SKRIPSI

PERBANDINGAN FITUR STATISTIK DAN LINGUISTIK PADA ALGORITMA *K-NEAREST NEIGHBORS* (KNN) & *NEURAL NETWORK* DALAM PENGKLASIFIKASIAN SMS SPAM

Oleh
Muhammad Raihan Rizal
07352011006

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KAHIRUN
TERNATE
2024

LEMBAR PENGESAHAN

PERBANDINGAN FITUR STATISTIK DAN LINGUISTIK PADA ALGORITMA K-NEAREST NEIGHBORS (KNN) & NEURAL NETWORK DALAM PENGKLASIFIKASIAN SMS SPAM

Oleh
Muhammad Raihan Rizal
07352011006

Skripsi ini telah disahkan
Tanggal 26 April 2024

Menyetujui
Tim Penguji

Ketua Penguji

SYARIFUDDIN N, KAPITA, S.Pd., M.Si.
NIP. 199103122024211001

Pembimbing I

MUHAMMAD FHADLI, S.Kom., M.Sc.
NIP. 199611232023211012

Anggota Penguji

ACHMAD FUAD, S.T., M.T.
NIP. 197606182005011001

Pembimbing II

YASIR MUIN, S.T., M.Kom.
NIDN. 9990582796

Anggota Penguji

MUHAMMAD SABRI AHMAD, S.Kom., M.Kom.
NIP. 198905092019031013

Mengetahui/Menyetujui

Koordinator Program Studi
Informatika

ROSIHAN, S.T., M.Cs.
NIP. 197607192010121001

Dekan Fakultas Teknik
Universitas Khairun

Ir. ENDAH HARISUN, S.T., M.T., CRP.
NIP. 197511302005011013

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama	: Muhammad Raihan Rizal
NPM	: 07352011006
Fakultas	: Teknik
Jurusan/Program Studi	: Informatika
Judul Skripsi	: Perbandingan Fitur Statistik dan Linguistik Pada Algoritma *K-Nearest Neighbors* (KNN) & *Neural Network* Dalam Pengklasifikasian SMS Spam

Dengan ini menyatakan bahwa penulisan Skripsi yang telah saya buat ini merupakan hasil karya sendiri dan benar keasliannya. Apabila ternyata di kemudian hari penulisan Skripsi ini merupakan hasil plagiat atau penjiplakan terhadap karya orang lain, maka saya bersedia mempertanggung jawabkan sekaligus bersedia menerima sanksi berdasarkan aturan tata tertib di Universitas Khairun.

Demikian pernyataan ini saya buat dalam keadaan sadar dan tidak dipaksakan.

Penulis

Muhammad Raihan Rizal

HALAMAN PERSEMBAHAN

Dengan Rahmat Allah Subhanahu Wa Ta'ala yanga Maha Pengasih lagi Maha Penyayang serta mengucapkan rasa syukur Alhamdulillah atas nimkat yang di berikan tanpa hentinya, penulis persembahkan skripsi ini kepada:

1. Alm. Rizal Syarbin dan Nasbiah Umasangadji, kedua orangtua yang hebat dan selalu mendukung selama penulis menyelesaikan skripsi ini. Alhamdulillah kini penulis bisa berada di tahap ini, menyelesaikan pendidikannya sampai sarjana. Terimakasih sudah mengantarkan penulis berada ditempat ini.
2. Terima kasih untuk teman-teman Panda Squad yaitu Cindy Rahmawati S. Hipy, Suci Ayu Maharani, Sasmita Hi. Sadek, Lisa Elisia Potale, Nurwana Iswan, Aprilia Silawane, Harlina Sapsuha, Nafra Aziqra Hi. A, Marhama Maynaka, Wahyudin Nurdin, Nirwandi Bahri, Ferliy, dan Rinaldi Abdul karena telah membantu dan menyemangati saya dalam menyelesaikan skripsi ini.
3. Terima kasih kepada diri sendiri yang selalu semangat untuk menyelesaikan skripsi ini. Terima kasih telah mengandalkan diri sendiri untuk tetap kuat tanpa bergantung kepada orang lain dan tidak pernah memutuskan untuk menyerah dalam situasi apapun.

MOTTO

"Hidup tanpa tujuan adalah hidup tanpa kemajuan"

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah *Subhanaahu Wata'ala* dan sholawat serta salam kepada Nabi Muhammad *Shallallahu'alaihi Wassallam*, atas hidayah serta karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul "Perbandingan Fitur Statistik dan Linguistik Pada Algoritma *K-Nearest Neighbors* (KNN) & *Neural Network* Dalam Pengklasifikasian SMS Spam".

Penyusunan skripsi penelitian ini dibuat untuk memenuhi salah satu persyaratan kelulusan yang ada di Universitas Khairun Ternate Fakultas Teknik Program Studi Informatika. Dalam penulisan laporan ini tentu tidak lepas dari dukungan dan dorongan dari banyak pihak, oleh karena itu penulis mengucapkan banyak terima kasih yang tulus kepada:

1. Bapak Dr. M. Ridha Ajam, S.H., M.Hum., selaku Rektor Universitas Khairun Ternate.
2. Bapak Endah Harisun, S.T., M.T., selaku Dekan Fakultas Teknik.
3. Bapak Rosihan, S.T., M.Cs., selaku Koordinator Program Studi Informatika.
4. Bapak Muhammad Fhadli, S.Kom., M.Sc., selaku Pembimbing I, yang telah bersedia meluangkan waktunya untuk memberikan bimbingan, dukungan serta saran juga kritik yang sangat berarti bagi penulis dalam proses penyelesaian skripsi penelitian ini.
5. Bapak Yasir Muin, S.T., M.Kom., selaku Pembimbing II, yang telah bersedia meluangkan waktunya untuk memberikan bimbingan, dukungan serta saran juga kritik yang sangat berarti bagi penulis dalam skripsi penelitian ini.
6. Bapak Syarifuddin N. Kapita, S.Pd., M.Si., selaku Penguji I, yang telah bersedia meluangkan waktu untuk menguji dan memberikan masukan perbaikan demi menyempurnakan skripsi ini.
7. Bapak Achmad Fuad, S.T., M.T., selaku Penguji II, yang telah bersedia meluangkan waktu untuk menguji dan memberikan masukan perbaikan skripsi ini.
8. Bapak Muhammad Sabri Ahmad, S.Kom., M.Kom., selaku Penguji III, yang telah bersedia meluangkan waktu untuk menguji dan memberikan masukan perbaikan skripsi ini.
9. Civitas akademika Fakultas Teknik Universitas Khairun yang telah membantu dalam pelaksanaan pembuatan hasil ini.
10. Kedua orang tua yang selalu memberikan semangat dan doa, serta dukungan dan

perhatian, sehingga penulis dapat menyelesaikan skripsi penelitian ini dengan baik.

11. Tidak lupa juga ucapan terima kasih dan semangat kepada teman-teman seperjuangan yang saling bekerja sama, memberikan saran dan membantu satu sama lain.
12. Semua pihak yang tidak dapat penulis sebutkan yang telah membantu baik secara langsung maupun tidak langsung dalam menyelesaikan laporan skripsi ini.

Akhir kata penulis ucapkan rasa terima kasih kepada semua pihak dan apabila ada yang tidak disebutkan namanya penulis mohon maaf, dan kepada semua pihak yang telah membantu dalam penulisan ini semoga amal dan kebaikannya mendapat balasan yang berlimpah dari Allah *Subhanaahu Wata'ala*.

Ternate, 26 April 2024

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

ABSTRAK

PERBANDINGAN FITUR STATISTIK DAN LINGUISTIK PADA ALGORITMA *K-NEAREST NEIGHBORS* (KNN) & *NEURAL NETWORK* DALAM PENGKLASIFIKASIAN SMS SPAM

Muhammad Raihan Rizal[1], Muhammad Fhadli[2] , Yasir Muin[3]
Program Studi Informatika, Fakultas Teknik, Universitas Khairun
Jl. Jati Metro, Kota Ternate Selatan
Email: mrrbrayhan016@gmail.com[1], muhammadfhadli@gmail.com[2], yasirmuin123@gmail.com[3]

SMS masih banyak digunakan, namun keberadaan SMS spam telah menjadi masalah serius. Menurut Laporan *Truecaller Insights* tahun 2020, Indonesia mencatatkan jumlah pesan spam tertinggi di Asia, dengan kontribusi besar dari sektor layanan keuangan. Penelitian ini bertujuan untuk membandingkan pengaruh fitur statistik dan linguistik dalam klasifikasi SMS spam, dengan menggunakan algoritma *K-Nearest Neighbors* (KNN) dan *Neural Network* (NN). Metodologi yang diterapkan meliputi tahapan identifikasi masalah, perencanaan, pemodelan, evaluasi model, implementasi model, dan pengujian. Dalam penelitian ini, data diproses dengan menggunakan fitur statistik (TF-IDF) dan fitur linguistik sebelum diterapkan pada model KNN dan NN. Kinerja model dievaluasi berdasarkan metrik *precision*, *precision*, F1-*score*, dan akurasi. Hasil penelitian menunjukkan bahwa model NN yang menggunakan fitur statistik memiliki performa dengan akurasi mencapai 98%, KNN dengan fitur statistik 95%, NN dengan fitur linguistik 85% dan KNN dengan fitur linguistik 82%. Secara keseluruhan, NN fitur statistik menunjukkan performa yang lebih baik dibandingkan KNN di semua jenis fitur yang diuji. Dari hasil evaluasi ini, dapat disimpulkan bahwa fitur statistik lebih efektif dibandingkan fitur linguistik dan metode NN lebih unggul dibandingkan dengan metode KNN.

Kata Kunci: SMS spam, *K-Nearest Neighbors*, *Neural Network*, fitur statistik, fitur linguistik, klasifikasi.

COMPARISON OF STATISTICAL AND LINGUISTIC FEATURES IN K-NEAREST NEIGHBORS (KNN) & NEURAL NETWORK ALGORITHMS FOR SMS SPAM CLASSIFICATION

SMS is still widely used, but the presence of spam SMS has become a serious problem. According to the 2020 Truecaller Insights Report, Indonesia recorded the highest number of spam messages in Asia, with a significant contribution from the financial services sector. This study aims to compare the influence of statistical and linguistic features in SMS spam classification using the K-Nearest Neighbors (KNN) and Neural Network (NN) algorithms. The methodology applied includes problem identification, planning, modeling, model evaluation, model implementation, and testing stages. In this research, data is processed using statistical features (TF-IDF) and linguistic features before being applied to the KNN and NN models. The performance of the models is evaluated based on precision, recall, F1-score, and accuracy metrics. The results show that the NN model using statistical features achieves an accuracy of 98%, KNN with statistical features 95%, NN with linguistic features 85%, and KNN with linguistic features 82%. Overall, the NN with statistical features outperforms KNN in all tested feature types. From this evaluation, it can be concluded that statistical features are more effective than linguistic features, and the NN method is superior to the KNN method.

Keywords: *SMS spam, K-Nearest Neighbors, Neural Network, statistical features, linguistic features, classification.*

BAB I

PENDAHULUAN

1.1. Latar Belakang

Short Message Service (SMS), yang juga dikenal sebagai layanan pesan singkat, tetap menjadi salah satu alat komunikasi jarak jauh yang banyak digunakan di era saat ini untuk mengirim pesan singkat. Seiring dengan kemajuan layanan pesan singkat ini, timbul dampak negatif berupa serangan pada perangkat seluler dalam bentuk SMS spam. SMS spam adalah pesan singkat yang tidak diinginkan oleh penerima, termasuk pesan berisi iklan dan upaya penipuan (*fraud*) (Dwiyansaputra, 2021).

Menurut laporan yang dipublikasikan dalam *Truecaller Insights Report* (2020), Indonesia mencatatkan jumlah pesan spam tertinggi di Asia pada tahun 2020. Angka pesan spam di Indonesia menurun sebanyak 34% dari 27,9% pada tahun 2019 menjadi 18,3%, dan mengalami penurunan peringkat ke urutan keenam di antara 20 negara dengan tingkat pesan spam tertinggi. Jenis pesan spam yang paling umum di Indonesia adalah layanan keuangan, mencapai 52% dari total pesan spam, diikuti oleh asuransi sebesar 25%, operator seluler 11%, penipuan (*scam*) 9%, dan tagihan hutang 3% (Herwanto, 2021). Selain itu, berdasarkan laporan terbaru yang disajikan dalam *Truecaller Insights Report* (2021), posisi Indonesia tetap tak berubah dari tahun sebelumnya, yaitu masih berada di urutan keenam dari 20 negara di Asia. Jenis pesan spam yang dominan di Indonesia, terutama dalam layanan keuangan, mengalami peningkatan signifikan, mencapai 80% dari sebelumnya 52%.

Dalam konteks ini tindakan proaktif dan langkah-langkah yang efektif dapat membantu meminimalkan dampak dari pesan-pesan spam yang mengganggu. Fitur

linguistik dan statistik memiliki peran yang signifikan dalam mengklasifikasi SMS spam. Fitur linguistik melibatkan analisis struktur, pola, dan penggunaan bahasa dalam teks, sementara fitur statistik merepresentasikan teks menjadi angka agar mengetahui bobot yang ada pada teks (AL-Jumaili, 2020).

Dalam konteks pengklasifikasian SMS spam. KNN merupakan teknik yang sering digunakan untuk klasifikasi data. Cara kerja dari metode ini adalah dengan mengklasifikasikan data berdasarkan seberapa dekat jarak antara satu data dengan data lain (Laksono, 2020). Dalam kasus SMS spam, KNN akan mengidentifikasi dan mengklasifikasi pesan baru berdasarkan fitur linguistik dan statistik.

Di sisi lain, *Neural Network* adalah model yang terinspirasi dari jaringan saraf manusia. Model ini terdiri dari *input layer*, satu atau lebih *hidden layer*, dan *output layer*. Setiap neuron dalam jaringan ini mengolah informasi menggunakan fungsi aktivasi dan menyesuaikan bobotnya berdasarkan data yang diberikan (Yuliana, 2019). Dalam konteks SMS spam, NN dapat mempelajari pola kompleks dalam data, memungkinkan identifikasi pesan spam dengan lebih akurat berdasarkan fitur linguistik dan statistik

Berdasarkan latar belakang tersebut kontribusi penelitian ini akan berfokus pada pemodelan untuk membandingkan fitur statistik dan linguistik pada algoritma KNN dan *Neural Network* dalam mengklasifikasi SMS spam dan evaluasi perbandingan algoritma menggunakan *classification report* guna mengetahui performa dari masing masing model.

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas, berikut rumusan masalah:

1. Bagaimana tahapan perbandingan fitur statistik dan linguistik pada algoritma KNN dan NN terhadap SMS spam?

2. Bagaimana hasil evaluasi perbandingan fitur statistik dan linguistik pada algoritma KNN dan NN terhadap SMS spam?

1.3. Batasan Masalah

Berdasarkan masalah yang dibahas, penelitian ini memiliki batasan-batasan tertentu sebagai berikut:

1. Terdapat jenis bahasa pada SMS yang digunakan yaitu Bahasa Indonesia maupun *slank*.
2. Fitur statistik dan fitur linguistik diterapkan pada masing masing algoritma klasifikasi.
3. Hasil penelitian dari perbandingan fitur statistik dan linguistik pada algoritma *K-Nearest Neighbors* dan *Neural Network* dalam pengklasifikasian SMS spam yaitu berupa *classification report,* dan mengetahui SMS berupa spam (penipuan dan promo) atau ham.
4. *Output* dari keluaran yaitu spam penipuan, spam promo dan ham (normal).

1.4. Tujuan Penelitian

Adapun tujuan penelitian yang diangkat. Berikut adalah tujuannya:

1. Mengetahui tahapan perbandingan fitur statistik dan linguistik pada algoritma KNN dan NN terhadap SMS spam.
2. Mengetahui hasil evaluasi perbandingan fitur statistik dan linguistik pada algoritma KNN dan NN terhadap SMS spam.

1.5. Manfaat Penelitian

Berdasarkan latar belakang yang dijelaskan, Adapun manfaat penelitian yaitu:

1. Menambah wawasan terhadap algoritma yang digunakan.
2. Dengan adanya penelitian ini, kita dapat meminimalisir gangguan dan kebingungan yang disebabkan oleh pesan-pesan tidak diinginkan.

1.6. Sistematika Penulisan

Untuk memudahkan pembahasan dalam skripsi ini, sistematika penulisan dibagi menjadi 5 (lima) bab yang terdiri dari:

BAB I PENDAHULUAN

Bab ini mencakup konteks dan alasan di balik penelitian (latar belakang), termasuk perumusan masalah, pembatasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bagian ini menguraikan berbagai teori dan konsep dari sumber-sumber terkait yang menjadi dasar referensi utama dalam melakukan penelitian ini.

BAB III METODOLOGI PENELITIAN

Dalam bab ini, dijelaskan tentang pendekatan dan teknik yang digunakan oleh penulis dalam menangani penelitian sesuai dengan masalah yang diangkat.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan hasil yang diperoleh berupa hasil evaluasi dari model yang dibuat serta pengujian dalam mengklasifikasi sebuah pesan baru.

BAB V PENUTUP

Bab ini berisi kesimpulan yang didapatkan dari analisis model yang didapatkan serta saran untuk peneliti yang ingin melanjutkan penelitian ini.

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Pada bab ini, penelitian yang dilakukan penulis bukanlah yang pertama dalam konteks klasifikasi SMS spam. Sebelumnya, telah ada sejumlah penelitian terkait yang membahas topik yang sama. Oleh karena itu, dalam bab ini, akan disajikan landasan teori yang relevan dengan penelitian yang akan dilakukan. Daftar penelitian terkait dapat ditemukan dalam tabel 2.1.

Tabel 2.1 Penelitian Terkait

No.	Nama dan Tahun	Judul	Hasil Penelitian
1.	(Wahid, 2021)	Identifikasi SMS Spam Menggunakan Metode Naive Bayes	Berdasarkan hasil penelitian: metode *Naive Bayes Classification* sangat tepat digunakan dalam pengklasifikasian pesan teks pada SMS inBox. Hasil filtrasi dengan menggunakan metode ini cukup akurat, dengan tingkat persentase kesalahan sebesar 1,33%. Hal ini menunjukkan bahwa metode ini dapat digunakan untuk menyaring spam pada pesan teks dengan tingkat akurasi yang tinggi. Namun, penelitian ini masih memiliki potensi untuk dikembangkan lebih lanjut, terutama dalam meningkatkan kualitas filtrasi pada pesan teks dan memperkaya data riset.

2.	(Setifani, 2020)	Perbandingan Algoritma *Naïve Bayes*, SVM, Dan *Decision Tree* Untuk Klasifikasi Sms Spam	Dari penelitian ini, diperoleh hasil sebagai berikut: 1. Algoritma *Naïve Bayes* memiliki nilai *precision* yang paling tinggi untuk klasifikasi SMS normal (1.00), sedangkan algoritma SVM memiliki nilai *precision* tertinggi untuk klasifikasi SMS *fraud* (0.90) dan SMS promo (0.93). 2. Algoritma *Naïve Bayes* memiliki nilai *precision* yang paling tinggi untuk klasifikasi SMS *fraud* (0.93) dan SMS promo (0.92), sedangkan algoritma SVM memiliki nilai *precision* tertinggi untuk klasifikasi SMS normal (0.99). 3. Algoritma *Naïve Bayes* dan SVM memiliki nilai f1-*score* yang lebih tinggi daripada algoritma *Decision Tree*. 4. Algoritma *Naïve Bayes* memiliki nilai akurasi tertinggi (0.94) dibandingkan dengan algoritma *Decision Tree* (0.87) dan SVM (0.93). Berdasarkan nilai *precision*, *precision*, f1-*score*, dan akurasi, dapat disimpulkan bahwa algoritma Naïve Bayes memiliki kemampuan terbaik dalam mengklasifikasikan SMS spam berbahasa Indonesia dibandingkan dengan algoritma SVM dan *Decision Tree*.
3.	(Reviantika, 2021)	Analisis Klasifikasi SMS Spam Menggunakan *Logistic Regression*	Penelitian ini bertujuan untuk membedakan atau mengklasifikasikan antara pesan SMS spam dan *non*-spam. Proses penelitian meliputi beberapa tahap, seperti *literature review*, pengumpulan data, *preprocessing*, pembobotan fitur menggunakan TF-IDF, analisis hasil, dan pengujian *dataset*. Tahap

			preprocessing meliputi beberapa teknik seperti *case folding*, tokenisasi, *filtering*, *normalization*, dan *stemming*. Metode *logistic regression* digunakan untuk klasifikasi, dan evaluasi hasil dilakukan menggunakan *confusion matrix*, akurasi, *precision*, *precision*, dan *F1-score*. Metode yang diusulkan mencapai akurasi sebesar 95%, yang lebih baik dibandingkan dengan metode *Naive Bayes Classifier*.
4.	(Abayomi-Alli, 2022)	*A deep learning method for automatic* SMS spam *classification: Performance of learning algorithms on indigenous dataset*	dibahas tentang klasifikasi SMS spam menggunakan beberapa algoritma *machine learning*. Penelitian ini menggunakan dua *dataset*, yaitu ExAIS_SMS *dataset* dan UCI *dataset*. Pada tahap *preprocessing*, dilakukan eliminasi bahasa asli dari database dan penghapusan *header*, *time stamp*, dan tanggal dari setiap pesan SMS. Metode yang diusulkan mencapai akurasi sebesar 95%.
5.	(Laksono, 2020)	Optimasi Nilai K pada Algoritma KNN untuk Klasifikasi Spam dan Ham Email	Penelitian ini bertujuan untuk mengetahui klasifikasi email spam dengan ham menggunakan metode KNN sebagai upaya mengurangi jumlah spam. Penelitian ini menggunakan *dataset* yang terdiri dari 27716 dokumen email berbahasa Inggris. Proses klasifikasi melibatkan pra-pemrosesan data (*case folding, data cleaning, stopword removal, stemming*, dan tokenisasi) sebelum memproses informasi melalui KNN. Hasil evaluasi klasifikasi menggunakan confusion matrix menunjukkan bahwa metode KNN dengan nilai K=1 memiliki akurasi tertinggi sebesar 91.4%.
6.	(Dwiyansaputra, 2021)	Deteksi SMS Spam Berbahasa Indonesia	Penelitian ini bertujuan untuk membangun model pembelajaran

		Menggunakan Tf-Idf dan *Stochastic Gradient descent Classifie*r (*Indonesian* SMS Spam *Detection Using* Tf-Idf a*nd Stochastic Gradient descent*)	mesin dengan akurasi yang lebih tinggi untuk mendeteksi spam SMS dalam bahasa Indonesia. Model yang dibangun berhasil mencapai akurasi sebesar 95% dalam mendeteksi SMS spam dan bukan spam. Studi ini mengatasi masalah spam SMS yang sering terjadi dan mengganggu, dengan menggunakan teknik klasifikasi pembelajaran mesin untuk menyaring spam SMS secara otomatis. Penelitian ini menunjukkan bahwa metode Stochastic *Gradient descent* Classifier, ketika digabungkan dengan metode pembobotan fitur TF-IDF, dapat meningkatkan performa dalam klasifikasi teks, khususnya dalam mendeteksi spam SMS berbahasa Indonesia.
7.	(Widyawati, 2019)	Perbandingan Algoritma *Naïve Bayes* Dan *Support Vector Machine* (Svm) Dalam Klasifikasi Sms Spam Berbahasa Indonesia	Penelitian ini bertujuan untuk menentukan metode klasifikasi yang lebih efektif dalam mengidentifikasi SMS spam. Penelitian ini menggunakan d*ataset* yang terdiri dari 1143 rekaman SMS, dengan 765 untuk pelatihan dan 378 untuk pengujian. Proses klasifikasi melibatkan pra-pemrosesan data (tokenisasi, penghapusan *stopword*, dan stemming) sebelum memproses informasi melalui *Naïve Bayes* dan SVM. Hasilnya menunjukkan bahwa *kinerja Naïve Bayes* melampaui SVM dalam hal *precision* (94%) dan presisi (95%).

Hubungan penelitian tabel 2.1 dengan penelitian ini yang menerapkan perbandingan fitur statistik dan fitur linguistik pada algoritma KNN dan NN dalam pengklasifikasian SMS spam. Berikut adalah Gapnya.

1. Metodologi: Penelitian ini membandingan fitur statistik dan linguistik, yang berbeda dari penelitian lain yang lebih fokus pada metode klasifikasi seperti *Naive Bayes*, SVM, *Decision Tree*, atau *logistic regression* dan menggunakan TF-IDF sebagai representasi dari sebuah pesan sebelum mengimplementasi ke algoritma.
2. Algoritma: penelitian ini menggunakan KNN dan *Neural Network*, sedangkan penelitian lain menggunakan KNN, *Naive Bayes*, SVM, *Decision Tree*, atau *logistic regression*. Dan belum ada yang menggunakan *Neural Network* (NN), hal ini memberikan perspektif baru dalam klasifikasi SMS spam.
3. Fitur Data: Penelitian Anda mengeksplorasi fitur statistik dan linguistik, khususnya untuk fitur linguistik yang memberikan wawasan lebih dalam tentang karakteristik SMS spam dibandingkan dengan pendekatan penelitian lain yang lebih fokus pada konten teks saja seperti fitur statistik yaitu tf-idf.
4. Evaluasi dan Performa: Penelitian ini memberikan hasil yang berbeda dalam hal akurasi, *precision*, dan *precision* dibandingkan dengan penelitian lain, karena berdasarkan *dataset* yang digunakan berbeda. Pada penelitian ini menggunakan data sebanyak 5000 data dan akan di gunakan fitur statistik dan linguistik dalam mengklasifikasikan SMS spam, tentunya akan menghasilkan akurasi, *precision*, *precision* dan f1-*score* yang berbeda.

Berdasarkan dari gap yang dijelaskan sebelumnya dapat dilihat perbedaannya antara penelitan saya dengan penelitian sebelumnya, baik dari segi metode, algoritma, fitur data hingga evaluasi dan performas.

2.2. Klasifikasi

Klasifikasi merupakan suatu teknik untuk menilai objek data serta mengelompokkan objek berdasarkan atribut – atribut atau ciri objek ke dalam salah satu kategori yang telah

didefinisikan. Klasifikasi melakukan pembelajaran model berdasarkan data latih yang telah di berikan label atau kelas target (Herwanto, 2021).

2.3. *Short Massage System* (SMS)

Short Message Service (SMS) atau layanan pesan singkat tetap menjadi salah satu sarana komunikasi jarak jauh yang digunakan secara luas di era saat ini untuk mengirim pesan singkat. Namun, seiring dengan evolusi layanan pesan singkat ini, muncul dampak negatif yang mengancam perangkat seluler, yaitu serangan berupa SMS spam (Dwiyansaputra, 2021).

2.4. SMS Spam

SMS spam merujuk pada pesan spam yang disebarkan melalui SMS. Isi pesan ini biasanya berupa penawaran atau promosi yang tidak diharapkan oleh penerima. SMS spam dapat mengganggu dan mengacaukan aktivitas pengguna ponsel. Oleh karena itu, penting untuk mendeteksi SMS spam dengan tujuan melindungi pengguna dari pesan yang tidak diinginkan (Dwiyansaputra, 2021).

2.5. *K-Nearest Neighbors* (KNN)

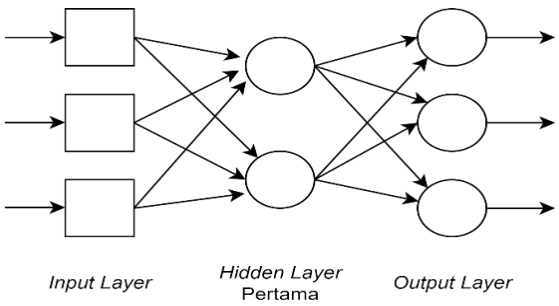
*K-Nearest Neighbo*r (KNN) merupakan sebuah metode klasifikasi data yang mengoperasikan dengan prinsip dasar penentuan label atau kelas data berdasarkan kedekatan jarak antara data yang akan diklasifikasikan dengan data-data lain dalam *dataset*. Dalam penggunaan algoritma *K-Nearest Neighbors*, penentuan jumlah tetangga terdekat (K) adalah langkah kritis yang perlu dipilih. Nilai K yang paling sering digunakan adalah bilangan ganjil, seperti K = 1, 3, 5, dan seterusnya. Penentuan nilai K dalam metode *K-Nearest Neighbors* (KNN) mempertimbangkan faktor jumlah data yang tersedia dan dimensi data. Ketika *dataset* memiliki banyak data, sebaiknya nilai K yang dipilih relatif rendah. Namun,

ketika dimensi data (jumlah fitur) meningkat, pemilihan nilai K yang lebih tinggi menjadi lebih disarankan (Laksono, 2020).

2.6. *Neural Network*

Neural Network adalah sistem yang digunakan untuk memproses informasi dan dianalogikan sebagai suatu bentuk kotak yang menerima masukan dan menghasilkan keluaran. Namun, prinsip dasarnya tetap sama: jaringan saraf buatan belajar dengan mengubah koneksi antara neuron-neuronnya. Jaringan semacam ini memiliki kemampuan untuk menjalankan berbagai tugas pemrosesan informasi yang beragam (Ramadhan, 2023).

Tentunya di dalam *Neural Network* terdapat beberapa jenis yang diantaranya terdapat *feed-forward Neural Network*, dan pada *feed-forward Neural Network* sendiri terdapat salah satu varian yaitu MLP (*Multilayer Perceptron*).. Untuk lebih jelas arsitekturnya dapat dilihat pada gambar 2.1 berikut.



Gambar 2.1 Arsitektur MLP

Berdasarkan gambar 2.1 yaitu arsitektur dari MLP (*Multilayer Perceptron*) atau sering disingkat sebagai MLP adalah salah satu varian dari *Neural Network feed-forward* yang memiliki satu atau lebih lapisan tersembunyi (*hidden layer*). Secara umum, MLP terdiri dari lapisan *input* yang berfungsi sebagai tempat untuk memasukkan data, setidaknya satu

lapisan tersembunyi yang berisi neuron komputasi, dan lapisan keluaran yang berfungsi sebagai tempat penyimpanan hasil perhitungan (Wibawa, 2018).

Gambar 2.1 menjelaskan terkait bagaimana alur arsitektur MLP melakukan proses didalamnya. Dalam jaringan saraf tiruan (MLP), terdapat aspek utama yang memiliki peran penting, yaitu fungsi aktivasi. Fungsi aktivasi digunakan untuk menghitung *output* pada suatu node berdasarkan berbagai *input* yang diterimanya.

Pada penelitian ini fungsi aktivasi yang digunakan yaitu *Rectified Linear Unit* (ReLU). Menurut (Wibawa, 2018) Fungsi aktivasi *Rectified Linear Unit* (ReLU) adalah jenis fungsi aktivasi yang memiliki perhitungan yang cukup sederhana. Dalam proses penerapannya, baik dalam tahap penerusan (*forward*) maupun tahap mundur (*backward*), ReLU menggunakan kondisi if sederhana. Salah satu keunggulan dari ReLU adalah bahwa tidak ada operasi eksponensial, perkalian, atau pembagian yang terlibat dalam perhitungannya.

Dalam ReLU, jika elemen *input* bernilai negatif, maka nilainya diubah menjadi 0. Pendekatan perhitungan yang sederhana seperti ini memiliki keuntungan dalam hal efisiensi komputasi. Karena tidak ada operasi matematis yang rumit, ReLU memerlukan waktu komputasi yang relatif sedikit saat digunakan dalam proses pelatihan (training) dan pengujian (*testing*) model *Neural Network*. Hal ini membuatnya menjadi pilihan yang populer dalam implementasi model jaringan saraf (*Neural Network*) modern.

2.7. Fitur Statistik

Fitur statistik adalah aspek penting dalam analisis teks yang digunakan untuk mengukur dan menggambarkan data teks secara numerik. Dalam konteks ini, salah satu metrik statistik yang sangat umum digunakan yaitu TF-IDF (AL-Jumaili, 2020).

TF-IDF adalah sebuah metrik statistik yang digunakan untuk mengevaluasi relevansi

kata-kata dalam sebuah dokumen. Metrik ini mengukur seberapa penting suatu kata dalam dokumen dengan cara memberikan bobot yang lebih tinggi pada kata-kata yang muncul lebih sering dalam dokumen tersebut. Dengan demikian, semakin sering kata tersebut muncul dalam dokumen, semakin tinggi bobotnya. Metrik ini berguna dalam analisis teks untuk menentukan kata-kata kunci yang paling relevan dalam sebuah dokumen (Indriyanto, 2019).

2.8. Fitur Linguistik

Fitur linguistik mengacu pada bagian dari penelitian yang berkaitan dengan fitur-fitur yang digunakan dalam analisis teks. Fitur-fitur ini adalah elemen-elemen penting dalam pemrosesan bahasa alami yang membantu dalam pemahaman dan analisis teks. Berikut adalah bagaimana fitur linguistik dapat digunakan dalam konteks penelitian ini untuk mengidentifikasi kelas (AL-Jumaili, 2020).

2.8.1. Fitur *Sentence Length*

Kalimat - kalimat yang terlalu pendek atau terlalu Panjang dapat mempengaruhi suatu pola pada pesan. Fitur **"***Sentence Length*" dihitung berdasarkan jumlah kata dalam kalimat sebelum dilakukan *preprocessing* (Indriyanto, 2019).

2.8.2. Fitur *Tittle Word*

Nilai fitur ini diperoleh dengan menghitung berapa banyak kata dengan penulisan seperti judul yang mana setiap kata diawali dengan huruf besar dalam sebuah kalimat (Indriyanto, 2019).

2.8.3. Fitur *Cue-Phrase*

Cue-phrase merupakan kumpulan kata yang memberi tanda bahwa sebuah kalimat memiliki kepentingan khusus. Sebagai ilustrasi, *cue-phrase* bisa berupa ungkapan seperti

"kesimpulannya", "oleh karena itu", atau "jadi" (Indriyanto, 2019). Terdapat dua fitur *cue-phrase* yang digunakan pada penelitian ini yaitu, promo dan penipuan.

2.9. *Preprocessing* Teks

Preprocessing teks adalah langkah pertama dalam mengolah data *input* sebelum melewati tahap klasifikasi (Firmansyah, 2020). Berikut adalah penjelasan mengenai serangkaian langkah yang tercakup dalam *Preprocessing* teks.

1. *Case Folding*

Case folding adalah langkah dalam *preprocessing* teks yang bertujuan untuk mengnormalisasi teks. Ini mencakup konversi semua huruf dalam kata-kata menjadi huruf kecil, menghapus angka dan tanda baca, menghilangkan spasi kosong, serta mengatasi kata-kata singkatan (Herwanto, 2021).

2. *Word Normalize*

Word normalize yaitu tahapan yang dilakukan untuk mengganti kata singkatan pada text menjadi kata asli setelah semua huruf dijadikan *lowercase* (Reviantika, 2021).

3. *Stopwords Removal*

Stopword removal adalah langkah dalam proses pemfilteran yang bertujuan untuk menghapus kata-kata yang memiliki peran yang tidak signifikan dalam suatu teks. Setiap kata atau frasa yang dihasilkan akan dibandingkan dengan daftar kata *stopword* (Herwanto, 2021).

4. *Stemming*

Stemming bertujuan untuk menemukan kata dasar (*root word*) dengan menghilangkan semua jenis imbuhan, termasuk imbuhan awalan (*prefix*), imbuhan akhiran (*suffix*), dan bahkan kombinasi imbuhan awalan dan akhiran (*confixes*) dari sebuah kata. Ini

dilakukan untuk menyederhanakan kata-kata dalam teks sehingga kata-kata yang memiliki bentuk yang berbeda tetapi merujuk pada konsep yang sama dapat diidentifikasi dengan cara yang lebih efisien (Herwanto, 2021).

2.10. *Classification Report*

Evaluasi kinerja model penelitian ini disajikan dalam bentuk laporan klasifikasi yang melibatkan penilaian akurasi, presisi, *precision*, serta *F1-score* untuk setiap kelas dan metrik evaluasi secara keseluruhan. Data ini membantu kami untuk mengevaluasi sejauh mana model *K-Nearest Neighbors* (KNN) yang telah dioptimasi mampu menghasilkan hasil klasifikasi yang memuaskan pada *dataset* yang digunakan (Ramadhan, 2023).

Nilai akurasi, presisi, *precision*, serta *F1-score* dikatakan cukup baik apabila berada pada rentang 50,01% - 75.00%. dan untuk rentang 75.01% - 100% dikatakan sangat baik (Wijaya, 2022).

Pada penelitian ini menggunakan *multiclass* karena klasifikasi yang dilakukan menggunakan 3 kelas yaitu kelas spam promo, penipuan dan ham (normal). Berikut penjelasan terkait akurasi, *precision*, *precision*, serta F1-*score*. *True negative* pada *multiclass* tidak digunakan karena data yang diprediksi melebihi 2 kelas.

Keterangan:

TP = *True Positive*

FP = *False Positive*

FN = *False Negative*

1. *Precision*

Precision adalah ukuran yang mengindikasikan sejauh mana suatu sistem atau model dapat memberikan informasi yang relevan untuk kebutuhan pengguna. *Precision*

menggambarkan tingkat ketepatan sistem dalam memberikan jawaban yang sesuai dengan permintaan pengguna. Semakin tinggi nilai *precision*, semakin baik kualitas metode yang digunakan oleh sistem (Setifani, 2020). *Precision* dihitung menggunakan rumus 2.1 berikut:

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad \text{.....(2.1)}$$

2. *Precision*

Precision adalah ukuran proporsi dokumen yang berhasil ditemukan oleh sebuah proses pencarian dalam sistem Informasi Relevansi (IR). Nilai *precision* mencerminkan sejauh mana sistem berhasil mengidentifikasi dan mengembalikan informasi atau dokumen yang relevan. Semakin tinggi nilai *precision*, semakin baik kemampuan sistem dalam menemukan kembali dokumen yang relevan. *Precision* dihitung dengan rumus 2.2 berikut (Setifani, 2020):

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad \text{.....(2.2)}$$

3. F1-*score*

F1-*score* adalah metrik evaluasi yang mewakili harmonic mean dari *precision* dan *precision*. Lebih sederhananya, F1-*score* adalah rata-rata yang dibobotkan dari *precision* dan *precision.* Rentang nilai F1-*score* dari 0 hingga 1 (Setifani, 2020). Berikut adalah persamaan rumus 2.3 berikut:

$$F1 - score = \frac{2\times(Precision\times Recall)}{(Precision+Recall)} \times 100\% \quad \text{.....(2.3)}$$

4. Akurasi

Akurasi merupakan ukuran yang menggambarkan sejauh mana hasil prediksi benar pada seluruh *dataset*. Akurasi memberikan pemahaman tentang sejauh mana prediksi benar dan prediksi salah dalam konteks klasifikasi (Setifani, 2020). Rumus 2.4 akurasi digunakan untuk membandingkan hasil prediksi yang tepat dengan keseluruhan data sebagai berikut:

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Data} \times 100\% \text{.....(2.4)}$$

2.11. *Python*

Python adalah bahasa pemrograman yang sangat beragam. Dengan menggunakan alat dan perpustakaan yang tepat, seseorang dapat menjadi inovator sejati. Proses belajar bahasa pemrograman membutuhkan nyali, kemauan, waktu, dan mungkin sejumlah minuman berenergi. Oleh karena itu, sangat penting untuk menetapkan tujuan dan memahami berbagai kegunaan *Python* sebelum memulai pembelajaran. Bahasa ini dirancang untuk menjadi fleksibel dan dapat digunakan dalam berbagai konteks pemrograman. Kejelasan dan logika kode *python* membuatnya cocok untuk proyek skala kecil maupun besar. Kemampuan bahasa ini untuk menerapkan berbagai prosedur komputer membuatnya menjadi alat yang sangat kuat dalam menghasilkan teknologi yang mengejutkan (Runimeirati, 2023).

2.12. *Library* Yang Digunakan

Dalam pemodelan, beberapa *library* yang esensial diperlukan untuk memfasilitasi berbagai aspek analisis data dan pemrosesan teks. Berikut adalah beberapa *library* yang digunakan. Untuk lebih jelas dapat dilihat pada tabel 2.2 berikut.

Tabel 2.2 *Library* yang Digunakan

No.	*Library*	Keterangan
1.	NLTK (*Natural Language Toolkit*)	digunakan untuk pemrosesan bahasa alami (NLP) pada teks, menyediakan algoritma seperti stemming, dan penghapusan *stopword*.
2.	Pandas	Berguna untuk manipulasi dan analisis data, memudahkan pembersihan data serta manipulasi struktur data seperti DataFrame.
3.	Sastrawi	Digunakan untuk melakukan proses stemming pada teks berbahasa Indonesia, membantu mengubah kata ke bentuk dasarnya.

4.	*Time*	Diperlukan untuk mengukur waktu eksekusi program, berguna untuk evaluasi kinerja program secara efisien.
5.	Re (*Regular Expression*)	Memungkinkan untuk melakukan operasi regular expression pada teks, mempermudah pencarian dan manipulasi pola teks.
6.	*String*	Digunakan untuk operasi pada string, seperti penghapusan karakter tertentu dari string.
7.	NumPy	Berperan dalam operasi matematika dan perhitungan saintifik pada array, matriks, dan struktur data lainnya. Berperan dalam operasi matematika dan perhitungan saintifik pada array, matriks, dan struktur data lainnya.
9.	*Scikit-learn (Sklearn)*	Menyediakan fungsi untuk *preprocessing* dan modeling pada data, termasuk algoritma klasifikasi dan pengolahan fitur.
10.	*Pickle*	Berguna untuk melakukan serialisasi dan deserialisasi objek *Python*, membantu dalam menyimpan dan memulihkan objek dengan mudah.
11.	*Streamlit*	Digunakan untuk membuat aplikasi web dengan mudah dan cepat. Dengan Streamlit, dapat membuat antarmuka pengguna (UI) untuk aplikasi data atau visualisasi data tanpa perlu pengetahuan mendalam tentang pengembangan web.

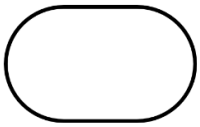
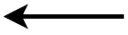
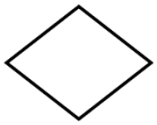
Dengan kombinasi *library* tersebut, pemodelan dan analisis data dapat dilakukan secara efektif dalam konteks pemrosesan teks dan bahasa alami.

2.13. *Flowchart*

Flowchart atau sering disebut juga bagan alir, berbentuk diagram yang menggambarkan alur dari berbagai algoritma dalam suatu program, sehingga memperlihatkan bagaimana program tersebut bergerak dan beralih dari satu langkah ke langkah berikutnya (Solikin, 2018). Dalam diagram alir (*flowchart*) pada tabel 2.3, terdapat beberapa simbol yang dapat dikenali adalah sebagai berikut (Said, 2020).

Tabel 2.3 Simbol *Flowchart*

No.	Simbol	Keterangan	Fungsi

1.		*Terminal*	Untuk memulai dan mengakhiri proses
2.		*Input*/*Output*	Untuk memasukkan dari luar atau hasil dari suatu proses
3.		Proses	Simbol yang menunjukkan setiap pengolahan yang dilakukan
4.		*Flowline*	Menunjukan alur proses yang dilakukan
5.		*Decision*	Menentukan keputusan dari suatu pertanyaan

Pada tabel 2.3 diatas merupakann simbol-simbol yang biasa digunakan dalam pembuatan diagram alir, khususnya pada penelitian ini yang digunakan 5 simbol, yakni *terminal*, *input/output*, proses, *flowline* dan *decision*.

2.14. *Black Box Testing*

Pengujian *black Box* adalah metode pengujian yang berfokus pada fungsi sistem, termasuk masukan dan keluaran dari sistem, tanpa memerlukan pengetahuan tentang bagaimana sistem tersebut bekerja secara internal (Rumlaklak, 2022).

BAB III

METODE PENELITIAN

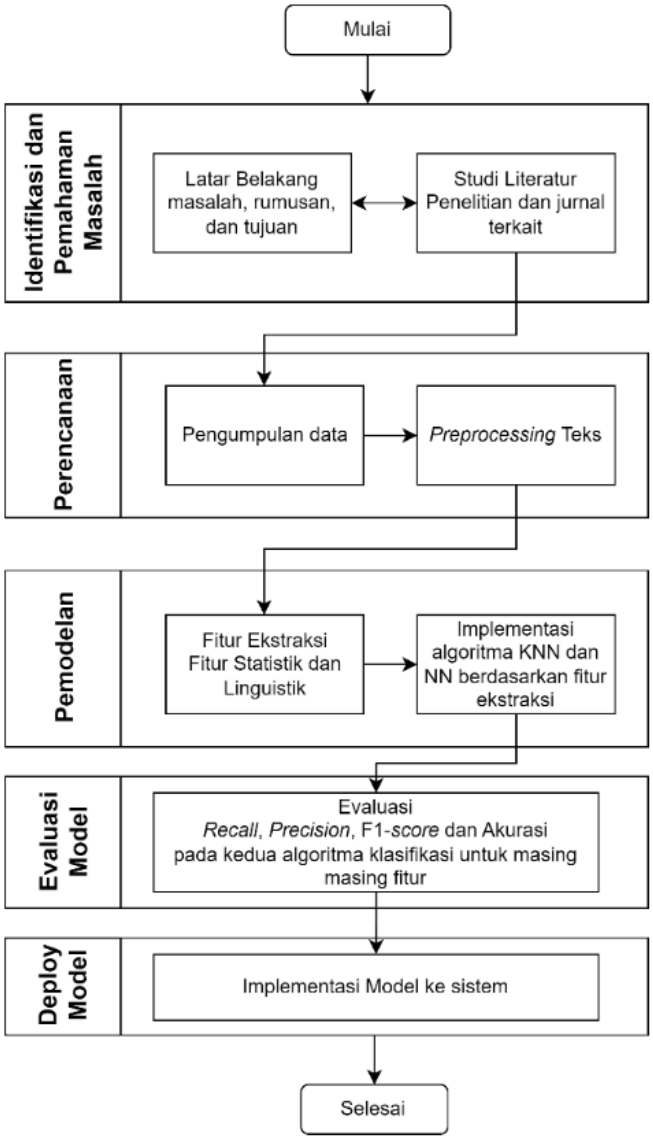
3.1. Jenis, Sifat dan Pendekatan Penelitian

Penelitian yang akan dilakukan menerapkan konsep penelitian kuantitatif, yang merupakan salah satu metode penelitian yang banyak digunakan dalam ilmu pengetahuan dan penelitian sosial. Dalam penelitian kuantitatif, dikumpulkan data berupa teks yang kemudian di konversi ke bentuk numerik dengan fitur statistik dan fitur linguistik kemudian di evaluasi data tersebut menggunakan algoritma KNN dan NN. Tujuan dari penelitian ini adalah untuk mendapatkan pemahaman yang lebih dalam tentang fenomena yang sedang diteliti.

Penelitian ini berupa eksperimen, yang berarti dilakukan serangkaian eksperimen untuk mendapatkan hasil akurasi yang optimal. Fokus dari eksperimen ini adalah membandingkan fitur statistik dan fitur linguistik pada kedua algoritma klasifikasi teks. Eksperimen ini dilakukan pada *dataset* yang memiliki jumlah 4000 data.

Pendekatan yang digunakan dalam penelitian ini adalah pendekatan kuantitatif. Ini berarti bahwa akan mengikuti langkah-langkah atau alur penelitian yang telah direncanakan sebelumnya. Pendekatan ini memungkinkan untuk memastikan bahwa penelitian dilakukan secara sistematis dan dengan metode yang teruji. Selain itu, pendekatan kuantitatif juga memungkinkan untuk mengukur dan menganalisis data, sehingga mendapatkan hasil yang dapat diandalkan dan akurat.

Adapun berdasarkan dari penjelasan diatas dapat dipresentasikan metode penelitiannya yang terdiri dari identifikasi dan permasalahan masalah, perencanaan, pemodelan, evaluasi model hingga *deploy* model sebagai berikut pada gambar 3.1.



Gambar 3.1 Metode Penelitian

1. Identifikasi dan Pemahaman Masalah

Pada identifikasi dan pemahaman masalah, memahami masalah yang terjadi dalam proses identifikasi masalah, digunakan berbagai metode dengan tujuan untuk mengarahkan analisis yang lebih mendalam guna mengidentifikasi permasalahan. Setiap metode yang diterapkan menghasilkan informasi yang berkontribusi pada rangkuman yang akan digunakan dalam upaya penyelesaian masalah. Salah satu metode yang digunakan dalam

proses identifikasi masalah adalah studi literatur. Metode ini melibatkan pengumpulan data dengan cara membaca dan menelusuri berbagai referensi seperti jurnal untuk memperoleh teori dan informasi yang relevan dengan permasalahan penelitian. Tahapan ini menjadi landasan penting dalam menentukan pendekatan pemecahan masalah yang tepat.

2. Perencanaan

Pada tahapan ini dilakukan pengumpulan data dan pengolahan data yang diperoleh atau *preprocessing* teks. Data diperoleh dengan cara mencari relawan yang bersedia memberikan data mereka. Tentunya peneliti tidak akan melanggar etika, tidak menyebarluaskan data yang diperoleh dan data yang diperoleh tidak salah gunakan dikhususkan pada penelitian saja, selain itu data juga diambil dari *dataset* publik 2018 yang disediakan oleh peneliti bernama Yudi Wibisino.

Setelah memperoleh data, dilakukan *preprocessing* teks untuk mengolah data mentah yang diperoleh menjadi *dataset* yang siap digunakan oleh program.

3. Pemodelan

Algoritma yang digunakan untuk membangun model adalah *K-Nearest Neighbors* (KNN) *Classifier* dan *Neural Network* (*MLP Classifier*). Kedua algoritma ini sering digunakan untuk pengklasifikasi text. Sebelum dilakukan diimplementasikan ke model yang dibuat maka diperlukan fitur statistik dan fitur linguistik setelah dilakukan pemrosesan *dataset* sebelumnya.

4. Evaluasi Model

Model yang dibuat akan dilakukan evalusi yakni dengan membandingkan *precision, precision,* F1-*Score* dan Akurasi untuk melihat tingkat keakuratan kedua model yang telah dibuat, diantara kedua model tersebut model mana yang lebih tinggi akurasinya dalam

melakukan pengklasifikasian SMS spam, setelah dilakukan komparasi kedua model tersebut selanjutnya dapat dievaluasi pada sistem sederhana apakah model yang dibuat dapat diterapkan atau tidak pada sistem.

5. *Deploy* Model

Setelah melakukan evaluasi terhadap perbandingan fitur statistik dan linguistik pada algoritma KNN dan NN, dipilih salah satu model terbaik dan diimplementasikan pada sebuah sistem sederhana dengan tambahan *Library.*

3.2. Metode Pengumpulan Data

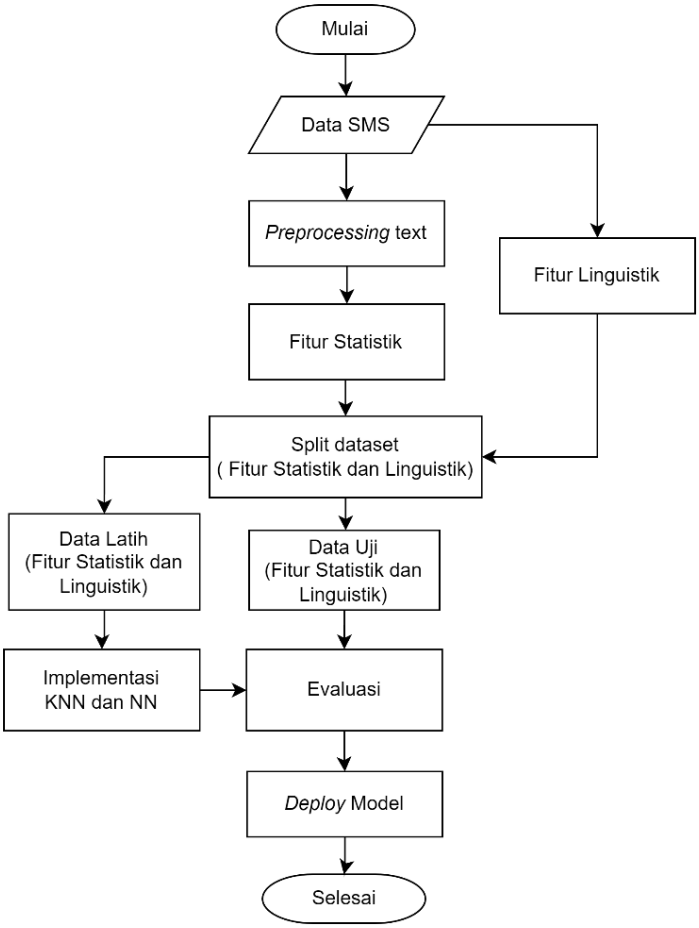
Pada tahapan ini dilakukan pengumpulan data. Data yang diperoleh peneliti dari *dataset* publik 2018 yang disediakan oleh peneliti bernama Yudi Wibisino.

3.3. Metode Analisis Data

Dataset yang diperoleh berdasarkan *dataset* publik yang mana *dataset* tersebut sudah dilakukan pelabelan, selain itu data yang diperoleh akan diambil dan digunakan untuk melatih model menggunakan fitur statistik ataupun fitur linguistik pada masing masing algoritma untuk melihat model mana yang lebih baik.

3.4. Alur Penelitian

Dalam penelitian ini *output* yang akan dihasilkan berupa 4 model diantaranya fitur statistik dengan KNN, fitur linguistik dengan KNN, fitur statistik dengan NN dan fitur lingusitik dengan NN. Dan diantara ke empat model tersebut akan diambil model yang terbaik berdasarkan hasil classification reportnya, baik itu akurasi, *precision*, *precision* dan F1-*score*. Dan kemudian di implementasikan ke dalam sebuah sistem menggunakan *library* dari *python* yaitu *streamlit*, untuk lebih jelasnya dapat dilihat tahapan pada gambar 3.2.



Gambar 3.2 Alur Penelitian

Berdasarkan penjelasan pada gambar 3.2 maka dapat dijabarkan sebagai berikut.

3.4.1. Data SMS

Data SMS adalah data yang dikumpulkan berdasarkan metode pengumpulan data yang dilakukan dan sudah dilabeli, data inilah yang akan dijadikan sebagai *dataset*. Meskipun begitu *dataset* ini masih bersifat mentah dan akan diolah Kembali pada tahapan selanjutnya yaitu pada tahapan *preprocessing* teks. Akan tetapi, sebelum di terapkan pada tahapan *preprocessing* teks dilakukan penerapan fitur linguistik terlebih dahulu guna untuk mengekstrak fitur fitur yang ada pada fitur linguistik. Karena nanti jika diolah setelah pada tahapan *preprocessing* teks, maka untuk memahami kata tersebut akan sulit karena sudah

menjadi *root word.* Berikut adalah contoh data yang akan diolah dapat dilihat pada tabel 3.1 berikut:

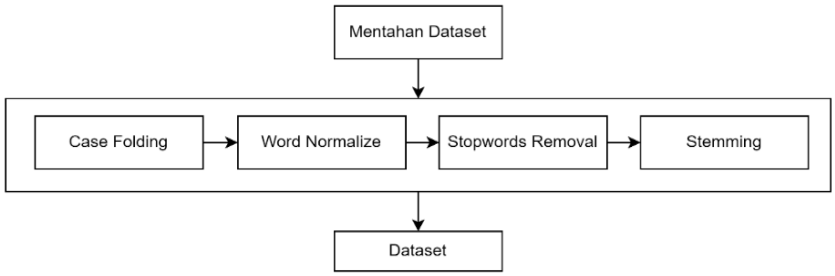
Tabel 3.1 Contoh Data yang Digunakan.

Kalimat	Label
Saya sudah transfer uangnya ke rekeningmu, jangan lupa cek saja nanti.	Normal
Dapatkan diskon 30% untuk semua pakaian di toko kami hingga akhir minggu ini!	Promo
Saya baru saja memenangkan lotre dan ingin berbagi keberuntungan dengan Anda, cukup kirimkan data pribadi Anda untuk klaim hadiah di 082212322243	Penipuan

Data tersebut akan diolah pada proses selanjutnya.

3.4.2. *Preprocessing* Teks

Pada tahapan ini diperlukan untuk mengurangi teks yang tidak diperlukan atau mengembalikan kata menjadi *root word* guna untuk menghindari kesamaan kata sebelum diproses atau diekstraksi dengan TF-IDF. Berikut adalah tahapan dalam melakukan *preprocessing* teks dapat dilihat pada gambar 3.3.



Gambar 3.3 Tahapan Proses *Preprocessing* Teks.

Berdasarkan gambar 3.3 untuk penjelasan lebih jelas dapat dilihat sebagai berikut:

1. *Case Folding*

Pada tahap ini, semua karakter dalam *dataset* diubah menjadi huruf kecil (*lowercase*), dan karakter-karakter lain dihapus atau dianggap sebagai pemisah. Misalnya, kata "Aku"

dan "aku" akan dianggap identik setelah melewati proses case folding karena jika tidak maka kedua kata tadi akan dianggap sama. Sebagai contoh berikut dapat dilihat pada gambar 3.4.

```
Raw Content     :  Aku senin udah ke tempat kerja. Minggu2 depan aku gaktau bisa/ngga :(
Case Folding    : aku senin udah ke tempat kerja minggu depan aku gaktau bisangga
```

Gambar 3.4 Contoh Data Sebelum dan Sesudah Dilakukan *Case Folding.*

Dapat dilihat pada gambar 3.4 kata “Aku” menjadi “aku” begitupun dengan kata lainnya.

2. *Word Normalize*

Word normalize yaitu tahapan yang dilakukan untuk mengganti kata singkatan pada text menjadi kata asli setelah semua huruf dijadikan *lowercase*, seperti contoh berikut pada gambar 3.5.

```
Raw Data        : Berarti saya ke palasari nya skrg .. abis itu langsung ke gik
Word Normalize  : berarti saya ke palasari nya sekarang .. habis itu langsung ke gik
```

Gambar 3.5 Contoh Data Sebelum dan Sesudah Dilakukan *Word Normalize.*

Dapat dilihat pada gambar 3.5 kata “skrg” menjadi “sekarang” begitupun dengan kata lainnya.

3. *Stopwords Removal*

Penghapusan kata yang mengandung kata-kata yang tidak memiliki makna penting dalam konteks analisis teks. Sebagai contoh berikut dapat dilihat pada gambar 3.6.

```
Raw Data              : Berarti saya ke palasari nya skrg .. abis itu langsung ke gik
Stopword Removal      : Berarti palasari nya skrg .. abis langsung gik
```

Gambar 3.6 Contoh Data Sebelum dan Sesudah Dilakukan *Stopword Removal.*

Dapat dilihat pada gambar 3.6 kata tidak memiliki makna penting seperti “saya ke” di hapus begitupun dengan kata lainnya.

4. *Stemming*

Menemukan kata dasar (*root word*) dengan menghilangkan semua jenis imbuhan, termasuk imbuhan awalan (*prefix*), imbuhan akhiran (*suffix*), dan bahkan kombinasi imbuhan awalan dan akhiran (*confixes*) dari sebuah kata. Ini dilakukan untuk menyederhanakan kata-kata dalam teks sehingga kata-kata yang memiliki bentuk yang berbeda tetapi merujuk pada konsep yang sama dapat diidentifikasi. Berikut contoh penggunaan *Stemming* dapat dilihat pada gambar 3.7:

```
Raw Data                    : Maaf saya tidak akan melakukan hal yang seperti itu lagi
Stemming                    : maaf saya tidak akan laku hal yang seperti itu lagi
```

Gambar 3.7 Contoh Data Sebelum dan Sesudah Dilakukan *Stemming.*

Dapat dilihat pada gambar 3.7 kata kata akan dikembalikan menjadi kata dasar seperti kata“melakukan” menjadi “laku” begitupun dengan kata lainnya.

3.4.3. Fitur Linguistik

Pada tahapan ini sebelum melakukan *Preprocessing* teks, diterapkan fitur linguistik. Terdapat 4 fitur linguistik yang digunakan yaitu:

1. *Sentence Length*

Menghitung banyak kata didalam sebuah kalimat. Untuk lebih jelas dapat dilihat pada tabel 3.2 berikut:

Tabel 3.2 Fitur *Sentence Length.*

Kalimat	***Sentence Length***
Saya sudah transfer uangnya ke rekeningmu, jangan lupa cek saja nanti.	11
Dapatkan diskon 30% untuk semua pakaian di toko kami hingga akhir minggu ini!	13

Saya baru saja memenangkan lotre dan ingin berbagi keberuntungan dengan Anda, cukup kirimkan data pribadi Anda untuk klaim hadiah di 082212322243	21

Dalam sebuah kalimat, kata yang dihitung yaitu teks, angka atau nomor yang terpisah oleh spasi dianggap sebagai satu kata.

2. Tittle word

Menghitung kata yang memiliki format penulisan seperti judul yang berawalan dengan huruf besar dalam sebuah kalimat. Untuk lebih jelas dapat dilihat pada tabel 3.3.

Tabel 3.3 Fitur *Tittle Word.*

Kalimat	***Tittle word***
Saya sudah transfer uangnya ke rekeningmu, Jangan lupa cek saja nanti.	2
Dapatkan diskon 30% untuk semua pakaian di toko kami hingga akhir minggu ini!	1
Saya baru saja memenangkan lotre dan ingin berbagi keberuntungan dengan Anda, cukup kirimkan data pribadi Anda untuk klaim hadiah di 082212322243	3

Untuk kalimat yang pertama dapat dilihat pada kata “Saya” yang memiliki kata yang berawalan dengan huruf besar. Begitupun pada dua kalimat lainnya yang memiliki kata yang berawal dengan huruf besar.

3. *Cue-phrase* promo

Pada fitur ini akan dicocokan dengan kata kunci promo yang berasal dari sebuah pakar, sebagai contoh kata “diskon” misalkan. Untuk lebih jelas dapat dilihat pada tabel 3.4 sebagai berikut:

Tabel 3.4 Fitur *Cue-Phrase* Promo.

Kalimat	***Cue-phrase promo***
Saya sudah transfer uangnya ke rekeningmu, Jangan lupa cek saja nanti.	0
Dapatkan diskon 30% untuk semua pakaian di toko kami hingga akhir minggu ini!	1

Saya baru saja memenangkan lotre dan ingin berbagi keberuntungan dengan Anda, cukup kirimkan data pribadi Anda untuk klaim hadiah di 082212322243	0

Pada kalimat kedua, karena terdapat kata kunci “diskon” di dalam kalimat tersebut sebanyak satu maka dihitung satu. Dan untuk kalimat kesatu dan ketiga karena tidak ada maka dianggap nol.

4. *Cue-phrase* penipuan

Pada fitur ini akan dicocokan dengan kata kunci promo yang berasal dari sebuah pakar, sebagai contoh kata “transfer” dan “hadiah” misalkan. Untuk lebih jelas dapat dilihat pada tabel 3.5 sebagai berikut:

Tabel 3.5 Fitur *Cue-Phrase* Penipuan.

Kalimat	***Cue-phrase* penipuan**
Saya sudah transfer uangnya ke rekeningmu, Jangan lupa cek saja nanti.	1
Dapatkan diskon 30% untuk semua pakaian di toko kami hingga akhir minggu ini!	0
Saya baru saja memenangkan lotre dan ingin berbagi keberuntungan dengan Anda, cukup kirimkan data pribadi Anda untuk klaim hadiah di 082212322243	1

Pada kalimat kesatu dan ketiga, karena terdapat kata kunci “transfer” dan “hadiah” di dalam kalimat tersebut sebanyak satu pada masing-masing kalimat maka dihitung satu. Dan untuk kalimat kedua karena tidak ada maka dianggap nol.

Dari fitur-fitur linguistik yang ada, nanti dikumpulkan dan menjadi data seperti berikut pada tabel 3.6.

Tabel 3.6 Fitur Linguistik.

Kalimat	***Sentence Length***	***Tittle word***	***Cue-phrase* penipuan**	***Cue-phrase* penipuan**
Saya sudah transfer uangnya ke rekeningmu,	11	2	1	1

Jangan lupa cek saja nanti.				
Dapatkan diskon 30% untuk semua pakaian di toko kami hingga akhir minggu ini!	13	1	0	0
Saya baru saja memenangkan lotre dan ingin berbagi keberuntungan dengan Anda, cukup kirimkan data pribadi Anda untuk klaim hadiah di 082212322243	21	3	1	1

Pada tabel 3.6 fitur-fitur inilah yang akan di implementasikan ke dalam algoritma KNN maupun algoritma NN. Dan selanjutnya dilanjutkan dengan tahapan *preprocessing* teks.

3.4.4. Fitur Statistik

Pada tahapan sebelumnya data yang telah diolah pada proses *preprocessing* teks akan menghasilkan sebuah *dataset* yang baru yang akan dilakukan proses fitur statistik pada tahapan ini.

TF-IDF digunakan untuk menghitung bobot kata-kata dalam teks yang digunakan sebagai fitur dalam model pengklasifikasian teks. Dengan kata lain, ini membantu model dalam memahami dokumen-dokumen teks dengan memberikan representasi numerik yang berasal dari nilai TF-IDF. Berikut adalah contoh Langkah-langkah penerapan TF-IDF untuk mencari bobot dari setiap kata pada dokumen korpus sebagai fitur statistik:

Berikut adalah contoh dokumen yang akan direpresentasikan, dapat dilihat pada tabel 3.7:

Tabel 3.7 Contoh Dokumen.

D ke-i	Kalimat	Label
D1	Saya Suka Belajar Bahasa	Suka
D2	Banyak Bahasa Yang Saya Suka	Suka

D3	Hari Ini Saya Belajar Matematika	Tidak Suka

Selanjutnya adalah Langkah-langkah TF-IDF:

1. Jadikan *vocabulary*

Berdasarkan dokumen yang ada pada tabel 3.1 dipisahkan dan diambil kata-kata tersebut dibuat menjadi satu *vocabulary,* yang berisikan kata-kata pada dokumen yang unik (tidak ada kata yang duplikat). Untuk lebih jelas akan terlihat seperti ini; Bahasa, Banyak, Belajar, Hari, Ini, Matematika, Saya, Suka, Yang.

2. Selanjutnya, menghitung frekuensi kata pada dokumen

Untuk lebih jelas dapat dilihat pada tabel 3.8:

Tabel 3.8 Menghitung Frekuensi Kata yang Ada Pada Korpus di Setiap Dokumen.

W_i / D_i	**Bahasa**	**Banyak**	**Belajar**	**Hari**	**Ini**	**Matematika**	**Saya**	**Suka**	**Yang**
D1	1	0	1	0	0	0	1	1	0
D2	1	1	0	0	0	0	1	1	1
D3	0	0	1	1	1	1	1	0	0

3. Setelah itu implementasikan rumus berikut untuk mencari *term-frequency* nya

Berikut persamaan rumus 3.1 TF sebagai berikut:

$$tf(w_i, d_j) = \frac{Jumlah\ kemunculan\ w_i\ pada\ dokumen\ d_j}{jumlah\ kata\ pada\ dokumen\ d_j} \text{.....(3.1)}$$

Penjelasan:

W_i = kata ke-i

d_i = dokumen ke-i

Untuk lebih jelas penggunaan rumusnya dapat dilihat pada tabel 3.9 berikut:

Tabel 3.9 Penerapan TF Pada Dokumen.

W_i / D_i	**Bahasa**	**Banyak**	**Belajar**	**Hari**	**Ini**	**Matematika**	**Saya**	**Suka**	**Yang**
D1	1/4	0	1/4	0	0	0	1/4	1/4	0

D2	1/5	1/5	0	0	0	0	1/5	1/5	1/5
D3	0	0	1/5	1/5	1/5	1/5	1/5	0	0

4. Kemudian, implementasikan rumus berikut untuk mencari IDF nya

   Berikut persamaan rumus 3.2 IDF sebagai berikut:

$$idf(i) = \log\left(\frac{jumlah\ dokumen}{jumlah\ dokumen\ yang\ ada\ kata\ i}\right) \quad \text{.....(3.2)}$$

   Untuk lebih jelas penggunaan rumusnya dapat dilihat pada tabel 3.10 berikut:

Tabel 3.10 Penerapan IDF Pada Dokumen.

W_i / D_i	**Bahasa**	**Banyak**	**Belajar**	**Hari**	**Ini**	**Matematika**	**Saya**	**Suka**	**Yang**
D1	1/4	0	1/4	0	0	0	1/4	1/4	0
D2	1/5	1/5	0	0	0	0	1/5	1/5	1/5
D3	0	0	1/5	1/5	1/5	1/5	1/5	0	0
n_t	2	1	2	1	1	1	3	2	1
Log (N/n_t)	0,17	0,48	0,17	0,48	0,48	0,48	0	0,17	0,48

5. Selanjutnya mengkalikan antara *term-frequency* dan IDF

   Berikut persamaan rumus 3.3 TF-IDF sebagai berikut:

$$TF - IDF = \mathrm{TF} \times \mathrm{IDF} \quad \text{.....(3.3)}$$

   Untuk lebih jelas penggunaan rumusnya dapat dilihat pada tabel 3.11 berikut:

Tabel 3.11 Penerapan TF-IDF Pada Dokumen.

W_i / D_i	**Bahasa**	**Banyak**	**Belajar**	**Hari**	**Ini**	**Matematika**	**Saya**	**Suka**	**Yang**
D1	0.0425	0	0.0425	0	0	0	0	0.0425	0
D2	0.034	0.096	0	0	0	0	0	0.034	0.096
D3	0	0	0.034	0.096	0.096	0.096	0	0	0

Pada tabel 3.11 merupakan fitur statistik yang digunakan dalam pengimplementasian ke dalam algoritma KNN dan algoritma NN. fitur statistik disini didapatkan dari menghitung bobot kata dalam teks dengan TF-IDF, kata kata tersebut merupakan fitur statistiknya.

3.4.5. *Split Dataset*

Pada tahapan ini *dataset* yang telah diektraksi dengan fitur ekstraksi akan dilakukan split *dataset*, *Dataset* akan dibagi 20% untuk data uji dan sisanya untuk data latih. Data latih ini bertujuan untuk melatih algoritma machine learning, setelah data tersebut dilatih akan dilakukan pengujian dengan data uji. *dataset* akan di *split* berdasarkan fitur statistik dan fitur linguistik.

3.4.6. Implementasi KNN dan NN

Berdasarkan hasil dari kedua fitur ekstraksi yaitu fitur statistik dan fitur linguistik. pada tahapan ini akan dilakukan pelatihan algoritma *machine learning* dengan data latih untuk model yang akan dibuat yaitu KNN dan NN. Dilakukan implementasi berdasarkan hasil dari fitur statistik ke kedua algoritma, begitupun juga dengan fitur linguistik ke kedua algoritma dengan menggunakan *dataset* yang sudah di split sebelumnya.

1. *K-Nearest Neighbors* (KNN)

Dalam mengimplementasi KNN tentunya memiliki beberapa tahapan didalamnya, berikut adalah tahapannya.

a. Menentukan nilai K, Jumlah K idealnya adalah bilangan ganjil (Laksono, 2020). Misalkan nilai K=1, untuk lebih jelas datanya ada pada tabel 3.12.

Tabel 3.12 Contoh Fitur Statistik yang Akan di Implementasikan

W_i / D_i	Bahasa	Banyak	Belajar	Hari	Ini	Matematika	Saya	Suka	Yang
D1	0.0425	0	0.0425	0	0	0	0	0.0425	0
D2	0.034	0.096	0	0	0	0	0	0.034	0.096
D3	0	0	0.034	0.096	0.096	0.096	0	0	0

Dari tabel 3.12 nilai K (sama dengan D2) adalah: K = D2 = [0.034, 0.096, 0, 0, 0, 0, 0, 0.034, 0.096]

b. Menghitung jarak menggunakan *Euclidean Distance* dengan rumus persamaan 3.4 berikut.

$$Jarak = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \text{.....(3.4)}$$

Jarak D2 ke D1:

$$Jarak$$
$$= \sqrt{(0.034 - 0.0425)^2 + (0.096 - 0)^2 + (0 - 0.0425)^2 + (0 - 0)^2 + (0 - 0)^2}$$
$$= \sqrt{+(0 - 0)^2 + (0 - 0)^2 + (0.034 - 0.0425)^2 + (0.096 - 0)^2}$$
$$Jarak = 0.1428$$

Jarak D2 ke D3:

$$Jarak$$
$$= \sqrt{(0.034 - 0)^2 + (0.096 - 0)^2 + (0 - 0.034)^2 + (0 - 0.096)^2 + (0 - 0.096)^2}$$
$$= \sqrt{+(0 - 0.096)^2 + (0 - 0)^2 + (0.034 - 0)^2 + (0.096 - 0)^2}$$
$$Jarak = 0.2226$$

Kemudian untuk Jarak D2 ke D2 tentunya 0. Karena tidak ada selisih jarak.

c. Mengambil nilai K terdekat sebanyak nilai K yang ditentukan diawal, hasil yang didapatkan sebelumnya kemudian disusun secara *ascending* atau dari kecil ke besar mulai dari atas kebawah. Untuk lebih jelas dapat dilihat pada tabel 3.13 berikut:

Tabel 3.13 Hasil KNN

D ke-i	Jarak	Label
D1	0.1428	Suka
D3	0.2226	Tidak Suka

Dari kedua data di tabel 3.13 dapat dilihat jarak D1 ke D2 memiliki jarak yang cukup dekat, dibanding dengan D3 ke D2. Oleh karena itu K=1, maka D1 yang merupakan hasil *predicted* nya. Yang artinya bahwa dokumen 1 lebih dekat dengan dokumen 2

dan dapat disimpulkan bahwa prediksinya yaitu suka

2. *Neural Network* (NN)

Dalam mengimplementasi MLP tentunya memiliki beberapa tahapan didalamnya, berikut adalah tahapannya, sebelum itu dapat dilihat pada tabel 3.14 data fitur linguistik.

Tabel 3.14 Data Fitur Linguistik.

Kalimat	***Sentence Length***	***Tittle word***	***Cue-phrase* promo**	***Cue-phrase* penipuan**	**Label**
Saya sudah transfer uangnya ke rekeningmu, Jangan lupa cek saja nanti.	11	2	1	1	Normal
Dapatkan diskon 30% untuk semua pakaian di toko kami hingga akhir minggu ini!	13	1	0	0	Spam
Saya baru saja memenangkan lotre dan ingin berbagi keberuntungan dengan Anda, cukup kirimkan data pribadi Anda untuk klaim hadiah di 082212322243	21	3	1	1	Spam

Tabel 3.14 menampilkan kalimat beserta dengan fitur linguistik dan label dari kalimat tersebut.

a. Menentukan *hidden layer*

Menentukan neuron pada *hidden layer* dan banyaknya *hidden layer* yang akan digunakan. Banyaknya *hidden layer* dapat mempengaruhi waktu dalam proses iterasi nantinya. Misalkan disini 1 *hidden layer* dan 2 *neuron*.

b. Inisialisasi bobot dan bias

Nilai pada *input layer* diisi dengan tiap fitur yang dihasilkan dari representasi TF-IDF maupun fitur linguistik pada setiap dokumen yang ada. dan untuk inisialisasi pembobotan dilakukan secara acak dengan *range* (-1) - 1.

Misalkan kita memiliki bobot dan bias yang sudah diinisialisasi (sebagai contoh):

Bobot *input* ke *hidden layer* (W_1):

$$W_1 = \begin{bmatrix} 0.1 & -0.2 \\ 0.3 & 0.4 \\ 0.5 & -0.6 \\ -0,7 & 0.8 \end{bmatrix}$$

Bias *hidden layer* (b_1):

$$b_1 = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

Bobot *hidden* ke *output layer* (w_2):

$$W_2 = \begin{bmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \end{bmatrix}$$

Bias *output layer* (b_2):

$$b_2 = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

c. *Feedforward*

Pada tahap ini dilakukan proses maju dari *input layer* hingga ke *output layer*. Selanjutnya dilakukan operasi linear pada *input* yang diterima dari lapisan sebelumnya. Ini melibatkan perkalian matriks antara bobot (W) dan *output* (O) dari lapisan sebelumnya (*hidden layer* sebelumnya jika terdapat lebih dari satu atau *input layer*), diikuti dengan penambahan bias. Berikut adalah rumus persamaan 3.5 yang digunakan:

$$Z = W . O + b \quad \text{.....(3.5)}$$

Dimana:

Z = hasil operasi linear (*input* untuk fungsi aktivasi).

W = matriks bobot.

O = vektor *output* dari lapisan sebelumnya.

b = vektor bias.

Berikut perhitungan untuk mencari nilai *output* dari *hidden layer*:

Input: [11,2,1,1]

Bobot *input* ke *hidden layer* (W_1):

$$\begin{bmatrix} 0.1 & -0.2 \\ 0.3 & 0.4 \\ 0.5 & -0.6 \\ -0.7 & 0.8 \end{bmatrix}$$

Bias *hidden layer* (b_1):

$$\begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

Kita hitung z^1 (nilai sebelum aktivasi di *hidden layer*) perhitungan Matriks:

$$Z_1 = W_1 \cdot Input + b_1$$

$$Z_1 = \left(\begin{bmatrix} 0.1 & -0.2 & 0.5 & -0.7 \\ 0.3 & 0.4 & -0.6 & 0.8 \end{bmatrix} \cdot \begin{bmatrix} 11 \\ 2 \\ 1 \\ 1 \end{bmatrix} \right) + \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

$$Z_1 = \begin{bmatrix} 0.1 * 11 - 0.2 * 2 + 0.5 * 1 - 0.7 * 1 \\ 0.3 * 11 + 0.4 * 2 - 0.6 * 1 + 0.8 * 1 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

$$Z_1 = \begin{bmatrix} 1.1 - 0.4 + 0.5 - 0.7 \\ 3.3 + 0.8 - 0.6 + 0.8 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

$$Z_1 = \begin{bmatrix} 0.5 \\ 4.5 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

$$Z_1 = \begin{bmatrix} 0.6 \\ 4.7 \end{bmatrix}$$

Setelah itu, *Output* dari operasi linear kemudian dilanjutkan ke fungsi aktivasi ReLU untuk memperkenalkan non-linearitas ke dalam jaringan. berikut dua persamaan 3.6

dan 3.7 dapat dilihat sebagai berikut.

*output*_tersembunyi = ReLU (Z)……(3.6)

ReLU(Z)=max(0,Z) ……(3.7)

Dalam kata lain, jika nilai *input x* positif atau nol, maka *output* ReLU akan sama dengan nilai *input x*. Namun, jika nilai *input x* negatif, maka *output* ReLU akan menjadi nol. Oleh karena itu, fungsi ReLU mematikan (menghasilkan nol) untuk nilai *input* yang negatif dan mempertahankan nilai *input* yang positif.

Fungsi aktivasi ReLU:

*output*_tersembunyi = ReLU(Z_1) = $\begin{bmatrix}\max(0, 0.6)\\ \max(0, 4.7)\end{bmatrix}$

*output*_tersembunyi = $\begin{bmatrix}0.6\\ 4.7\end{bmatrix}$

Selanjutnya pada *output layer* dilakukan perhitungan yang sama yang dimana guna mencari nilai dari *output layer* dan langkah yang dilakukan sama seperti mencari *output* dari *hidden layer*, disini digunakan fungsi aktivasi *softmax* karena lebih dari dua kelas yakni tiga (*Multiclass*). Berikut adalah rumus persamaan 3.8 yang digunakan:

input_output = bobot_*output* × *output*_tersembunyi + bias_*output*……..(3.8)

Menentukan nilai *input* pada *output layer*

$$Z_2 = W_2 \cdot output\_tersembunyi + b_2$$

$$Z_2 = \left(\begin{bmatrix}0.1 & 0.2\\ 0.3 & 0.4\end{bmatrix} \cdot \begin{bmatrix}0.6\\ 4.7\end{bmatrix}\right) + \begin{bmatrix}0.1\\ 0.2\end{bmatrix}$$

$$Z_2 = \begin{bmatrix}0.1 * 0.6 + 0.2 * 4.7\\ 0.3 * 0.6 + 0.4 * 4.7\end{bmatrix} + \begin{bmatrix}0.1\\ 0.2\end{bmatrix}$$

$$Z_2 = \begin{bmatrix}1.0\\ 2.26\end{bmatrix}$$

Selanjutnya dilakukan implementasi *softmax*, dengan menggunakan fungsi aktivasi *softmax* berikut persamaan rumus 3.9:

$$softmax(\mathrm{Z}_i) = \frac{e^{Z_i}}{\sum_{j=1}^{K} e^{Z_j}} \text{.....(3.9)}$$

Dimana:

e = nilai ketetapan Euler (2.71828)

K = banyaknya kelas yang diklasifikasi

Perhitungan *softmax*:

$$softmax(Z_2) = \left[\frac{e^{1.0}}{e^{1.0} + e^{2.26}}, \frac{e^{2.26}}{e^{1.0} + e^{2.26}}\right]$$

$$softmax(Z_2) \approx [0.268, 0.732]$$

Hasil dari *output layer* pada setiap node akan dibandingkan nilai probablitasnya, yang paling tinggi dapat dikatakan sebagai klasifikasi di kelas tersebut.

d. Hitung *Loss*

Hitung nilai *loss* atau fungsi kerugian berdasarkan perbandingan antara prediksi model dan label yang sebenarnya. Berikut adalah rumus persamaan 3.10 yang digunakan:

$$Cross - Entropy\ Loss\ = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{i,j} \log(P_{i,j}) \text{.....(3.10)}$$

N = jumlah sampel.

K = jumlah kelas

$y_{i,j}$ = label yang sebenarnya untuk sampel i pada kelas j

$P_{i,j}$ = probabilitas prediksi sampel i pada kelas j

Nilai Loss yang paling baik adalah nilai loss yang paling rendah nilainya. Berikut Perhitungannya:

Cross-Entropy Loss untuk Kalimat 1:

L = -Σ(label * log(*output_softmax*))

L = -(1 * log(0.268) + 0 * log(0.732))

L = -(log(0.268)) = -(-0,5719) = 0,5719

e. Backpropagation

Tahapan ini dilakukan hitung gradien dari fungsi kerugian terhadap setiap parameter (bobot dan bias) menggunakan aturan rantai (*chain rule*). Berikut adalah persamaan yang digunakan untuk menghitung gradien dari fungsi kerugian pada persamaan rumus 3.11 dan 3.12 berikut.

$$\frac{\partial L}{\partial Z_2} = softmax(Z) - label \quad \text{.....(3.11)}$$

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Z_2} \times output\_tersembunyi^T \quad \text{.....(3.12)}$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial Z} \quad \text{.....(3.13)}$$

Dimana:

Label = kelas dari label yang di dapatkan 0 dan 1.

Softmax(Z) = fungsi aktivasi *softmax* terhadap Z

$output\_tersembunyi^T$ = transpose dari *output* tersembunyi.

Hitung *Gradien Loss* terhadap w_2 dan b_2:

1. *Gradien Loss* terhadap Z_2 (*softmax output*)

$$\frac{\partial L}{\partial Z_2} = softmax(Z_2) - label$$

$$\frac{\partial L}{\partial Z_2} = [0.268 \quad 0.732] - [1 \;\; 0] \; = \; [-0.732 \quad 0.732]$$

2. *Gradien Loss* terhadap w_2:

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial Z_2} \cdot output\_tersembunyi^T$$

$$\frac{\partial L}{\partial W_2} = [-0.732 \quad 0.732] \cdot [0.6 \quad 4.7]^T$$

$$\frac{\partial L}{\partial W_2} = [-0.732 \quad 0.732] \cdot \begin{bmatrix} 0.6 \\ 4.7 \end{bmatrix}$$

$$\frac{\partial L}{\partial W_2} = [-0.732 \times 0.6 + 0.732 \times 4.7]$$

$$\frac{\partial L}{\partial W_2} = 3.0021$$

3. *Gradien Loss* terhadap b_2:

$$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial Z_2}$$

$$\frac{\partial L}{\partial b_2} = [-0.732 \quad 0.732]$$

Hitung *Gradien Loss* terhadap Z_1 (*output*_tersembunyi):

$$\frac{\partial L}{\partial Z_1} = W_2^T \cdot \frac{\partial L}{\partial Z_2}$$

$$\frac{\partial L}{\partial Z_1} = \begin{bmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \end{bmatrix}^T \cdot \begin{bmatrix} -0.732 \\ 0.732 \end{bmatrix}$$

$$\frac{\partial L}{\partial Z_1} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \cdot \begin{bmatrix} -0.732 \\ 0.732 \end{bmatrix}$$

$$\frac{\partial L}{\partial Z_1} = \begin{bmatrix} 0.1 \times -0.732 + 0.2 \times 0.732 \\ 0.3 \times -0.732 + 0.4 \times 0.732 \end{bmatrix}$$

$$\frac{\partial L}{\partial Z_1} = \begin{bmatrix} -0.0732 + 0.1464 \\ -0.2196 + 0.2928 \end{bmatrix}$$

$$\frac{\partial L}{\partial Z_1} = \begin{bmatrix} 0.0732 \\ 0.0732 \end{bmatrix}$$

Hitung *Gradien Loss* terhadap W_1 dan b_1:

1. *Gradien Loss* terhadap W_1:

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial Z_1} \cdot Input^T$$

$$\frac{\partial L}{\partial W_1} = \begin{bmatrix} 0 \\ 0.1464 \end{bmatrix} \cdot \begin{bmatrix} 11 \\ 2 \\ 1 \\ 1 \end{bmatrix}^T$$

$$\frac{\partial L}{\partial W_1} = \begin{bmatrix} 0 \\ 0.1464 \end{bmatrix} \cdot \begin{bmatrix} 11 & 2 & 1 & 1 \end{bmatrix}$$

$$\frac{\partial L}{\partial W_1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.1464 * 11 & 0.1464 * 2 & 0.1464 * 1 & 0.1464 * 1 \end{bmatrix}$$

$$\frac{\partial L}{\partial W_1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1.6104 & 0.2928 & 0.1464 & 0.1464 \end{bmatrix}$$

2. *Gradien Loss* terhadap b_1:

$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial Z_1}$$

$$\frac{\partial L}{\partial b_1} = \begin{bmatrix} 0.0732 \\ 0.0732 \end{bmatrix}$$

Setelah mendapatkan hasil dari gradian selanjutnya akan dilakukan pembaruan bobot dan bias, tentunya dalam melakukan *update* bobot dan bias terdapat penggunaan *Gradient descent*. *Gradient descent* yang digunakan memperbarui bobot.

Perbarui Bobot dan Bias dengan *Gradient descent*: Misalkan learning rate (laju pembelajaran) adalah α.

1. Perbarui dan b_2:

Misalkan α = 0.01, maka perbarui w_2 dan b_2:

$$W_{2\,baru} = W_2 - \alpha \cdot \frac{\partial L}{\partial W_2}$$

$$W_{2\,baru} = \begin{bmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \end{bmatrix} - 0.01 \times 3.0012 \cdot \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$W_{2\,baru} = \begin{bmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \end{bmatrix} - \begin{bmatrix} 0.030012 & 0.030012 \\ 0.030012 & 0.030012 \end{bmatrix}$$

$$W_{2\,baru} = \begin{bmatrix} 0.1 - 0.030012 & 0.3 - 0.030012 \\ 0.2 - 0.030012 & 0.4 - 0.030012 \end{bmatrix}$$

$$W_{2\,baru} = \begin{bmatrix} 0.069988 & 0.269988 \\ 0.169988 & 0.369988 \end{bmatrix}$$

$$b_{2\,baru} = b_2 - \alpha \cdot \frac{\partial L}{\partial b_2}$$

$$b_{2\,baru} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} - 0.01 \cdot \begin{bmatrix} -0.732 \\ 0.732 \end{bmatrix}$$

$$b_{2\,baru} = \begin{bmatrix} 0.10732 \\ 0.19268 \end{bmatrix}$$

2. Perbarui W_1 dan b_1:

Untuk kolom pertama dari W_1:

$$W_{1\,baru} = W_1 - \alpha \cdot \frac{\partial L}{\partial W_1}$$

$$W_{1\,baru} = \begin{bmatrix} 0.1 & -0.2 \\ 0.3 & 0.4 \\ 0.5 & -0.6 \\ -0.7 & 0.8 \end{bmatrix} - 0.01 \times \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1.6104 & 0.2928 & 0.1464 & 0.1464 \end{bmatrix}$$

$$W_{1\,baru} = \begin{bmatrix} 0.1 & -0.2 \\ 0.3 - 0.016104 & 0.4 - 0.002928 \\ 0.5 - 0.001464 & -0.6 - 0.001464 \\ -0.7 - 0.001464 & 0.8 - 0.001464 \end{bmatrix}$$

$$W_{1\,baru} = \begin{bmatrix} 0.1 & -0.2 \\ 0.283896 & 0.397072 \\ 0.498536 & -0.601464 \\ -0.701464 & 0.798536 \end{bmatrix}$$

$$b_{1\,baru} = b_1 - \alpha \cdot \frac{\partial L}{\partial b_1}$$

$$b_{1\,baru} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} - 0.01 \cdot \begin{bmatrix} 0.732 \\ 0.732 \end{bmatrix}$$

$$b_{1\,baru} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} - \begin{bmatrix} 0.000732 \\ 0.000732 \end{bmatrix}$$

$$b_{1\,baru} = \begin{bmatrix} 0.099267 \\ 0.199267 \end{bmatrix}$$

f. Iterasi

Pada tahapan ini akan secara berulang ulang mulai dari proses *feedforward* dengan nilai yang sudah di *update* seperti bias dan lain-lain sampai dengan *backpropagation*. Setelah itu dicek Kembali apakah sudah konvergen atau tidak, jika tidak akan dilakukan

iterasi sampai konvergen.

3.4.7. Evaluasi KNN dan NN

Pada tahapan ini dilakukan evaluasi terhadap masing masing model. Guna untuk mengetahui perbedaan model mana yang lebih baik. Dapat dilihat dari akurasi, *precision*, *precision*, hingga F1-*Score* nya. Berikut adalah contoh penerapan perhitungan akurasi:

1. Diasumsikan telah melakukan klasifikasi pada sejumlah sampel dan hasilnya dapat dijabarkan dalam bentuk *confusion matrix* sebagai berikut pada tabel 3.13.

Tabel 3.15 Dokumen yang sesuai aktual dan prediksinya.

Actual \ Predicted	A	B	C
A	5	1	2
B	2	6	1
C	1	1	7

Pada tabel 3.15 pada bagian baris menunjukkan kelas sebenarnya (*Actual*) dan kolom menunjukkan prediksi yang diberikan oleh model (*Predicted*).

2. Setelah itu dilakukan perhitungan untuk mencari nilai *Precision*

Untuk kelas A:

$$presisi\ A\ = \frac{5}{5+1+2}\ = 62.5\%$$

Untuk kelas B:

$$presisi\ B\ = \frac{6}{1\ +\ 6\ +1} =\ 66.67\%$$

Untuk kelas C:

$$presisi\ C\ = \frac{7}{2\ +1+7}\ =\ 70\%$$

3. Dilanjutkan dengan perhitungan untuk mencari nilai dari *Precision*

Untuk kelas A:

$$Recall\ A\ = \frac{5}{5+1+2}\ =\ 62.5\%$$

Untuk kelas B:

$$Recall\ B\ = \frac{6}{2\ +\ 6\ +1}\ =\ 66.67\%$$

Untuk kelas C:

$$Recall\ C\ = \frac{7}{1\ +1+7}\ =\ 77.78\%$$

4. Setelah mendapatkan hasil dari nilai *precision* dan nilai *precision* tiap kelas kemudian di implementasikan pada persamaan rumus untuk mencari nilai dari *F1-Score*, berikut rumus persamaan 3.13 berikut.

$$F1\text{-}Score\ = 2\ X\ \frac{Precision\ X\ Recall}{Precision\ +\ Recall} \quad \text{.....(3.13)}$$

Untuk kelas A:

$$F1-Score\ A\ = 2\ \times\ \frac{0.625\times 0.625}{0.625+0.625}\ =\ 62.5\%$$

Untuk kelas B:

$$F1-Score\ B\ = 2\ \times\ \frac{0.6667\times 0.6667}{0.6667+0.6667}\ =\ 66.67\%$$

Untuk kelas C:

$$F1-Score\ C\ = 2\ \times\ \frac{0.7\times 0.7778}{0.7+0.7778}\ =\ 73.72\%$$

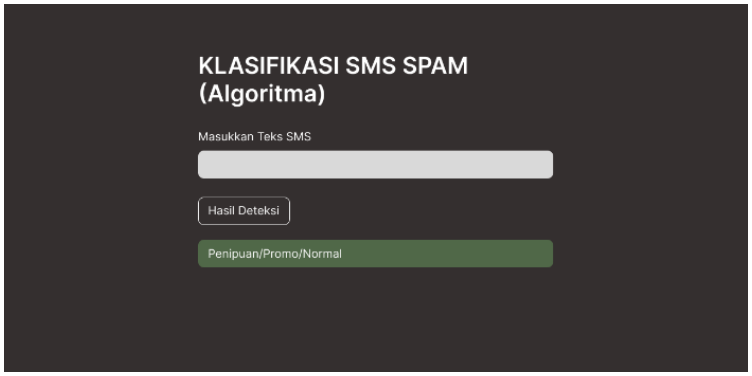
5. Selanjutnya dilakukan perhitungan untuk mencari nilai akurasi

$$Akurasi\ =\ \frac{5+6+7}{(5+1+2)+(2+6+1)+(1+1+7)}\ =\ \frac{18}{27}\ =\ 66.67\ \%$$

Setelah mendapatkan hasilnya, baik itu *precision*, *precision*, f1-*score* dan akurasinya akan dilihat model mana yang lebih baik dari keempat parameter tersebut.

3.4.8. *Deploy* Model

Pada tahapan *deploy* model, model yang telah dibuat pada tahapan sebelumnya akan dilihat dari nilai parameternya dari model yang telah dibuat yakni *precision*, *precision*, f1-*score* dan akurasi. Jika sudah dapat maka tahapan selanjutnya *deploy* pada sebuah sistem sederhana dengan *library python* yaitu *streamlit* untuk melakukan pengetesan terhadap pengklasifikasian SMS spam. Berikut adalah *prototype* dari *streamlit* yang digunakan. Berikut tampilan *prototype* dari *streamlit* dapat dilihat pada gambar 3.8 berikut.

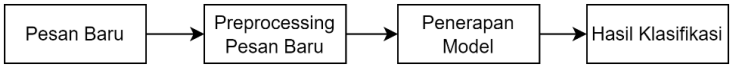


Gambar 3.8 Tampilan *Prototype* Pada *Streamlit*

Pada gambar 3.8 terdapat 2 kolom dan 1 tombol dimana kolom pertama digunakan untuk meng*input*kan pesan untuk mengklasifikasikan pesan tersebut, kemudian jika sudah diisikan kemudian tekan tombol hasil deteksi yang kemudian nanti muncul hasilnya pada kolom berwarna hijau tergantung jenis klasifikasinya, jika spam penipuan akan ditampilkan "penipuan", spam promo dengan kata "promo" dan ham atau normal jika tidak dianggap sebagai pesan spam.

3.5. Klasifikasi

Klasifikasi tentunya merupakan pengujian sebuah model dalam memberikan label baru atau klasifikasi pada sebuah pesan baru, berikut blok diagram proses klasifikasi pada sebuah pesan baru pada gambar 3.9.



Gambar 3.9 Blok Diagram Proses Klasifikasi Pada Data Baru.

Berdasarkan gambar 3.9 dapat dilihat tahapan pada pesan baru yang dimana dilakukan *input*an sebuah pesan baru kemudian pesan baru tersebut dilakukan preprocessing sesuai dengan tahapan model sebelumnya sebagaimana untuk menyesuaikan dengan model yang telah dibuat. Kemudian pada penerapan model dilihat bagaimana pola dari sebuah model yang telah dibuat, kemudian dari pola tersebut sistem model melakukan pelabelan atau klasifikasi pada pesan baru sehingga hasil tersebutlah yang merupakan hasil klasifikasinya.

3.6. Pengujian *Black Box*

Model terbaik dari ke empat model tersebut kemudian dilakukan pengujian sebaik apa model melakukan klasifikasi terhadap pesan baru. klasifikasi dimana modelnya dipilih terlebih dahulu berdasarkan ke 4 model yang ada dan akan menghasilkan *output* yang berupa kelas yakni normal, spam penipuan dan spam promo.

3.7. Alat dan Bahan Penelitian

Peneliti menggunakan perangkat keras dan perangkat lunak sebagai sarana yang mendukung dalam pelaksanaan penelitian.

3.5.1. *Hardware* (Perangkat Keras)

Berikut adalah spesifikasi dari perangkat keras yang digunakan untuk penunjang penelitian, dapat dilihat pada tabel 3.16 berikut.

Tabel 3.16 Spesifikasi Perangkat Keras.

Jenis	Spesifikasi
Processor	AMD Ryzen 5 5600H *with* Radeon *Graphics*
RAM	16 GB DDR4

SSD	500 GB
System Type	64-*bit*

3.5.2. *Software* (Perangkat Lunak)

Berikut adalah spesifikasi dari perangkat lunak yang digunakan untuk penunjang penelitian, dapat dilihat pada tabel 3.17.

Tabel 3.17 Spesifikasi Perangkat Lunak.

Jenis	Tipe	Keterangan
OS	Windows 11	Untuk mengintegrasikan perangkat lunak dengan perangkat keras.
Bahasa Pemrograman	*Python* V3.11.5	Teknologi yang diterapkan untuk mengkonversi bahasa manusia menjadi format yang dapat dipahami oleh mesin
Text Editor	*Visual Studio Code*	Untuk mengintegrasikan algoritma yang dibuat

BAB IV

HASIL DAN PEMBAHASAN

4.1. Analisis Data

Pada tahapan ini menjelaskan tentang pembuatan model yang bertujuan untuk membedakan pesan, seperti spam promo, spam penipuan, atau pesan normal. Model ini akan diuji untuk me ngetahui seberapa akurat dan efektifnya dalam mengidentifikasi jenis-jenis pesan tersebut dengan mencari nilai *precision*, *precision*, *accuracy*, dan *f1-score* sebanyak empat model diantaranya menggunakan fitur statistik dengan KNN, fitur linguistik dengan KNN, fitur statistik dengan NN dan fitur linguistik dengan NN. Kemudian akan dibandingkan diantara keempat model tersebut berdasarkan dari nilai *precision*, *precision*, *accuracy*, dan *f1-score*. Data yang didapatkan terdapat sebanyak 4000 data yang digunakan dalam penelitian ini, yang dimana *dataset*nya diambil dari *dataset* publik yang dibuat oleh seorang peneliti bernama Yudi Wibisono. Untuk dapat melakukan pembacaan *dataset* pada *python* perlu terlebih dahulu melakukan pemasang *library* pandas dan kemudian simpan ke sebuah variabel. Agar lebih jelas dapat dilihat pada gambar 4.1 berikut.

```
import pandas as pd

data = pd.read_csv('Dataset_Pesan_Campuran.csv')
```

Gambar 4.1 Kodingan Pembacaan *Dataset* (file.csv) dengan *Library Pandas*.

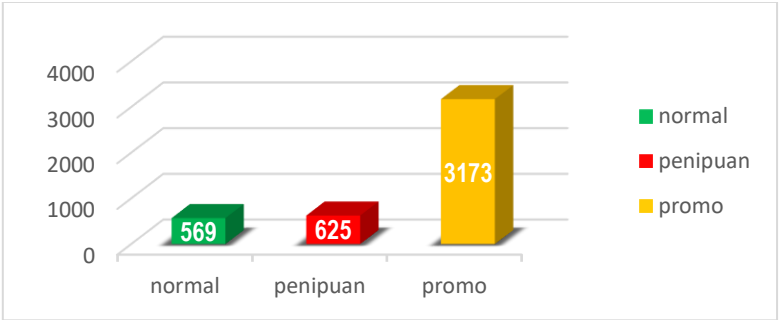
Dari gambar 4.1 dapat dilihat terdapat pemanggilan *library pandas* dengan inisial pd, dan diteruskan dengan variabel data sebagai tempat penampung. Yang kemudian akan mendapatkan hasil seperti contoh *dataset* pesan yang dipakai dalam penelitian ini dapat dilihat pada gambar 4.2 berikut.

	Pesan	label
2	Di kfc yg deket enhaii ada dy	0
3	Maaf jika ada janji yang belum terpenuhi, jika ada janji boleh mengingatkan saya.	0
4	*ngsih bunga ato coklat min	0
5	.sambl nunggu itu.. Gimana kalo ngerjain form formnya aja dulu	0
6	[Akademik] Untuk perhatian tuk jadwal kontrak kuliah buat ang 2012 itu bukan sebenarny, kuliah ikut jadwal kur yg baru ang 2013 ato keatas 2014,2015,2016.. sehingga jadwal yg ada adalah jadwal bayangan tuk tujuan kontrak kuliah..Hal diatas ga berlaku tuk mk yg memang berjalan di ang kalian spt skripsi dan sidang..demikian harap diperhatikan..	0
7	[INFO PLA]Assalam..Bagi yg ambil mk pla wajib kumpul pada hari/tgl kamis,25 agustus 2016 pukul 10.00 WIBB..Agenda: pembekalan dan penjelasan PLA.Tempat : Gdg JICA lt.3 R.S301.Demikian informasiny..harap diperhatikan d disebarkan	0
8	[Info] bagi teman2 yg mengumpulkan data asisten lab/matakuliah bukti berupa surat keterangannya sudah bisa diambil ke pak andri	0
9	2016/08/21 12:27:17 Anda telah terdaftar sebagai pelanggan TCASH full service. Tingkatkan saldo Anda dg isi saldo TCASH di merchant berlogo TCASH terdekat.	0
10	2016/08/21 12:27:32 PIN baru TCash anda adalah 605598. Silahkan menggunakan PIN baru anda untuk menikmati layanan TCash Telkomsel	0
11	22nya mad. Aku bingung bkn laporannya ðŸ˜	0
12	aamiin, makasih semuanya moga cepat lulus juga	0
13	Aamiin. Makasih semuanya..semoga teman2 cepat nyusul dan semoga segala urusannya diberikan kelancaran oleh Allah.aamiin	0
14	Abis maghrib yaa	0
15	Ada belakang warkop deket indomaret gerum 500/bulan	0
16	Ada di bip yang studio fotonya gitu	0
17	Ada nama1 . Tpi gatau msh berfungsi atau engga, udh lama ga dipake sih	0
18	Ada diruangan nya tadi	0
19	Ada internet min alamatnya dmn?	0
20	Ada mba, harganya 1,6jt/bln. Fasilitasnya air hangat, kamar mandi dalam, tv, tv kabel, wifi, meja, tempat tdr, dan lemari.	0
21	Ada nya instagram	0
22	Ada perasaan nyesel gt sedikit. Waktu kuliah. Asa ga merhatin. Ga ngerangkul semua. Aku pribadi hmm	0
23	Ada semua di panduan pla..dishare di grup ini dan fb	0
24	Ada sih yg nikah muda tp (keliatan) bahagia, tp one of a million kayaknya kalo temen2 ku hmm	0
25	Ada teh dikpad, ini kontaknya ####	0
26	Ada yang 400 rb yang kamar mandinya sama2 kalau yang ada kamar mandi di dalam kalau yang kecil 600 dan yang agak besar 700 rb	0
27	Ada yg lain kok ikan tahu tempe	0
28	Ada yg nikah muda, ada yg gagal nikah, ada yg pacaran lama eh putus tapi alesannya karna terlalu sayang jd gak mau nyakitin (pret), ada yg jagain jodoh org, ada yg ah banyak lah wkwk	0
29	Ada yg tau ruangan pak daya yg di fpmipa b?	0
30	Ada yg tdk bisa jam tertentu? Kemungkinan pagi jam 8, tapi nanti diumumkan.	0
31	Aduh aku ge ini pusing yg mau lamaran siapa yg nyari2 kebutuhan siapa	0
32	Aigoo banyak banget chatnyaa aku baru liat hape hehehe	0
33	Airnya asa kurang lancar, besok ku tanya papa yah,	0
34	Ajakin ke soekarno hatta aja nama1 . Makan di eatboss soekarno hatta wkwkw	0
35	Aku ada spagetti, tp usah kaldaluarsa belum yah	0
36	Aku bawa daging yg dah direbus gapapa? Yg mentah tinggal tulang2an wkwk	0
37	Aku besok dtg agak telat jam 8 boleh ngga? Kan jam 10 ya mulainya?	0
38	aku delete aja ya, ntar nama1 pull dulu atau sync dulu, ntar masukin yang nama1 , baru push.	0
39	Aku depan pasca skrng	0
40	Aku di pt. DI	0
4329	Selamat tambahan Flash Anda sudah aktif, berlaku s.d. 2016-08-07 23:59:59.	2
4330	Selamat! Anda berhak membeli PAKET PAS untuk Nelpon, SMS, Internetan, Video, dan NSP. Pilih paketnya MULAI Rp500 di *100*999#, hanya berlaku HARI INI!	2
4331	Selamat! Anda sudah mengaktifkan INSTINK utk dpt info menarik dari Telkomsel. Info berhenti program:*600*600#. CS:133/188	2
4332	Selamat!!anda telah mendapat gratis 200 sms dari SMSGIFT dari no. 6285258952502. Info CS : 133/188	2
4333	Selamat, Anda mendapatkan kuota Tsel sebesar 100SMS. Cek kuota melalui *889#. Info: www.telkomsel.com	2
4334	Selamat, Anda telah membeli Paket Combo Kenyang Internet As. Cek bonus di *889#.	2
4335	Sensasi seru main game yang belum pernah ada. Ayo bergabung dengan para heroes di Game HIT. Download Gratis di http://bit.ly/2c6jsXN	2
4336	Seru banget main 8 Ball Pool, pakai pulsa bisa banget loh Bro untuk beli koin games lainnya, cek caranya di http://tsel.me/jajanonline	2
4337	Smartfren Gratis Mi-Fi/modem 4G Tahun baru saatnya ganti gadget baru Bawa pulang Mi-Fi 4Gm-mu dg tunjukan sms ini ke Galeri SF & isi pulsa 100rb disana s/d 13Jan Info:888	2
4338	Special PROMO! BONUS KUOTA 500MB (1hr)! Cukup isi ulang mulai dari 5,000 hari ini! Ayo buruan isi pulsa sekarang juga. T580	2
4339	Spesial untuk Anda! Paket Flash Volume Ultima, kuota 1,8GB/30hr, kuota lebih banyak! Harga mulai Rp60rb di *100*473#. Tarif&lokasi cek di tsel.me/FL	2
4340	Spesial untuk Anda! Paket Internet Hemat Kuota berlimpah. Hanya Rp35 Ribu kuota 2.5 GB/30 hari. Aktifkan sekarang juga di *550*905#. Buruan..! SKB	2
4341	SPESIAL utk kamu pelanggan setia Indosat Ooredoo. Bonus 14 hari masa aktif, 10 menit telp dan 100 SMS, CUMA isi ulang s/d 16 Desember 2015. (bonus berlaku	2
4342	Sticker LINE yang lucu-lucu banyak banget loh, beli yuk dengan pulsamu, gampang banget, coba yuk	2
4343	SUPER Wow! Gratis 12 GB buat kamu, ckup dgn isi pulsa minimun Rp 100 Ribu s.d 22 November. Hanya berlaku satu kali kesempatan & berlaku akumulasi loh!	2
4344	Tambahan THR dr MatahariMall. Kode THR6 utk Diskon 6% Elektronik, kode THR15 utk Diskon 15% Lifestyle, kode THR25 utk Diskon 25% Fashion. http://mthr.ml/TSELTHR	2
4345	Terima Kasih telah menggunakan Paket Data dari Indosat. Info detail tekan *123#	2
4346	Terima kasih telah menjadi pelanggan setia Indosatooredoo. Download Video + BONUS GRATIS Nelpon 60 Menit, hubungi *900*8#	2
4347	Terima kasih telah menjadi Sahabat IndosatOoredoo, dapatkan Video & Game seru + Gratis telp di *323*2#	2
4348	Terima kasih telah menjadi Sahabat IndosatOoredoo. Kenalan tanpa batas dan BONUS GRATIS NELPON 60 menit, hub *955*1# lalu YA/OK	2
4349	Terima kasih telah menjadi Sahabat IndosatOoredoo. Mau tahu RAHASIA2 TERHEBOH ARTIS-ARTIS TOP + bonus gratis nelpon 60 menit?Tlp *700*1#	2
4350	Terima kasih telah menjadi Sahabat IndosatOoredoo. Download Video + BONUS GRATIS Nelpon 60 Menit, hubungi *900*8#	2
4351	Terima kasih tlh melakukan isi ulang tgl. 20.02.2016. GRATIS NELPON 60Menit+VIDEO, tekan *900*9# lalu YES/OK	2
4352	Terkini, ungkapin apa yang ada dipikiranmu lewat sticker line, beli pake pulsa TSel https://store.line.me/ download di stickershop	2
4353	Terus Isi saldo TCASHmu dan nikmati aneka promo di merchants partner TCASH! Cek promo terbaru TCASH di tsel.me/tappromo	2
4354	Thank you for your appreciation & trust in using Telkomsel service. For further feedback please advice & text sms to 1166 (FREE)(for int'l charged)	2
4355	Tingkatkan nilai isi ulang Anda selanjutnya minimal Rp10ribu untuk bisa mendapatkan Paket MURAH PAS di *100*999#. Info 188.	2
4356	Tri berBAGI bonus PULSA di bulan pernuh berkah! Isi pulsa min. 5rb dapet bonus pulsa 10rb utk nelp ke nmr Tri! Hanya hari ini saja! Ayo buruan isi pulsa! T496	2
4357	TrmKsh Anda Tlh Berhasil Mengisi Pulsa SMARTFREN 10.000.Mau Jualan PULSA/BAYAR LISTRIK,SPEEDY/TELKOM Hub:Agen BAROKAH/Mp Reload.Pasti untung?	2
4358	Udah top up coins di LINE utk beli sticker atau themes? Beli skrg jg pake pulsa utk tambah koleksi-mu! Cek caranya di http://tsel.me/jajanonline	2
4359	Use my invite code, nxmvf722ue, and get a free ride up to Rp75,000. Redeem it at https://www.uber.com/invite/nxmvf722ue	2
4360	Wow! Diskon 50% utk pembelian paket Flash 6GB + BONUS 60Menit CUMA Rp60rb utk 30hr di *999*606# SEKARANG! Tunggu apalagi buruan! S&K berlaku	2
4361	Wow! Promo diperpanjang sd 13 Agt. Khusus kamu! Tukar HP Smartfrenmu dengan Andromax E LTE Cuma 709rb dr 999rb. Ayo datangi Gallery Smartfren terdekat.	2
4362	Yang lain penuh batasan,kami beri kebebasa n. DOUBLE KUOTA BONUS 10GB(4G), UNLIMITED Nelp&SMS, Musik streaming gak pakai Kuotalcuma dg FREEDOM COMBO. Daftar*123#	2
4363	Yuk dengerin secara langsung update keseharian Seleb dr Blog Suaranya! Tlp *5094* GRATIS 3hr lalu Rp2500/mgg.CS:838. VAS280	2
4364	Yuk INTERNET-an NGEBUT utk akses FB, Twitter, & Chatting dgn beli Paket Flash 500MB/7hr (sesuai zona lokasi) mulai Rp25rb di *100*433# cek tarif di tsel.me/FL	2
4365	Yuk temen belanja di google play, mudah banget loh, pakai pulsa juga bisa, http://tsel.me/jajanonline	2
4366	Yuk tetap gunakan Flash Volume Ultima utk update informasi Anda, kuota 60MB/7hr mulai Rp7rb di *100*431#. Tarif&lokasi cek di tsel.me/FL	2
4367	Mau nonton bioskop gratis bersama keluarga? Cigna berikan khusus untuk Anda. Tunggu telepon dari kami. Info Cigna Klik bit.ly/tentangCIGNA	2
4368	Yuks internetan seru-seruan dg Flash Volume Ultima kuota 1.8GB/30hr mulai dari Rp60rb sesuai zona lokasi. Hub *363*145#. Info tarif & lokasi lihat di	2

Gambar 4.2. *Dataset* Publik (Sumber: Yudi Wibisono)

Berdasarkan data pada gambar 4.2 dapat dikelompokkan datanya sebagai berikut

dapat dilihat pada gambar 4.3.



Gambar 4.3. Banyaknya Data Berdasarkan Label.

4.2. Fitur Linguistik

Pada tahapan ini dilakukan representasi fitur linguistik berdasarkan *dataset*. Terdapat 4 fitur linguistik yang digunakan pada penelitian ini yaitu, *sentence length, tittle word, cue-phrase* promo dan *cue-phrase* penipuan. Berikut tahapannya.

1. *Sentence Length*

Pada tahapan ini dilakukan pembacaan pesan dari *dataset* untuk menghitung banyaknya kata dari kalimat yang ada, dapat dilihat pada gambar 4.4 berikut.

```
def sentence_length(text):
    words = text.split()
    return len(words)
```

Gambar 4.4 *Sentence Length.*

Dapat dilihat pada gambar 4.4 penggunaan text.split() untuk membagi teks menjadi sebuah list berdasarkan spasi. Yang kemudian dihitung menggunakan *len*(words).

2. *Tittle word*

Pada tahapan ini dilakukan pembacaan pesan dari *Dataset* untuk menghitung banyaknya huruf kapital di awal kalimat, dapat dilihat pada gambar 4.5 berikut.

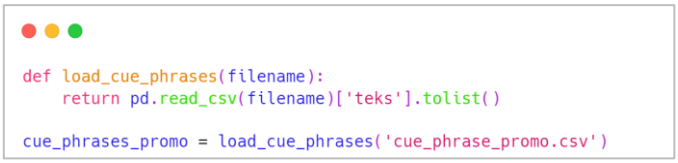
```
def count_capitalized_words(text):
    return sum(1 for word in re.findall(r'\b[A-Z][a-z]*\b', text))
```

Gambar 4.5 *Tittle Word*.

Dapat dilihat pada gambar 4.5 penggunaan re atau *regular expression* untuk menghitung banyaknya kata yang berawalan dengan huruf kapital.

3. *Cue-phrase* promo

Pada tahapan ini dilakukan pembacaan pesan dari *dataset* untuk menghitung kesamaan kata dengan kata yang memiliki makna terselubung di dalam pesan promo, dapat dilihat pada gambar 4.6 berikut.



```
def load_cue_phrases(filename):
    return pd.read_csv(filename)['teks'].tolist()

cue_phrases_promo = load_cue_phrases('cue_phrase_promo.csv')
```

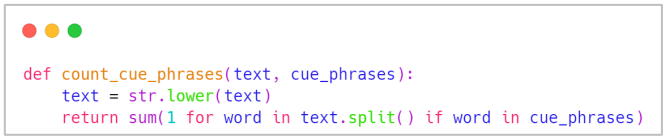
Gambar 4.6 Membaca kata *cue-phrase* promo.

Dapat dilihat pada gambar 4.6 pembacaan file.csv yang berisikan *cue-phrase* promo, lebih jelas dapat dilihat pada tabel 4.1 berikut.

Tabel 4.1 *Cue-phrase* promo.

Cue-phrase promo
eksklusif, terbatas, segera, urgent, kesempatan, langka, special, bonus, diskon, cashback, jackpot, keberuntungan, cepat, hanya, terakhir, jangan lewatkan, sensasional, luar biasa, menguntungkan, jaminan, potongan, hemat, berhadiah, unik, langsung, terbaru, fitur, upgrade, premiere, terbatas waktu, kouta, internet, beli, premium, prioritas, pilihan.

Berdasarkan tabel 4.1 kemudian di implementasikan seperti pada gambar 4.7 berikut.



```
def count_cue_phrases(text, cue_phrases):
    text = str.lower(text)
    return sum(1 for word in text.split() if word in cue_phrases)
```

Gambar 4.7 Menghitung *Cue-Phrase* Promo.

4. *Cue-phrase* penipuan

Pada tahapan ini dilakukan pembacaan pesan dari *dataset* untuk menghitung kesamaan kata dengan kata yang memiliki makna terselubung di dalam pesan penipuan, dapat dilihat pada gambar 4.8 berikut.

```
def load_cue_phrases(filename):
    return pd.read_csv(filename)['teks'].tolist()

cue_phrases_fraud = load_cue_phrases('cue_phrase_penipuan.csv')
```

Gambar 4.8 Membaca *Cue-Phrase* Penipuan.

Dapat dilihat pada gambar 4.8 pembacaan file.csv yang berisikan *cue-phrase* penipuan, lebih jelas dapat dilihat pada tabel 4.2 berikut.

Tabel 4.2 *Cue-Phrase* Penipuan.

Cue-phrase penipuan
hadiah, menang, undian, klaim, gratis, terpilih, transfer, verifikasi, kode, rahasia, bank, pembayaran, dapatkan, klik, tautan, konfirmasi, akun, pinjaman, investasi, keamanan, garansi, eksklusivitas, cepat, terbatas, keuntungan, resmi, asli, terpercaya, cek, hadiah besar, pemberian, mendapatkan, transaksi, batas, peringatan, syarat

Berdasarkan tabel 4.2 akan dihitung sesuai dengan kata-kata diatas dengan kata yang ada pada pesan, lebih jelas dapat di implementasikan seperti pada gambar 4.9 berikut.

```
def count_cue_phrases(text, cue_phrases):
    text = str.lower(text)
    return sum(1 for word in text.split() if word in cue_phrases)
```

Gambar 4.9 Menghitung *Cue-Phrase* Penipuan

Berdasarkan tahapan tahapan fitur linguistik akan memperoleh hasil sebagai berikut. Lebih jelas dapat dilihat pada tabel 4.3 berikut.

Tabel 4.3 Fitur Linguistik

Index Pesan	*Sentence Length*	*Tittle Word*	*Cue-phrase* promo	*Cue-phrase* penipuan
0	7	1	0	0
1	13	1	0	0
2	5	0	0	0
3	10	1	0	0
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
4363	14	1	0	0
4364	20	6	0	1
4365	19	7	0	2
4366	24	6	0	0

Hasil yang didapatkan berdasarkan fitur yang didapat sesuai dengan tabel 4.3, kemudian disimpan dan akan diterapkan pada tahapan implementasi fitur linguistik pada algoritma KNN dan NN.

4.3. *Preprocessing* Teks

Pada tahapan ini dilakukan pemrosesan teks untuk mengolah pesan pada dokumen menjadi sebuah kata dasar. Berikut tahapannya.

1. Case Folding

Pada tahapan ini *dataset* sebelumnya pada kolom pesan akan dihapus baik itu tanda titik, koma, simbol maupun url, atau link-link lainnya. Berikut adalah kodingan pengimplementasiannya dapat dilihat pada gambar 4.10.

```
def casefolding(text):
    text = text.lower()
    text = re.sub(r'http[s]?:?://\S+|www\.\S+(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\\(\\),]|(?:%[0-9a-fA-F][0-
9a-fA-F]))+', '', text)
    text = re.sub(r'[-]?[0-9]+', '', text)
    text = re.sub(r'[+]?', '', text)
    text = re.sub(r'[^\w\s]', '', text)
    text = text.strip()
    return text
```

Gambar 4.10 Kodingan *Case Folding.*

Dari kodingan pada gambar 4.4, selanjutnya implementasikan ke pesan dengan *case folding*. Lebih jelas dapat dilihat pada gambar 4.11 berikut.

```
data['case_folding'] = data['content'].apply(casefolding)
case_folding = data['case_folding']
print(case_folding)
```

Gambar 4.11 Implementasi *Case Folding*

Dari gambar 4.11 akan mendapatkan hasil seperti pada tabel 4.4 berikut.

Tabel 4.4 *Case Folding*

No.	Hasil *Case Folding*
1	di kfc yg deket enhaii ada dy
2	maaf jika ada janji yang belum terpenuhi jika ada janji boleh mengingatkan saya
3	ngsih bunga ato coklat min
4	sambl nunggu itu gimana kalo ngerjain form formnya aja dulu
5	akademik untuk perhatian tuk jadwal kontrak kuliah buat ang itu bukan sebenarny kuliah ikut jadwal kur yg baru ang ato keatas sehingga jadwal yg ada adalah jadwal bayangan tuk tujuan kontrak kuliahhal diatas ga berlaku tuk mk yg memang berjalan di ang kalian spt skripsi dan sidangdemikian harap diperhatikan
6	info plaassalambagi yg ambil mk pla wajib kumpul pada haritgl kamis agustus pukul wibbagenda pembekalan dan penjelasan platempat gdg jica lt rsdemikian informasinyharap diperhatikan d disebarkan
7	info bagi teman yg mengumpulkan data asisten labmatakuliah bukti berupa surat keterangannya sudah bisa diambil ke pak andri
8	anda telah terdaftar sebagai pelanggan tcash full service tingkatkan saldo anda dg isi saldo tcash di merchant berlogo tcash terdekat
9	pin baru tcash anda adalah silahkan menggunakan pin baru anda untuk menikmati layanan tcash telkomsel
10	nya mad aku bingung bkn laporannya ðÿ
11	aamiin makasih semuanya moga cepat lulus juga
12	aamiin makasih semuanyasemoga teman cepat nyusul dan semoga segala urusannya diberikan kelancaran oleh allahaamiin
.	.
.	.
.	.
4363	yuk dengerin secara langsung update keseharian seleb dr blog suaranya tlp gratis hr lalu rpmggcs vas

4364	yuk internetan ngebut utk akses fb twitter chatting dgn beli paket flash mbhr sesuai zona lokasi mulai rprb di cek tarif di tselmefl
4364	yuk temen belanja di google play mudah banget loh pakai pulsa juga bisa
4365	yuk tetap gunakan flash volume ultima utk update informasi anda kuota mbhr mulai rprb di tariflokasi cek di tselmefl
4366	mau nonton bioskop gratis bersama keluarga cigna berikan khusus untuk anda tunggu telepon dari kami info cigna klik bitlytentangcigna
4367	yuks internetan seruseruan dg flash volume ultima kuota gbhr mulai dari rprb sesuai zona lokasi hub info tarif lokasi lihat di tselmefl

Berdasarkan tabel 4.4 adalah hasil yang didapatkan melalui proses *case folding*.

2. *Word Normalize*

Pada tahapan ini dilakukan untuk mengganti kata yang berupa singkatan pada text menjadi kata asli setelah semua huruf dijadikan *lowercase*, lebih jelas dapat dilihat berikut pada gambar 4.12.

```
def text_normalize(text):
    text = ' '.join([key_norm[key_norm['singkat'] == word]['hasil'].values[0]
    if (key_norm['singkat'] == word).any()
    else word for word in text.split()
    ])
    text = str.lower(text)
    return text
```

Gambar 4.12 Kodingan *Text Normalize.*

Dari kodingan pada gambar 4.12, selanjutnya implementasikan ke pesan dengan *text normalize*. Lebih jelas dapat dilihat pada gambar 4.13 berikut.

```
data['text_normalize'] = data['case_folding'].apply(text_normalize)
text_normalize = data['text_normalize']
print(text_normalize)
```

Gambar 4.13 Implementasi *Text Normalize*

Dari gambar 4.13 akan mendapatkan hasil seperti pada tabel 4.5 berikut.

Tabel 4.5 *Text Normalize.*

No.	Hasil *Text Normalize*
1	di kfc yang dekat enhaii ada dia
2	maaf jika ada janji yang belum terpenuhi jika ada janji boleh mengingatkan saya

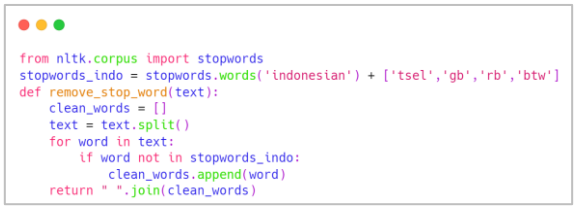
3	ngasih bunga ato cokelat min
4	sambl nunggu itu bagaimana kalau mengerjakan form formnya saja dulu
5	akademik untuk perhatian tuk jadwal kontrak kuliah buat ang itu bukan sebenarny kuliah ikut jadwal kur yang baru ang ato keatas sehingga jadwal yang ada adalah jadwal bayangan tuk tujuan kontrak kuliahhal diatas tidak berlaku tuk mk yang memang berjalan di ang kalian seperti skripsi dan sidangdemikian harap diperhatikan
6	informasi plaassalambagi yang ambil mk pla wajib kumpul pada haritgl kamis agustus pukul wibbagenda pembekalan dan penjelasan platempat gdg jica lantai rsdemikian informasinyharap diperhatikan d disebarkan
7	informasi bagi teman yang mengumpulkan data asisten labmatakuliah bukti berupa surat keterangannya sudah bisa diambil ke pak andri
8	anda telah terdaftar sebagai pelanggan tcash full servis tingkatkan saldo anda dengan isi saldo tcash di merchant berlogo tcash terdekat
9	pin baru tcash anda adalah silahkan menggunakan pin baru anda untuk menikmati layanan tcash telkomsel
10	nya mad saya bingung bukan laporannya ðÿ
11	aamiin terimakasih semuanya moga cepat lulus juga
12	aamiin terimakasih semuanyasemoga teman cepat nyusul dan semoga segala urusannya diberikan kelancaran oleh allahaamiin
.	.
.	.
.	.
4363	yuk internetan mengebut untuk akses fb twitter chatting dengan beli paket flash mbhr sesuai zona lokasi mulai rprb di cek tarif di tselmefl
4364	yuk temen belanja di google play mudah banget loh pakai pulsa juga bisa
4364	yuk tetap gunakan flash volume ultima untuk update informasi anda kuota mbhr mulai rprb di tariflokasi cek di tselmefl
4365	mau nonton bioskop gratis bersama keluarga cigna berikan khusus untuk anda tunggu telepon dari kami informasi cigna klik bitlytentangcigna
4366	yuks internetan seruseruan dengan flash volume ultima kuota gbhr mulai dari rprb sesuai zona lokasi hubungi informasi tarif lokasi li
4367	yuk internetan mengebut untuk akses fb twitter chatting dengan beli paket flash mbhr sesuai zona lokasi mulai rprb di cek tarif di tselmefl

Berdasarkan tabel 4.5 adalah hasil yang didapatkan melalui proses *text normalize*.

3. Stopwords Removal

Pada tahapan ini dilakukan penghapusan kata yang mengandung kata-kata yang tidak memiliki makna penting dalam konteks analisis teks. lebih jelas dapat dilihat berikut

pada gambar 4.14.



```
from nltk.corpus import stopwords
stopwords_indo = stopwords.words('indonesian') + ['tsel','gb','rb','btw']
def remove_stop_word(text):
    clean_words = []
    text = text.split()
    for word in text:
        if word not in stopwords_indo:
            clean_words.append(word)
    return " ".join(clean_words)
```

Gambar 4.14 Kodingan *Stopwords Removal.*

Dari kodingan pada gambar 4.14 melakukan pengambilan daftar kata-kata yang biasanya tidak terlalu penting dalam sebuah kalimat, seperti "dan", "saya", atau "di". Kata-kata ini sering kali diabaikan saat komputer memahami atau menganalisis teks karena mereka tidak menambahkan banyak makna, selanjutnya implementasikan ke pesan dengan *stopword removal*. Lebih jelas dapat dilihat pada gambar 4.15 berikut.

```
data['stopwords_removal'] = data['text_normalize'].apply(stemming)
stopwords_removal = data['stopwords_removal']
print(stopwords_removal)
```

Gambar 4.15 Implementasi *Stopwords Removal.*

Dari gambar 4.15 akan mendapatkan hasil seperti pada tabel 4.6 berikut.

Tabel 4.6 *Stopwords Removal.*

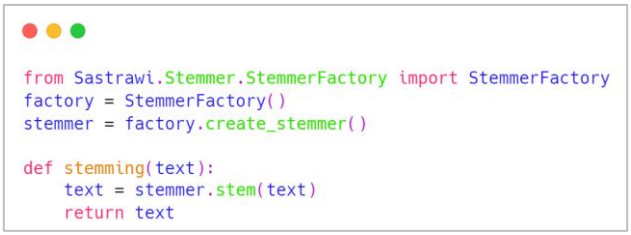
No.	Hasil *Stopwords Removal*
1	di kfc yang dekat enhaii ada dia
2	maaf jika ada janji yang belum terpenuhi jika ada janji boleh mengingatkan saya
3	ngasih bunga ato cokelat min
4	sambl nunggu itu bagaimana kalau mengerjakan form formnya saja dulu
5	akademik untuk perhatian tuk jadwal kontrak kuliah buat ang itu bukan sebenarny kuliah ikut jadwal kur yang baru ang ato keatas sehingga jadwal yang ada adalah jadwal bayangan tuk tujuan kontrak kuliahhal diatas tidak berlaku tuk mk yang memang berjalan di ang kalian seperti skripsi dan sidangdemikian harap diperhatikan
6	informasi plaassalambagi yang ambil mk pla wajib kumpul pada haritgl kamis agustus pukul wibbagenda pembekalan dan penjelasan platempat gdg jica lantai rsdemikian informasinyharap diperhatikan d disebarkan

7	informasi bagi teman yang mengumpulkan data asisten labmatakuliah bukti berupa surat keterangannya sudah bisa diambil ke pak andri
8	anda telah terdaftar sebagai pelanggan tcash full servis tingkatkan saldo anda dengan isi saldo tcash di merchant berlogo tcash terdekat
9	pin baru tcash anda adalah silahkan menggunakan pin baru anda untuk menikmati layanan tcash telkomsel
10	nya mad saya bingung bukan laporannya ðÿ
11	aamiin terimakasih semuanya moga cepat lulus juga
12	aamiin terimakasih semuanyasemoga teman cepat nyusul dan semoga segala urusannya diberikan kelancaran oleh allahaamiin
.	.
.	.
.	.
4364	yuk temen belanja di google play mudah banget
4365	yuk tetap guna flash volume ultima untuk update
4366	mau nonton bioskop gratis sama keluarga cigna
4367	yuks internetan seruseruan dengan flash volume

Berdasarkan tabel 4.6 adalah hasil yang didapatkan melalui proses *text normalize*.

4. Stemming

Menemukan kata dasar (*root word*) dengan menghilangkan semua jenis imbuhan, termasuk imbuhan awalan (*prefix*), imbuhan akhiran (*suffix*), dan bahkan kombinasi imbuhan awalan dan akhiran (*confixes*) dari sebuah kata. Ini dilakukan untuk menyederhanakan kata-kata dalam teks sehingga kata-kata yang memiliki bentuk yang berbeda tetapi merujuk pada konsep yang sama dapat diidentifikasi. Lebih jelas dapat dilihat pada gambar 4.16 berikut.



```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stemming(text):
    text = stemmer.stem(text)
    return text
```

Gambar 4.16 Kodingan *Stopwords Removal.*

Dari kodingan pada gambar 4.16 melakukan *import StemmerFactory* dari modul *Sastrawi.Stemmer* yang ada dalam *library Sastrawi*. *Sastrawi* digunakan untuk melakukan

proses stemming pada teks bahasa Indonesia. Yang dimana untuk mengubah kata-kata dalam teks ke bentuk dasar, misalnya mengubah kata "membaca", "dibaca", "membacakan" menjadi bentuk dasar "baca". *StemmerFactory* merupakan kelas yang digunakan untuk membuat objek stemmer yang akan melakukan proses *stemming* ini, selanjutnya implementasikan ke pesan dengan *stemming*. Lebih jelas dapat dilihat pada gambar 4.17 berikut.

```
data['stopwords_removal'] = data['text_normalize'].apply(stemming)
stopwords_removal = data['stopwords_removal']
print(stopwords_removal)
```

Gambar 4.17 Implementasi *Stemming.*

Dari gambar 4.17 akan mendapatkan hasil seperti pada tabel 4.7 berikut.

Tabel 4.7 *Stemming.*

No.	Hasil S*temming*
1	di kfc yang dekat enhaii ada dia
2	maaf jika ada janji yang belum terpenuhi jika ada janji boleh mengingatkan saya
3	ngasih bunga ato cokelat min
4	sambl nunggu itu bagaimana kalau mengerjakan form formnya saja dulu
5	akademik untuk perhatian tuk jadwal kontrak kuliah buat ang itu bukan sebenarny kuliah ikut jadwal kur yang baru ang ato keatas sehingga jadwal yang ada adalah jadwal bayangan tuk tujuan kontrak kuliahhal diatas tidak berlaku tuk mk yang memang berjalan di ang kalian seperti skripsi dan sidangdemikian harap diperhatikan
6	informasi plaassalambagi yang ambil mk pla wajib kumpul pada haritgl kamis agustus pukul wibbagenda pembekalan dan penjelasan platempat gdg jica lantai rsdemikian informasinyharap diperhatikan d disebarkan
7	informasi bagi teman yang mengumpulkan data asisten labmatakuliah bukti berupa surat keterangannya sudah bisa diambil ke pak andri
8	anda telah terdaftar sebagai pelanggan tcash full servis tingkatkan saldo anda dengan isi saldo tcash di merchant berlogo tcash terdekat
9	pin baru tcash anda adalah silahkan menggunakan pin baru anda untuk menikmati layanan tcash telkomsel
10	nya mad saya bingung bukan laporannya ðÿ
11	aamiin terimakasih semuanya moga cepat lulus juga

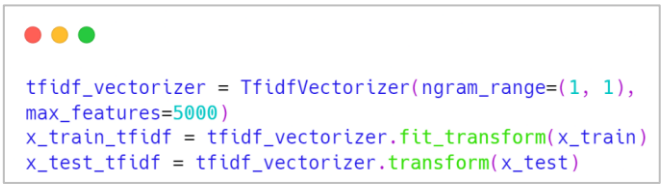
12	aamiin terimakasih semuanyasemoga teman cepat nyusul dan semoga segala urusannya diberikan kelancaran oleh allahaamiin
.	.
.	.
.	.
4364	yuk tetap guna flash volume ultima untuk updat...
4365	mau nonton bioskop gratis sama keluarga cigna ...
4366	yuks internetan seruseruan dengan flash volume...
4367	yuk temen belanja di google play mudah banget ...

Berdasarkan tabel 4.7 adalah hasil yang didapatkan melalui proses *stemming*.

4.4. Fitur Statistik

Pada tahapan ini, setelah mendapatkan hasil dari *preprocessing* teks kemudian dilakukan proses TF-IDF untuk mencari nilai bobot dari sebuah dokumen pada korpus yang ada untuk digunakan sebagai sebuah fitur statistik.

Berikut kodingan untuk menerapkan fitur statistik yaitu tf-idf. Untuk lebih jelas dapat dilihat pada gambar 4.18.



```
tfidf_vectorizer = TfidfVectorizer(ngram_range=(1, 1),
max_features=5000)
x_train_tfidf = tfidf_vectorizer.fit_transform(x_train)
x_test_tfidf = tfidf_vectorizer.transform(x_test)
```

Gambar 4.18 Tf-idf.

Berdasarkan gambar 4.18 akan mendapatkan hasil berikut. Dapat dilihat pada tabel 4.8 berikut.

Tabel 4.8 Fitur Statistik

Index Pesan	Hasil Fitur Statistik
0	0.0, 0.0, 0.0, …….. , 0.1897183788418123, …….. , 0.0.
1	0.0, 0.0, 0.0, …….. , 0. 0.14812848238646548, …….. , 0.0.
2	0.0, 0.0, 0.0, …….. , 0. 0.27167991909641626, …….. , 0.0.
.	.
.	.
.	.
4364	0.0, 0.0, 0.0, …….. , 0.20536668111376213, …….. , 0.0.
4365	0.0, 0.0, 0.0, …….. , 0.24772080073672192, …….. , 0.0.

4366	0.0, 0.0, 0.0, …….. , 0.3028273355833191, …… , 0.0.

Pada tabel 4.8 karena memiliki fitur yang dibatasi sebanyak 5000 fitur maka yang diambil hanya yang memiliki nilai atau perwakilan saja seperti pada tabel 4.8.

4.5. *Split Dataset*

Selanjutnya, setelah mendapatkan nilai fitur dari fitur statistik dan fitur linguistik. Akan dilakukan proses *split dataset* untuk membagi *dataset* 80% data latih sedangkan sisanya sebagai data uji atau sebanyak 20%. Pembagian ini dilakukan untuk masing-masing fitur yakni, split *dataset* untuk statistik dan split *dataset* untuk linguistik. Lebih jelas berikut kodingan untuk melakukan proses pembagian dapat dilihat pada gambar 4.19.

```
x = data['clean_teks'] #statistik
x = features_linguistic #linguistik
y = data['label']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42
```

Gambar 4.19 *Split Dataset*.

Hasil yang didapatkan berdasarkan gambar 4.19 berupa pembagian *dataset* yang ada sebelumnya dengan test size 0.2 atau 20% dan sisanya untuk data train. Isi dari data tersebut berupa acak. Adapun fitur linguistik dengan membagi pembacaan dari variabel x nya dengan *feature_linguistik* dan statistik dengan *clean_teks* yang dimana pembagian secara terpisah dengan kodingan yang sama seperti gambar 4.19 sebelumnya.

4.6. Implementasi KNN dan NN

Setelah melakukan *split dataset* berdasarkan fitur statistik dan linguistik, kemudian dilanjutkan pada proses pengimplementasi algoritma KNN dan NN menggunakan data latih yang sudah dibagi pada tahapan sebelumnya. Berikut sampel dari masing masing fitur yang sudah melalui tahapan ekstraksi fitur dapat dilihat pada tabel 4.9.

Tabel 4.9 Sampel Fitur Statistik dari *Dataset*

Pesan (sudah di *preprocessing*)	**Label**	**Fitur TF – IDF**
salah satu kuota internet anda telah habis pastikan anda memiliki kuota lainnya cek secara berkala kuota dan beli paket di tselmetsel atau hub	2	anda:0.2819338021058919, atau:0.12869061082851369, beli:0.15734335724205817, berkala:0.274749217672005, cek:0.1710128618504541, dan:0.1710128618504541, di:0.0975743060505912, habis:0.23434588467305734, hub:0.16987936897816722, internet:0.16835968193481882, kuota:0.4148607357038051, lainnya:0.20787868066406873, memiliki:0.23482592175425965, paket:0.12090898020822247, pastikan:0.24852039817620059, salah:0.26709006646175654, satu:0.23110304272441426, secara:0.25589809952614856, telah:0.18302934545847718, tselmetsel:0.2302121113097295
hanya hari ini kamu bisa dpt diamond freefire hanya rp loh maksimal x trxnomorhari beli pakai pulsa telkomselmu di tselmepfh skb	2	beli:0.12934614939521338, bisa:0.1623046746330956, di:0.08021222496312963, diamond:0.29626173295823455, dpt:0.1938419178257319, freefire:0.335855267464546, hanya:0.2994560083867521, hari:0.1318694348986634, ini:0.15060988516745544, kamu:0.16994996870301832, loh:0.23745302711370572, maksimal:0.27333293182818097, pakai:0.21673035401320456, pulsa:0.12397516204029022, rp:0.09366455006729327, skb:0.11450971114688897, telkomselmu:0.335855267464546, trxnomorhari:0.335855267464546, tselmepfh:0.335855267464546

Untuk sampel dari fitur linguistik dapat dilihat pada tabel 4.10 berikut.

Tabel 4.10 Sampel Fitur Linguistik dari *Dataset*.

Kalimat	**Panjang Kalimat**	**Jumlah Kata dengan Huruf Besar**	**Jumlah Cue-Phrase Promo**	**Jumlah Cue-Phrase Penipuan**
Pulsa 63amper habis. Segera isi pulsa di MyTelkomsel untuk mendapatkan promo	17	3	1	0

menarik lainnya!				
Yg mau ngampus aku pengen titip bawain skl aku. Thanks yaa	11	1	0	0
modal 10rb bisa menang Jut44n rupiah pasang T*toa sekarang!	18	4	1	1

Berdasarkan sampel tabel diata akan di terapkan pada masing masing model. Berikut penjelasan terkait bagaimana pengimplementasiannya:

1. Fitur statistik dengan KNN

Pada penerapan fitur statistik dengan algoritma KNN, yang dimana hasil dari representasi tf-idf sebelumnya digunakan sebagai fitur dalam penerapan pada KNN. Berikut 2 data yang diambil *dataset* yang sudah melakukan *pre-processing* dan penerapan fitur statistik, dapat dilihat pada tabel 4.9.

Dari data diatas diambil 2 sampel yang akan dilakukan pengemplementasian ke dalam KNN dengan fitur statik dimana A adalah data pertama, dan B data kedua.

X=[0.2819338021058919,0.12869061082851369,0.15734335724205817,0.274749217672005,0.1710128618504541,0.1710128618504541,0.0975743060505912,0.23434588467305734,0.16987936897816722,0.16835968193481882,0.4148607357038051,0.20787868066406873,0.23482592175425965,0.12090898020822247,0.24852039817620059,0.26709006646175654,0.23110304272441426,0.25589809952614856,0.18302934545847718,0.2302121113097295]

Y=[0.12934614939521338,0.1623046746330956,0.08021222496312963,0.296261732958
23455,0.1938419178257319,0.335855267464546,0.2994560083867521,0.131869434898
6634,0.15060988516745544,0.16994996870301832,0.23745302711370572,0.273332931
82818097,0.21673035401320456,0.12397516204029022,0.09366455006729327,0.11450
971114688897,0.335855267464546,0.335855267464546,0.335855267464546,0.3358552
67464546]

Selanjutnya menggunakan rumus *Euclidean Distance*: $\mathrm{d}(\mathrm{X},\mathrm{Y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$ berikut penerapannya.

(0.2819338021058919 - 0.12934614939521338)^2 = 0.023311376657769477

(0.12896061082851369 - 0.1623046746330956) ^{2} = 0.001125522436002205

(0.15734335724205817 - 0.08021222496312963) ^{2} = 0.005948047843040882

(0.274749217672005 - 0.29262173295823455) ^{2} = 0.0004627826472773861 6

(0.1710128618504541 - 0.1938419178257319) ^{2} = 0.0005326862599280093

(0.1710128618504541 - 0.33585267464546) ^{2} = 0.0272429132494693

(0.0975743060505912 - 0.2994560083867521) ^{2} = 0.040793678073208106

(0.2344588467305734 - 0.1318694348986634) ^{2} = 0.010511994626666817

(0.16987936897816722 - 0.15060988516745544) ^{2} = 0.0003699312592576146 4

(0.16835968193481882 - 0.1694996870301832) ^{2} = 2.5297021857845587e-06

(0.4148607357038051 - 0.23745302711370572) ^{2} = 0.03157900912568367

(0.20787868066406873 - 0.27333293182818097) ^{2} = 0.004264042693596114

(0.23482592175425965 - 0.2167035401320456) ^{2} = 0.0003267496667028352

(0.1209089802282247 - 0.12397516204029022) ^{2} = 9.39217192001938e-06

(0.24852039817620059 - 0.09366455006729327) ^{2} = 0.024022526665769527

(0.26709006646176554 - 0.11450971114688897) ^{2} = 0.023000968962646743

(0.23110304272441426 - 0.33585267464546) ^{2} = 0.010980599936637672

(0.255899392614856 - 0.33585267464546) ^{2} = 0.006382959237771679

(0.1830923454584778 - 0.33585267464546) ^{2} = 0.02303407291715316

(0.2302121113097295 - 0.33585267464546) ^{2} = 0.01115558811495254

Kemudian dijumlahkan seluruh fitur yang ada.

∑=0.023311376657769477+0.001125522436002205+0.005948047843040882+0.000462
78264727738616+0.0005326862599280093+0.02724292132494693+0.04079367807320
8106+0.010511994626666817+0.0003699312592576146 4+2.5297021857845587e−06+0.
03157900912568367+0.00426404269359611 4+0.0003267496667028352+9.3921719200
19338e−06+0.024022526665769527+0.023300968962646743+0.010980599936637672+
0.006382595237771679+0.023303407291715316+0.01115558811495254

Selanjutnya dilakukan proses akar dari nilai yang telah dijumlahkan.

$$d(A,B) = \sqrt{0.2456244718165509} \approx 0.4956$$

Hasil yang didapatkan diatas merupakan sampel dalam pengimplementasian KNN secara manual pada data yang dimiliki khususnya pada fitur statistik, namun untuk fiturnya yang diambil hanyalah yang memiliki nilai selebihnya tidak diambil.

Untuk penerapan pada kodingan lebih jelas penerapan dalam kodingan dapat dilihat pada gambar 4.20 berikut.

```
from sklearn.neighbors import KNeighborsClassifier

n_neighbors_knn = 3
text_algorithm_knn = KNeighborsClassifier(n_neighbors=n_neighbors_knn)
model_knn = text_algorithm_knn.fit(x_train_tfidf, y_train)
predicted_knn = model_knn.predict(x_test_tfidf)
```

Gambar 4.20 Impelementasi KNN Dengan Fitur Statistik.

Pada gambar 4.20 dilakukan penggunaan library KNN dimana digunakan untuk penerapan algoritma. Selanjutnya diteruskan dengan penentuan nilai k-nya atau nilai tetangganya. Disini K yang digunakan yaitu 3. Dimana dilanjutkan dengan pembuatan variabel untuk menyimpan algoritma penyelesaian KNN dengan nilai K-nya yang sudah ditentukan sebelumnya. Seterusnya dilakukan pelatihan dengan menyimpan pada variabel model_knn untuk modelnya sendiri. Dan dilakukan predicted_knn untuk menguji model tersebut dengan data uji.

2. Fitur linguistik dengan KNN

Pada penerapan fitur linguistik dengan algoritma KNN, yang dimana hasil dari fitur linguistik yakni panjang kalimat, jumlah kata dengan huruf besar, jumlah *cue-phrase* promo dan jumlah *cue-phrase* penipuan. Sebelumnya digunakan sebagai fitur dalam penerapan pada KNN, Berikut 2 data yang diambil *dataset* yang sudah melakukan precprocessing dan penerapan fitur statistik, dapat dilihat pada tabel 4.10.

Dari data diatas diambil 3 sampel yang akan dilakukan pengimplementasian ke dalam KNN dengan fitur linguistik.

Hitung jarak dari Kalimat 1 ke Kalimat 2:

$$d_{(1,2)} = \sqrt{(17-11)^2 + (3-1)^2 + (1-0)^2 + (0-0)^2}$$

$$d_{(1,2)} = \sqrt{6^2 + 2^2 + 1^2 + 0^2}$$

$$d_{(1,2)} = \sqrt{36 + 4 + 1 + 0}$$

$$d_{(1,2)} = \sqrt{41}$$

$$d_{(1,2)} = 6{,}4$$

Hitung jarak dari Kalimat 1 ke Kalimat 3:

$$d_{(1,3)} = \sqrt{(17-18)^2 + (3-4)^2 + (1-1)^2 + (0-1)^2}$$

$$d_{(1,3)} = \sqrt{(-1)^2 + (-1)^2 + 0^2 + (-1)^2}$$

$$d_{(1,3)} = \sqrt{1 + 1 + 0 + 1}$$

$$d_{(1,3)} = \sqrt{3}$$

$$d_{(1,3)} = 1{,}73$$

Hasil yang didapatkan diatas merupakan sampel dalam pengimplementasian KNN secara manual pada data yang dimiliki, namun untuk fiturnya yang diambil hanyalah yang memiliki nilai selebihnya tidak diambil.

Lebih jelas dapat dilihat pada gambar 4.21 berikut.

```
from sklearn.neighbors import KNeighborsClassifier

n_neighbors_knn = 3
model_knn = KNeighborsClassifier(n_neighbors=n_neighbors_knn).fit(x_train, y_train)
predicted_knn = model_knn.predict(x_test)
```

Gambar 4.21 Impelementasi KNN Dengan Fitur Linguistik.

Pada gambar 4.21 dilakukan penggunaan library KNN dimana digunakan untuk penerapan algoritma. Selanjutnya diteruskan dengan penentuan nilai k-nya atau nilai tetangganya. Disini K yang digunakan yaitu 3. Dimana dilanjutkan dengan pembuatan variabel untuk menyimpan algoritma penyelesaian KNN dengan nilai K-nya yang sudah ditentukan sebelumnya. Seterusnya dilakukan pelatihan dengan menyimpan pada variabel model_knn untuk modelnya sendiri. Dan dilakukan predicted_knn untuk menguji model tersebut dengan data uji.

3. Fitur statistik dengan NN

Pada penerapan fitur statistik dengan algoritma NN, yang dimana hasil dari representasi tf-idf sebelumnya digunakan sebagai fitur dalam penerapan pada NN, Berikut implementasi dalam bentuk manual ke dalam NN.

Inisialisasi bobot:

Bobot *input* ke *hidden layer* (W1):

$$W1 = \begin{pmatrix} 0.1 & -0.2 & 0.5 & -0.7 \\ 0.3 & 0.4 & -0.6 & 0.8 \end{pmatrix}$$

Bias *hidden layer* (b1):

$$b1 = \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$$

Bobot *hidden* ke *output layer* (W2):

$$W2 = \begin{pmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \\ 0.5 & -0.3 \end{pmatrix}$$

Bias *output layer* (b2):

$$b2 = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}$$

Feedforward:

Perhitungan *hidden layer*:

$$Z1 = \left(\begin{pmatrix} 0.1 & -0.2 & 0.5 & -0.7 \\ 0.3 & 0.4 & -0.6 & 0.8 \end{pmatrix} \cdot \begin{pmatrix} 17 \\ 3 \\ 1 \\ 0 \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$$

$$Z1 = \left(\begin{pmatrix} 0.1 \cdot 17 + (-0.2) \cdot 3 + 0.5 \cdot 1 + (-0.7) \cdot 0 \\ 0.3 \cdot 17 + 0.4 \cdot 3 + (-0.6) \cdot 1 + 0.8 \cdot 0 \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$$

$$Z1 = \left(\begin{pmatrix} 1.7 - 0.6 + 0.5 + 0 \\ 5.1 + 1.2 - 0.6 + 0 \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$$

$$Z1 = \left(\begin{pmatrix} 1.6 \\ 5.7 \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$$

$$Z1 = \begin{pmatrix} 1.7 \\ 5.9 \end{pmatrix}$$

Fungsi aktivasi ReLU:3

$$\mathrm{ReLU}(x) = \max(0, x)$$

$$\text{output\_tersembunyi} = \text{ReLU}(Z1) = \begin{pmatrix} \max(0, 1.7) \\ \max(0, 5.9) \end{pmatrix}$$

$$\text{output\_tersembunyi} = \begin{pmatrix} 1.7 \\ 5.9 \end{pmatrix}$$

Perhitungan *output layer:*

$$Z2 = W2 \cdot \text{output\_tersembunyi} + b2$$

$$Z2 = \left(\begin{pmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \\ 0.5 & -0.3 \end{pmatrix} \cdot \begin{pmatrix} 1.7 \\ 5.9 \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}$$

$$Z2 = \left(\begin{pmatrix} 0.1 \cdot 1.7 + 0.3 \cdot 5.9 \\ 0.2 \cdot 1.7 + 0.4 \cdot 5.9 \\ 0.5 \cdot 1.7 + (-0.3) \cdot 5.9 \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}$$

$$Z2 = \left(\begin{pmatrix} 0.17 + 1.77 \\ 0.34 + 2.36 \\ 0.85 - 1.77 \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}$$

$$Z2 = \left(\begin{pmatrix} 1.94 \\ 2.70 \\ -0.92 \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}$$

$$Z2 = \begin{pmatrix} 2.04 \\ 2.90 \\ -0.62 \end{pmatrix}$$

Fungsi aktivasi *softmax:*

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Hitung eksponen dari nilai Z $_{2:}$

$$e^{2.04} \approx 7.69$$
$$e^{2.90} \approx 18.18$$
$$e^{-0.62} \approx 0.54$$

Maka,

$$\text{softmax}(Z2) = \left[\frac{7.69}{7.69+18.18+0.54}, \frac{18.18}{7.69+18.18+0.54}, \frac{0.54}{7.69+18.18+0.54} \right]$$

$$\text{softmax}(Z2) \approx \left[\frac{7.69}{26.41}, \frac{18.18}{26.41}, \frac{0.54}{26.41}\right]$$

$$\text{softmax}(Z2) \approx [0.291, 0.689, 0.020]$$

Prediksi Kelas:

Hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; promo: 0.324, normal: 0.375, dan penipuan: 0.301

Dalam penerapan ke dalam kodingan lebih jelas dapat dilihat pada gambar 4.22 berikut.

```
from sklearn.neural_network import MLPClassifier

text_algorithm_nn = MLPClassifier(hidden_layer_sizes=(100,), solver="adam",
activation="relu")
model_nn = text_algorithm_nn.fit(x_train_tfidf, y_train)
predicted_nn = model_nn.predict(x_test_tfidf)
```

Gambar 4.22 Impelementasi NN Dengan Fitur Statistik.

Pada gambar 4.22 dilakukan penggunaan library NN (*MLPClassifier*) dimana digunakan untuk penerapan algoritma. Selanjutnya diteruskan dengan penentuan nilai *hidden layer*-nya. Disini *hidden layer* yang digunakan yaitu 100 dengan solver adam dan aktivasi relu. Setelah itu dilakukan pelatihan dengan data train untuk menyimpan model pada variabel *model_nn*. Selanjutnya dilakukan *predicted_nn* untuk menguji model tersebut dengan data uji.

4. Fitur linguistik dengan NN

Pada penerapan fitur linguistik dengan algoritma NN, yang dimana hasil dari fitur linguistik sebelumnya digunakan sebagai fitur dalam penerapan pada NN. Berikut implementasi dalam bentuk manual ke dalam NN.

Inisialisasi bobot:

Bobot *input* ke *hidden layer* (W1):

$$W1 = \begin{pmatrix} 0.1 & -0.2 \\ 0.3 & 0.4 \\ 0.5 & -0.6 \\ -0.7 & 0.8 \end{pmatrix}$$

Bias *hidden layer* (b1):

$$b1 = \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$$

Bobot *hidden* ke *output layer* (W2):

$$W2 = \begin{pmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \\ 0.5 & -0.3 \end{pmatrix}$$

Bias *output layer* (b2):

$$b2 = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}$$

Feedforward:

Perhitungan *hidden layer*:

$$Z1 = \left(\begin{pmatrix} 0.1 & -0.2 & 0.5 & -0.7 \\ 0.3 & 0.4 & -0.6 & 0.8 \end{pmatrix} \cdot \begin{pmatrix} 0.2819338021058919 \\ 0.12869061082851369 \\ 0.15734335724205817 \\ 0.274749217672005 \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$$

$$Z1 = \left(\begin{pmatrix} -0.11119751570448797 \\ 0.26144974475554214 \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$$

$$Z1 = \begin{pmatrix} -0.011197515704487967 \\ 0.46144974475554214 \end{pmatrix}$$

Fungsi aktivasi ReLU:

$$\mathrm{ReLU}(x) = \max(0, x)$$

$$\text{output\_tersembunyi} = \text{ReLU}(Z1) = \begin{pmatrix} \max(0, -0.011197515704487967) \\ \max(0, 0.46144974475554214) \end{pmatrix}$$

$$\text{output\_tersembunyi} = \begin{pmatrix} 0 \\ 0.46144974475554214 \end{pmatrix}$$

Perhitungan *output layer:*

$$Z2 = W2 \cdot \text{output\_tersembunyi} + b2$$

$$Z2 = \left(\begin{pmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \\ 0.5 & -0.3 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0.46144974475554214 \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}$$

$$Z2 = \left(\begin{pmatrix} 0.1 \cdot 0 + 0.3 \cdot 0.46144974475554214 \\ 0.2 \cdot 0 + 0.4 \cdot 0.46144974475554214 \\ 0.5 \cdot 0 + (-0.3) \cdot 0.46144974475554214 \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}$$

$$Z2 = \left(\begin{pmatrix} 0 + 0.13843492342666264 \\ 0 + 0.18457989790221686 \\ 0 + (-0.13843492342666264) \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}$$

$$Z2 = \left(\begin{pmatrix} 0.13843492342666264 \\ 0.18457989790221686 \\ -0.13843492342666264 \end{pmatrix} \right) + \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}$$

$$Z2 = \begin{pmatrix} 0.23843492342666264 \\ 0.38457989790221686 \\ 0.16156507657333736 \end{pmatrix}$$

Fungsi aktivasi *softmax:*

$$e^{0.23843492342666264} \approx 1.269$$
$$e^{0.38457989790221686} \approx 1.469$$
$$e^{0.16156507657333736} \approx 1.176$$

Maka,

$$\text{softmax}(Z2) = \left[\frac{1.269}{1.269+1.469+1.176}, \frac{1.469}{1.269+1.469+1.176}, \frac{1.176}{1.269+1.469+1.176} \right]$$

$$\text{softmax}(Z2) = \left[\frac{1.269}{3.914}, \frac{1.469}{3.914}, \frac{1.176}{3.914} \right]$$

$$\text{softmax}(Z2) \approx [0.324, 0.375, 0.301]$$

Prediksi Kelas:

Hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; promo: 0.291, normal: 0.689, dan penipuan: 0.020. Dalam penerapan ke dalam kodingan lebih jelas dapat dilihat pada gambar 4.23 berikut.

```
from sklearn.neural_network import MLPClassifier

model_nn = MLPClassifier(hidden_layer_sizes=(100,), solver='adam', activation='relu',
max_iter=200).fit(x_train, y_train)
predicted_nn = model_nn.predict(x_test)
```

Gambar 4.23 Impelementasi NN Dengan Fitur Linguistik.

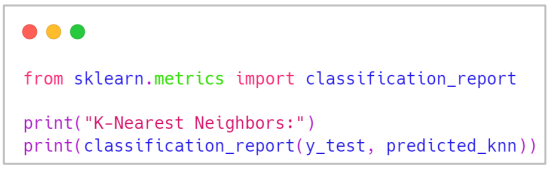
Pada gambar 4.23 dilakukan penggunaan *library* NN (*MLPClassifier*) dimana digunakan untuk penerapan algoritma. Selanjutnya diteruskan dengan penentuan nilai *hidden layer*-nya. Disini *hidden layer* yang digunakan yaitu 100 dengan solver adam dan aktivasi relu. Setelah itu dilakukan pelatihan dengan data train untuk menyimpan model pada variabel *model_nn*. Selanjutnya dilakukan *predicted_nn* untuk menguji model tersebut dengan data uji.

4.7. Evaluasi Model

Pada tahapan ini, masuk pada proses implementasi algoritma KNN dan NN menggunakan data uji untuk mengevaluasi model. Berikut tahapan evaluasi model:

1. Model fitur statistik dengan KNN

Setelah melalui proses pelatihan model yang telah dibuat. Selanjutnya dilakukan pengevaluasian model dengan data *actual* atau y_test (label data *test*) dengan prediksinya. Lebih jelas dapat dilihat pada gambar 4.24 berikut.



```
from sklearn.metrics import classification_report

print("K-Nearest Neighbors:")
print(classification_report(y_test, predicted_knn))
```

Gambar 4.24 Classification Report KNN Dengan Fitur Statistik.

Berdasarkan gambar 4.24 akan mendapatkan hasil evaluasi sebagai berikut, dapat dilihat pada tabel 4.11.

Tabel 4.11 Hasil Evaluasi KNN Dengan Fitur Statistik.

Kelas	*Precision*	*Precision*	*F1-score*
Normal	0.81	0.94	0.87
Penipuan	0.94	0.87	0.90
Promo	0.99	0.97	0.98
Akurasi	0.95		

Berdasarkan hasil dari tabel 4.11 *classification report* dari model *K-Nearest Neighbors* (KNN) dengan fitur statistik, dapat dilihat bahwa terdapat tiga kelas yang dievaluasi, diidentifikasi dengan label 0, 1, dan 2.

Untuk kelas 0, *precision* sebesar 0.81 menunjukkan bahwa dari total prediksi yang dilakukan untuk kelas 0, sekitar 81% benar-benar merupakan kelas 0. *Precision* sebesar 0.94 mengindikasikan bahwa dari total data yang sebenarnya adalah kelas 0, model berhasil mengidentifikasi sekitar 94% dari data tersebut sebagai kelas 0. F1-*score*, yang menggabungkan presisi dan *precision*, memiliki nilai 0.87 untuk kelas 0.

Untuk kelas 1, presisi sebesar 0.94 menandakan bahwa sekitar 94% dari prediksi yang diklasifikasikan sebagai kelas 1 memang benar-benar merupakan kelas 1. *Precision* sebesar 0.87 menunjukkan bahwa sekitar 87% dari data kelas 1 berhasil diidentifikasi dengan benar oleh model. F1-*score* untuk kelas 1 adalah 0.90.

Sementara untuk kelas 2, presisi sebesar 0.99 menunjukkan tingkat keakuratan yang

sangat tinggi dalam mengklasifikasikan data ke kelas 2. *Precision* sebesar 0.97 mengindikasikan bahwa model berhasil mengidentifikasi sebagian besar data yang sebenarnya kelas 2. F1-*score* untuk kelas 2 adalah 0.98.

Dari sisi akurasi (*accuracy*), model KNN dengan fitur statistik mencapai nilai 0.95, yang menunjukkan bahwa sekitar 95% dari seluruh data yang dievaluasi telah diklasifikasikan dengan benar oleh model.

Secara keseluruhan, hasil ini menunjukkan bahwa model KNN dengan fitur statistik memiliki kinerja yang baik dalam mengklasifikasikan data ke dalam ketiga kelas yang dievaluasi, dengan presisi, *precision*, dan F1-*score* yang relatif tinggi untuk setiap kelasnya. Akurasi model juga cukup tinggi, mencapai 95%.

2. Model fitur linguistik dengan KNN

Setelah mendapatkan hasil prediksi sebelumnya kemudian dilakukan pengevaluasian model dengan data *actual* atau y_test (label data test) dengan prediksinya. Lebih jelas dapat dilihat pada gambar 4.25 berikut.

```
from sklearn.metrics import classification_report

print("K-Nearest Neighbors:")
print(classification_report(y_test, predicted_knn))
```

Gambar 4.25 Classification Report KNN Dengan Fitur Linguistik.

Berdasarkan gambar 4.25 akan mendapatkan hasil evaluasi sebagai berikut, dapat dilihat pada tabel 4.12.

Tabel 4.12 Hasil Evaluasi KNN Dengan Fitur Linguistik.

Kelas	*Precision*	*Precision*	*F1-score*
Normal	0.73	0.82	0.77
Penipuan	0.53	0.37	0.43

Promo	0.89	0.92	0.91
Akurasi	0.82		

Berdasarkan hasil evaluasi *K-Nearest Neighbors* (KNN) dengan fitur linguistik, yang terdapat dalam Tabel 4.12, dapat dilihat bahwa terdapat tiga kelas yang dievaluasi, diidentifikasi dengan label 0, 1, dan 2.

Untuk kelas 0, presisi sebesar 0.73 menunjukkan bahwa dari total prediksi yang diklasifikasikan sebagai kelas 0, sekitar 73% benar-benar merupakan kelas 0. *Precision* sebesar 0.82 menunjukkan bahwa dari total data yang sebenarnya adalah kelas 0, model berhasil mengidentifikasi sekitar 82% dari data tersebut sebagai kelas 0. F1-*score* untuk kelas 0 adalah 0.77.

Kelas 1 memiliki presisi sebesar 0.53, yang menandakan bahwa sekitar 53% dari prediksi yang diklasifikasikan sebagai kelas 1 memang benar-benar merupakan kelas 1. *Precision* sebesar 0.37 mengindikasikan bahwa sekitar 37% dari data kelas 1 berhasil diidentifikasi dengan benar oleh model. F1-*score* untuk kelas 1 adalah 0.43.

Kelas 2 memiliki presisi sebesar 0.89, menunjukkan tingkat keakuratan yang tinggi dalam mengklasifikasikan data ke kelas 2. *Precision* sebesar 0.92 mengindikasikan bahwa model berhasil mengidentifikasi sebagian besar data yang sebenarnya kelas 2. F1-*score* untuk kelas 2 adalah 0.91.

Akurasi (*accuracy*) model KNN dengan fitur linguistik adalah 0.82, yang menunjukkan bahwa sekitar 82% dari seluruh data yang dievaluasi telah diklasifikasikan dengan benar oleh model.

Secara keseluruhan, hasil ini menunjukkan bahwa model KNN dengan fitur linguistik memiliki kinerja yang bervariasi untuk setiap kelasnya. Kelas 2 menunjukkan kinerja yang

baik dengan presisi, *precision*, dan F1-*score* yang tinggi, sementara kelas 0 juga memiliki kinerja yang relatif baik. Namun, kelas 1 menunjukkan kinerja yang lebih rendah, dengan presisi, *precision*, dan F1-*score* yang lebih rendah. Akurasi model juga sedikit lebih rendah dibandingkan dengan model KNN sebelumnya dengan fitur statistik. Hal ini menunjukkan bahwa penggunaan fitur linguistik dalam model KNN mungkin tidak seefektif fitur statistik dalam kasus ini.

3. Model fitur statistik dengan NN

Setelah mendapatkan hasil prediksi sebelumnya kemudia dilakukan pengevaluasian model dengan data actual atau y_test (label data test) dengan prediksinya. Lebih jelas dapat dilihat pada gambar 4.26 berikut.

```
from sklearn.metrics import classification_report

print("Neural Network:")
print(classification_report(y_test, predicted_nn))
```

Gambar 4.26 Classification Report NN Dengan Fitur Statistik.

Berdasarkan gambar 4.26 akan mendapatkan hasil evaluasi sebagai berikut, dapat dilihat pada tabel 4.13.

Tabel 4.13 Hasil evaluasi NN dengan fitur statistik.

Kelas	*Precision*	*Precision*	*F1-score*
0	0,98	0,95	0,96
1	0,95	0,92	0,93
2	0,98	1,00	0,99
Akurasi	0,98		

Berdasarkan hasil evaluasi *Neural Network* (NN) dengan fitur statistik yang terdapat dalam Tabel 4.13, terlihat bahwa terdapat tiga kelas yang dievaluasi, diidentifikasi dengan

label 0, 1, dan 2.

Untuk kelas 0, presisi sebesar 0.98 menunjukkan bahwa dari total prediksi yang diklasifikasikan sebagai kelas 0, sekitar 98% benar-benar merupakan kelas 0. *Precision* sebesar 0.95 mengindikasikan bahwa dari total data yang sebenarnya adalah kelas 0, model berhasil mengidentifikasi sekitar 95% dari data tersebut sebagai kelas 0. F1-*score* untuk kelas 0 adalah 0.96.

Kelas 1 memiliki presisi sebesar 0.95, yang menandakan bahwa sekitar 95% dari prediksi yang diklasifikasikan sebagai kelas 1 memang benar-benar merupakan kelas 1. *Precision* sebesar 0.92 mengindikasikan bahwa sekitar 92% dari data kelas 1 berhasil diidentifikasi dengan benar oleh model. F1-*score* untuk kelas 1 adalah 0.93.

Kelas 2 memiliki presisi sebesar 0.98, menunjukkan tingkat keakuratan yang tinggi dalam mengklasifikasikan data ke kelas 2. *Precision* sebesar 1.00 mengindikasikan bahwa model berhasil mengidentifikasi semua data yang sebenarnya kelas 2. F1-*score* untuk kelas 2 adalah 0.99.

Akurasi (*accuracy*) model NN dengan fitur statistik adalah 0.98, yang menunjukkan bahwa sekitar 98% dari seluruh data yang dievaluasi telah diklasifikasikan dengan benar oleh model.

Secara keseluruhan, hasil ini menunjukkan bahwa model NN dengan fitur statistik memiliki kinerja yang sangat baik untuk setiap kelasnya. Presisi, *precision*, dan F1-*score* yang tinggi menunjukkan kemampuan model untuk mengklasifikasikan data dengan akurat ke dalam setiap kelas. Akurasi model juga sangat tinggi, mencapai 98%, menunjukkan kemampuan model dalam memprediksi label kelas dengan sangat baik.

4. Model fitur linguistik dengan NN

Setelah mendapatkan hasil prediksi sebelumnya kemudia dilakukan pengevaluasian model dengan data actual atau y_test (label data test) dengan prediksinya. Lebih jelas dapat dilihat pada gambar 4.27 berikut.

```
from sklearn.metrics import classification_report

print("Neural Network:")
print(classification_report(y_test, predicted_nn))
```

Gambar 4.27 Classification Report NN Dengan Fitur Linguistik.

Berdasarkan gambar 4.27 akan mendapatkan hasil evaluasi sebagai berikut, dapat dilihat pada tabel 4.14.

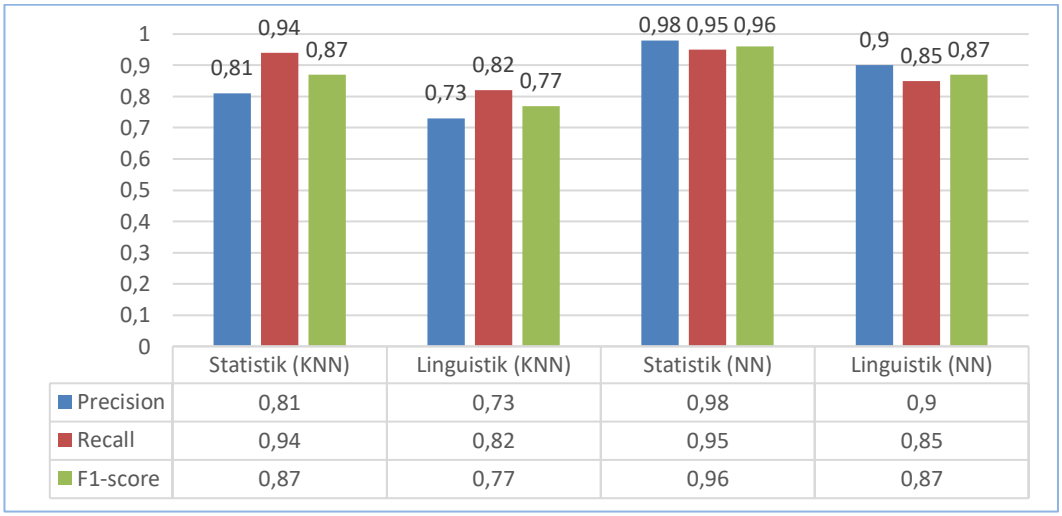
Tabel 4.14 Hasil Evaluasi NN Dengan Fitur Linguistik.

Kelas	*Precision*	*Precision*	*F1-score*
0	0.85	0.85	0.85
1	0.78	0.28	0.41
2	0.85	0.97	0.91
Akurasi	0.85		

Berdasarkan tabel 4.14 diperoleh nilai kelas 0 yang memiliki presisi sangat tinggi (0.90) dan *precision* yang cukup tinggi (0.85), menghasilkan F1-*score* yang tinggi (0.87). Ini menunjukkan bahwa model sangat kompeten dalam mengidentifikasi kelas 0 dengan tepat dan jarang salah memprediksi kelas lain sebagai kelas 0. kelas 1 memiliki presisi rendah (0.70) dan *precision* yang sangat rendah (0.31), menghasilkan F1-*score* yang rendah (0.43). Ini menandakan bahwa model sering salah memprediksi kelas lain sebagai kelas 1 dan gagal mengidentifikasi sebagian besar kasus positif aktual kelas 1. kelas 2 menunjukkan kinerja yang sangat baik dengan presisi tinggi (0.85) dan *precision* yang sangat tinggi (0.97), menghasilkan F1-*score* yang sangat tinggi (0.91). Ini menunjukkan bahwa model sangat

efektif dalam mengidentifikasi kelas 2, dengan sedikit kesalahan positif dan berhasil mengidentifikasi hampir semua kasus positif aktual. Akurasi keseluruhan dari model adalah 0.85, menandakan bahwa secara keseluruhan, 85% dari prediksi yang dibuat oleh model adalah benar.

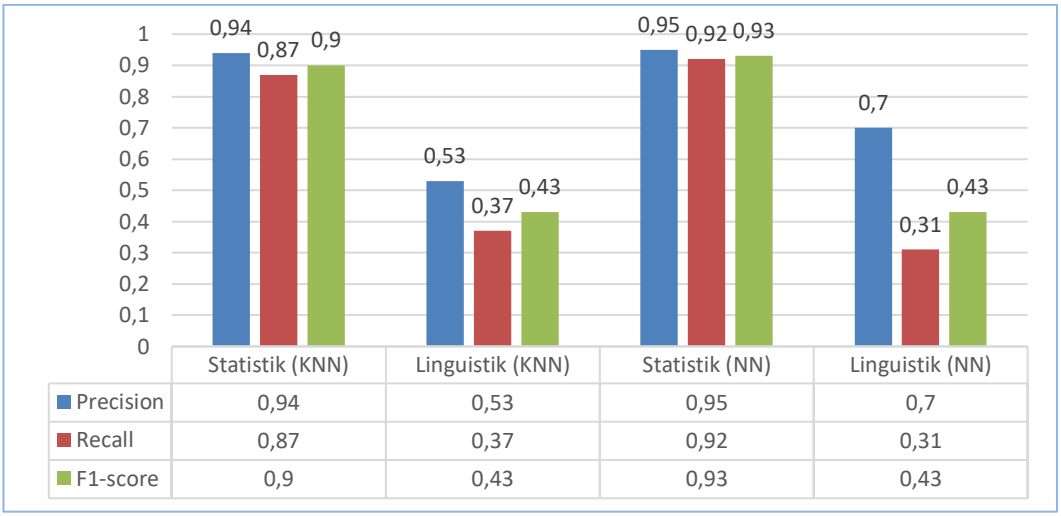
Berdasarkan dari ke empat model tersebut, maka akan diperoleh data perbandingan pada gambar 4.28 untuk pesan normal, 4.29 untuk pesan penipuan dan 4.30 untuk pesan promo. Berikut gambar 4.28.



Gambar 4.28 Pesan Normal.

Dari gambar 4.28, dapat dilihat bahwa model *Neural Network* (NN) dengan fitur statistik memiliki kinerja yang paling baik dalam mengklasifikasikan pesan normal, dengan presisi sebesar 0.98, *precision* sebesar 0.95, dan F1-*score* sebesar 0.96. Sementara itu, model *K-Nearest Neighbors* (KNN) dengan fitur statistik menunjukkan kinerja yang cukup baik dengan presisi 0.81, *precision* 0.94, dan F1-*score* 0.87. Namun, model NN dengan fitur linguistik juga menunjukkan kinerja yang baik dengan presisi 0.90, *precision* 0.85, dan F1-*score* 0.87, sedangkan model KNN dengan fitur linguistik memiliki kinerja yang lebih rendah dengan presisi 0.73, *precision* 0.82, dan F1-*score* 0.77.

Secara keseluruhan, model NN cenderung memberikan kinerja yang lebih baik daripada model KNN dalam mengklasifikasikan pesan normal, terutama ketika menggunakan fitur statistik. Namun, fitur linguistik juga memberikan hasil yang dapat diterima, terutama ketika digunakan dalam model NN. Berikut gambar 4.29 untuk pesan penipuan.

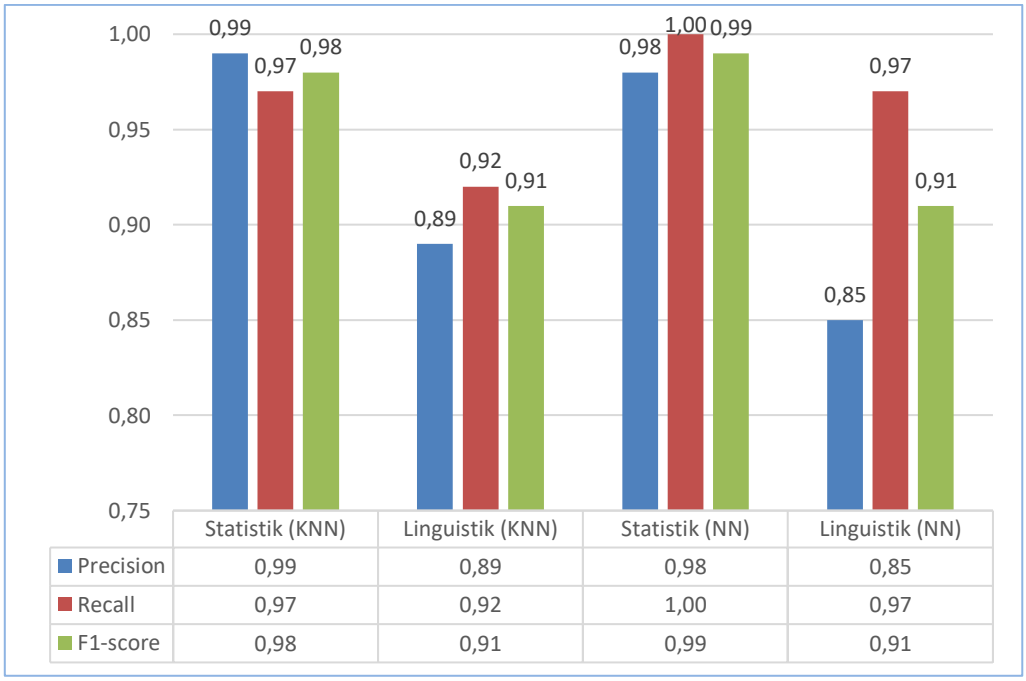


Gambar 4.29 Pesan Penipuan.

Dari gambar 4.29, dapat dilihat bahwa model *Neural Network* (NN) dengan fitur statistik memiliki kinerja yang paling baik dalam mengklasifikasikan pesan penipuan, dengan presisi sebesar 0.95, *precision* sebesar 0.92, dan F1-*score* sebesar 0.93. Sementara itu, model *K-Nearest Neighbors* (KNN) dengan fitur statistik juga menunjukkan kinerja yang baik dengan presisi 0.94, *precision* 0.87, dan F1-*score* 0.90. Namun, model NN dengan fitur linguistik juga menunjukkan kinerja yang baik dengan presisi 0.70, *precision* 0.31, dan F1-*score* 0.43, sedangkan model KNN dengan fitur linguistik memiliki kinerja yang lebih rendah dengan presisi 0.53, *precision* 0.37, dan F1-*score* 0.43.

Secara keseluruhan, model NN cenderung memberikan kinerja yang lebih baik daripada model KNN dalam mengklasifikasikan pesan penipuan, terutama ketika

menggunakan fitur statistik. Namun, model NN dengan fitur linguistik juga memberikan hasil yang dapat diterima, meskipun kinerja yang lebih rendah daripada model KNN dengan fitur statistik.Berikut gambar 4.30 untuk pesan promo.

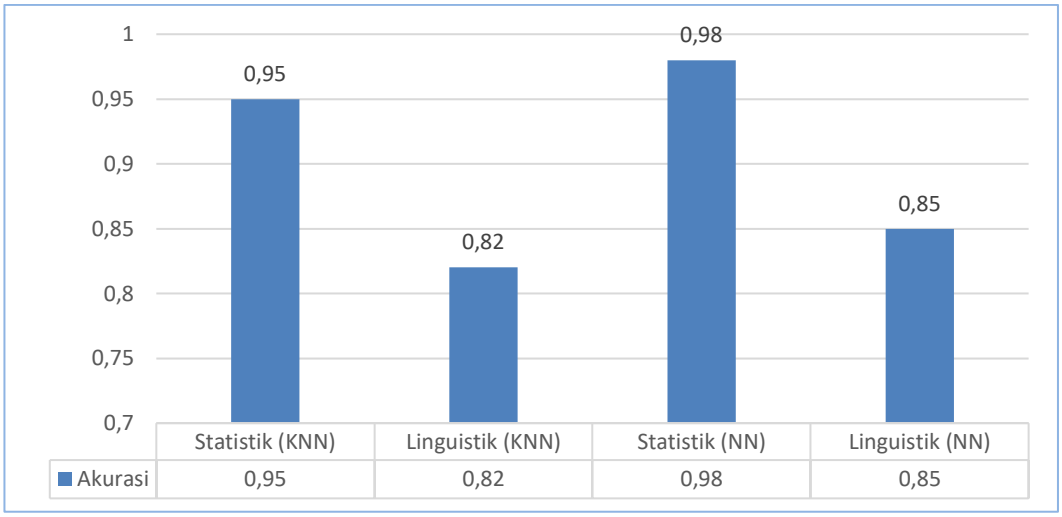


Gambar 4.30 Pesan Promo.

Dari gambar 4.30, dapat dilihat bahwa model *Neural Network* (NN) dengan fitur statistik memiliki kinerja yang paling baik dalam mengklasifikasikan pesan promo, dengan presisi sebesar 0.98, *precision* sebesar 1.00, dan F1-*score* sebesar 0.99. Sementara itu, model *K-Nearest Neighbors* (KNN) dengan fitur statistik juga menunjukkan kinerja yang sangat baik dengan presisi 0.99, *precision* 0.97, dan F1-*score* 0.98. Namun, model NN dengan fitur linguistik juga menunjukkan kinerja yang baik dengan presisi 0.85, *precision* 0.97, dan F1-*score* 0.91, sedangkan model KNN dengan fitur linguistik memiliki kinerja yang sedikit lebih rendah dengan presisi 0.89, *precision* 0.92, dan F1-*score* 0.91.

Secara keseluruhan, model NN cenderung memberikan kinerja yang lebih baik daripada model KNN dalam mengklasifikasikan pesan promo, terutama ketika

menggunakan fitur statistik. Namun, model NN dengan fitur linguistik juga memberikan hasil yang dapat diterima, meskipun kinerja yang lebih rendah daripada model KNN dengan fitur statistik. Untuk akurasinya dapat dilihat pada gambar 4.31 berikut.



Gambar 4.31 Akurasi.

Dari gambar 4.31, dapat dilihat bahwa model *Neural Network* (NN) dengan fitur statistik memiliki akurasi tertinggi, yaitu 0.98. Model *K-Nearest Neighbors* (KNN) dengan fitur statistik juga menunjukkan akurasi yang tinggi, dengan nilai 0.95. Namun, model NN dengan fitur linguistik memiliki akurasi yang lebih rendah, yaitu 0.85. Sementara itu, model KNN dengan fitur linguistik memiliki akurasi yang paling rendah, yaitu 0.82.

Secara keseluruhan, model NN cenderung memberikan akurasi yang lebih tinggi daripada model KNN, terutama ketika menggunakan fitur statistik. Sedangkan penggunaan fitur linguistik cenderung menghasilkan akurasi yang lebih rendah untuk kedua jenis model, meskipun model NN masih menunjukkan akurasi yang lebih tinggi daripada model KNN.

Dari hasil evaluasi yang disajikan dalam tabel-tabel sebelumnya, terlihat bahwa model *Neural Network* (NN) cenderung memberikan performa yang lebih baik secara konsisten dibandingkan dengan model *K-Nearest Neighbors* (KNN), terutama ketika menggunakan

fitur statistik. Model NN memiliki akurasi yang tinggi di semua kelas, serta presisi, *precision*, dan F1-*score* yang baik untuk setiap kelas.

Secara khusus, model NN dengan fitur statistik menunjukkan performa yang sangat baik dengan akurasi 0.98 dan presisi, *precision*, serta F1-*score* yang tinggi untuk setiap kelas. Meskipun model NN dengan fitur linguistik juga memberikan hasil yang baik, namun performanya sedikit di bawah model NN dengan fitur statistik.

Dengan demikian, berdasarkan evaluasi yang telah dilakukan, dapat disimpulkan bahwa model *Neural Network* dengan fitur statistik adalah yang paling baik secara keseluruhan dalam mengklasifikasikan pesan menjadi kelas normal, penipuan, dan promo.

4.8. *Deploy* Model

Pada tahapan ini, dilihat model mana yang paling baik dari segi *precision*, *precision*, *accuracy* dan *f1-score*. Kemudian di *deploy* modelnya dengan *streamlit*. Berdasarkan hasil pada evaluasi model dapat dilihat bahwa model terbaik yaitu: model yang menggunakan fitur statistik pada algoritma *Neural Network.* Untuk melakukan *deploy* ke *streamlit* dapat dilihat pada gambar 4.32 berikut.

```
import pickle

# Menyimpan model Neural Network
with open('model_nn.pkl', 'wb') as file:
    pickle.dump(model_nn, file)

# Menyimpan TfidfVectorizer
with open('tfidf_vectorizer.pkl', 'wb') as file:
    pickle.dump(tfidf_vectorizer, file)
```

Gambar 4.32 Penggunaan Library *Pickle* Untuk Menyimpan Model.

Model yang telah disimpan selanjutnya akan di gunakan ke streamlit, untuk lebih jelas dapat dilihat pada gambar 4.33 berikut.

```
import streamlit as st
import pandas as pd
import pickle
import re
from nltk.corpus import stopwords
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

factory = StemmerFactory()
stemmer = factory.create_stemmer()

key_norm = pd.read_csv('key_norm.csv')

def casefolding(text):
    text = text.lower()
    text = re.sub(r'http[s]?:?://\S+|www\.\S+(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\\(\\),]|(?:%[0-
9a-fA-F][0-9a-fA-F]))+', ' ', text)
    text = re.sub(r'[-]?[0-9]+', ' ', text)
    text = re.sub(r'[+]?', '', text)
    text = re.sub(r'[^\w\s]', '', text)
    text = text.strip()
    return text

def text_normalize(text):
    text = ' '.join([key_norm[key_norm['singkat'] == word]['hasil'].values[0]
                     if (key_norm['singkat'] == word).any()
                     else word for word in text.split()])
    text = str.lower(text)
    return text

def remove_stop_word(text):
    stopwords_indo = stopwords.words('indonesian') + ['tsel', 'gb', 'rb', 'btw']
    clean_words = []
    text = text.split()
    for word in text:
        if word not in stopwords_indo:
            clean_words.append(word)
    return " ".join(clean_words)

def stemming(text):
    text = stemmer.stem(text)
    return text

def preprocess_text(text):
    text = casefolding(text)
    text = text_normalize(text)
    text = remove_stop_word(text)
    text = stemming(text)
    return text
```

Gambar 4.33 *Preprocessing* Teks Untuk Data Baru.

Sesuai dengan gambar 4.33 dapat dilihat terdapat proses *preprocessing teks* untuk mengubah teks atau pesan baru yang nantinya akan diklasifikasi, tidak lupa dengan penambahan library. Selanjutnya dilanjutkan pada tahapan implementasinya. Dapat dilihat pada gambar 4.34 berikut.

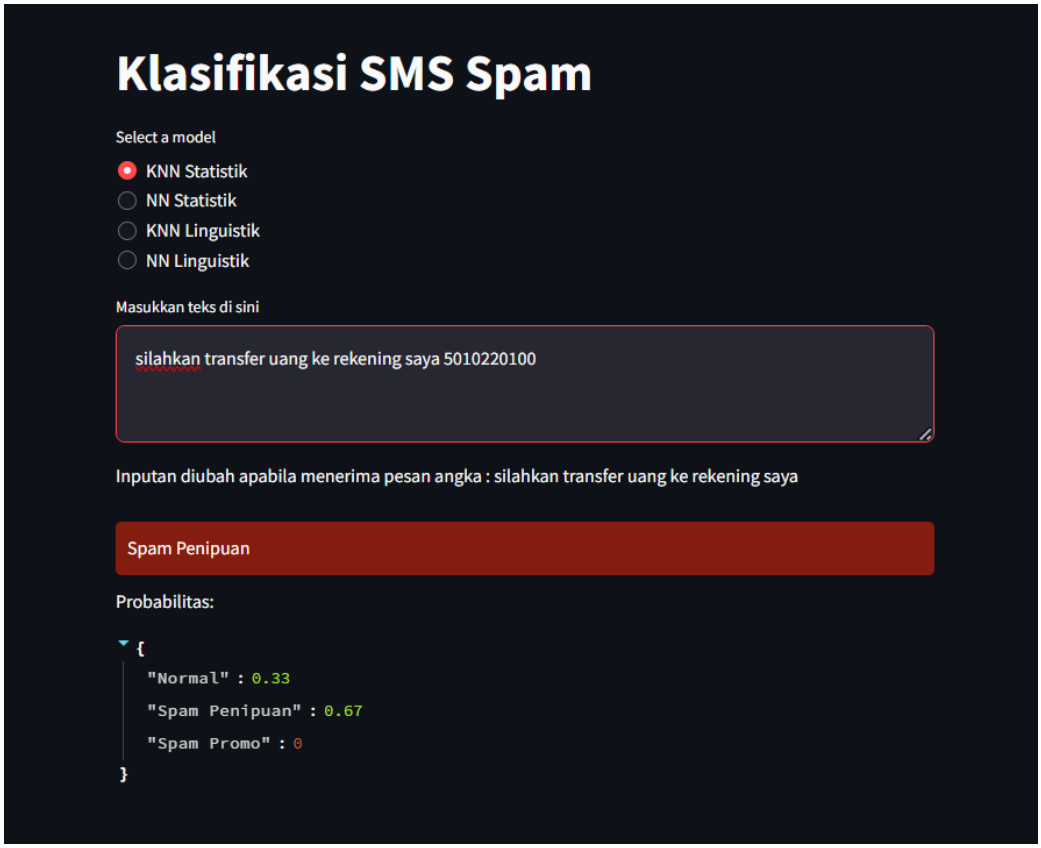
```python
# Memuat model dan TfidfVectorizer
with open('model_nn.pkl', 'rb') as model_file:
    model_nn = pickle.load(model_file)

with open('tfidf_vectorizer.pkl', 'rb') as vectorizer_file:
    tfidf_vectorizer = pickle.load(vectorizer_file)

# Membuat antarmuka pengguna dengan Streamlit
st.title('Klasifikasi SMS Spam Menggunakan Neural Network (Fitur Statistik)')

# Menerima input teks dari pengguna
user_input = st.text_input("Masukkan teks di sini")

if st.button('Prediksi'):
    # Preprocess input pengguna
    preprocessed_text = preprocess_text(user_input)
    # Vectorize teks
    vectorized_text = tfidf_vectorizer.transform([preprocessed_text])
    if prediction[0] == 1:
        detection = 'SMS Penipuan'
    elif prediction[0] == 2:
        detection = 'SMS Promo'
    else:
        detection = 'SMS Normal'

    st.success(detection)
```

Gambar 4.34 Implementasi *Streamlit.*

Berdasarkan gambar 4.34 dilakukan pembacaan model yang kemudian dilanjutkan dengan penambahan atribut seperti *title*, *text_input* untuk klasifikasi pesan baru, dan juga *button* sebagai pengeksekusi model yang akan diklasifikasi. Berikut tampilan dari *streamlit.* Dapat dilihat pada gambar 4.35 berikut.

Gambar 4.35 Tampilan *Streamlit.*

Jika dilakukan percobaan klasifikasi dengan contoh teks “silahkan transfer uang ke rekening saya 5010220100”, hasilnya akan seperti gambar 4.36 berikut.



Gambar 4.36 Hasil Klasifikasi Pada Tampilan *Streamlit*.

4.9. Pengujian *Black Box*

Pengujian *black Box* dilakukan untuk menguji hasil klasifikasi dari ke empat model yang telah dibuat.

4.9.1. Model NN Dengan Fitur Statistik

Dalam model ini dilakukan pengujian terhadap skenario yang dibuat untuk mengetahui bagaimana model dalam melakukan klasifikasi. hasil klasifikasi model NN (statistik) didasari dengan probablitas. Dengan mengetahui probablitas dapat diketahui berapa persentase dari hasil pengujian dimana untuk NN dapat dilihat pada akhir *softmax* pada *output layer.* Hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; promo: 0.291, normal: 0.689, dan penipuan: 0.020. Hasil pengujian *black Box*nya dapat dilihat pada

tabel 4.15 berikut.

Tabel 4.15 *Black Box Testing* Hasil Klasifikasi Model NN (Statistik)

No	Deskripsi Skenario	*Input* (Contoh SMS)	*Output* yang Diharapkan	Hasil Pengujian	Kesimpulan
1	SMS normal tanpa unsur spam	"Siap untuk pertemuan besok? Jangan lupa bawa dokumennya ya."	Normal	Normal	Valid
2	SMS promo dari toko terkenal tanpa persetujuan penerima	"Diskon besar-besaran hanya minggu ini di Toko XYZ! Dapatkan diskon hingga 50%!"	Spam (Promo)	Spam (Promo)	Valid
3	SMS mengandung informasi pribadi yang sah	"Ini adalah pengingat bahwa tagihan listrik Anda jatuh tempo besok. Silakan bayar untuk menghindari pemutusan."	Normal	Spam (Penipuan)	Tidak Valid
4	SMS penipuan yang meminta data pribadi	"Verifikasi akun Anda sekarang dengan mengklik link berikut atau akun Anda akan diblokir!"	Spam (Penipuan)	Spam (Penipuan)	Valid
5	SMS undangan acara dari organisasi yang dikenal	"Anda diundang! Bergabunglah dengan kami dalam seminar kesehatan XYZ gratis minggu ini. Daftar sekarang!"	Normal	Spam (Penipuan)	Tidak Valid
6	SMS promo yang mengatasnamakan bank tanpa izin	"Selamat! Anda terpilih menerima kartu kredit ABC dengan limit hingga 100 juta. Klik di sini untuk aktivasi!"	Spam (Penipuan)	Spam (Penipuan)	Valid

Berdasarkan pada tabel 4.15 Berikut adalah analisis untuk masing-masing baris:

1. SMS normal tentang pertemuan bisnis: Hasil pengujian sesuai dengan *output* yang

diharapkan, Berikut hasil yang di dapatkan dari *output layer*.

Normal = $e^{4.4657}$ = 86.956

Spam penipuan = $e^{-3.8687}$ = 0.02085

Spam promo = $e^{-2.2378}$ = 0.10735

Total = 86.956 + 0.02085 + 0.10735 ≈ 87.0842

Fungsi aktivasi *softmax*:

Normal = $\frac{86.956}{87.0842}$ ≈ 0.9985 => 1

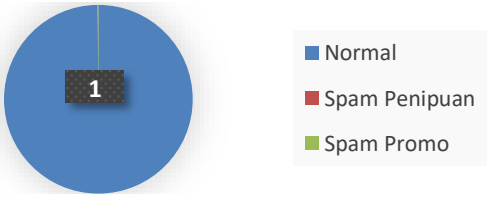
Spam penipuan = $\frac{0.02085}{87.0842}$ ≈ 0.00024 => 0

Spam promo = $\frac{0.10735}{87.0842}$ ≈ 0.00123 => 0

Maka,

$softmax \approx [1, 0, 0]$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.37 berikut.



Gambar 4.37 Grafik Pengujian Pesan Ke-1 Dengan Model NN (Statistik)

Berdasarkan gambar 4.37 hasil yang di dapat, dilihat probablitas yang di peroleh. Hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; normal: 1, penipuan: 0, dan promo: 0. Hasil yang didapatkan menunjukkan bahwa sistem dengan benar mengidentifikasi ini sebagai SMS normal.

2. SMS promosi dari toko terkenal: Sistem dengan benar mengidentifikasinya sebagai spam promosi, yang sesuai dengan *output* yang diharapkan. Berikut hasil yang di dapatkan

dari *output layer*.

Normal = $e^{-2.284}$ = 0.1019

Spam penipuan = $e^{-2.3709}$ = 0.0934

Spam promo = $e^{2.8118}$ = 16.657

Total = 0.1019 + 0.0934 + 16.657 ≈ 16.8523

Fungsi aktivasi *softmax*:

Normal = $\frac{0.1019}{16.8523} \approx 0.0060$

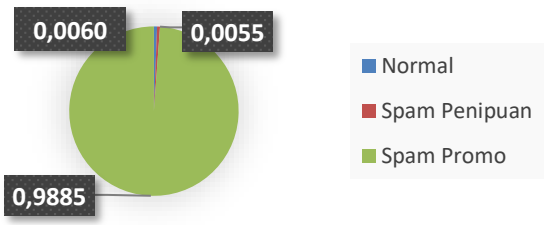
Spam penipuan = $\frac{0.0934}{16.8523} \approx 0.0055$

Spam promo = $\frac{16.657}{16.8523} \approx 0.9885$

Maka,

$softmax \approx [0.0060, 0.0055, 0.9885]$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.38 berikut.



Gambar 4.38 Grafik pengujian Pesan Ke-2 Dengan Model NN (Statistik)

Berdasarkan gambar 3.38, hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; normal: 0.0060, penipuan: 0.0055, dan promo: 0.9885. Hasil yang diperoleh menunjukkan bahwa sistem dengan benar mengidentifikasi ini sebagai SMS spam promo.

3. SMS yang mengandung informasi pribadi yang sah: Disini terjadi kesalahan dalam pengujian, dimana sistem salah mengidentifikasi SMS sah sebagai penipuan. Berikut hasil yang di dapatkan dari *output layer*.

Normal = $e^{-1.9086}$ = 0.1486

Spam penipuan = $e^{2.0866}$ = 8.056

Spam promo = $e^{-1.0264}$ = 0.3585

Total = 0.1486 + 8.056 + 0.3585 ≈ 8.5631

Fungsi aktivasi *softmax*:

Normal = $\frac{0.1486}{8.5631} \approx 0.0174$

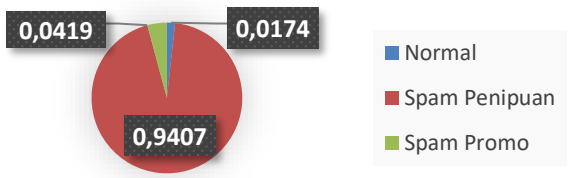
Spam penipuan = $\frac{8.056}{8.5631} \approx 0.9407$

Spam promo = $\frac{0.3585}{8.5631} \approx 0.0419$

Maka,

$softmax \approx [0.0174, 0.9407, 0.0419]$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.39 berikut.



Gambar 4.39 Grafik Pengujian Pesan Ke-3 Dengan Model NN (Statistik).

Berdasarkan gambar 3.39, hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; normal: 0.0174, penipuan: 0.9407, dan promo: 0.0419. Hasil ini menunjukkan bahwa sistem perlu diperbaiki dalam hal ini karena hasil yang dihasilkan tidak valid.

4. SMS penipuan yang meminta data pribadi: Hasil pengujian tidak sesuai dengan *output* yang diharapkan. Berikut hasil yang di dapatkan dari *output layer*.

Normal = $e^{-0.491}$ = 0.6118

Spam penipuan = $e^{2.9076}$ = 18.3195

Spam promo = $e^{-3.1436}$ = 0.0431

Total = 0.6118 + 18.3195 + 0.0431 ≈ 18.9744

Fungsi aktivasi *softmax*:

Normal = $\frac{0.6118}{18.9744}$ ≈ 0.0323 => 0.03

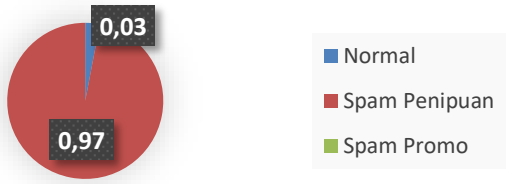
Spam penipuan = $\frac{18.3195}{18.9744}$ ≈ 0.9655 => 0.97

Spam promo = $\frac{0.0431}{18.9744}$ ≈ 0.0023 => 0

Maka,

$softmax \approx [0.03, 0.97, 0]$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.40 berikut.



Gambar 4.40 Grafik Pengujian Pesan Ke-4 Dengan Model NN (Statistik)

Berdasarkan gambar 4.40, hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; normal: 0.03, penipuan: 0.97, dan promo: 0. SMS ini di identifikasi benar sebagai sms spam penipuan. Dapat dilihat probablitas yang di peroleh pada gambar 4.40 berikut.

5. SMS undangan acara dari organisasi yang dikenal: Hasil pengujian tidak sesuai dengan *output* yang diharapkan, Hasil pengujian tidak sesuai dengan *output* yang diharapkan. Berikut hasil yang di dapatkan dari *output layer*.

Normal = $e^{0.0754}$ = 1.0783

Spam penipuan = $e^{0.3719}$ = 1.4501

Spam promo = $e^{-1.8685}$ = 0.1543

Total = 1.0783 + 1.4501 + 0.1543 ≈ 2.6827

Fungsi aktivasi *softmax*:

Normal = $\frac{1.0783}{2.6827} \approx 0.4019 \Rightarrow 0.40$

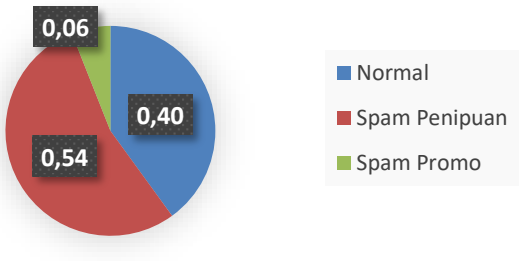
Spam penipuan = $\frac{1.4501}{2.6827} \approx 0.5404 \Rightarrow 0.54$

Spam promo = $\frac{0.1543}{2.6827} \approx 0.0575 \Rightarrow 0.06$

Maka,

$softmax \approx [0.40, 0.54, 0.06]$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.41 berikut.



Gambar 4.41 Grafik Pengujian Pesan Ke-5 Dengan Model NN (Statistik)

Berdasarkan gambar 4.41, hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; normal: 0.40, penipuan: 0.54, dan promo: 0.06. Hasil menunjukkan sistem mengidentifikasi salah sebagai SMS normal. Dapat dilihat probablitas yang di peroleh pada gambar 4.41 berikut.

6. SMS promosi yang mengatasnamakan bank tanpa izin: Hasil pengujian menunjukkan sistem salah mengidentifikasi SMS penipuan sebagai spam promosi. Hasil pengujian tidak sesuai dengan *output* yang diharapkan. Berikut hasil yang di dapatkan dari *output layer*.

Normal = $e^{-2.2131}$ = 0.1091

Spam penipuan = $e^{1.0011}$ = 2.7217

Spam promo = $e^{-0.4234}$ = 0.6547

Total = 0.1091 + 2.7217 + 0.6547 ≈ 3.4855

Fungsi aktivasi *softmax*:

Normal = $\frac{0.1091}{3.4855} \approx 0.0313$

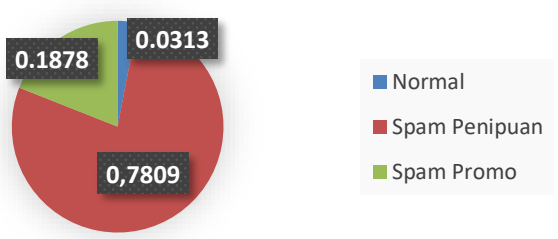
Spam penipuan = $\frac{2.7217}{3.4855} \approx 0.7809$

Spam promo = $\frac{0.6547}{3.4855} \approx 0.1878$

Maka,

$softmax \approx [0.0313, 0.7809, 0.1878]$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.42 berikut.



Gambar 4.42 Grafik Pengujian Pesan Ke-6 Dengan Model NN (Statistik)

Berdasarkan hasil pada gambar 4.42 hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; normal: 0.0313, penipuan: 0.7809, dan promo: 0.1878. Hasil menunjukkan bahwa sistem mungkin memerlukan penyesuaian untuk membedakan antara penipuan dan promosi yang sah.

4.9.2. Model KNN Dengan Fitur Statistik

Selanjutnya untuk *black Box testing* hasil klasifikasi model KNN (statistik) dengan tambahan probablitasnya untuk mengetahui berapa persentase dari hasil pengujian dimana diterapkan kodingan ini di dalamnya untuk mengetahui hasil dari 3 k atau tetangga terdekat sebagai berikut pada gambar 4.43 berikut.

```
# show top-3 nearest neighbors
distances, indices = model_knn_statistik.kneighbors(vectorized_text,
n_neighbors=3)

print("Top-3 Nearest Neighbors:")
for i, (distance, index) in enumerate(zip(distances[0], indices[0])):
    neighbor_class = kelas[model_knn_statistik._y[index]]
    print(f"{i+1}: Class = {neighbor_class}, Distance = {distance}")
```

Gambar 4.43 Menampilkan 3 Tetangga Terdekat.

Berdasarkan gambar 4.43 akan menampilkan 3 tetangga terdekat untuk melakukan perhitungan probablitas berdasarkan 3 label yang di dapatkan. Berikut hasil pengujiannya dapat dilihat pada tabel 4.16 berikut.

Tabel 4.16 *Black Box Testing* Hasil Klasifikasi Model KNN (Statistik)

No	Deskripsi Skenario	*Input* (Contoh SMS)	*Output* yang Diharapkan	Hasil Pengujian	Kesimpulan
1	SMS normal tanpa unsur spam	"Siap untuk pertemuan besok? Jangan lupa bawa dokumennya ya."	Normal	Normal	Valid
2	SMS promo dari toko terkenal tanpa persetujuan penerima	"Diskon besar-besaran hanya minggu ini di Toko XYZ! Dapatkan diskon hingga 50%!"	Spam (Promo)	Spam (Promo)	Valid
3	SMS mengandung informasi pribadi yang sah	"Ini adalah pengingat bahwa tagihan listrik Anda jatuh tempo besok. Silakan bayar untuk menghindari pemutusan."	Normal	Normal	Valid
4	SMS penipuan yang meminta data pribadi	"Verifikasi akun Anda sekarang dengan mengklik link berikut atau akun Anda akan diblokir!"	Spam (Penipuan)	Spam (Penipuan)	Valid
5	SMS undangan acara dari organisasi yang dikenal	"Anda diundang! Bergabunglah dengan kami dalam seminar kesehatan XYZ gratis minggu ini. Daftar sekarang!"	Normal	Normal	Valid

6	SMS promo yang mengatas namakan bank tanpa izin	"Selamat! Anda terpilih menerima kartu kredit ABC dengan limit hingga 100 juta. Klik di sini untuk aktivasi!"	Spam (Penipuan)	Spam (Penipuan)	Valid

Berdasarkan pada tabel 4.16 Berikut adalah analisis untuk masing-masing baris:

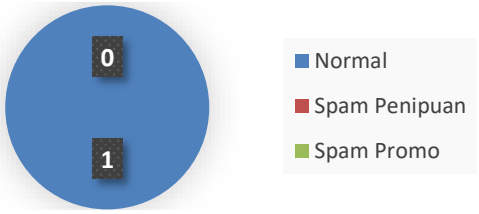
1. SMS normal tentang pertemuan bisnis: Hasil pengujian sesuai dengan *output* yang diharapkan, Berikut 3 tetangga terdekat.

Jarak terdekat ke - 1 kelas = normal, jarak = 1.0
Jarak terdekat ke - 2: kelas = normal, jarak = 1.12
Jarak terdekat ke - 3: kelas = normal, jarak = 1.16

Dilakukan perhitungan probablitas:

Terdapat 3 label normal, maka $\frac{3}{3} = 1$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.44 berikut.



Gambar 4.44 Grafik Pengujian Pesan Ke-1 Dengan Model KNN (Statistik).

Berdasarkan hasil pada gambar 4.44 pesan diklasifikasikan berdasarkan kelas yang paling banyak pada tetangga terdekat yaitu pesan normal. Berdasarkan hasil yang didapatkan menunjukkan bahwa sistem dengan benar mengidentifikasi ini sebagai SMS normal.

2. SMS promosi dari toko terkenal: Sistem dengan benar mengidentifikasinya sebagai spam promosi, Berikut 3 tetangga terdekat.

Jarak terdekat ke - 1 kelas = promo, jarak = 1.0

Jarak terdekat ke - 2: kelas = promo, jarak = 1.17

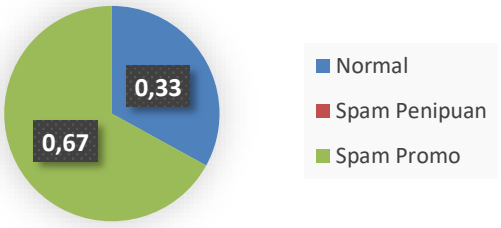
Jarak terdekat ke - 3: kelas = normal, jarak = 1.17

Dilakukan perhitungan probablitas:

Terdapat 1 label normal, maka $\frac{1}{3}$ = 0,33

Terdapat 2 label promo, maka $\frac{2}{3}$ = 0,67

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.45 berikut.



Gambar 4.45 Grafik Pengujian Pesan Ke-2 Dengan Model KNN (Statistik).

Berdasarkan hasil pada gambar 4.45 pesan diklasifikasikan berdasarkan kelas yang paling banyak pada tetangga terdekat yaitu pesan promo. Berdasarkan hasil yang didapatkan menunjukan bahwa sesuai dengan *output* yang diharapkan.

3. SMS yang mengandung informasi pribadi yang sah: Disini terjadi kesalahan dalam pengujian, Berikut 3 tetangga terdekat.

Jarak terdekat ke - 1 kelas = normal, jarak = 1.0

Jarak terdekat ke - 2: kelas = penipuan, jarak = 1.04

Jarak terdekat ke - 3: kelas = promo, jarak = 1.17

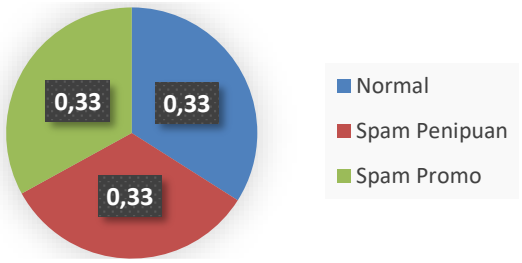
Dilakukan perhitungan probablitas:

Terdapat 1 label normal, maka $\frac{1}{3}$ = 0,33

Terdapat 1 label penipuan, maka $\frac{1}{3}$ = 0,33

Terdapat 1 label promo, maka $\frac{1}{3}$ = 0,33

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.46 berikut.



Gambar 4.46 Grafik Pengujian Pesan Ke-3 Dengan Model KNN (Statistik).

Berdasarkan hasil pada gambar 4.46 pesan diklasifikasikan berdasarkan yang paling mendekati yaitu pesan normal, maka diklasifikasikan sebagai pesan normal. Berdasarkan hasil yang didapatkan menunjukan dimana sistem salah mengidentifikasi SMS sah sebagai penipuan. Ini menunjukkan bahwa sistem perlu diperbaiki dalam hal ini.

4. SMS penipuan yang meminta data pribadi: Hasil pengujian tidak sesuai dengan *output* yang diharapkan. Berikut 3 tetangga terdekat.

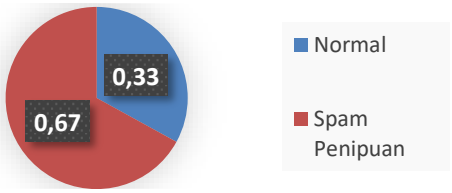
Jarak terdekat ke - 1 kelas = penipuan, jarak = 1.0
Jarak terdekat ke - 2: kelas = normal, jarak = 1.14
Jarak terdekat ke - 3: kelas = penipuan, jarak = 1.20

Dilakukan perhitungan probablitas:

Terdapat 1 label normal, maka $\frac{1}{3}$ = 0,33

Terdapat 2 label penipuan, maka $\frac{2}{3}$ = 0,67

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.47 berikut.



Gambar 4.47 Grafik Pengujian Pesan Ke-4 Dengan Model KNN (Statistik).

Berdasarkan hasil pada gambar 4.47 pesan diklasifikasikan berdasarkan kelas yang paling banyak pada tetangga terdekat yaitu pesan penipuan. Berdasarkan hasil yang

didapatkan SMS ini sesuai diidentifikasi sebagai spam penipuan.

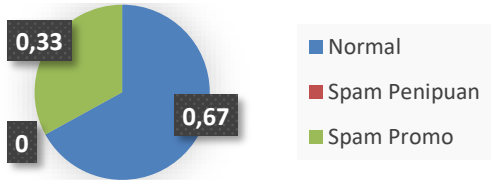
5. SMS undangan acara dari organisasi yang dikenal: Hasil pengujian sesuai dengan *output* yang diharapkan, Berikut 3 tetangga terdekat.

Jarak terdekat ke - 1 kelas = normal, jarak = 1.0
Jarak terdekat ke - 2: kelas = penipuan, jarak = 1.22
Jarak terdekat ke - 3: kelas = promo, jarak = 1.25

Dilakukan perhitungan probablitas:

Terdapat 2 label normal, maka $\frac{2}{3}$ = 0,67
Terdapat 1 label promo, maka $\frac{1}{3}$ = 0,33

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.48 berikut.



Gambar 4.48 Grafik Pengujian Pesan Ke-5 Dengan Model KNN (Statistik).

Berdasarkan hasil pada gambar 4.48 pesan diklasifikasikan berdasarkan kelas yang paling banyak pada tetangga terdekat yaitu pesan normal. Berdasarkan hasil yang didapatkan menunjukkan sistem mengidentifikasi dengan benar sebagai SMS normal.

6. SMS promosi yang mengatasnamakan bank tanpa izin: Hasil pengujian menunjukkan sistem salah mengidentifikasi SMS penipuan sebagai spam promosi. Berikut 3 tetangga terdekat.

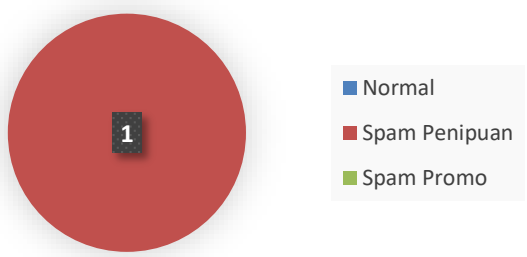
Jarak terdekat ke - 1 kelas = penipuan, jarak = 1.0
Jarak terdekat ke - 2: kelas = normal, jarak = 1.10
Jarak terdekat ke - 3: kelas = penipuan, jarak = 1.21

Dilakukan perhitungan probablitas:

Terdapat 1 label normal, maka $\frac{1}{3}$ = 0,33

Terdapat 2 label penipuan, maka $\frac{2}{3}$ = 0,67

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.49 berikut.



Gambar 4.49 Grafik Pengujian Pesan Ke-6 Dengan Model KNN (Statistik).

Berdasarkan hasil pada gambar 4.49 pesan diklasifikasikan berdasarkan kelas yang paling banyak pada tetangga terdekat yaitu pesan penipuan. Berdasarkan hasil yang didapatkan ini menunjukkan bahwa sistem sesuai mengidentifikasi pesan penipuan.

4.9.3. Model NN Dengan Fitur Linguistik

Selanjutnya untuk *black Box testing* hasil klasifikasi model NN (linguistik) dengan probablitasnya untuk mengetahui berapa persentase dari hasil pengujian dimana dapat dilihat pada akhir *softmax* pada *output layer.* Hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; promo: 0.291, normal: 0.689, dan penipuan: 0.020. Hasil pengujian *black Box*nya dapat dilihat pada tabel 4.17 berikut.

Tabel 4.17 *Black Box Testing* Hasil Klasifikasi Model NN (Linguistik)

No	Deskripsi Skenario	*Input* (Contoh SMS)	*Output* yang Diharapkan	Hasil Pengujian	Kesimpulan
1	SMS normal tanpa unsur spam	"Siap untuk pertemuan besok? Jangan lupa bawa dokumennya ya."	Normal	Normal	Valid

2	SMS promo dari toko terkenal tanpa persetujuan penerima	"Diskon besar-besaran hanya minggu ini di Toko XYZ! Dapatkan diskon hingga 50%!"	Spam (Promo)	Spam (Promo)	Valid
3	SMS mengandung informasi pribadi yang sah	"Ini adalah pengingat bahwa tagihan listrik Anda jatuh tempo besok. Silakan bayar untuk menghindari pemutusan."	Normal	Normal	Valid
4	SMS penipuan yang meminta data pribadi	"Verifikasi akun Anda sekarang dengan mengklik link berikut atau akun Anda akan diblokir!"	Spam (Penipuan)	Spam (Penipuan)	Valid
5	SMS undangan acara dari organisasi yang dikenal	"Anda diundang! Bergabunglah dengan kami dalam seminar kesehatan XYZ gratis minggu ini. Daftar sekarang!"	Normal	Spam (Promo)	Tidak Valid
6	SMS promo yang mengatasnamakan bank tanpa izin	"Selamat! Anda terpilih menerima kartu kredit ABC dengan limit hingga 100 juta. Klik di sini untuk aktivasi!"	Spam (Penipuan)	Spam (Promo)	Tidak Valid

Berdasarkan pada tabel 4.17 Berikut adalah analisis untuk masing-masing baris:

1. SMS normal tentang pertemuan bisnis: Hasil pengujian sesuai dengan *output* yang diharapkan, Berikut hasil yang di dapatkan dari *output layer*.

Normal = $e^{-1.3682}$ = 6.4425

Spam penipuan = $e^{0.0464}$ = 0.5697

Spam promo = $e^{0.4016}$ = 0.2341

Total = 6.4425 + 0.5697 + 0.2341 ≈ 7.2463

Fungsi aktivasi *softmax*:

Normal = $\frac{6.4425}{7.2463} \approx 0.8892$

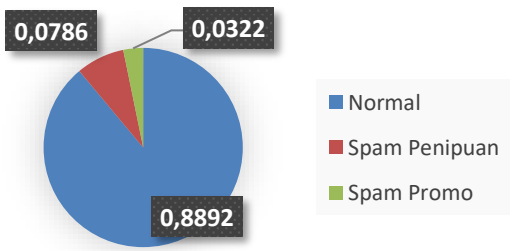
Spam penipuan = $\frac{0.5697}{7.2463} \approx 0.0786$

Spam promo = $\frac{0.2341}{7.2463} \approx 0.0322$

Maka,

$softmax \approx [0.8892, 0.0786, 0.0322]$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.50 berikut.



Gambar 4.50 Grafik Pengujian Pesan Ke-1 Dengan Model NN (Linguistik).

Berdasarkan hasil pada gambar 4.50 hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; normal: 0.8892, penipuan: 0.0786, dan promo: 0.0322. Hasil yang didapatkan menunjukkan bahwa sistem salah mengidentifikasi ini sebagai SMS normal.

2. SMS promosi dari toko terkenal: Sistem dengan benar mengidentifikasinya sebagai spam promosi, yang sesuai dengan *output* yang diharapkan. Berikut hasil yang di dapatkan dari *output layer*.

Normal = $e^{-0.8916}$ = 0.4106

Spam penipuan = $e^{-0.9957}$ = 0.3695

Spam promo = $e^{0.7326}$ = 2.0806

Total = 0.4106 + 0.3695 + 2.0806 ≈ 2.8607

Fungsi aktivasi *softmax*:

Normal = $\frac{0.4106}{2.8607} \approx 0.1436$

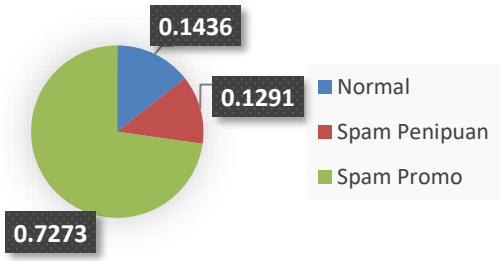
Spam penipuan = $\frac{0.3695}{2.8607} \approx 0.1291$

Spam promo = $\frac{2.0806}{2.8607} \approx 0.7273$

Maka,

$softmax \approx [0.1436, 0.1291, 0.7273]$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.51 berikut.



Gambar 4.51 Grafik Pengujian Pesan Ke-2 Dengan Model NN (Linguistik).

Berdasarkan hasil pada gambar 4.51 hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; normal: 0.1436, penipuan: 0.1291, dan promo: 0.7273. Dapat dilihat probablitas yang di peroleh pada gambar 4.51 berikut.

3. SMS yang mengandung informasi pribadi yang sah: Sistem dengan benar mengidentifikasinya sebagai normal, yang sesuai dengan *output* yang diharapkan. Berikut hasil yang di dapatkan dari *output layer*.

Normal = $e^{0.4716}$ = 1.6021

Spam penipuan = $e^{-0.6099}$ = 0.5436

Spam promo = $e^{-0.3849}$ = 0.6805

Total = 1.6021 + 0.5436 + 0.6805 = 2.8262

Fungsi aktivasi *softmax*:

Normal = $\frac{1.6021}{2.8262} \approx 0.5670$

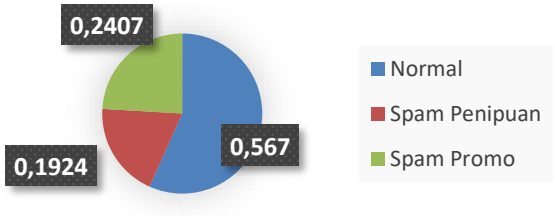
Spam penipuan = $\frac{0.5436}{2.8262} \approx 0.1924$

Spam promo = $\frac{0.6805}{2.8262} \approx 0.2407$

Maka,

$softmax \approx [0.5670, 0.1924, 0.2407\ ]$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.52 berikut.



Gambar 4.52 Grafik Pengujian Pesan Ke-3 Dengan Model NN (Linguistik).

Berdasarkan hasil pada gambar 4.52 hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; normal: 0.5670, penipuan: 0.1924, dan promo: 0.2407. Ini menunjukkan bahwa sistem perlu benar dalam hal ini. Dapat dilihat probablitas yang di peroleh pada gambar 4.52 berikut.

4. SMS penipuan yang meminta data pribadi: Hasil pengujian sesuai dengan *output* yang diharapkan. Berikut hasil yang di dapatkan dari *output layer*.

Normal = $e^{-2.4405}$ = 0.0869

Spam penipuan = $e^{1.4726}$ = 4.3600

Spam promo = $e^{0.012}$ = 0.1856

Total = 0.0869+4.3600+1.0121≈5.4590

Fungsi aktivasi *softmax*:

Normal = $\frac{0.0869}{5.4590} \approx 0.0159$

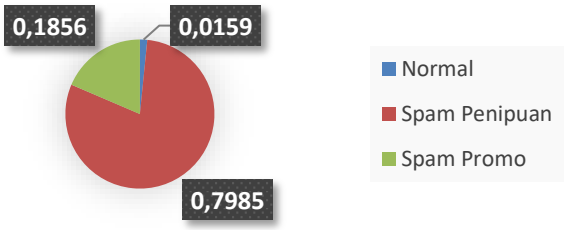
Spam penipuan = $\frac{4.3600}{5.4590} \approx 0.0786$

Spam promo = $\frac{0.1856}{5.4590} \approx 0.1856$

Maka,

$softmax \approx [0.0159, 0.7985, 0.1856]$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.53 berikut.



Gambar 4.53 Grafik Pengujian Pesan Ke-4 Dengan Model NN (Linguistik).

Berdasarkan hasil pada gambar 4.53 hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; normal: 0.0159, penipuan: 0.7985, dan promo: 0.1856. Hasil klasifikasi sesuai sebagai spam penipuan.

5. SMS undangan acara dari organisasi yang dikenal: Hasil pengujian tidak sesuai dengan *output* yang diharapkan, Berikut hasil yang di dapatkan dari *output layer*.

Normal = $e^{-1.3682}$ = 0.2546

Spam penipuan = $e^{0.0464}$ = 1.0475

Spam promo = $e^{0.4016}$ = 1.4948

Total = 0.2546 + 1.0475 + 1.4948 ≈ 2.7969

Fungsi aktivasi *softmax*:

Normal = $\frac{0.2546}{2.7969} \approx 0.0910$

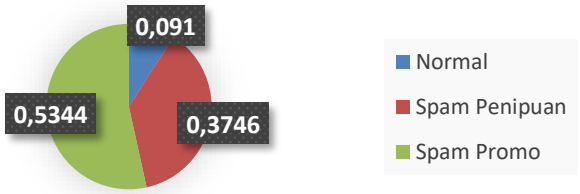
Spam penipuan = $\frac{1.0475}{2.7969} \approx 0.3746$

Spam promo = $\frac{1.4948}{2.7969} \approx 0.5344$

Maka,

$softmax \approx [0.0910, 0.3746, 0.5344]$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.54 berikut.



Gambar 4.54 Grafik Pengujian Pesan Ke-5 Dengan Model NN (Linguistik).

Berdasarkan hasil pada gambar 4.54 hasil dari output layer adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk input tersebut. Probabilitas kelas; normal: 0.0910, penipuan: 0.3746, dan promo: 0.5344. menunjukkan sistem mengidentifikasi salah sebagai SMS normal.

6. SMS promosi yang mengatasnamakan bank tanpa izin: Hasil pengujian menunjukkan sistem salah mengidentifikasi SMS penipuan sebagai spam promosi. Berikut hasil yang di dapatkan dari *output layer*.

Normal = $e^{-3.345}$ = 0.0352

Spam penipuan = $e^{1.3232}$ = 3.7549

Spam promo = $e^{0.7814}$ = 2.1851

Total = 0.0352 + 3.7549 + 2.1851 ≈ 5.9752

Fungsi aktivasi *softmax*:

Normal = $\frac{0.0352}{5.9752} \approx 0.0059$

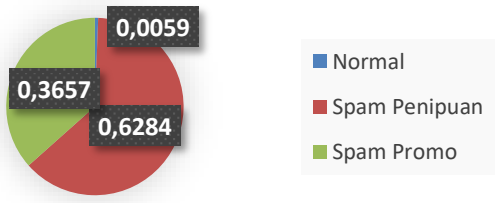
Spam penipuan = $\frac{3.7549}{5.9752} \approx 0.6284$

Spam promo = $\frac{0.2341}{5.9752} \approx 0.3657$

Maka,

$softmax \approx [0.0059, 0.6284, 0.3657]$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.55 berikut.



Gambar 4.55 Grafik Pengujian Pesan Ke-6 Dengan Model NN (Linguistik).

Berdasarkan hasil pada gambar 4.55 hasil dari *output layer* adalah probabilitas untuk setiap kelas. Kelas dengan probabilitas tertinggi adalah prediksi untuk *input* tersebut. Probabilitas kelas; normal: 0.0059, penipuan: 0.6284, dan promo: 0.3657. Ini menunjukkan bahwa sistem mungkin memerlukan penyesuaian untuk membedakan antara penipuan dan promosi yang sah.

4.9.4. Model KNN Dengan Fitur Linguistik

Selanjutnya untuk *black box testing* hasil klasifikasi model KNN (linguistik) dengan probablitasnya untuk mengetahui berapa persentase dari hasil pengujian dimana diterapkan kodingan ini di dalamnya untuk mengetahui hasil dari 3 k atau tetangga terdekat sebagai berikut pada gambar 4.56 berikut.

```
# show top-3 nearest neighbors
distances, indices = model_knn_statistik.kneighbors(vectorized_text,
n_neighbors=3)

print("Top-3 Nearest Neighbors:")
for i, (distance, index) in enumerate(zip(distances[0], indices[0])):
    neighbor_class = kelas[model_knn_statistik._y[index]]
    print(f"{i+1}: Class = {neighbor_class}, Distance = {distance}")
```

Gambar 4.56 Menampilkan 3 Tetangga Terdekat.

Berdasarkan gambar 4.56 akan menampilkan 3 tetangga terdekat untuk melakukan perhitungan probablitas berdasarkan 3 label yang di dapatkan. Dalam pengujian *black Box*nya dapat dilihat pada tabel 4.18 berikut.

Tabel 4.18 *Black Box Testing* Hasil Klasifikasi Model KNN (Linguistik)

No	Deskripsi Skenario	*Input* (Contoh SMS)	*Output* yang Diharapkan	Hasil Pengujian	Kesimpulan
1	SMS normal tanpa unsur spam	"Siap untuk pertemuan besok? Jangan lupa bawa dokumennya ya."	Normal	Normal	Valid
2	SMS promo dari toko terkenal tanpa persetujuan penerima	"Diskon besar-besaran hanya minggu ini di Toko XYZ! Dapatkan diskon hingga 50%!"	Spam (Promo)	Normal	Tidak Valid
3	SMS mengandung informasi pribadi yang sah	"Ini adalah pengingat bahwa tagihan listrik Anda jatuh tempo besok. Silakan bayar untuk menghindari pemutusan."	Normal	Normal	Valid
4	SMS penipuan yang meminta data pribadi	"Verifikasi akun Anda sekarang dengan mengklik link berikut atau akun Anda akan diblokir!"	Spam (Penipuan)	Spam (Penipuan)	Valid
5	SMS undangan acara dari organisasi yang dikenal	"Anda diundang! Bergabunglah dengan kami dalam seminar kesehatan XYZ gratis minggu ini. Daftar sekarang!"	Normal	Normal	Tidak Valid
6	SMS promo yang mengatasna makan bank tanpa izin	"Selamat! Anda terpilih menerima kartu kredit ABC dengan limit hingga 100 juta. Klik di sini untuk aktivasi!"	Spam (Penipuan)	Spam (Penipuan)	Valid

Berdasarkan pada tabel 4.18 Berikut adalah analisis untuk masing-masing baris:

1. SMS normal tentang pertemuan bisnis: Hasil pengujian sesuai dengan *output* yang diharapkan, Berikut 3 tetangga terdekat.

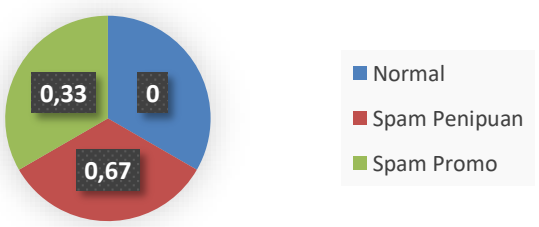
Jarak terdekat ke - 1 kelas = penipuan, jarak = 0
Jarak terdekat ke - 2: kelas = promo, jarak = 1.0
Jarak terdekat ke - 3: kelas = penipuan, jarak = 1.41

Dilakukan perhitungan probablitas:

Terdapat 1 label promo, maka $\frac{1}{3}$ = 0,33

Terdapat 2 label penipuan, maka $\frac{2}{3}$ = 0,67

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.57 berikut.



Gambar 4.57 Grafik Pengujian Pesan Ke-1 Dengan Model KNN (Linguistik).

Berdasarkan hasil pada gambar 4.57 pesan diklasifikasikan berdasarkan kelas yang paling banyak pada tetangga terdekat yaitu pesan penipuan. Berdasarkan hasil yang didapatkan yang menunjukkan bahwa sistem dengan benar mengidentifikasi ini sebagai SMS normal.

2. SMS promosi dari toko terkenal: Sistem dengan benar mengidentifikasinya sebagai spam promosi, Hasil pengujian sesuai dengan *output* yang diharapkan, Berikut 3 tetangga terdekat.

Jarak terdekat ke - 1 kelas = normal, jarak = 1,73
Jarak terdekat ke - 2: kelas = promo, jarak = 1.73
Jarak terdekat ke - 3: kelas = penipuan, jarak = 2

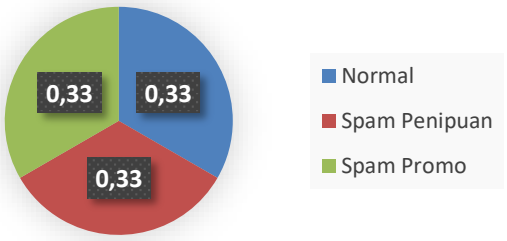
Dilakukan perhitungan probablitas:

Terdapat 1 label normal, maka $\frac{1}{3}$ = 0,33

Terdapat 1 label penipuan, maka $\frac{1}{3}$ = 0,33

Terdapat 1 label promo, maka $\frac{1}{3}$ = 0,33

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.58 berikut.



Gambar 4.58 Grafik Pengujian Pesan Ke-2 Dengan Model KNN (Linguistik)

Berdasarkan hasil pada gambar 4.58 pesan diklasifikasikan berdasarkan kelas yang paling terdekat yaitu pesan normal. Berdasarkan hasil yang didapatkan yang menunjukkan bahwa sistem dengan salah mengidentifikasi ini sebagai SMS normal dimana seharusnya diidentifikasi sebagai pesan promo. Dapat dilihat probablitas yang di peroleh pada gambar 4.58 berikut.

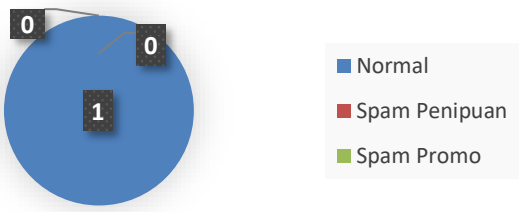
3. SMS yang mengandung informasi pribadi yang sah: Disini terjadi kesalahan dalam pengujian, Berikut 3 tetangga terdekat.

Jarak terdekat ke - 1 kelas = normal, jarak = 0
Jarak terdekat ke - 2: kelas = normal, jarak = 0
Jarak terdekat ke - 3: kelas = normal, jarak = 1

Dilakukan perhitungan probablitas:

Terdapat 3 label normal, maka $\frac{3}{3}$ = 1

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.59 berikut.



Gambar 4.59 Grafik Pengujian Pesan Ke-3 Dengan Model KNN (Linguistik)

Berdasarkan hasil pada gambar 4.59 pesan diklasifikasikan berdasarkan kelas yang paling banyak pada tetangga terdekat yaitu pesan normal. Berdasarkan hasil yang didapatkan yang menunjukkan dimana sistem benar mengidentifikasi SMS sah sebagai normal.

4. SMS penipuan yang meminta data pribadi: Hasil pengujian tidak sesuai dengan *output* yang diharapkan. Hasil pengujian sesuai dengan *output* yang diharapkan, Berikut 3 tetangga terdekat.

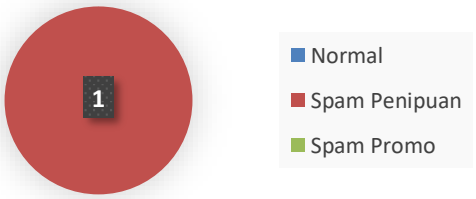
Jarak terdekat ke - 1 kelas = promo, jarak = 1
Jarak terdekat ke - 2: kelas = penipuan, jarak = 1.41
Jarak terdekat ke - 3: kelas = penipuan, jarak = 1.73

Dilakukan perhitungan probablitas:

Terdapat 2 label penipuan, maka $\frac{2}{3}$ = 0,67

Terdapat 1 label promo, maka $\frac{1}{3}$ = 0,33

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.60 berikut.



Gambar 4.60 Grafik Pengujian Pesan Ke-4 Dengan Model KNN (Linguistik)

Berdasarkan hasil pada gambar 4.60 pesan diklasifikasikan berdasarkan kelas yang paling banyak pada tetangga terdekat yaitu pesan penipuan. Berdasarkan hasil yang didapatkan yang menunjukkan SMS ini seharusnya diidentifikasi sebagai spam penipuan, tetapi sistem mengidentifikasinya sebagai normal, yang menunjukkan kegagalan dalam mendeteksi *phishing* atau penipuan.

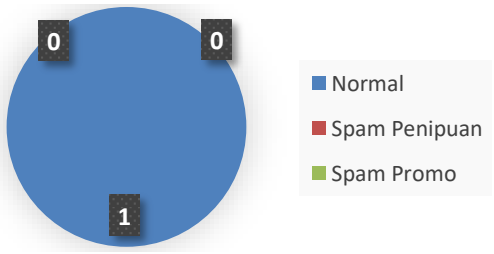
5. SMS undangan acara dari organisasi yang dikenal: Hasil pengujian tidak sesuai dengan *output* yang diharapkan, Berikut 3 tetangga terdekat.

Jarak terdekat ke - 1 kelas = normal, jarak = 1.0
Jarak terdekat ke - 2: kelas = normal, jarak = 1.0
Jarak terdekat ke - 3: kelas = normal, jarak = 1.0

Dilakukan perhitungan probablitas:

Terdapat 1 label normal, maka $\frac{3}{3} = 1$

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.61 berikut.



Gambar 4.61 Grafik Pengujian Pesan Ke-5 Dengan Model KNN (Linguistik).

Berdasarkan hasil pada gambar 4.61 pesan diklasifikasikan berdasarkan kelas yang paling banyak pada tetangga terdekat yaitu pesan normal. Berdasarkan hasil yang didapatkan yang menunjukkan menunjukkan sistem mengidentifikasi sesuai sebagai SMS normal.

6. SMS promosi yang mengatasnamakan bank tanpa izin: Hasil pengujian menunjukkan sistem salah mengidentifikasi SMS penipuan sebagai spam promosi. Berikut 3 tetangga

terdekat.

Jarak terdekat ke - 1 kelas = penipuan, jarak = 0

Jarak terdekat ke - 2: kelas = promo, jarak = 1.0

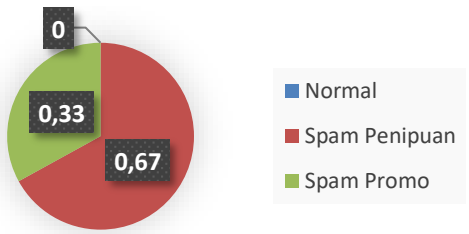
Jarak terdekat ke - 3: kelas = penipuan, jarak = 1.41

Dilakukan perhitungan probablitas:

Terdapat 1 label promo, maka $\frac{1}{3}$ = 0,33

Terdapat 2 label penipuan, maka $\frac{2}{3}$ = 0,67

Berikut hasil probablitas pengujian, dapat dilihat pada gambar 4.62 berikut.



Gambar 4.62 Grafik Pengujian Pesan Ke-6 Dengan Model KNN (Linguistik).

Berdasarkan hasil pada gambar 4.62 pesan diklasifikasikan berdasarkan kelas yang paling banyak pada tetangga terdekat yaitu pesan penipuan. Berdasarkan hasil yang didapatkan yang menunjukkan bahwa sistem benar dalam mengidentifikasi pesan penipuan.

4.10. Hasil Analisis

Setelah melakukan evaluasi model dan pengujian *black box* terhadap ke empat model yang berbeda baik itu dengan fitur statistik ataupun fitur linguistik dengan berdasarkan KNN dan *Neural Network* (NN). Berikut adalah analisis keseluruhan dari performa model tersebut dapat dilihat pada tabel 4.19.

Tabel 4.19 Hasil Analisis.

Model	Evaluasi Model	Pengujian *Black Box*
Model *K-Nearest Neighbors*	Model ini memberikan akurasi tinggi sebesar 95% dan	Model ini berhasil mengidentifikasi semua jenis

(KNN) dengan Fitur Statistik	menunjukkan *precision*, *precision*, dan F1-*score* yang sangat baik di seluruh kategori, dengan performa terbaik di kelas Promo. Ini menunjukkan bahwa model ini efektif dalam mengklasifikasikan dan membedakan antara berbagai jenis pesan.	pesan dengan benar, menunjukkan konsistensi yang kuat dalam pengenalan berbagai skenario tes.
Model *K-Nearest Neighbors* (KNN) dengan Fitur Linguistik	Meskipun memiliki akurasi yang lebih rendah (82%), model ini menunjukkan variasi yang signifikan dalam *precision* dan *precision*, terutama di kelas Penipuan dengan skor yang rendah. Hal ini menunjukkan bahwa model ini kurang efisien dalam mengklasifikasikan pesan secara akurat ketika hanya mengandalkan fitur linguistik.	Model ini mengalami beberapa kesulitan, khususnya dalam mengenali SMS yang mengandung informasi pribadi, sering kali salah mengidentifikasinya sebagai promo. Ini menunjukkan kebutuhan akan penyesuaian dan peningkatan pada model ini untuk meningkatkan keakuratan dalam situasi yang lebih variatif.
Model *Neural Network* (NN) dengan Fitur Statistik	Dengan akurasi yang sangat tinggi (98%), model ini menunjukkan *precision*, *precision*, dan F1-*score* yang luar biasa di semua kategori. Khususnya, kelas Promo mendapatkan nilai sempurna, yang menandakan kemampuan model yang luar biasa dalam mengidentifikasi dan mengklasifikasikan pesan secara akurat.	Model ini menunjukkan performa yang sangat tinggi, dengan tingkat keberhasilan yang tinggi dalam mengenali semua jenis pesan secara akurat dalam semua skenario tes yang diberikan, menandakan bahwa model ini sangat efektif dan handal.
Model *Neural Network* (NN) dengan Fitur Linguistik	Model ini mencapai akurasi yang baik (85%) tetapi menunjukkan kelemahan pada beberapa kategori, khususnya di kelas Penipuan dengan *precision* yang sangat rendah. Hal ini menunjukkan bahwa model ini mungkin tidak sepenuhnya efektif dalam menginterpretasi fitur linguistik untuk kategori tertentu.	Meskipun lebih baik dari KNN dengan fitur linguistik, model ini masih menunjukkan beberapa kelemahan dalam mengidentifikasi secara akurat jenis pesan seperti undangan acara dan informasi pribadi, seringkali mengklasifikasikannya sebagai promo atau spam, yang menunjukkan kebutuhan untuk pengoptimalan lebih lanjut.

Berdasarkan tabel 4.19 Model *Neural Network* dengan fitur statistik menonjol sebagai

pilihan terbaik, berdasarkan evaluasi dan hasil pengujian *black Box*. Model ini menunjukkan keandalan, akurasi, dan konsistensi yang sangat tinggi dalam mengenali berbagai jenis pesan. Model ini sangat direkomendasikan untuk implementasi lebih lanjut dalam aplikasi nyata. Sementara itu, model lain, khususnya yang menggunakan fitur linguistik, memerlukan peninjauan dan penyesuaian lebih lanjut untuk meningkatkan keakuratan dan kehandalan dalam situasi praktis yang lebih beragam.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa kesimpulan yang dapat diambil dari penelitian ini:

1. Fitur statistik ataupun fitur linguistik berhasil merepresentasikan sebuah pesan menjadi sebuah angka untuk diterapkan ke dalam algoritma.
2. Algoritma KNN dan NN berhasil melakukan pelatihan model dengan fitur statistik ataupun fitur linguistik. Dan selain itu dapat melakukan klasifikasi pada sebuah pesan baru.
3. Fitur statistik ataupun fitur linguistik terbukti fitur statistik lebih efektif dibandingkan dengan fitur linguistik. Karena berdasarkan hasil evaluasi model untuk fitur statistik dengan KNN dan NN memiliki nilai akurasi 98% dan 95%. Sedangkan model yang menggunakan fitur linguistik memiliki nilai dibawah fitur statistik yakni KNN dengan akurasi 82% dan NN dengan akurasi 85%. Dalam penelitian ini fitur statistik jelas lebih efektik dalam performa model khususnya pada akurasi.
4. Dalam mengklasifikasi sebuah pesan baru dapat dilihat bahwa model KNN fitur statistik lebih unggul. Berdasarkan pengujian *black Box* dilakukan pengujian pesan baru sebanyak 6 pesan dengan skenario yang berbeda dan menghasilkan klasifikasi yang diperoleh seluruhnya valid, sedangkan untuk model yang lain berdasarkan skenario yang sama masih mengalami keterangan tidak valid dimana , NN fitur statistik mengalami 2 skenario dengan keterangan tidak valid, KNN fitur linguistik dengan 2 skenario dengan keterangan tidak valid dan NN fitur linguistik mendapatkan 3 skenario dengan keterangan tidak valid.

5. Hasil *classification report* yang diperoleh antara ke empat model tersebut memiliki keunggulan masing-masing khususnya penggunaan NN dengan menggunakan fitur statistik yang dimana memiliki keunggulan nilai *precision*: pesan normal 98%; penipuan 95%; promo 98%, *precision*: pesan normal 96%; penipuan 92%; promo 100%, *f1-score*: pesan normal 96%; penipuan 93%; promo 99% dan akurasi 98% pada setiap kelas terkecuali pada kelas promo, nilai *precision* nya dimana pada penggunaan fitur statistik dengan algoritma KNN yang lebih unggul dengan nilai 99%. Oleh karena itu model NN dengan fitur statistik lebih unggul dibanding ketiga model.

6. Metode NN terbukti efektif dibandingkan dengan KNN karena dapat dilihat NN sendiri unggul pada *precision*, *precision*, f-1 *score* dan akurasi baik itu dengan fitur statistik sendiri ataupun dengan fitur linguistik. Untuk kelas normal baik itu KNN statistik dengan NN statistik ataupun KNN linguistik dengan NN linguistik dapat dilihat bahwa nilai *precision*; NN statistik 98% sedangkan KNN statistik 81% ataupun NN linguistik 90% sedangkan KNN linguistik 73%, *precision*; NN statistik 95% sedangkan KNN statistik 94% ataupun NN linguistik 85% sedangkan KNN linguistik 82%, dan *f1-score;* NN statistik 96% sedangkan KNN statistik 87% ataupun NN linguistik 87% sedangkan KNN linguistik 77%. Untuk kelas spam penipuan baik itu KNN statistik dengan NN statistik ataupun KNN linguistik dengan NN linguistik dapat dilihat bahwa nilai *precision*; NN statistik 95% sedangkan KNN statistik 94% ataupun NN linguistik 70% sedangkan KNN linguistik 53%, *precision*; NN statistik 92% sedangkan KNN statistik 87% ataupun NN linguistik 31% sedangkan KNN linguistik 37%, dan *f1-score;* NN statistik 93% sedangkan KNN statistik 90% ataupun NN linguistik 43% sedangkan KNN linguistik 43%. Untuk kelas spam promo baik itu KNN statistik dengan NN statistik ataupun KNN linguistik dengan NN linguistik dapat dilihat bahwa nilai *precision*; NN statistik 98%

sedangkan KNN statistik 99% ataupun NN linguistik 85% sedangkan KNN linguistik 89%, *precision*; NN statistik 100% sedangkan KNN statistik 97% ataupun NN linguistik 97% sedangkan KNN linguistik 92%, dan *f1-score;* NN statistik 99% sedangkan KNN statistik 98% ataupun NN linguistik 91% sedangkan KNN linguistik 91%. Dan Untuk Akurasi baik itu KNN statistik dengan NN statistik ataupun KNN linguistik dengan NN linguistik dapat dilihat bahwa nilai akurasinya; NN statistik 98% sedangkan KNN statistik 95% ataupun NN linguistik 85% sedangkan KNN linguistik 82%. Berdasarkan hasil yang diperoleh maka secara garis besar bahwa algoritma NN dalam mengklasifikasi pesan spam lebih baik performanya dibandingkan KNN. Alasan lain mengapa NN lebih unggul dibandingkan KNN yaitu dikarenakan NN dalam pengelolaan data, NN mampu dalam menangani data yang sangat kompleks dengan berbagai fitur. Dalam halnya pada penelitian ini yang memiliki data sebanyak 5000 lebih data, tentunya ketika diproses ke fitur statistik yang memiliki banyak fitur, dimana tf idf membuat fitur berdasarkan per kata yang ada pada tiap kalimat pada seluruh *dataset* yang ada. sedangkan KNN dalam mengelola datanya lebih sederhana, sedangkan data yang dimiliki sangat kompleks khususnya pada fitur yang melimpah.

7. Analisis hasil klasifikasi pada ke empat model menunjukan model *Neural Network* (NN) dan *K-Nearest Neighbors* (KNN) yang menggunakan fitur statistik berhasil mengenali SMS spam, terutama yang berisi promosi atau penipuan. Namun, model NN dengan fitur statistik sering salah menganggap SMS yang sebenarnya aman sebagai spam. Di sisi lain, model NN dan KNN yang berfokus pada analisis bahasa kurang tepat, sering kali salah mengklasifikasikan SMS yang tidak bermasalah sebagai spam dan gagal mengidentifikasi SMS penipuan. Secara umum, semua model terlalu mudah menganggap SMS yang sebenarnya tidak masalah sebagai spam dan model NN lebih baik dalam mendeteksi

penipuan dibandingkan model KNN.

8. Secara keseluruhan model yang didapatkan, Metode NN dengan fitur statistik tampaknya terbukti efektif. Evaluasi model menunjukan nilai *precision*, *recall*, f1-*score*, dan akurasi yang didapatkan menunjukkan bahwa model tersebut dengan metode NN dengan fitur statistik baik.

5.2 Saran

Berdasarkan hasil penelitian, beberapa saran untuk penelitian selanjutnya dapat menggunakan *ataset*, algoritma ataupun fitur statistik maupun fitur linguistik yang lain ataupun terbaru dalam melakukan pengklasifikasian SMS Spam. Dan mampu melakukan implementasi yang lebih kompleks lagi dalam implementasi seperti pada android misalnya pada *whatsapp* dan platform lainnya agar dapat meminimalisir penipuan dari pesan spam. Disarankan juga agar dapat lebih teliti dalam mengambil data khususnya pada data penipuan. Karena banyak penipu yang selalu mencari cara dalam melakukan penipuan tentunya penipu juga selalu mengalami yang namanya perkembangan setiap saat. Ditambah teknologi selalu mengalami yang namanya perkembangan. Maka dari itu perlu melakukan analisis kembali terkait jenis penipuan baru yang dialami orang sekitar.

DAFTAR PUSTAKA

Abayomi-Alli, O., Misra, S. dan, Abayomi-Alli, A., 2022. *A Deep Learning Method For Automatic* SMS Spam *Classification*: *Performance of Learning Algorithms on Indigenous Dataset*. *Concurrency and Computation: Practice and Experience*, Vol. 11, *Issue*. 1 *Februar*, 2020

AL-Jumaili, A. S. A. dan, Tayyeh, H. K., 2020. *A Hybrid Method of Linguistic and Statistical Features For Arabic Sentiment Analysis*. Baghdad *Science Journal*, ISSN:2078-8665, Vol. 17, *Issue*.1 *March*, 2020

Dwiyansaputra, R., Nugraha, G. S., Bimantoro, F. dan, Aranta, A., 2021. Deteksi SMS Spam Berbahasa Indonesia Menggunakan TF-IDF dan *Stochastic Gradient descent Classifier* (*Indonesian* SMS Spam *Detection Using* TF-IDF *and Stochastic Gradient descent*. Jurnal Teknologi Informasi, Komputer dan Aplikasinya, ISSN:2657-0327, Vol. 3, *Issue*. 2 September, 2021

Fhadli, M., Fauzi, M. A. dan, Afirianto, T., 2017. Peringkasan Literatur Ilmu Komputer Bahasa Indonesia Berbasis Fitur Statistik dan Linguistik Menggunakan Metode *Gaussian Naïve Bayes*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, ISSN: 2548-964X, Vol. 1, *Issue*. 4 April, 2017

Firmansyah, M. R., Ilyas, R., & Kasyidi, F., 2020. Klasifikasi Kalimat Ilmiah Menggunakan *Recurrent Neural Network*. *Prosiding The 11th Industrial Research Workshop and National* Seminar, Vol. 11, *Issue*. 1 *Agustus,* 2020

Herwanto, H., Chusna, N. L. dan, Arif, M. S., 2021. Klasifikasi SMS Spam Berbahasa Indonesia Menggunakan Algoritma *Multinomial Naïve Bayes*. Jurnal Media Informatika Budidarma, ISSN:2548-8368, Vol. 5, *Issue*. 4 Oktober, 2021.

Laksono, E. P., Basuki, A. dan, Abdurrachman Bachtiar, F., 2020. Optimasi Nilai K pada Algoritma KNN Untuk Klasifikasi Spam dan Ham Email. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), ISSN:2580-0760, Vol. 1, *Issue*. 1 April, 2020.

Ramadhan, A., Lindawati, L. dan, Rose, M. M., 2023. Komparasi Algoritma *Neural Network* dan *K-Nearest Neighbor* dalam Mendeteksi *Malware* Android. *Building of Informatics Technology and Science* (BITS), ISSN:2685-3310, Vol. 5, *Issue*. 1, Juni 2023.

Reviantika, F., Azhar, Y. dan, Indah Marthasari, G., 2021. Analisis Klasifikasi SMS Spam

Menggunakan *Logistic Regression*. Jurnal Sistem Cerdas, ISSN:2622-8254, Vol. 4, *Issue*. 3, 2021.

Rumlaklak, N. D., Fanggidae, A., & Polly, Y. T., 2022. Klasifikasi Penentuan Status Zona di Kota Kupang Menggunakan Algoritma *Naive Bayes Classifier*. Jurnal Komputer Dan Informatika, ISSN:2654-4091, Vol. 10, *Issue*. 1 Maret, 2022.

Runimeirati, Muis, A. dan, Muhammad, F., 2023. Pelatihan *Text Mining* Menggunakan Bahasa Pemrograman *Python*. Abdimas Langkanae, ISSN:2808-7682, Vol. 3, *Issue*. 1, 2023.

Said, M. S. dan, Yusti, Y., 2020. Penerapan Algoritma *K-Means* Dalam Penentuan Jurusan Siswa SMAN 05 Bombana. Jurnal Sistem Informasi dan Teknik Komputer, ISSN:2502-5899, Vol. 5, *Issue*. 2, 2020.

Setifani, N. A., Fitriana, D. N. dan, Yusuf, A., 2020. Perbandingan Algoritma *Naïve Bayes*, SVM, dan *Decision Tree* Untuk Klasifikasi SMS Spam. Jurnal Sistem Informasi Musirawas, Vol. 5, *Issue.* 2 Desember, 2020

Solikin, I., 2018. Implementasi E-Modul Pada Program Studi Manajemen Informatika Universitas Bina Darma Berbasis Web *Mobile*. Jurnal Rekayasa Sistem dan Teknologi Informasi, ISSN:2580-0760, Vol. 2, *Issue*. 2 Juni, 2018.

Wahid, A., Baharulloh, M., Kahfiansyah, R., Abrilianto, T., Saifudin, A. dan, Mulyati, S. (2021). Identifikasi SMS Spam Menggunakan Metode *Naive Bayes*. Jurnal Informatika Universitas Pamulang, ISSN:2622-4615, Vol. 6, *Issue*. 3 September, 2021.

Wibawa, M. S. dan, Maysanjaya, I. M. D., 2018. *Multi Layer Perceptron* dan *Principal Component Analysis* Untuk Diagnosa Kanker Payudara. Jurnal Nasional Pendidikan Teknik Informatika, ISSN:2548-4265, Vol. 7, *Issue*. 1 Maret, 2018

Widyawati dan, Susanto., 2019. Perbandingan Algoritma *Naïve Bayes* dan *Support Vector Machine* (SVM) Dalam Klasifikasi SMS Spam Berbahasa Indonesia. Jurnal Ilmiah Sains dan Teknologi, ISSN:2622-6391, Vol. 3, *Issue*. 2 Agustus, 2019

Wijaya, D. P., Murti, L. D., & Rachman, M. R., 2022. *Precision* dan *Precision* pada *Online* Public Access Catalog (OPAC) Dinas Arsip dan Perpustakaan Kota Bandung. VISI PUSTAKA*:* Buletin Jaringan Informasi Antar Perpustakaan, ISSN:2088-2025, Vol. 24, *Issue*. 1 Mei, 2022

Yuliana, D., Purwanto dan, Supriyanto, C., 2019. Klasifikasi Teks Pengaduan Masyarakat

Dengan Menggunakan Algoritma *Neural Network*. Jurnal KomtekInfo, ISSN:2502-8758, Vol. 5, *Issue*. 3 April, 2019

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	JUMAT, 08 MARET 2024
Pukul	:	10:30 - 12:30
Tempat	:	RUANG PRODI

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa	:	MUHAMMAD RAIHAN RIZAL
NPM	:	07352011006
Judul	:	PERBANDINGAN FITUR STATISTIK DAN LINGUISTIK PADA ALGORITMA K-NEAREST NEIGHBOR (KNN) & NEURAL NETWORK DALAM PENGKLASIFISIKAN SMS SPAM

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

- Pahami program

Acc
24-09-2024

Dosen Pembimbing I,

MUHAMMAD FHADLI, S.Kom., M.Sc.
NIP. 199611232023211012

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	JUMAT, 08 MARET 2024
Pukul	:	10:30 - 12:30
Tempat	:	RUANG PRODI

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa	:	MUHAMMAD RAIHAN RIZAL
NPM	:	07352011006
Judul	:	PERBANDINGAN FITUR STATISTIK DAN LINGUISTIK PADA ALGORITMA K-NEAREST NEIGHBOR (KNN) & NEURAL NETWORK DALAM PENGKLASIFISIKAN SMS SPAM

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

- Lakukan perbaikan sesuai permintaan penguji

ACC OK tgl 29/09/2024 Stop Chia..

Dosen Pembimbing II,

YASIR MUIN, S.T., M.Kom.
NIDN. 9990582796

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	JUMAT, 08 MARET 2024
Pukul	:	10:30 - 12:30
Tempat	:	RUANG PRODI

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa	:	MUHAMMAD RAIHAN RIZAL
NPM	:	07352011006
Judul	:	PERBANDINGAN FITUR STATISTIK DAN LINGUISTIK PADA ALGORITMA K-NEAREST NEIGHBOR (KNN) & NEURAL NETWORK DALAM PENGKLASIFISIKAN SMS SPAM

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

1. ganti warna spam menjadi merah
2. manual program tidak ada, jadi dibuat metode secara manual dan program manual kemudian di uji dgn library sebagai uji validasi
3. alur atau tahap penelitian dicantumkan dalam naskah

Ttd, 03/04

Dosen Penguji I,

SYARIFUDDIN N. KAPITA, S.Pd., M.Si.
NIDN. 0012039105

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	JUMAT, 08 MARET 2024
Pukul	:	10:30 - 12:30
Tempat	:	RUANG PRODI

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa	:	MUHAMMAD RAIHAN RIZAL
NPM	:	07352011006
Judul	:	PERBANDINGAN FITUR STATISTIK DAN LINGUISTIK PADA ALGORITMA K-NEAREST NEIGHBOR (KNN) & NEURAL NETWORK DALAM PENGKLASIFISIKAN SMS SPAM

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

1. perhatikan dengan teliti pustaka/referensi dan format penulisan
2. pertajam kembali kesimpulan dan analisis akhir penelitian
3. diagram hasil perbandingan??
4. hasil klasifikasi knn dan nn utk semua fitur?
5. implementasi matematis knn dan nn dalam klasifikasi perlu lebih jelas!
6. perhatikan cataan ujian, bawa saat asistensi

Dosen Penguji II,

ACHMAD FUAD, S.T., M.T.
NIP. 197606182005011001

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	JUMAT, 08 MARET 2024
Pukul	:	10:30 - 12:30
Tempat	:	RUANG PRODI

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa	:	MUHAMMAD RAIHAN RIZAL
NPM	:	07352011006
Judul	:	PERBANDINGAN FITUR STATISTIK DAN LINGUISTIK PADA ALGORITMA K-NEAREST NEIGHBOR (KNN) & NEURAL NETWORK DALAM PENGKLASIFISIKAN SMS SPAM

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

Bawa aplikasinya ke saya ketika konsultasi perbaikan.

23/4.2024

Dosen Penguji II,

MUHAMMAD SABRI AHMAD, S.Kom., M.Kom.
NIP. 198905012019031013

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI
UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK PROGRAM STUDI INFORMATIKA
Kampus III Universitas Khairun, Kelurahan Jati Kota Ternate Selatan
http://if.unkhair.ac.id, http://unkhair.ac.id Group FB: if.unkhair

KARTU BIMBINGAN HASIL

Nama Mahasiswa : Muhammad Raihan Rizal
NIM : 07352011006
Dosen Pembimbing I : Muhammad Fhadli, S.Kom., M.Sc.
Judul : Perbandingan Fitur Statistik Dan Linguistik Pada Algoritma K-Nearest Neighbors (KNN) & Neural Network Dalam Pengklasifikasian SMS Spam

NO	Tanggal	Uraian	Paraf
1.	21/02/2024	Selesaikan Program	
2.	22/02/2024	Selesaikan laporan	
3.	22/02/2024	Acc	

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI
UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK PROGRAM STUDI INFORMATIKA
Kampus III Universitas Khairun, Kelurahan Jati Kota Ternate Selatan
http://if.unkhair.ac.id, http://unkhair.ac.id Group FB: if.unkhair

KARTU BIMBINGAN HASIL

Nama Mahasiswa : Muhammad Raihan Rizal
NIM : 07352011006
Dosen Pembimbing II : Yasir Muin, S.T., M.Kom.
Judul : Perbandingan Fitur Statistik Dan Linguistik Pada Algoritma K-Nearest Neighbors (KNN) & Neural Network Dalam Pengklasifikasian SMS Spam

NO	Tanggal	Uraian	Paraf
1.	23/02/2024	Tambahkan model untuk deteksi kalimat	
2.	26/02/2024	Deteksi kalimatnya belum sesuai	
		ok siap ujian Tgl 27/02/2024	

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	JUMAT, 26 APRIL 2024
Pukul	:	09:00 - 10:30
Tempat	:	RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa	:	MUHAMMAD RAIHAN RIZAL
NPM	:	07352011006
Judul	:	PERBANDINGAN FITUR STATISTIK DAN LINGUISTIK PADA ALGORITMA K-NEAREST NEIGHBOR (KNN) & NEURAL NETWORK DALAM PENGKLASIFIKASIKAN SMS SPAM

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

- Pahami source code

28-6-2024

Dosen Pembimbing I,

MUHAMMAD FHADLI, S.Kom., M.Sc.
NIP. 199611232023211012

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

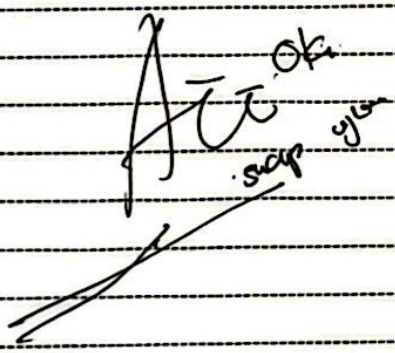
Hari / tanggal	:	JUMAT, 26 APRIL 2024
Pukul	:	09:00 - 10:30
Tempat	:	RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa	:	MUHAMMAD RAIHAN RIZAL
NPM	:	07352011006
Judul	:	PERBANDINGAN FITUR STATISTIK DAN LINGUISTIK PADA ALGORITMA K-NEAREST NEIGHBOR (KNN) & NEURAL NETWORK DALAM PENGKLASIFIKASIKAN SMS SPAM

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

- Lakukan Perbaikan yang diminta oleh penguji



Dosen Pembimbing II,

YASIR MUIN, S.T., M.Kom.
NIDN. 9990582796

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	JUMAT, 26 APRIL 2024
Pukul	:	09:00 - 10:30
Tempat	:	RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa	:	MUHAMMAD RAIHAN RIZAL
NPM	:	07352011006
Judul	:	PERBANDINGAN FITUR STATISTIK DAN LINGUISTIK PADA ALGORITMA K-NEAREST NEIGHBOR (KNN) & NEURAL NETWORK DALAM PENGKLASIFIKASIKAN SMS SPAM

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

1. perbaiki tabel tambahkan Wi
2. Pelajari lagi menentukan nilai negatif dari logaritma
3. diferensiasi pelajari lagi aturan rantai
4. tidak ada matriks transpose yang dicantumkan sehingga nilai perkalian tidak jelas

Ttd, 22/05/2024

Dosen Penguji I,

SYARIFUDDIN N. KAPITA, S.Pd., M.Si.
NIDN. 0012039105

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	JUMAT, 26 APRIL 2024
Pukul	:	09:00 - 10:30
Tempat	:	RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa	:	MUHAMMAD RAIHAN RIZAL
NPM	:	07352011006
Judul	:	PERBANDINGAN FITUR STATISTIK DAN LINGUISTIK PADA ALGORITMA K-NEAREST NEIGHBOR (KNN) & NEURAL NETWORK DALAM PENGKLASIFIKASIKAN SMS SPAM

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

1. kesimpulan, analisis akhir, saran dan abstrak diperjelas dan dipertajam kembali!
2. semua masukan/catatan saat ujian ditindaklanjuti dan diasistensi!
3. hasil perbandingan perlu lebih teliti dan dicek kembali, tambahkan perhitungan manual dengan diagramnya!
4. perhatikan format penulisan!

Kasi Tanda 28/06/2024
lanjut penjilidan

Acc Perbaikan Skripsi
08/07/2024

Dosen Penguji II,

ACHMAD FUAD, S.T., M.T.
NIP. 197606182005011001

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	JUMAT, 26 APRIL 2024
Pukul	:	09:00 - 10:30
Tempat	:	RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa	:	MUHAMMAD RAIHAN RIZAL
NPM	:	07352011006
Judul	:	PERBANDINGAN FITUR STATISTIK DAN LINGUISTIK PADA ALGORITMA K-NEAREST NEIGHBOR (KNN) & NEURAL NETWORK DALAM PENGKLASIFIKASIKAN SMS SPAM

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

Tambahkan simbolik, angka dalam pemrosesan aplikas

Dosen Penguji III,

MUHAMMAD SABRI AHMAD, S.Kom., M.Kom.
NIP. 198905092019031013