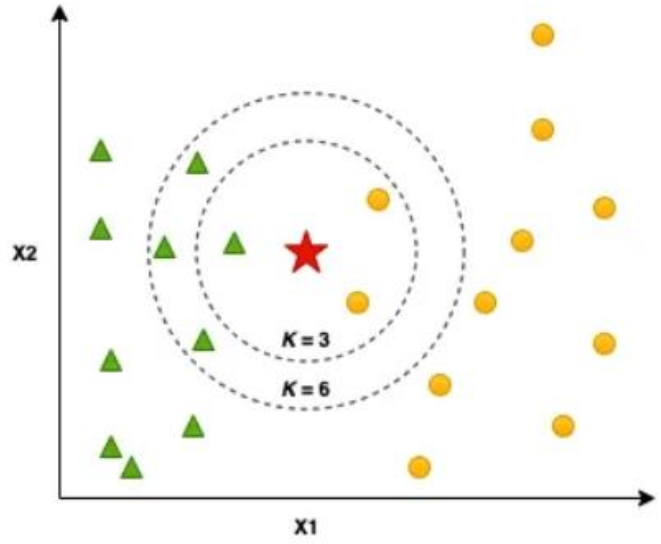


Gambar 2.4 Proses SVM Non-Linear[19]

Pada gambar 2.4, SVM Non-Linear bekerja dengan menggunakan fungsi kernel yang dapat membentuk ruang klasifikasi ke dalam dimensi yang lebih tinggi. Dalam gambar 2.4, SVM non-linear dibentuk ke dalam 3 dimensi yang memisahkan kedua kubu menjadi kubu atas dan bawah. Dengan menggunakan fungsi kernel, dapat dilakukan klasifikasi data yang kompleks[19].

2.3.2 K-Nearest Neighbors (KNN)

KNN atau K-Nearest Neighbors merupakan salah satu algoritma metode *data mining* yang sederhana dan umum digunakan. Klasifikasi data dilakukan melalui tingkat kemiripan 1 objek dengan objek lainnya[4].

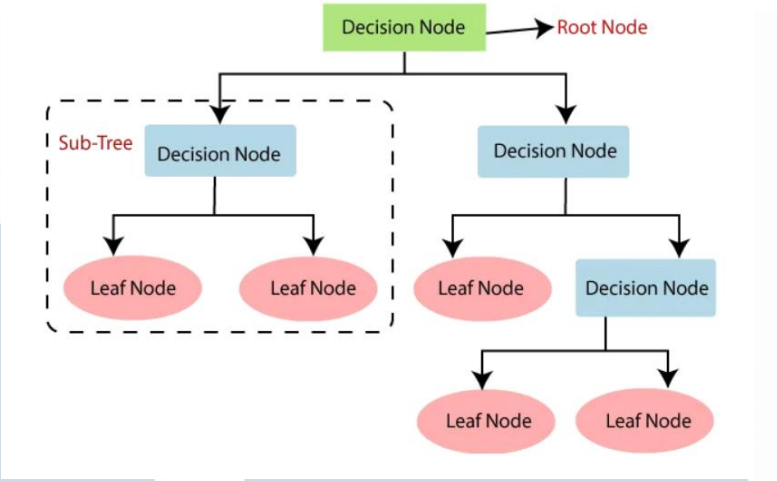


Gambar 2.3 Proses KNN[20]

Pada gambar 2.3, proses KNN diatas menggunakan 2 indikasi nilai yaitu nilai 1 digambar sebagai lingkaran kuning, dan 0 sebagai segitiga hijau. Nilai sebuah data uji yang berupa Bintang merah dikategorikan tergantung nilai K yang dimasukkan. Pada gambar 2.3, nilai data uji akan termasuk 1 jika K yang diungkapkan berupa 3, sedangkan data uji akan terkategorikan 0 jika K bernilai 6[20].

2.3.3 Decision Tree

Decision Tree merupakan algoritma *machine learning* yang tidak menggunakan asumsi terhadap distribusi sebuah data (non-parametric). Decision Tree dapat digunakan sebagai klasifikasi juga regresi data. Decision Tree merepresentasikan data kedalam bentuk struktur pohon.



Gambar 2.5 Proses Decision Tree[21]

Pada gambar 2.5, Decision Tree membentuk struktur pohon yang membagikan nilai utama sebagai *root node*, ke seluruh pilihan-pilihannya (*decision node*) yang seperti pola pikir manusia terhadap sebuah pilihan. Baik pilihan tersebut terdapat jawaban ya, tidak, atau terdapat pilihan baru lagi. Pada gambar 2.5, setelah mendapatkan Keputusan ya dan tidak dari sebuah pilihan akan terbentuk *sub-tree* yang tidak dapat membentuk cabang lagi. Pilihan yang membentuk cabang baru akan berlanjut hingga membentuk *sub-tree* baru.

2.3.4 Naïve Bayes

Naïve bayes merupakan Teknik klasifikasi yang menggunakan teorema bayes yang mengasumsikan bahwa seluruh fitur yang menjadi faktor prediksi tidak berhubungan satu sama lainnya. Cara kerja Naïve Bayes dengan menghitung probabilitas tertinggi setiap kelas yang akan dipilih untuk ditampilkan. Model ini juga sangat cocok untuk digunakan dengan permasalahan NLP (*Natural Languange Processing*). Pada model analisis klasifikasi digunakan tipe Naïve Bayes yaitu Gaussian, tipe ini bekerja dengan mengasumsikan keseluruhan fitur mengikuti distribusi yang normal[22].

2.3.5 CRISP-DM

CRISP-DM merupakan metode yang umum digunakan untuk memberikan proses terstruktur dalam melakukan *data mining.* Memiliki prosedur yang terstruktur, dimulai dengan memahami sebuah permasalahan dari data, proses ini akan membentuk tujuan yang ingin dicapai dengan data yang digunakan. Kemudian memahami isi dari data yang akan dianalisis dan data dipersiapkan dengan cara dibersihkan, konversi tipe data, juga pemilihan variabel terbaik. Setelah mempersiapkan data, akan dibentuk model algoritma untuk melakukan analisis terhadap data. Model ini akan menghasilkan prediksi terhadap klasifikasi yang dilakukan dan dievaluasi akurasi dari setiap algoritma yang digunakan. Proses terakhir, yaitu dengan dilakukan *deployment* atau mengembangkan model yang dibentuk menjadi program nyata yang dapat berfungsi untuk melaksanakan prediksi klasifikasi data[23].

2.3.6 Confusion Matrix

Confusion matrix merupakan metode yang dikembangkan untuk mengukur performa dan mengevaluasi model *machine learning* terhadap data yang diuji. Metode ditampilkan secara dua dimensi, mencakup nilai asli dan prediksi dari hasil. Dimensi ini menghasilkan indikasi hasil, berupa TP (*True Positive*), TN (*True Negative)*, FP (*False Positive*), dan FN (*False Negative*). Indikasi ini akan dihasilkan pengukuran *Accuracy, Precision, Recall,* dan F1-Score dalam bentuk persamaan sebagai berikut [12].

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Rumus 2. 1 Rumus *Accuracy*[24].

Pada rumus 2.1, terdapat rumus *accuracy* yang merupakan sebuah metrik pada *confusion matrix.* Berfungsi untuk menghitung jumlah prediksi yang tepat pada model secara keseluruhan [24].

$$Precision = \frac{TP}{TP + FP}$$

Rumus 2. 2 Rumus *Precision*[24]

Pada rumus 2.2, berupa rumus *precision* yang digunakan untuk menghitung prediksi positif yang benar pada model [24].

$$Recall = \frac{TP}{TP + FN}$$

Rumus 2. 3 Rumus *Recall*[24]

Pada rumus 2.3, *recall* berfungsi untuk mencari tahu rasio dari *true positive*. Dari keseluruhan data yang diprediksi positif, berapa yang terhitung benar[24].

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$$

Rumus 2. 4 Rumus *F1-Score*[24].

Pada rumus 2.4, *F1-Score* merupakan evaluasi metrik yang menghubungkan rumus *precision* dan *recall*. *F1-score* akan mengukur keseimbangan antara kedua rumus, yang berperan penting dalam kasus nilai data yang tidak seimbang[24].

2.4 Tools Penelitian

2.4.1 Python

Python adalah bahasa pemrograman tingkat tinggi yang populer digunakan untuk berbagai tugas, termasuk analisis data, kecerdasan buatan, komputasi ilmiah, dan pengembangan web. Python merupakan pilihan yang disukai baik bagi pemula maupun ahli karena kemudahannya, keterbacaannya, dan kemudahan penggunaannya. Selain itu, Python adalah bahasa sumber terbuka yang dapat digunakan dan dibagikan secara gratis[25]

2.4.2 Google Colab

Google Colab merupakan Software berbasis web untuk melakukan pemrograman *code* berbasis *python*. Google Colab juga telah menyediakan sebagian besar pustaka (*library*) untuk melakukan analisis, seperti, Keras, TensorFlow, NumPy, Pandas, Matplotlib dan pendukung lainnya. Versi Python yang diberikan juga beragam untuk mengatur kompatibilitas terhadap tersedia *library* yang membutuhkan versi spesifik. Google Colab menyediakan server untuk melaksanakan komputasi dan penyimpanan menggunakan *drive*. Proses komputasi model dilakukan tanpa menggunakan *hardware* pengguna, dan dapat mengurangi kendala pemrosesan selama berada di jangkauan internet[26].

2.4.3 Visual Studio Code

Visual Studio Code merupakan *software* buatan Microsoft untuk menjalankan kode pemrograman ringan namun *powerfull*. *Software* ini mendukung penggunaan bahasa pemrograman yang beragam, seperti Python, C++, JavaScript, dan lainnya. VS Code memberikan fitur untuk meningkatkan fleksibilitas pengguna dalam melakukan pemrograman, seperti IntelliSense (autocompletion), *integrated debugger,* GitControl, dan masih banyak lagi. VS Code juga menerapkan fungsi ekstensi yang dapat diinstal untuk memberikan fungsi lebih dalam meningkatkan efektifitas dan fleksibilitas saat melakukan pemrograman[27].