PERBANDINGAN PERFORMANSI MODEL NAÏVE BAYES DALAM MENGANALISIS SENTIMEN MENGENAI E-SPORT MOBILE LEGENDS



Disusun Oleh:

N a m a : Muhammad Aqil Arkanudin

NIM : 19523083

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2025

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PERBANDINGAN PERFORMANSI MODEL NAÏVE BAYES DALAM MENGANALISIS SENTIMEN MENGENAI E-SPORT MOBILE LEGENDS

TUGAS AKHIR

Disusun Oleh:

N a m a : Muhammad Aqil Arkanudin

NIM : 19523083

Yogyakarta, 10 Agustus 2025

Pembimbing,

(Ir. Irving Vitra Paputungan, S.T., M.Sc., Ph.D.)

HALAMAN PENGESAHAN DOSEN PENGUJI

PERBANDINGAN PERFORMANSI MODEL NAÏVE BAYES DALAM MENGANALISIS SENTIMEN MENGENAI E-SPORT MOBILE LEGENDS

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 10 Agustus 2025

Tim Penguji

Ir. Irving Vitra Paputungan, S.T., M.Sc., Ph.D. _____

Anggota 1

Ari Sujarwo, S.Kom., M.I.T. _____

Anggota 2

Ir. Chandra Kusuma Dewa S.Kom., M.Cs., Ph.D. _____

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	:	Muhammad Aqil Arkanudin
NIM	:	19523083

Tugas akhir dengan judul:

PERBANDINGAN PERFORMANSI MODEL NAÏVE BAYES DALAM MENGANALISIS SENTIMEN MENGENAI E-SPORT MOBILE LEGENDS

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 10 Agustus 2025



(Muhammad Aqil Arkanudin)

HALAMAN PERSEMBAHAN

Bismillahirrahmaanirrahiim… Alhamdulillaahirabbil'aalamiin… segala puji bagi Allah SWT atas segala nikmat dan karunia yang telah diberikan kepada saya sehingga saya mendapatkan kekuatan dan kesabaran dalam menyelesaikan skripsi. Puji dan syukur saya panjatkan kepada Allah SWT yang selalu memberikan yang terbaik kepada hamba-Nya. Tidak lupa sholawat serta salam semoga selalu terlimpahkan kepada Nabi Muhammad SAW.

Dengan rahmat Allah Yang Maha Pengasih Lagi Maha Penyayang, skripsi ini dipersembahkan kepada :

1. Ibu terhebat yang sangat saya cintai. Terima kasih untuk setiap doa, pengorbanan, dukungan, motivasi, serta kasih sayang yang telah diberikan kepada saya sehingga saya bisa sampai pada tahap ini. Semoga Ibu selalu diberi kekuatan, kesabaran, kesehatan, dan kelancaran rezeki oleh Allah SWT. Aamiin.
2. Almarhum Ayah saya yang sangat saya sayangi, yang belum sempat saya tunjukkan hal membanggakan dari diri ini. Terima kasih atas segala pengorbanan, doa, bimbingan, kasih sayang serta hal-hal lainnya yang telah diberikan. Semoga Allah mengampuni segala dosamu, menerima segala amal sholehmu dan memberikan tempat terbaik kepadamu di sisi-Nya. Aamiin.
3. Kedua saudara perempuan saya, Ulfa dan Bita. Terimakasih telah memberikan warna pada kehidupan saya dengan celotehan dan keributannya sehari-hari. Terima kasih atas doa dan dukungannya selama ini. Semoga selalu diberikan kesehatan lahir batin dan dapat menjadi anak yang sholehah dan tercapai cita-citanya. Aamiin.
4. Keluarga besar saya, yang selalu memberikan semangat dan motivasi.
5. Dosen pembimbing saya, Bapak Irving Vitra Paputungan, yang telah membimbing dan memberikan arahan dalam penyusunan tugas akhir ini.
6. Teman-teman yang selalu menjadi bagian dari perjalanan ini, baik dalam suka maupun duka.
7. Dan untuk diri saya sendiri, sebagai bentuk pencapaian dan pembuktian atas perjuangan selama ini.

HALAMAN MOTO

"*Dalam hidup, akan selalu ada rintangan. Tapi mentalitas seorang pemenang adalah bagaimana kamu bangkit setelah terjatuh, bukan seberapa sering kamu terjatuh. Teruslah berusaha, percaya pada proses, dan nikmati setiap perjalanan.*"

(Bambang Pamungkas)

"Betapapun sulitnya hidup, selalu ada sesuatu yang dapat kamu lakukan dan berhasil"

(Stephen Hawking)

"Carpe diem"

(Horace)

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan berkat dan tuntunan-Nya, sehingga penyusunan skripsi dengan judul "Perbandingan Performansi Model Naïve Bayes dalam Menganalisis Sentimen Mengenai E-Sport Mobile Legends" ini akhirnya dapat terselesaikan. Dalam penulisan skripsi ini penulis tidak lepas dari berbagai hambatan dan rintangan, namun berkat bantuan, bimbingan, petunjuk dan saran dari berbagai pihak maka hambatan dan rintangan tersebut dapat teratasi. Banyak pihak yang membantu dalam penyusunan skripsi ini. Pada kesempatan ini penulis ingin menyampaikan terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun secara tidak langsung hingga terselesaikannya skripsi ini.

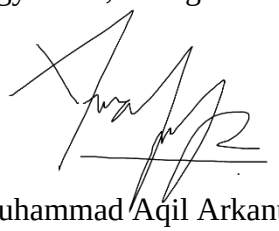
Ucapan terima kasih yang tulus penulis ucapkan kepada :

1. Prof. Dr. Ir. Hari Purnomo, M.T., IPU, ASEAN.Eng selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Dr. Ir. Raden Teguh Dirgahayu, S.T., M.Sc. selaku Ketua Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Progam Studi Informatika – Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Ir. Irving Vitra Paputungan, S.T., M.Sc., Ph.D. selaku dosen pemimbing tugas akhir. Terima kasih atas bimbingan, saran, dukungan, dan ilmu selama penulisan laporan tugas akhir ini.
5. Elyza Gustri Wahyuni, S.T. M.Cs. selaku dosen pemimbing akademik saya di Program Studi Informatika – Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
6. Seluruh Bapak/Ibu dosen Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia yang telah memberikan pengetahuan dan ilmu yang sangat bermanfaat.
7. Seluruh sahabat dan teman saya yang selalu memberi dukungan, semangat, serta saran selama masa perkuliahan.
8. Semua pihak yang tidak bisa dituliskan satu persatu namun senantiasa memberikan dukungan baik secara langsung maupun tidak langsung.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, maka dari itu penulis membutuhkan saran dan kritik yang bersifat membangun dari berbagai pihak,

sehingga dapat dipergunakan untuk memperbarui skripsi ini maupun penelitian selanjutnya. Akhirnya penulis berharap semoga skripsi ini bisa bermanfaat bagi semua pihak.

Yogyakarta, 10 Agustus 2025



(Muhammad Aqil Arkanudin)

SARI

Penelitian ini bertujuan untuk membandingkan performansi tiga varian model Naïve Bayes, yaitu Gaussian, Multinomial, dan Bernoulli, dalam menganalisis sentimen mengenai e-sport Mobile Legends. Proses penelitian mencakup eksplorasi dan pengumpulan data, pra-pemrosesan dan pelabelan data, pembobotan menggunakan TF-IDF, pelatihan model, evaluasi menggunakan confusion matrix, serta visualisasi hasil. Hasil evaluasi menunjukkan bahwa model Bernoulli Naïve Bayes memperoleh akurasi tertinggi sebesar 0,729, diikuti oleh Multinomial sebesar 0,724, dan Gaussian sebesar 0,481. Berdasarkan hasil tersebut, Bernoulli Naïve Bayes dinilai paling efektif untuk analisis sentimen dalam konteks e-sport Mobile Legends.

Kata kunci: analisis sentimen, Naïve Bayes, Mobile Legends, e-sport, TF-IDF

GLOSARIUM

Glosarium memuat daftar kata tertentu yang digunakan dalam laporan dan membutuhkan penjelasan, misalnya kata serapan yang belum lazim digunakan. Urutkan sesuai abjad. Contoh penulisannya seperti di bawah ini:

Analisis Sentimen	proses mengidentifikasi dan mengkatogerikan opini atau emosi dalam teks.
Machine Learning	pembelajaran mesin yang merupakan cabang kecerdasan buatan yang memungkinkan sistem belajar dari data tanpa perlu diprogram secara eksplisit
Naïve Bayes	algoritma klasifikasi berbasis *probabilistic* yang menggunakan Teorema Bayes dengan asumsi independensi antar fitur
Gaussian	varian *Naïve Bayes* yang digunakan untuk data numerik kontinu dan mengasumsikan distribusi normal.
Multinomial	varian *Naïve Bayes* yang digunakan untuk data teks dalam bentuk frekuensi atau *TF-IDF*.
Bernoulli	varian *Naïve Bayes* yang digunakan untuk data biner, seperti keberadaan atau ketiadaan fitur.
TF-IDF	singkatan dari *Term Frequency-Inverse Document Frequency*, yaitu metode pembobotan kata dalam dokumen untuk mencerminkan pentingnya suatu kata dalam kumpulan dokumen.
Confusion Matrix	matriks yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan membandingkan beberapa komponen penilaian.
Accuracy	akurasi, ukuran performa model berdasarkan proporsi prediksi yang benar terhadap keseluruhan data.
Precision	presisi, ukuran seberapa banyak prediksi positif yang benar dari semua prediksi positif yang dihasilkan model.
Recall	ukuran kemampuan model dalam menemukan seluruh data yang termasuk dalam kelas positif sebenarnya.
F1-Score	rata-rata harmonis antara *precision* dan *recall*, memberikan keseimbangan antara keduanya.
Pra-pemrosesan	tahapan awal dalam analisis teks yang mencakup pembersihan, tokenisasi, penghapusan *stopword*, dan *stemming*.

E-Sport	istilah untuk kompetisi permainan video profesional yang dilakukan secara daring atau luring
Mobile Legends	salah satu permainan video bergenre MOBA (Multiplayer Online Battle Arena) yang popular di kalangan e-sport

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI ix
GLOSARIUM x
DAFTAR ISI xii
DAFTAR TABEL xiv
DAFTAR GAMBAR xv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah dan Pertanyaan Penelitian 3
1.3 Batasan Masalah dan Lingkup Penelitian 3
1.4 Tujuan Penelitian 4
1.5 Manfaat Penelitian 4
1.6 Metodologi Penelitian 4
1.7 Sistematika Penulisan 5
BAB II KAJIAN PUSTAKA 7
2.1 Analisis Sentimen 7
2.2 Klasifikasi 9
2.3 Klasifikasi Naïve Bayes 13
2.4 *Cross Validation* 16

2.5 *Performance Evaluation Measure* dengan *Confusion Matrix*.....17
BAB III PERANCANGAN ANALISIS SENTIMEN.....21
3.1 Eksplorasi Data.....22
3.2 Pengumpulan Data.....23
3.3 Pra-pemrosesan Data.....24
3.4 Pembobotan TF-IDF.....29
3.5 Pelatihan Model.....33
3.6 Evaluasi Model.....41
3.7 Visualisasi Data.....49
BAB IV HASIL DAN PEMBAHASAN.....52
4.1 Eksplorasi data.....52
4.2 Pengumpulan Data.....52
4.3 Implementasi Pra-pemrosesan Data.....54
4.4 Implementasi Pembagian Data dan Pembototan TF-IDF.....64
4.5 Pelatihan Model Naïve Bayes.....67
4.6 Evaluasi Model Naïve Bayes.....69
4.7 Visualisasi Data.....78
BAB V KESIMPULAN DAN SARAN.....83
5.1 Kesimpulan.....83
5.2 Saran.....84
DAFTAR PUSTAKA.....85

DAFTAR TABEL

Tabel 2. 1 Perbandingan metode klasifikasi 11
Tabel 2. 2 Confusion Matrix 19

Tabel 3. 1 Contoh Hasil Penghapusan Tanda Baca 26
Tabel 3. 2 Contoh pesan yang terdupliklasi 26
Tabel 3. 3 Contoh proses case folding 27
Tabel 3. 4 Contoh proses tokenizing 27
Tabel 3. 5 Contoh proses stopword removal 28
Tabel 3. 6 Contoh proses stemming 28
Tabel 3. 7 Contoh penghapusan emoji yang terkonversi menjadi teks 29
Tabel 3. 8 Contoh nilai TF-IDF 31

Tabel 4. 1 Sampel pesan yang terduplikasi 58
Tabel 4. 2 Contoh hasil penghapusan tanda baca 59
Tabel 4. 3 Contoh hasil proses case folding 60
Tabel 4. 4 Contoh hasil tokenizing 60
Tabel 4. 5 Contoh hasil proses penghapusan stopword 61
Tabel 4. 6 Contoh hasil proses stemming 62
Tabel 4. 7 Contoh hasil penghapusan emoji yang terkonversi menjadi teks 62
Tabel 4. 8 Contoh hasil pelabelan data dengan label positif 63
Tabel 4. 9 Contoh hasil pelabelan data dengan label negatif 63
Tabel 4. 10 Contoh hasil pelabelan data dengan label netral 63
Tabel 4. 11 Contoh hasil pembobotan dengan TF-IDF 67
Tabel 4. 12 Hasil evaluasi dari model Gaussian 72
Tabel 4. 13 Hasil evaluasi dari model Multinomial 74
Tabel 4. 14 Hasil evaluasi dari model Bernoulli 75

DAFTAR GAMBAR

Gambar 3. 1 Diagram Alir Perbandingan Performansi 3 Model Naive Bayes 21
Gambar 3. 2 Contoh cuplikan dataset 23
Gambar 3. 3 Contoh sebelum pembersihan kolom yang tidak diperlukan 25
Gambar 3. 4 Contoh setelah penghapusan kolom yang tidak diperlukan 25
Gambar 3. 5 Jumlah dataset berdasarkan kelas sentimen 50
Gambar 3. 6 Jumlah data latih dan data uji 50
Gambar 3. 7 WordCloud kata yang paling sering muncul 51

Gambar 4. 1 Ekstensi Comment Stack 52
Gambar 4. 2 Siaran Grand Final MSC 2023 53
Gambar 4. 3 Hasil ekspor data live chat 53
Gambar 4. 4 Kode program untuk menginstal package 54
Gambar 4. 5 Kode program untuk mengimpor library 54
Gambar 4. 6 Kode program untuk mengakses dataset 55
Gambar 4. 7 Contoh dataset saat diakses pada Google Colab 56
Gambar 4. 8 Kode program untuk menghapus kolom yang tidak diperlukan 56
Gambar 4. 9 Hasil setelah penghapusan kolom yang tidak diperlukan 57
Gambar 4. 10 Kode program untuk menghapus pesan yang sama 57
Gambar 4. 11 Hasil setelah penghapusan kolom dengan pesan yang sama 58
Gambar 4. 12 Kode program untuk menghapus tanda baca 59
Gambar 4. 13 Kode program untuk mengubah menjadi huruf kecil 59
Gambar 4. 14 Kode program untuk melakukan tokenizing 60
Gambar 4. 15 Kode program untuk menghapus stopword 61
Gambar 4. 16 Kode program untuk melakukan stemming 62
Gambar 4. 17 Dataset setelah proses pre-processing 64
Gambar 4. 18 Kode program untuk mengimpor library yang digunakan 65
Gambar 4. 19 Kode program untuk membagi dataset menjadi data latih dan data uji 65
Gambar 4. 20 Hasil pembagian data latih dan data uji 66
Gambar 4. 21 Kode program untuk melakukan pembobotan data 67
Gambar 4. 22 Diagram batang data latih dan data uji berdasarkan label sentimen 68
Gambar 4. 23 Kode program untuk membuat dan melatih model 68

Gambar 4. 24 Kode program untuk mengevaluasi performansi model 70
Gambar 4. 25 Confusion matrix dari model Gaussian 71
Gambar 4. 26 Confusion matrix dari model Multinomial 73
Gambar 4. 27 Confusion matrix dari model Bernoulli 74
Gambar 4. 28 Hasil evaluasi 3 model Naive Bayes 77
Gambar 4. 29 WordCloud pesan dengan label positif 78
Gambar 4. 30 WordCloud pesan dengan label negatif 79
Gambar 4. 31 WordCloud pesan dengan label netral 80
Gambar 4. 32 WordCloud keseluruhan data 81

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi dan popularitas media sosial telah menciptakan ruang bagi masyarakat untuk mengekspresikan pendapat dan pandangan mereka terhadap berbagai topik termasuk *e-sport*. Salah satu game yang memiliki komunitas besar di dunia *e-sport* adalah game Mobile Legends yang menjadi subjek diskusi publik yang intens baik dalam bentuk dukungan maupun kritik (E. P. Putra et al., 2023). Dukungan dari para penggemar sering kali terlihat dalam bentuk pujian atas inovasi game, sedangkan kritik biasanya berfokus pada aspek *gameplay* atau kebijakan pengembang. Oleh karena itu analisis sentimen terhadap game Mobile Legends semakin penting untuk memahami persepsi masyarakat, serta bagaimana game ini mempengaruhi komunitas pemainnya (Sinaga & Jatmoko, 2020).

Analisis sentimen adalah proses penting untuk menggali opini dari teks seperti komentar di media sosial, ulasan, dan postingan forum. Teknik ini biasanya digunakan untuk mengkategorikan sentimen menjadi positif, negatif, atau netral (Sari & Wibowo, 2019). Di sinilah peran penting model *machine learning* terutama dalam pemrosesan bahasa alami (NLP) (Surya, 2023). Salah satu pendekatan yang sering digunakan untuk analisis sentimen adalah Naive Bayes yang dikenal dengan kesederhanaannya serta efisiensi dalam menangani data teks. Naive Bayes digunakan pada penelitian ini karena mudah diimpelementasikan. Naive Bayes dipilih karena memiliki beberapa keunggulan seperti kecepatan yang tinggi dalam pelatihan dan prediksi, serta kemampuannya untuk bekerja dengan baik meskipun hanya menggunakan sedikit data.

Model Naive Bayes adalah kelompok algoritma probabilistik yang berdasarkan pada Teorema Bayes dengan asumsi independensi antar fitur. Terdapat beberapa model dari Naive Bayes yang digunakan dalam berbagai konteks analisis sentimen antara lain *Gaussian Naive Bayes, Multinomial Naive Bayes,* dan *Bernoulli Naive Bayes.* Masing-masing model ini memiliki karakteristik tersendiri dalam mengolah data dan menghasilkan prediksi terutama dalam konteks data teks (Lowd & Domingos, 2005).

Gaussian Naive Bayes biasanya digunakan ketika fitur input dianggap mengikuti distribusi normal (Gaussian). Meskipun lebih umum digunakan untuk data numerik, *Gaussian Naive Bayes* terkadang diterapkan pada analisis sentimen untuk memproses fitur yang diturunkan dari teks yang dikonversi menjadi representasi numerik tertentu (Nurul A'ayunnisa,

Salim, & Azis, 2022). *Multinomial Naive Bayes* sangat cocok untuk data yang berupa frekuensi atau jumlah kejadian, seperti analisis teks di mana dokumen diubah menjadi representasi vektor frekuensi kata. Ini menjadikannya pilihan utama dalam analisis sentimen karena karakteristik data teks yang sering kali direpresentasikan dalam bentuk *bag-of-words* atau *tf-idf* (Yuyun, Nurul Hidayah, & Supriadi Sahibu, 2021). *Bernoulli Naive Bayes* bekerja dengan baik pada data biner yang di mana fitur teks dikonversi menjadi representasi biner (kehadiran atau ketidakhadiran kata). Hal ini memungkinkan *Bernoulli Naive Bayes* untuk menangkap fitur yang didasarkan pada apakah sebuah kata muncul dalam dokumen atau tidak, tanpa mempertimbangkan frekuensinya (Yodi Prayoga, Id Hadiana, & Rakhmat Umbara, 2021).

Ketiga model ini memberikan performansi yang berbeda dalam konteks analisis sentimen. *Multinomial Naive Bayes* lebih sering memberikan hasil yang baik ketika bekerja dengan data teks yang diwakili dalam bentuk frekuensi kata, sementara *Bernoulli Naive Bayes* lebih sensitif terhadap kehadiran kata dan *Gaussian Naive Bayes* memberikan performa optimal untuk fitur berkelanjutan, meskipun penerapannya dalam analisis sentimen masih memerlukan evaluasi lebih lanjut (Gunawan, Sastypratiwi, & Pratama, 2018).

Adanya kebutuhan untuk mendapatkan pemahaman yang lebih baik tentang bagaimana persepsi masyarakat terhadap Mobile Legends dapat memengaruhi popularitas dan keberlanjutan game tersebut menjadi motivasi dalam penelitian ini. Pemahaman ini tidak hanya penting bagi pengembang game, tetapi juga bagi penyelenggara turnamen dan pelaku industri *e-sport* lainnya untuk mengembangkan strategi yang lebih efektif dalam mengelola komunitas pemain. Selain itu penelitian ini berusaha mengisi kesenjangan literatur yang ada dengan menyediakan analisis yang komprehensif mengenai model Naive Bayes dalam konteks analisis sentimen mengenai esport. Meskipun terdapat berbagai metode klasifikasi lain seperti Support Vector Machine (SVM) dan Decision Tree yang juga umum digunakan dalam analisis sentimen, penelitian ini secara spesifik berfokus pada perbandingan performa di antara tiga varian model Naïve Bayes (Gaussian, Multinomial, dan Bernoulli). Fokus ini dipilih untuk mengisi kesenjangan literatur yang ada dengan menyediakan analisis yang lebih komprehensif dan mendalam mengenai performa varian Naïve Bayes dalam konteks analisis sentimen pada e-sport .

Penelitian ini bertujuan untuk membandingkan performansi ketiga varian utama dari model Naive Bayes dalam menganalisis sentimen terhadap *e-sport* Mobile Legends. Dengan mengevaluasi akurasi, presisi, *recall*, dan metrik evaluasi lainnya, penelitian ini akan membantu menentukan model Naive Bayes mana yang paling efektif dalam memahami persepsi publik terhadap Mobile Legends. Dengan semakin berkembangnya esport dan

meningkatnya volume opini yang dipublikasikan oleh penggemar, pemain, dan pengamat Mobile Legends, penting untuk memiliki alat analisis yang efisien dan akurat. Hasil perbandingan ini diharapkan dapat memberikan wawasan mendalam bagi pengembang game, penyelenggara turnamen, serta pelaku industri esport secara keseluruhan dalam merespons sentimen publik dan mengembangkan strategi yang lebih baik. Kontribusi utama dari penelitian ini adalah menyediakan panduan yang jelas mengenai model Naive Bayes mana yang paling efektif untuk analisis sentimen di bidang *e-sport*, sehingga dapat digunakan oleh para praktisi dan peneliti di masa mendatang.

1.2 Rumusan Masalah dan Pertanyaan Penelitian

Dengan semakin berkembangnya *e-sport* dan meningkatnya volume opini yang dipublikasikan oleh penggemar, pemain, dan pengamat, penting untuk mengetahui analisis sentimennya, selain itu, masih terdapat adanya kesenjangan dalam literatur yang ada sehingga perlu menyediakan analisis komprehensif mengenai model Naïve Bayes dalam konteks analisis sentimen pada esport.

Berdasarkan rumusan masalah tersebut, berikut merupakan pertanyaan dalam penelitian ini :

a. Bagaimana performa 3 model Naïve Bayes dalam analisis sentiment terhadap *e-sport* Mobile Legends?
b. Model Naïve Bayes mana yang paling efektif dalam mengklasifikasikan sentimen publik terhadap *e-sport* Mobile Legends berdasarkan akurasi, presisi, *recall*, dan metrik evaluasi lainnya?
c. Apa pemahaman dan pengembangan strategi yang perlu dilakukan dalam mengelola komunitas pemain game Mobile Legends dan industri *e-sport*?

1.3 Batasan Masalah dan Lingkup Penelitian

Batasan-batasan masalah dalam penelitian ini adalah sebagai berikut:

a. Penelitian ini menggunakan tiga varian model Naïve Bayes, yaitu *Gaussian Naïve Bayes, Multinomial Naïve Bayes, dan Bernoulli Naïve Bayes*.
b. Data yang digunakan dalam penelitian ini terbatas pada teks yang diambil dari kolom *live chat* siaran resmi Grand Final MSC (MLBB Southeast Asia Cup) 2023 dengan judul "Bahasa Indonesia | Grand Final MSC | MSC 2023" pada kanal Youtube Mobile Legends: Bang Bang.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk membandingkan performansi tiga varian model Naïve Bayes dalam menganalisis sentiment mengenai *e-sport* Mobile Legends, mengevaluasi akurasi, presisi, recall, dan metrik evaluasi lainnya untuk tiap model guna menentukan model yang paling efektif dalam mengklasifikasikan konteks ini.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah menjadi penambah referensi mengenai penerapan model-model Naïve Bayes dalam analisis sentimen. Menyediakan analisis komparatif mengenai kinerja berbagai model Naïve Bayes dalam konteks *e-sports* Mobile Legends. Penelitian ini juga diharapkan dapat memberikan insight bagi penyelenggara acara untuk memahami sentimen reaksi komunitas atau audiens melalui analisis sentimen ini.

1.6 Metodologi Penelitian

Terdapat beberapa tahapan yang akan dilakukan dalam proses penelitian analisis sentiment ini. Tahapan yang dilakukan adalah sebagai berikut:

a. Eksplorasi data

   Tahapan awal dalam proses penelitian ini adalah penentuan atau pemilihan data yang akan digunakan

b. Pengumpulan data

   Tahap pengumpulan data merupakan proses pengambilan data yang akan digunakan

c. Pra-pemrosesan Data dan Pelabelan Data

   Tahap ketiga adalah pra-pemrosesan data yang merupakan proses yang dilakukan untuk mengubah data mentah menjadi lebih bersih dan siap digunakan dalam model *machine learning* yang diharapkan dapat membantu meningkatkan kualitas data dan nilai akurasi hasil analisis. Pada proses ini proses yang dilakukan antara lain adalah *cleaning, case folding, tokenizing, stopword removal, stemming*, pembersihan data manual, serta pelabelan data.

d. Pemisahan Data dan Pembobotan

   Pemisahan data merupakan proses pembagian data menjadi data latih dan data uji yang kemudian akan digunakan untuk melatih dan menguji model. Pembobotan kata adalah proses identifikasi kata-kata yang paling relevan dalam dokumen, mengurangi noise dari kata-kata umum yang tidak penting yang nantinya akan membantu meningkatkan akurasi model machine learning.

e. Pelatihan Model

Pelatihan model merupakan proses melatih model menggunakan data latih yang kemudian akan dievaluasi menggunakan data uji untuk mengukur performanya

f. Evaluasi Model

Evaluasi model merupakan proses mengukur performa atau kinerja model setelah pelatihan dengan tujuan menilai seberapa baik model dapat membuat prediksi pada data yang tidak atau belum dipelajari sebelumnya (data uji)

g. Visualisasi Data

Visualisasi data merupakan proses penyajian hasil analisis dan evaluasi model dalam bentuk visual seperti grafik dan *WordCloud* untuk membantu dalam menginterpretasikan hasil dan menyampaikan temuan secara lebih jelas.

1.7 Sistematika Penulisan

Sistematika penulisan pada laporan tugas akhir penelitian ini dimuat dalam 5 bab, masing-masing bab memberi penjelasan mengenai informasi yang berfungsi untuk memudahkan pembaca dalam mendapatkan informasi yang disampaikan oleh penulis, sistematika penulisan terdiri dari:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan

BAB II KAJIAN PUSTAKA

Bab ini berisi teori-teori yang digunakan dalam penelitan analisis sentimen mengenai Game Online Mobile Legends pada kolom *live chat* siaran Grand Final MSC 2023 di Youtube.

BAB III RANCANGAN ANALISIS SENTIMEN

Bab ini berisi bagan alur penelitian beserta penjelasannya. Bab ini juga memuat teknik atau metode yang digunakan dalam penelitian serta data yang akan digunakan sebagai data kajian dalam analisis sentimen mengenai Game Online Mobile Legends pada kolom *live chat* siaran Grand Final MSC 2023 di Youtube.

BAB IV HASIL DAN PEMBAHASAN

Bab ini memaparkan pembahasan mengenai hasil penelitian yang berisi pengolahan data yang terlah dilakukan dengan mengacu pada teori dan alur penelitian yang telah dirancang dan dijelaskan sebelumnya.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan akhir dari analisis sentimen mengenai Game Online Mobile Legends pada kolom *live chat* siaran Grand Final MSC 2023 di Youtube menggunakan 3 model dari Naïve Bayes serta saran atau perbaikan untuk penelitian-penelitian berikutnya.

BAB II
KAJIAN PUSTAKA

2.1 Analisis Sentimen

Analisis sentimen merupakan disiplin ilmu yang mempelajari pandangan, perasaan, evaluasi, sikap, dan emosi individu yang diungkapkan melalui bahasa tertulis. Bidang ini semakin mendapat perhatian setelah penelitian yang dilakukan oleh Pang, Turney, Goldberg, dan Zhu. Teknik analisis sentimen berperan penting dalam mendukung berbagai keputusan dalam beragam situasi. Penelitian ini mengkategorikan sentimen ke dalam tiga atribut, yaitu positif, netral, dan negatif, karena komentar yang ditemukan di internet dapat berupa salah satu dari ketiga kategori tersebut (Saputra et al., 2015). Selanjutnya, terdapat berberapa pendekatan utama dalam analisis sentimen: *lexicon-based*, regresi sentimen, dan *Machine Learning* (Chedia Dhaoui, Cynthia M. Webster, 2017).

Pendekatan lexicon dalam analisis sentimen disebut juga pendekatan kamus (Kolchyna, Souza, Treleaven, & Aste, 2015). Proses klasifikasi sentimen adalah memberikan nilai tiap kata pada opini berdasarkan kamus atau corpus yang dilengkapi bobot baik positif maupun negatif pada tiap katanya (Musfiroh, Khaira, Utomo, & Suratno, 2021). Regresi dalam *Sentiment Analysis* adalah metode statistik yang digunakan untuk memprediksi nilai sentimen atau opini dalam bentuk numerik. Pendekatan ini bertujuan untuk mengukur intensitas sentimen, bukan hanya mengklasifikasikannya sebagai positif, negatif, atau netral. Dengan regresi, model dapat memprediksi tingkat kekuatan sentimen dari teks yang dianalisis. Teknik ini sering digunakan dalam berbagai aplikasi, seperti analisis ulasan produk, evaluasi kinerja layanan, dan penilaian opini publik terhadap isu-isu tertentu (Pang & Lee, 2008). Machine learning merupakan cabang ilmu yang berfokus pada pengembangan dan analisis algoritma yang memungkinkan komputer untuk belajar. Algoritma dalam *machine learning* bersifat umum, sehingga mampu menghasilkan informasi yang menarik atau berguna dari data tanpa memerlukan penulisan kode khusus. Ketika diberikan sejumlah data, algoritma umum ini mampu membentuk aturan atau model inferensi dari data tersebut (Id, 2021)

Metode *lexicon-based* memiliki beberapa kelebihan yang membuatnya sering digunakan dalam berbagai studi, yaitu: kemampuannya untuk menganalisis sentimen tanpa memerlukan data pelatihan yang besar, mudah diimplementasikan dan dapat diterapkan pada berbagai bahasa tanpa banyak modifikasi, dan ketahanannya terhadap data baru, di mana performanya tidak banyak terpengaruh oleh perubahan dalam pola data, sehingga cocok untuk aplikasi dengan data dinamis (Ravi & Ravi, 2015). Salah satu kelemahan utama metode *lexicon-based* adalah ketergantungan pada daftar kata-kata yang sudah ditentukan sebelumnya untuk

mendeteksi sentimen, tidak akurat ketika menangani teks dengan ambiguitas, metode ini juga seringkali mengabaikan pengaruh intensitas sentimen, dan sulitnya menjaga relevansi dan akurasi lexicon karena bahasa terus berkembang (B. Liu, 2020).

Kelebihan regresi sentimen adalah kemampuan untuk menangkap variasi intensitas sentimen pada skala kontinu, regresi sentimen juga memungkinkan analisis yang lebih kaya ketika digunakan untuk memantau tren sentimen seiring waktu atau membandingkan sentimen di antara berbagai kelompok data (B. Liu, 2020). Kekurangan utama dalam regresi sentimen adalah ketidakmampuannya untuk menangkap nuansa emosi dan konteks yang lebih kompleks dalam teks, regresi sentimen biasanya memerlukan data pelatihan yang besar dan berkualitas tinggi untuk menghasilkan model yang akurat, dan kinerja model dapat menurun jika dataset tidak seimbang atau memiliki bias (Pang & Lee, 2008).

Machine learning memiliki sejumlah kelebihan yang membuatnya semakin populer di berbagai bidang. Salah satu kelebihannya adalah kemampuannya untuk menangani data dalam jumlah besar dan kompleks secara efisien. Algoritma machine learning dapat mempelajari pola dari data yang banyak dan beragam, kemudian membuat prediksi atau keputusan berdasarkan data tersebut tanpa perlu pemrograman eksplisit untuk setiap kemungkinan. *Machine learning* mampu melakukan otomatisasi dalam proses pengambilan keputusan yang cepat dan akurat, sehingga sangat bermanfaat di bidang seperti pengenalan gambar, pengolahan bahasa alami, dan analisis prediktif. Dengan kemampuan untuk terus memperbaiki kinerjanya seiring bertambahnya data dan waktu, machine learning memberikan fleksibilitas dan efisiensi yang tinggi dibandingkan dengan metode tradisional (Sarker, 2021). Kekurangan *machine learning* (ML) adalah kebutuhan akan data yang sangat besar dan berkualitas tinggi untuk pelatihan model, Kurangnya data yang representatif atau data yang bias dapat menghasilkan hasil yang tidak akurat atau diskriminatif, Algoritma *machine learning* sering dianggap sebagai *"black box"* karena sulit untuk memahami bagaimana model tersebut mencapai kesimpulannya, dan machine learning juga masih menghadapi beberapa hambatan dalam penggunaannya yang lebih luas dan efektif (Jordan & Mitchell, 2015). Penggunaan data dalam proses pembelajaran memungkinkan komputer untuk mempelajari pola yang ada dan menghasilkan prediksi yang lebih akurat (Hua, Cuiqin, & Lijuan, 2009). Sehingga cara *machine learning* dipilih karena dapat menghasilkan prediksi dengan nilai akurasi yang tinggi. Dalam prosesnya, *machine learning* melibatkan berbagai jenis algoritma yang dapat menangani berbagai jenis masalah, termasuk pengelompokan, prediksi, dan pengenalan pola. Salah satu tugas utama yang sering diselesaikan oleh *machine learning* adalah klasifikasi.

Analisis sentimen berfokus pada pengelompokan teks, seperti ulasan dan komentar, ke dalam kategori sentimen positif, negatif, atau netral. Proses ini sering disebut sebagai klasifikasi sentimen, di mana algoritma *machine learning* digunakan untuk mendeteksi pola dalam data dan memprediksi kelas sentimen yang sesuai. Teknik-teknik klasifikasi ini memungkinkan model untuk memahami hubungan antara fitur-fitur dalam teks, sehingga dapat memberikan prediksi yang lebih akurat tentang opini atau emosi yang terkandung di dalamnya. Dengan demikian, klasifikasi sentimen berperan penting dalam mempermudah analisis pandangan publik dan tren sentimen dalam berbagai konteks.

2.2 Klasifikasi

Klasifikasi adalah merupakan satu teknik yang digunakan untuk mengelompokkan data ke dalam kategori atau kelas yang telah ditentukan berdasarkan fitur-fitur yang diberikan (Kotsiantis, Zaharakis, & Pintelas, 2006). Tujuannya adalah untuk membuat model yang dapat mempelajari pola-pola dari data latihan yang diberikan dan kemudian memprediksi kategori atau kelas dari data baru yang belum pernah dilihat sebelumnya. Proses klasifikasi melibatkan beberapa langkah. Pertama, data latihan yang terdiri dari input dan labelnya yang terdiri atas kelas atau kategori yang diinginkan guna melatih model (Soofi & Awan, 2017). Model ini bisa berupa berbagai algoritma seperti *Naïve Bayes*, *Decision Trees*, dan *Support Vector Machines* (SVM). Selama pelatihan, model belajar mengenali pola atau hubungan antara fitur-fitur dari data dengan kelas atau kategori yang sesuai.

Naive Bayes Classifier adalah metode klasifikasi berbasis teori probabilitas Bayes yang digunakan untuk memprediksi kategori suatu data. Algoritma ini disebut *"naive"* karena diasumsikan bahwa semua fitur dalam dataset saling independen, meskipun pada kenyataannya mungkin ada ketergantungan antar fitur (Domingos & Pazzani, 1997). *Support Vector Machine* (SVM) adalah metode dalam pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. SVM bekerja dengan mencari *hyperplane* atau batas pemisah yang optimal untuk memisahkan data ke dalam kelas-kelas yang berbeda. *Hyperplane* ini dipilih sedemikian rupa sehingga jarak antara *hyperplane* dan data terdekat dari masing-masing kelas (disebut margin) adalah yang terbesar. Dengan cara ini, SVM memastikan bahwa data dari setiap kelas berada pada sisi yang benar dari hyperplane dengan margin yang sebesar mungkin, sehingga meningkatkan kemampuan generalisasi model (Cortes & Vapnik, 2009). *Decision tree* adalah salah satu metode dalam *machine learning* yang digunakan untuk membuat model prediktif berdasarkan serangkaian keputusan. Pohon keputusan (*decision tree*) adalah salah satu pendekatan data mining yang banyak diterapkan untuk membangun sistem klasifikasi dari berbagai kovariat

atau merancang algoritma prediksi untuk variabel target. Pendekatan ini membagi populasi data ke dalam beberapa segmen bercabang yang membentuk struktur pohon terbalik, dengan elemen utama berupa simpul akar, simpul internal, dan simpul daun sebagai hasil akhir klasifikasi. (Song & Lu, 2015).

Naive Bayes Classifier memiliki beberapa kelebihan yang membuatnya populer dalam berbagai aplikasi, terutama dalam klasifikasi teks dan analisis sentimen, yaitu: Algoritma ini sangat sederhana dan mudah diimplementasikan, *Naive Bayes* memiliki kecepatan tinggi dalam proses pelatihan dan prediksi karena hanya memerlukan sedikit data untuk membuat model. Algoritma ini juga tahan terhadap *data noise* dan sangat efektif untuk masalah klasifikasi yang tidak memerlukan asumsi distribusi data yang kompleks (Bayes, 1968). Salah satu kelemahan utama *Naive Bayes* adalah asumsi bahwa semua fitur bersifat independen satu sama lain, padahal dalam banyak kasus, fitur-fitur tersebut saling bergantung. *Naive Bayes* sering kali kurang efektif untuk dataset kecil atau data yang tidak seimbang, di mana jumlah sampel dari beberapa kelas lebih banyak daripada kelas lain, sehingga hasilnya bisa bias terhadap kelas mayoritas. Keterbatasan lainnya adalah ketika bekerja dengan data numerik yang sangat bervariasi, karena algoritma ini lebih cocok untuk data kategorikal atau diskrit (Bayes, 1968).

Support Vector Machine (SVM) memiliki beberapa kelebihan yaitu: SVM juga sangat baik dalam kasus yang melibatkan margin klasifikasi yang jelas antara kategori. SVM juga sangat fleksibel karena dapat menggunakan *kernel trick*, yang memungkinkan pengklasifikasian data yang tidak linier dengan mengubah ruang fitur asli ke dalam dimensi yang lebih tinggi (Cortes & Vapnik, 2009). Salah satu kelemahan utama SVM adalah bahwa algoritma ini kurang efisien saat menangani dataset berukuran besar. SVM juga cenderung sensitif terhadap pemilihan parameter kernel yang tepat. Algoritma ini juga sulit untuk diinterpretasikan karena hasil klasifikasinya tidak memberikan probabilitas, tetapi hanya kategori, sehingga sulit memahami alasan di balik keputusan klasifikasi. Kelemahan lainnya adalah SVM tidak berkinerja baik saat digunakan untuk dataset yang memiliki banyak *noise* atau data yang saling tumpang tindih antara kelas satu dengan kelas lainnya (Cortes & Vapnik, 2009).

Decision tree menawarkan visualisasi yang jelas dan mudah dipahami mengenai proses pengambilan keputusan, *decision tree* dapat menangani data dengan baik meskipun ada fitur yang hilang atau tidak lengkap, *decision tree* mampu menangani data yang bersifat kategorikal maupun numerikal, kemampuannya dalam melakukan pemilihan fitur yang penting secara otomatis, sehingga membantu dalam penyederhanaan model dan meningkatkan efisiensi analisis (Maimon & Rokach, 2014). Salah satu kelemahan utama dari *decision tree* adalah

kecenderungannya untuk *overfitting*, *Overfitting* terjadi ketika model sangat sesuai dengan data pelatihan tetapi tidak dapat menggeneralisasi dengan baik pada data baru, decision tree seringkali tidak stabil terhadap perubahan kecil dalam data (Maimon & Rokach, 2014). Informasi mengenai perbandingan masing-masing metode klasifikasi dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Perbandingan metode klasifikasi

Metode	Kelebihan	Kekurangan	Kecocokan untuk Analisis Sentimen
Naïve Bayes Classifier	- Sederhana dan mudah diimplementasikan - Kecepatan tinggi dalam proses pelatihan dan prediksi - Memerlukan sedikit data untuk membuat model - Tahan terhadap *noise* - Efektif untuk masalah klasifikasi tanpa asumsi distribusi data kompleks	- Asumsi bahwa semua fitur independen - Kurang efektif untuk dasatet kecil atau tidak seimbang - Bias terhadap kelas mayoritas pada data tidak seimbang - Kurang cocok untuk data numerik yang sangat bervariasi	- Sangat cocok untuk analisis sentimen, terutama dalam klasifikasi teks - Efektif untuk dataset besar dengan fitur kategorikal - Kecepatan dan kesederhanaan membuat NBC sangat popular untuk tugas analisis sentimen
Support Vector Machine	- Baik dalam kasus dengan margin klasifikasi yang jelas - Fleksibel dengan penggunaan *kernel trick* - Dapat menangani data non-linear - Efektif dalam ruang berdimensi tinggi	- Kurang efisien untuk dataset berukuran besar - Sensitive terhadap pemilihan parameter kernel - Sulit diinterpretasikan karena tidak memberikan probabilitas - Kinerja buruk pada dataset dengan banyak *noise* atau tumpeng tindih antar kelas	- Cocok untuk analisis senntimen, terutama ketika ada pemisahan yang jelas antar kelas sentimen - Efektif untuk dataset menengah dengan fitur kompleks tetapi interpretasi hasil bisa menjadi sebuah tantangan
Decision Tree	- Visualisasi yang jelas dan mudah dipahami - Dapat menangani fata dengan fitur yang hilang - Mampu menangani data kategorikal dan numerikal	- Kecenderungan *overfitting* - Tidak stabil terhadap perubahan kecil dalam data - Bisa menjadi kompleks untuk dataset besar	- Cukup cocok untuk analisis sentiment, terutama ketika interpretasi model sangat penting - Bagus untuk dataset dengan campuran fitur kategorikal dan

	- Pemilihan fitur penting secara otomatis		numerikal. Namun perlu cermat terhadap *overfitting* pada dataset yang kompleks

Beberapa poin penting dari tabel tersebut:

1. *Naive Bayes Classifier* sangat cocok untuk analisis sentimen, terutama untuk klasifikasi teks dan dataset yang tidak berukuran kecil.
2. *Support Vector Machine* (SVM) efektif untuk analisis sentimen dengan pemisahan yang jelas antara kategori, tetapi interpretasinya bisa menjadi tantangan.
3. *Decision Tree* menawarkan interpretasi yang mudah dan bisa menangani berbagai jenis data, namun perlu berhati-hati terhadap overfitting.

Klasifikasi dalam *machine learning* digunakan dalam berbagai aplikasi, mulai dari pengenalan pola, analisis teks, pengenalan wajah, deteksi spam, hingga diagnosis medis. Salah satu algoritma yang populer dan sering digunakan untuk tugas klasifikasi adalah Naive Bayes.

Naïve Bayes adalah model machine learning yang sederhana namun efektif, terutama untuk tugas klasifikasi teks seperti analisis sentimen. Dibandingkan dengan model machine learning lain seperti *Support Vector Machines* (SVM) atau *Decision Tree, Naïve Bayes* memiliki keunggulan dalam hal kecepatan pelatihan dan prediksi, serta kemudahan implementasi. Namun, akurasinya cenderung lebih rendah dibandingkan dengan model yang lebih kompleks seperti SVM. Meskipun demikian, dalam penelitian ini, pembahasan kedua algoritma tersebut disajikan sebagai bagian dari tinjauan literatur untuk memberikan konteks komparatif terhadap pendekatan yang digunakan. *Naïve Bayes* tetap menjadi pilihan yang kompetitif untuk dataset yang memiliki fitur teks terstruktur seperti representasi TF-IDF.

Hasil penelitian menunjukkan bahwa metode *Naïve Bayes* memberikan kinerja analisis sentimen dengan tingkat akurasi sebesar 82,27%, *precision* sebesar 86,36%, dan *recall* sebesar 76,93% (Syahril Dwi Prasetyo, Shofa Shofiah Hilabi, & Fitri Nurapriani, 2023). Penelitian lain yang menggunakan metode klasifikasi berupa *Naive Bayes* dan *Support Vector Machine* menunjukkan bahwa Nilai akurasi implementasi klasifikasi yang diperoleh algoritma Naive Bayes sebesar 63,02% dan akurasi *support vector machine* sebesar 70,82%. Dapat disimpulkan bahwa algoritma support vector machine mempunyai nilai akurasi yang paling tinggi (Ningsih, Alfianda, Rahmaddeni, & Wulandari, 2024). Penelitian lain yang juga menggunakan metode klasifikasi berupa *Naive Bayes* dan *Support Vector Machine* menunjukkan bahwa *Naive Bayes*

Classifier mengungguli *Support Vector Machine* dalam hal klasifikasi bias mental dengan faktor 94,37%, sedangkan *Naive Bayes Classifier* mengungguli *Support Vector Machine* dengan faktor 86,87%. (Dwi Putra et al., 2023).

2.3 Klasifikasi Naïve Bayes

Klasifikasi Naïve Bayes adalah metode klasifikasi dalam machine learning yang berdasarkan pada Teorema Bayes dengan asumsi sederhana bahwa setiap fitur dalam data adalah independen satu sama lainnya, meskipun jarang terpenuhi sepenuhnya dalam dunia nyata (Berrar, 2018). Meskipun demikian, asumsi ini membuat proses perhitungan probabilitas kelas atau label menjadi lebih sederhana dan efisien.

Rumus dasar Teorema Bayes yang digunakan dalam *Naïve Bayes* adalah:

$$P(c|x) = \frac{P(x|c) \cdot P(c)}{P(x)}$$

dimana:

- $P(c|x)$ adalah probabilitas kelas c diberikan fitur x (posterior probability).
- $P(x|c)$ adalah probabilitas fitur x diberikan kelas c (likelihood).
- $P(c)$ adalah probabilitas prior dari kelas c.
- $P(x)$ adalah probabilitas prior dari fitur x (evidence).

Naïve Bayes mengasumsikan bahwa setiap fitur dalam x bersifat independen terhadap kelasnya, sehingga rumus dapat disederhanakan menjadi:

$$P(c|x) = \frac{P(x_1|c) \cdot P(x_2|c) \cdot \ldots \cdot P(x_n|c) \cdot P(c)}{P(x)}$$

Proses pelatihan dalam *Naïve Bayes* melibatkan perhitungan probabilitas dari setiap fitur dalam data latihan untuk setiap kelas yang ada. Saat melakukan prediksi terhadap data uji, model menghitung probabilitas dari setiap kelas berdasarkan fitur-fitur yang diberikan dan memilih kelas dengan probabilitas tertinggi sebagai prediksi kelas dari data tersebut.

Meskipun *Naïve Bayes* memiliki asumsi yang cukup sederhana, namun sering memberikan hasil yang baik dalam banyak kasus, terutama ketika memiliki jumlah fitur yang besar dan dataset yang cukup besar. Algoritma ini sering digunakan dalam klasifikasi teks seperti analisis sentimen, klasifikasi dokumen, atau deteksi spam, namun juga digunakan dalam berbagai bidang lainnya karena sifatnya yang cepat dan efisien (Kusnanda & Permana, 2023).

Pada proses pelatihan model ini, algoritma yang digunakan adalah Naïve Bayes Classifier dengan menggunakan model *Gaussian Naïve Bayes, Multinomial Naïve Bayes, serta Bernoulli Naïve Bayes*.

1. *Multinomial Naïve Bayes*

Multinomial naïve bayes ditentukan oleh serangkaian term, menggunakan istilah lain jumlah term diperhitungkan. Peluang antara term satu menggunakan yang lain merupakan independent (tidak bergantung). Pada Naïve Bayes, nilai peluang dokumen d pada kelas c bisa dihitung misalnya persamaan :

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

Keterangan :

c : kelas c

d : dokumen

P : peluang

Dengan memperhitungkan term nilai peluang dalam dokumen d pada kelas c bisa dihitung menggunakan persamaan :

$P(c \mid d) = P\ (c)\prod_{1\leq k\leq nd} P(fk|c)$

Keterangan :

P(c|d) : peluang kategori c pada dokumen d

P(c) : nilai peluang kategori c

P(fk|c) : peluang term fk yang ada pada kelas c

Parameter P(fk|c) di hitung dengan persamaan :

$$P(fk|c) = \frac{Tct + 1}{\sum t' \in VTct'+BF}$$

Tct merupakan peristiwa term t dalam sebuah dokumen menurut kelas c ∑t'ϵVTct' merupakan jumlah term yg masih ada dalam seluruh dokumen pada kelas c B merupakan jumlah variasi kata (vocabulary) yg masih ada pada data latih.

2. *Bernoulli Naïve Bayes*

Pada model Bernoulli, setiap data uji dihitung nilai probabilitas setiap kata untuk setiap kelas. Perbedaannya dengan contoh pertama adalah perhitungan probabilitas menggunakan jumlah data yang mengandung kata t (Tct), bukan frekuensi kemunculan kata t . Selain itu, jika contoh Multinomial menggunakan jumlah kata dan jumlah kosakata, model Bernoulli menggunakan jumlah data pelatihan per kelas (Tc)

dan jumlah kelas (Σc). Probabilitas setiap kata dihitung dengan menggunakan persamaan :

$$\mathrm{P(fk|c)} = \frac{\mathrm{Tct} + 1}{\mathrm{Tc} + \sum s}$$

Setelah diperoleh probabilitas setiap kata, maka dihitung nilai perkalian invers untuk probabilitas setiap kata pada data latih berbeda dengan kata pada data uji . Misalnya, terdapat kumpulan kata pada data latih d dan kumpulan kata pada data uji f. Kemudian temukan kata dalam data pelatihan d independen dari kata dalam data uji f. Dari kata-kata tersebut kemudian diturunkan nilai kata peluang (dalam hal ini dilambangkan P(fk'|c)) dan total kata (M). Nilai probabilitas kalimat untuk suatu kelas dihitung sebesar dengan persamaan :

$$P(c|d) = P(c) \prod_{i=1}^{N} P(fk|c)\ x\ \prod_{i+1}^{M} (1 - P(fk'\,|c))$$

3. Gausian Naïve Bayes

Ketika terdapat data kontinu, biasanya diasumsikan bahwa nilai kontinu yang terikat pada setiap kelas terdistribusi normal (Gaussian). Gaussian Naïve Bayes sangat cocok untuk membuat prediksi berdasarkan karakteristik distribusi normal. Probabilitas fitur yang diharapkan dihitung menggunakan persamaan.

$$\mathrm{P(xi\ |c)}\ = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(-\frac{(\mathrm{xi}-\mu\mathrm{c}\,)2}{2\sigma_c^2}\right)$$

Keterangan:

xi : likelihood or value (atribut|xi)

σ : standar deviasi atribut (atribut|xi)

μ : rata-rata (atribut|xi)

Setelah ketiga model *Naive Bayes* (*Gaussian, Multinomial, dan Bernoulli*) diterapkan untuk klasifikasi, evaluasi kinerja menjadi langkah krusial untuk membandingkan performa masing-masing model. Evaluasi ini dilakukan untuk menentukan model mana yang paling efektif dalam menganalisis data yang belum pernah dilihat sebelumnya. Metode *cross-validation* sering digunakan untuk mengevaluasi kemampuan generalisasi setiap model. Selain itu, metrik evaluasi seperti akurasi untuk mengukur proporsi prediksi yang benar, presisi untuk menilai ketepatan prediksi kelas tertentu, recall untuk mengukur sensitivitas terhadap data kelas tertentu, dan F1-score sebagai kombinasi keduanya digunakan untuk membandingkan

keandalan dan keseimbangan performa masing-masing model dalam memprediksi kelas pada data baru.

2.4 *Cross Validation*

Cross validation adalah teknik evaluasi yang digunakan untuk memperkirakan kinerja model machine learning pada data yang belum pernah dilihat sebelumnya (Ramezan, Warner, & Maxwell, 2019). Tujuannya adalah untuk memastikan bahwa model yang dibuat mampu melakukan generalisasi dengan baik terhadap data baru. *Cross-validation* juga bertujuan untuk mengoptimalkan parameter model dan mencegah overfitting dengan mengevaluasi model pada data yang belum pernah dilihat selama pelatihan (Y. Liu & Liao, 2017). Salah satu bentuk cross validation yang umum digunakan adalah *k-Fold cross validation*.

Cross-validation adalah metode evaluasi model yang sangat penting dalam machine learning, terutama karena kemampuannya untuk memberikan estimasi performa model yang lebih akurat dan menghindari overfitting. Salah satu kelebihan utama dari cross-validation adalah kemampuannya untuk menggunakan data secara lebih efisien, karena seluruh dataset dapat digunakan baik untuk pelatihan maupun pengujian secara bergantian, *Cross-validation* dapat diterapkan pada berbagai algoritma machine learning, dari model sederhana hingga yang kompleks, dan Model yang dioptimalkan melalui *cross-validation* cenderung memiliki generalisasi yang lebih baik ketika diterapkan pada data baru yang belum pernah dilihat sebelumnya, sehingga meningkatkan keandalan hasil prediksi (Kohavi & Edu, 1995).

Model yang dilatih dengan *cross-validation* umumnya lebih *generalizable* dan dapat bekerja dengan baik pada data baru yang tidak terlihat sebelumnya. *Generalizability* adalah kemampuan model untuk memprediksi dengan baik pada data yang berbeda dari data yang digunakan untuk melatih model. *Cross-validation* dapat digunakan untuk membandingkan kinerja model yang berbeda pada dataset yang sama. Hal ini dapat membantu untuk memilih model yang terbaik untuk tugas yang diberikan (Hafid, 2023).

Terdapat beberapa jenis cross validation yang umum digunakan, antara lain:

a. *K-fold cross validation*: Data dibagi menjadi k sub-set yang sama besar. Model dilatih pada k-1 sub-set dan diuji pada sub-set yang tersisa. Proses ini diulang k kali dengan menggunakan sub-set yang berbeda sebagai data latih dan data uji. Nilai k yang umum digunakan adalah 5, 10, atau 20.

b. *Leave-one-out cross validation*: Pada metode ini, model dilatih pada semua data kecuali satu data point. Kemudian, model diuji pada data point yang tersisa. Proses ini diulang untuk setiap data point dalam dataset.
c. *Stratified cross validation*: Metode ini digunakan untuk dataset yang tidak seimbang. Data dibagi menjadi sub-set berdasarkan proporsi kelas target dalam dataset. Kemudian, cross validation dilakukan pada sub-set yang telah terstratifikasi.
d. *Repeated cross validation*: Pada metode ini, cross validation diulang beberapa kali dengan menggunakan pembagian data yang berbeda setiap kali. Hal ini dapat membantu untuk mengurangi variabilitas dalam hasil *cross validation*.

Pada *k-Fold cross validation*, dataset dibagi menjadi k bagian (*fold*) yang seukuran. Secara umum, langkah-langkahnya adalah sebagai berikut (Fushiki, 2011):

a. Pembagian Dataset
   Dataset dibagi menjadi k bagian dengan ukuran yang sama.
b. Iterasi Training dan Evaluasi
   Model dilatih dan dievaluasi sebanyak k kali. Pada setiap iterasi, salah satu *fold* dijadikan data uji, sementara *k-1 fold* lainnya digunakan sebagai data latih. Model dilatih pada data latih dan dievaluasi pada data uji.
c. Pengukuran Performa
   Performa model diukur menggunakan metrik evaluasi seperti akurasi, presisi, *recall*, atau *F1-score*.
d. Perhitungan Rata-rata Performa

Nilai performa yang dihasilkan dari setiap iterasi dapat dijumlahkan dan dirata-ratakan untuk mendapatkan estimasi performa keseluruhan model.

2.5 *Performance Evaluation Measure* dengan *Confusion Matrix*

Performance evaluation measures merupakan metrik atau ukuran yang digunakan untuk mengukur kinerja dan akurasi dari model ML dalam melakukan prediksi atau klasifikasi pada data yang belum pernah dilihat sebelumnya (Jiao & Du, 2016). Tujuan utamanya adalah untuk memberikan pemahaman yang lebih mendalam tentang sejauh mana model tersebut dapat memprediksi dengan tepat dan akurat terhadap data baru (Ferri, Hernández-Orallo, & Modroiu, 2009).

Performance Evaluation Measure biasanya digambarkan dengan *confusion matrix* yang merupakan tabel yang memungkinkan pengguna untuk menilai sejauh mana model klasifikasi

berhasil memprediksi kelas-kelas yang benar dalam dataset tertentu. Matriks ini terdiri dari empat komponen utama: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) (Fakhruddin, Kom, & Ridwan, 2023).

Confusion matrix memiliki tiga keunggulan utama dalam evaluasi kinerja model klasifikasi. Pertama, *confusion matrix* memberikan gambaran detail mengenai performa model dengan menunjukkan distribusi prediksi yang benar dan salah melalui elemen-elemen seperti *True Positive, False Positive, True Negative, dan False Negative*. Ini membantu evaluator memahami kesalahan model dan memberikan wawasan lebih mendalam dibandingkan metrik akurasi saja, terutama pada dataset yang tidak seimbang. Kedua, *confusion matrix* memungkinkan perhitungan berbagai metrik seperti *precision, recall, F1-score,* dan *specificity*, yang penting dalam konteks tertentu seperti bidang kesehatan, di mana recall lebih penting daripada akurasi untuk menghindari kesalahan prediksi yang fatal. Ketiga, *confusion matrix* fleksibel dalam menangani data tidak seimbang, memberikan evaluasi lebih adil dengan memperlihatkan performa model dalam memprediksi kelas minoritas yang sering terabaikan oleh metrik akurasi (Saito & Rehmsmeier, 2015).

Performance Evaluation Measure biasanya digambarkan dengan *confusion matrix* seperti yang diperlihatkan pada Tabel 2.2.

Tabel 2. 2 Confusion Matrix

	Predicted Class		
True Class	**Positive**	**Negative**	**Neutral**
Positive	TP	FN1	FNe1
Negative	FP1	TN	FNe2
Neutral	FP2	FN2	TNe

dengan:

- TP adalah *True Positives*
- TN adalah *True Negatives*
- TNe adalah *True Neutrals*
- $FP1$ adalah *False Positive* 1
- $FP2$ adalah *False Positive* 2
- $FN1$ adalah *False Negative* 1
- $FN2$ adalah *False Negative* 2
- $FNe1$ adalah *False Neutral* 1
- $FNe2$ adalah *False Neutral* 2

Beberapa metrik evaluasi yang umum digunakan meliputi aspek seperti akurasi, presisi, *recall*, serta *f1-score*:

a. Akurasi (*Accuracy*)

Merupakan metrik yang mengukur seberapa sering model memberikan prediksi yang benar secara keseluruhan terhadap seluruh data. Akurasi dihitung dengan membagi jumlah prediksi yang benar dengan total prediksi yang dilakukan.

Rumusnya adalah sebagai berikut:

$$Akurasi = \frac{TP + TN + TNe}{Total}$$

b. Presisi (*Precision*)

Menunjukkan seberapa tepat model dalam memprediksi positif. Presisi dihitung dengan membagi jumlah prediksi positif yang benar dengan total prediksi positif yang dilakukan oleh model.

Rumusnya adalah sebagai berikut:

$$Presisi = \frac{TP}{TP + FP1 + FP2}$$

c. *Recall* (*Sensitivity*)

Mengukur seberapa baik model dapat mendeteksi nilai positif dari seluruh nilai yang benar positif yang ada dalam dataset. *Recall* dihitung dengan membagi jumlah prediksi positif yang benar dengan total nilai benar positif.

Rumusnya adalah sebagai berikut:

$$Recall = \frac{TP}{TP + FN1 + FNe1}$$

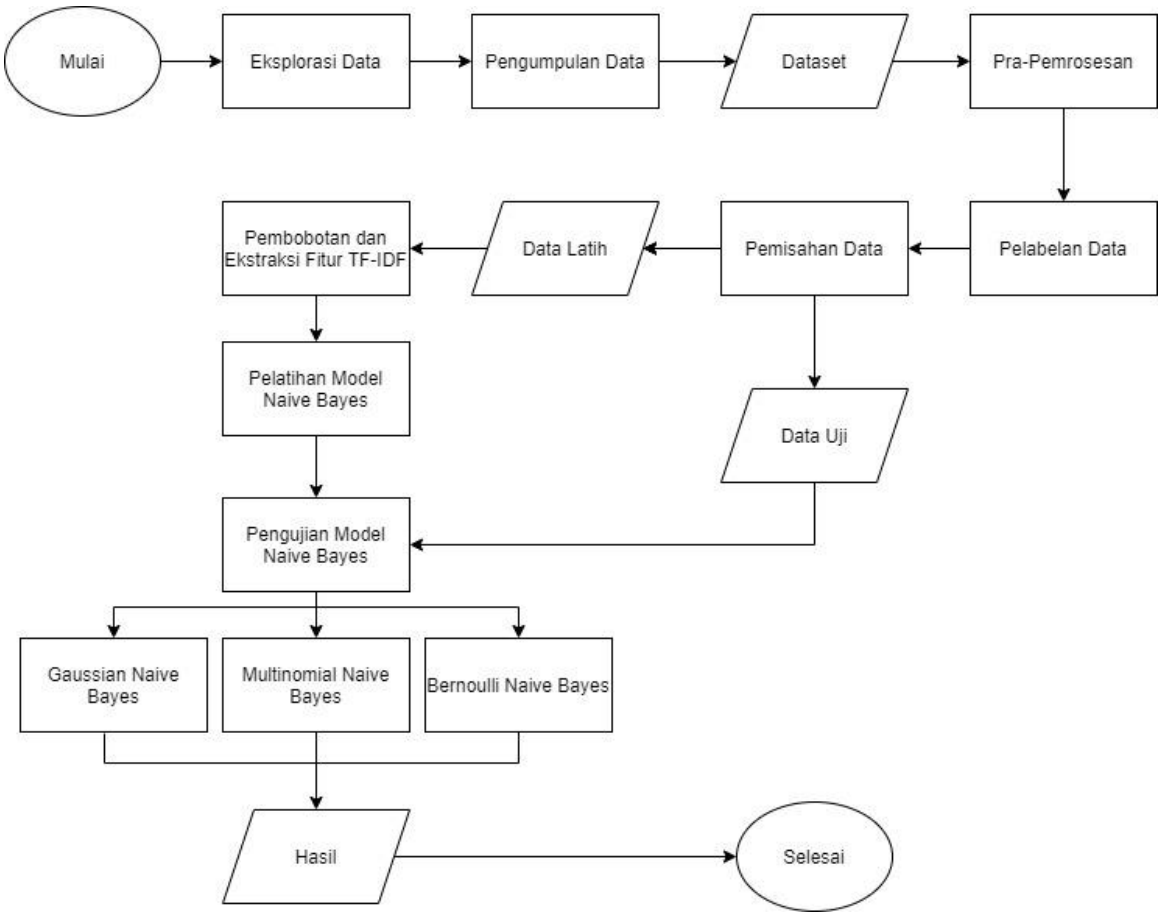
d. F1-Score

F1-Score adalah ukuran kinerja model klasifikasi yang digunakan untuk menyeimbangkan antara *precision* (ketepatan) dan *recall* (cakupan) suatu model. *F1-Score* merupakan harmonisasi rata-rata dari *precision* dan *recall*, memberikan bobot yang sama pada kedua metrik ini. Rumus *F1-Score* adalah sebagai berikut:

$$f1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

BAB III
PERANCANGAN ANALISIS SENTIMEN

Perancangan analisis sentimen merupakan proses-proses yang akan dilakukan sebelum pelaksanaan penelitian. Perancangan ini dijadikan acuan selama penelitian agar hasil yang diharapkan dapat tercapai. Proses yang akan dilakukan dapat dilihat pada Gambar 3.1 berikut.



Gambar 3. 1 Diagram Alir Perbandingan Performansi 3 Model Naive Bayes

Pada penelitian ini ada beberapa tahapan atau proses yang dilakukan mulai dari tahap eksplorasi data dari akun Youtube Mobile Legends: Bang Bang. Kemudian dilanjutkan dengan tahap pengumpulan data dengan melakukan scrapping data pada kolom *live chat* siaran Grand Final MSC 2023 pada akun Youtube tersebut menggunakan extension berupa *CommentStack* pada browser Microsoft Edge. Tahap berikutnya adalah pra-pemrosesan data yang meliputi *cleaning, case folding, tokenizing, stopword removal, stemming*, serta pembersihan data secara manual dan pelabelan data. Setelah pra-pemrosesan data, tahap yang dilakukan adalah melakukan pembobotan menggunakan algoritma *Term Frequency – Inverse Document Frequency (TF-IDF).* Tahap berikutnya pelatihan model yang merupakan proses melatih

model menggunakan data latih yang kemudian akan dievaluasi menggunakan data uji untuk mengukur performanya. Setelah pelatihan model, tahap yang dilakukan adalah evaluasi model menggunakan *Performance Evaluation Measure* beserta *Cross Validation* yang kemudian dapat menampilkan hasil evaluasi dari model tersebut yang merupakan accuracy, precision, dan recall. Tahap terakhir adalah visualisasi data dimana hasil dari analisis sentimen dan evaluasi model disajikan dalam bentuk visual seperti grafik maupun *WordCloud*. Visualisasi ini membantu dalam mengintepretasikan hasil dan menyampaikan temuan penelitian secara lebih jelas dan informatif.

3.1 Eksplorasi Data

Data yang digunakan untuk penelitian ini merupakan data primer. Data primer merupakan hasil pencarian data yang diambil secara langsung. Pada penelitian ini, data diambil secara langsung dari rekaman siaran langsung pertandingan Grand Final MSC (MLBB Southeast Asia Cup) 2023 pada platform Youtube. Data yang digunakan merupakan pesan-pesan pada *live chat* siaran Grand Final MSC 2023 berbahasa Indonesia yang disiarkan di channel YouTube Mobile Legends: Bang Bang pada tanggal 18 Juni 2023.

Siaran Grand Final MSC 2023 tersebut menayangkan pertandingan antara Onic Indonesia dan Blacklist International. Onic Indonesia, yang merupakan salah satu perwakilan dari Indonesia memenangkan pertandingan final tersebut sekaligus membawa kembali gelar juara MSC ke Indonesia setelah 4 tahun. Berdasarkan data dari E-Sport Charts, siaran Grand Final MSC 2023 mencapai *peak viewers* dengan jumlah 3.650.305 penonton pada berbagai platform. Platform terpopuler pada MSC 2023 adalah Youtube, sedangkan untuk bahasa terpopulernya adalah Bahasa Indonesia. Oleh karena itu, dataset yang diambil dari siaran Grand Final MSC berbahasa Indonesia tersebut dianggap cukup representatif untuk menganalisis sentimen komunitas Mobile Legends karena memiliki jumlah penonton yang besar, merupakan acara puncak dari salah satu kompetisi paling bergengsi di kancah *e-sport* Mobile Legends, dan sesuai dengan bahasa yang digunakan mayoritas komunitas *e-sport* Mobile Legends yaitu bahasa Indonesia.

Pada Langkah pertama adalah memahami struktur data dengan meninjau jumlah total pesan dan tipe data yang ada. Pembersihan data dilakukan dengan menghapus pesan duplikat dan menangani nilai yang hilang untuk memastikan keakuratan analisis. Analisis teks dilakukan dengan tokenisasi untuk memecah pesan menjadi kata-kata atau frasa. Identifikasi pola dan tren melibatkan analisis perubahan sentimen seiring waktu dan korelasi antara kata-kata tertentu dengan sentimen yang dihasilkan. Ekstraksi fitur dilakukan dengan menghitung

frekuensi kata dan menggunakan metode *TF-IDF (Term Frequency-Inverse Document Frequency)* untuk mengukur pentingnya kata dalam pesan. Statistik deskriptif digunakan untuk menghitung frekuensi kata dan distribusi sentimen (positif, negatif, netral), yang divisualisasikan melalui grafik dan word cloud untuk memberikan gambaran visual tentang kata-kata yang sering muncul.

3.2 Pengumpulan Data

Metode yang digunakan untuk pengumpulan data dari penelitian ini adalah penggunaan CommentStack untuk perekaman live chat Youtube. Pada penelitian ini platform CommentStack digunakan sebagai alat untuk mengumpulkan data dari rekaman *live chat* Youtube. CommentStack memungkinkan peneliti untuk mengakses dan merekam percakapan yang terjadi selama siaran live di Youtube yang kemudian dapat langsung diekspor tanpa perlu menyalin secara manual. Pengumpulan data yang digunakan dalam penelitian ini dilakukan pada tanggal 27 Agustus 2023.

Gambar 3.2 berikut merupakan contoh file .csv yang berhasil diunduh:

	A	B	C	D	E	F	G	H	I	J	K
1	Author	Date	Author Na	Role	Message	Video Tim	ID				
2	https://yt	2023-08-2	SYAIFUS SHOLIHIN O		fans evos	01:36:50	ChwKGkNNaXZ5X0hXelA4Q0ZWREN3Z1FkM2pzSVpR				
3	https://yt	2023-08-2	Hizkia effendy		everyone	01:36:50	ChwKGkNNbi0zdkhXelA4Q0ZkSEN3Z1FkUTZVTHB3				
4	https://yt	2023-08-2	Fendi Khan_chiL		text-greer	01:36:51	ChwKGkNOeIlqZkxXelA4Q0ZhN0R3Z1Fkb09vRlB3				
5	https://yt	2023-08-2	NIAA`03		kebanyak	01:36:51	ChwKGkNNQ0FsdkxXelA4Q0ZWSEJ3Z1FkRlZRSjRn				
6	https://yt	2023-08-2	Haerul suruga		jasjus ena	01:36:51	ChwKGkNJeXdtZkxXelA4Q0ZTY19yUVlkc0NzSTJR				
7	https://yt	2023-08-2	Alfin Mahendra putr		intro nya l	01:36:51	ChwKGkNOYmpvZkxXelA4Q0ZhVEF3Z1FkN3pzSU1B				
8	https://yt	2023-08-2	Radenmas Yosi		?????????	01:36:51	ChwKGkNOYmNvX0xXelA4Q0ZiTEN3Z1FkczRjQUxn				
9	https://yt	2023-08-2	marco		YANG SAR	01:36:51	ChwKGkNNSGZvX0xXelA4Q0ZVOF9yUVlkM2UwUHF3				
10	https://yt	2023-08-2	Akang Muhamadriza		kapan mu	01:36:51	ChwKGkNLdWZxX0xXelA4Q0ZiUEN3Z1FkTEUwTjdB				
11	https://yt	2023-08-2	Rakha Agustira		pada rasis	01:36:51	ChwKGkNPbVJydkxXelA4Q0ZVVVpmUW9kakk0RmNn				
12	https://yt	2023-08-2	Gabriel Yuda		jam berap	01:36:51	ChwKGkNLX2NzUExXelA4Q0ZiUEN3Z1FkTEUwTjdB				
13	https://yt	2023-08-2	Indra99		kingdom?	01:36:52	ChwKGkNOT2x4dkxXelA4Q0ZWREN3Z1FkM2pzSVpR				
14	https://yt	2023-08-2	Tegar Tegar		face-turqu	01:36:52	ChwKGkNKRzl6UExXelA4Q0ZiTEN3Z1FkczRjQUxn				
15	https://yt	2023-08-2	Lukman Bae		kpn main	01:36:52	ChwKGkNKN0wydkxXelA4Q0ZZekN3Z1FkdkJ3TVNR				
16	https://yt	2023-08-2	Yusof Zaky		ban estes	01:36:53	ChwKGkNNajI2UExXelA4Q0ZVOF9yUVlkM2UwUHF3				

Gambar 3. 2 Contoh cuplikan dataset

3.3 Pra-pemrosesan Data

Tahap pra-pemrosesan merupakan langkah yang dilakukan setelah proses pengumpulan data. *Pre-processing* adalah tahap persiapan data dalam analisis atau pemrosesan data sebelum diterapkan pada algoritma atau model tertentu. Kualitas data memainkan peran penting dalam menentukan kualitas keluaran yang dihasilkan ketika menggunakan algoritma apa pun (Roy, Sharma, Nath, Bhattacharyya, & Kalita, 2018). Tujuan dari *pre-processing* sendiri adalah untuk mengubah data mentah menjadi bentuk yang lebih sesuai untuk analisis dan model yang akan diterapkan.

Untuk memastikan bahwa *pre-processing* tidak menghilangkan informasi penting yang relevan bagi analisis sentimen, digunakan pendekatan standar dalam pembersihan data teks, termasuk penerapan daftar *stopword* dan proses stemming dari *library* Sastrawi. Pendekatan ini diharapkan dapat membersihkan teks dari elemen-elemen yang tidak signifikan dan menyederhanakan kata-kata ke bentuk dasar tanpa mengorbankan informasi penting yang diperlukan untuk analisis.

Berikut adalah langkah-langkah *pre-processing* yang digunakan dalam penelitian ini yang terdiri dari *cleaning, case folding, tokenizing, stopword removal, stemming*, dan pembersihan serta pelabelan data

3.3.1 Cleaning

Cleaning data adalah salah satu langkah dalam menciptakan data berkualitas (Herbert & Wang, 2007). Biasanya menangani dua situasi yang menimbulkan masukan yang tidak konsisten, nilai yang hilang, salah ejaan, dll. Pertama berkaitan dengan nilai yang hilang dan yang kedua adalah masalah duplikasi data (Roy et al., 2018). *Cleaning* data merupakan tahap pembersihan data dari hal yang tidak diperlukan seperti kolom, tanda baca, hashtag, *link*, serta nama pengguna.

Berikut adalah dataset sebelum dilakukan penghapusan kolom yang tidak diperlukan, terdapat kolom yang tidak diperlukan seperti *Author*, *Date*, *Author Name*, *Role*, *Video Time*, dan kolom *ID*:

Tampilan dataset sebelum penghapusan kolom yang tidak diperlukan dapat dilihat pada Gambar 3.3.

	Author	Date	Author Name	Role	Message	Video Time
0	https://yt4.ggpht.com/AIQaVBOHKRvkEPwah5VzNZkS...	2023-08-25T18:36:00.000Z	SYAIFUS SHOLIHIN OFFICIAL	NaN	fans evos 100% dukung blacklist	01:36:50
1	https://yt4.ggpht.com/ytc/AOPolaSoeC3InbKXHrVj...	2023-08-25T18:36:00.000Z	Hizkia effendy	NaN	everyone know kairi and coach yeb from philipines	01:36:50
2	https://yt4.ggpht.com/ytc/AOPolaTY8QxglXNwqc31...	2023-08-25T18:36:00.000Z	Fendi Khan_chiL	NaN	text-green-game-overtext-green-game-overtext-g...	01:36:51
3	https://yt4.ggpht.com/SKKXFRAu0DTzJhumxImvJqsg...	2023-08-25T18:36:00.000Z	NIAA`03	NaN	kebanyakan bocil rasisss ??????	01:36:51
4	https://yt4.ggpht.com/ytc/AOPolaRnvXQzUubrd6dO...	2023-08-25T18:36:00.000Z	Haerul suruga	NaN	jasjus enak	01:36:51
...	...	...	...	...	...	...
45330	https://yt4.ggpht.com/ytc/AOPolaRX6YwlpftFt3B9...	2023-08-25T23:06:00.000Z	Calvin Chandra	NaN	ZEYS MENYESAL, LBH MILIH DREAMS DRPD KIBOY	06:06:02

Gambar 3. 3 Contoh sebelum pembersihan kolom yang tidak diperlukan

Tampilan dataset setelah penghapusan kolom yang tidak diperlukan dapat dilihat pada Gambar 3.4.

	Message
0	fans evos 100% dukung blacklist
1	everyone know kairi and coach yeb from philipines
2	text-green-game-overtext-green-game-overtext-g...
3	kebanyakan bocil rasisss ??????
4	jasjus enak
...	...
45330	ZEYS MENYESAL, LBH MILIH DREAMS DRPD KIBOY
45331	MVP ROAMER NI BOS. BUKAN KALENG2 KAYAK IESF. P...
45332	Congrats Onic on behalf of Filipino Supporter ...
45333	FRENGKY .ANA
45334	gg

45335 rows × 1 columns

Gambar 3. 4 Contoh setelah penghapusan kolom yang tidak diperlukan

Sebagai contoh, hasil penghapusan tanda baca dari pesan-pesan tersebut disajikan pada Tabel 3.1

Tabel 3. 1 Contoh Hasil Penghapusan Tanda Baca

Pesan	Seteleah Proses Penghapusan Tanda Baca
Go Indo, udah bosan PH pegang terus. From Malaysia	Go Indo udah bosan PH pegang terus From Malaysia
#ONICWIN #GOONIC	ONICWIN GOONIC
ONIC TAKE OUR ENERGY!!!???????	ONIC TAKE OUR ENERGY
KASIH PAHAM BOY ????????????	KASIH PAHAM BOY
GO ONIC GO ONIC GOO..... DARI FANS LEMON INDONESIA	GO ONIC GO ONIC GOO DARI FANS LEMON INDONESIA

Beberapa sampel pesan yang terduplikasi dan kemudian dihapus dalam tahap pra-pemrosesan dapat dilihat pada Tabel 3.2

Tabel 3. 2 Contoh pesan yang terdupliklasi

Sampel pesan yang terduplikasi
gg onic
kelas
bismillah
onic win
lama
lag
burik
indopride
lemon gay
blacklist

3.3.2 Case Folding

Case Folding merupakan proses yang digunakan untuk mengubah semua huruf dalam teks menjadi bentuk yang sama yaitu huruf kapital atau huruf kecil semua. Dalam penelitian ini, semua huruf yang ada dalam data rekaman live chat diseragamkan menjadi huruf kecil. Contoh proses case folding dapat dilihat pada Tabel 3.2.

Tabel 3. 3 Contoh proses *case folding*

Pesan	Setelah Proses Case Folding
KASIH PAHAM BOY	kasih paham boy
GO ONIC GO ONIC GOO	go onic go onic goo
Support dari malaysia	support dari malaysia
HARUS CURI POINT game 1	harus curi point game 1

3.3.3 Tokenizing

Tokenizing merupakan adalah proses memecah dokumen menjadi kumpulan kata. Proses yang digunakan untuk membagi teks menjadi unit-unit terpisah, yang disebut sebagai token dengan memisahkan setiap kata berdasarkan karakter spasi. *Tokenizing* digunakan untuk mendapatkan urutan token yang kemudian dapat diproses lebih lanjut. *Tokenization* dapat dilakukan dengan menghilangkan tanda baca dan memisahkannya per spasi. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua token ke bentuk huruf kecil (*lower case*) (Nurjannah & Fitri Astuti, 2013). Contoh proses tokenisasi dapat dilihat pada Tabel 3.4.

Tabel 3. 4 Contoh proses *tokenizing*

Pesan	Setelah Proses Tokenizing
fans evos 100 dukung blacklist	[fans, evos, 100, dukung, blacklist]
bendera indo mana oy	[bendera, indo, mana, oy]
ban estes digi	[ban, estes, digi]
go onic go	[go, onic, go]
bismilah onic	[bismilah, onic]

3.3.4 Stopword Removal

Stopword merupakan kata-kata umum yang biasanya muncul dalam jumlah banyak dan dianggap tidak memiliki makna. Tujuan dihilangkannya *stopwords* adalah agar diperoleh kata yang berkaitan dengan analisis dokumen (Jannah & Prasetyo, 2022). Pada proses *stopwords removal* ini, peneliti menghapus kata-kata yang umum dan tidak memiliki nilai informatif dari teks, seperti “dan”, “atau”, “dari”, “itu”, dan sebagainya. Contoh hasil penghapusan kata-kata yang tidak penting dari pesan-pesan ditunjukkan pada Tabel 3.5

Tabel 3. 5 Contoh proses *stopword removal*

Pesan	Setelah Penghapusan Stopword
ada pak pulung dan ranger emas pertandingan semakin seru	pak pulung ranger emas pertandingan semakin seru
menang atau kalah itu wajar	menang kalah wajar
dari jam 4 sampai sekarang	jam 4 sekarang
seperti biasa meta pause	biasa meta pause
pada sara isi komen	sara isi komen

3.3.5 Stemming

Stemming merupakan proses normalisasi teks. proses mengembalikan kata menjadi kata dasarnya. Hal ini bisa dilakukan dengan cara menghilangkan akhiran atau awalan dari sebuah kata. bertujuan untuk mengubah kata menjadi ke dalam bentuk dasar atau akar kata sesuai KBBI. Proses stemming menggunakan *library* Python Sastrawi. Tujuan utama dari proses ini adalah untuk menghilangkan imbuhan agar kata-kata yang memiliki makna sama tapi dengan bentuk yang berbeda dapat dianggap identik dalam analisis teks (Nurjannah & Fitri Astuti, 2013). Contoh proses stemming dapat dilihat Tabel 3.6.

Tabel 3. 6 Contoh proses stemming

Pesan	Setelah Proses Stemming
udah menang unggul kebiasaan dah	udah menang unggul biasa dah
pertandingan semakin seru	tanding makin seru
berguna buat game	guna buat game
rrq kemana	rrq mana
kalahkan negara pelangi	kalah negara pelangi

3.3.6 Pembersihan Data dan Pelabelan Data

Pembersihan dan pelabelan data dilakukan secara manual. Pembersihan dan pelabelan data secara manual dibutuhkan untuk memastikan kualitas dan keandalan data sebelum dilanjutkan ke tahap analisis atau pemodelan lebih lanjut. Kemampuan manusia untuk memahami konteks dan menangani masalah yang kompleks membuat pembersihan data secara manual tetap menjadi pendekatan yang penting dalam pengolahan data. Pelabelan data dilakukan dengan mencocokkan teks *live chat* dengan konteks yang ditayangkan dalam siaran langsung di YouTube. Proses ini bertujuan untuk memastikan bahwa pelabelan sentimen sesuai dengan konteks situasi atau acara yang sedang berlangsung. Dengan cara ini, setiap komentar atau teks yang muncul dapat lebih tepat dikategorikan ke dalam kelas sentimen yang sesuai, karena memperhitungkan konteks visual dan verbal dari tayangan yang terkait. Contoh

pembersihan data manual yang dapat dilakukan adalah menghapus emoji yang sudah terkonversi menjadi teks dan menghapus komentar-komentar yang tidak relevan dengan objek penelitian ini. Contoh penghapusan emoji dapat dilihat pada Tabel 3.7

Tabel 3. 7 Contoh penghapusan emoji yang terkonversi menjadi teks

Emoji	Emoji yang terkonversi menjadi teks
	textgreengameover
	elbowbump
	buffering
	eyespurplecrying
	chillwdog
	facebluesmiling

Setelah melalui proses pre-processing, data yang tersisa menjadi 19.877 pesan dari data awal yang berjumlah 45.335 pesan.

Pelabelan dataset dilakukan secara manual yang dilakukan oleh peneliti. Ada 3 jenis label yang digunakan dalam pelabelan data yaitu label positif, negatif dan netral. Pelabelan dilakukan dengan cara melihat teks pesan beserta konteksnya dalam rekaman siaran langsung Grand Final MSC 2023.

3.4 Pembobotan TF-IDF

Setelah melewati tahap pre-processing, data yang akan digunakan harus berbentuk angka agar data yang masih berupa kata-kata dapat diolah dan dihitung. Data yang masih belum berupa angka selanjutnya akan diubah menjadi vector kemudian diberi nilai dan bobot untuk setiap katanya yang kemudian akan akan diolah menggunakan algoritma prediksi. Pada tahap pembobotan kata, algoritma yang digunakan adalah *Term Frequency – Inverse Document Frequency* (TF-IDF).

Metode ini merupakan algoritma yang melakukan penggabungan dua metode yaitu konsep frekuensi kemunculan *term* dalam sebuah dokumen dan *inverse* frekuensi dokumen yang mengandung kata tersebut, akan mampu meningkatkan proporsi jumlah dokumen yang dapat ditemukan kembali dan yang dianggap relevan secara sekaligus. Sehingga kriteria term yang paling tepat adalah *term* yang sering muncul dalam dokumen secara individu, namun jarang dijumpai pada dokumen lainnya (Widyasanti, Darma Putra, & Dwi Rusjayanthi, 2018).

Metode *TF-IDF* merupakan suatu cara untuk memberikan bobot hubungan suatu kata *(term)* terhadap dokumen. Frekuensi kemunculan kata di dalam dokumen yang diberikan

menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut yang rendah pada kumpulan dokumen (Nurjannah & Fitri Astuti, 2013).

Sebelum dilakukan pembobotan, data yang akan digunakan akan dibagi menjadi data latih dan data uji melauli proses yang bernama *Data Splitting*. Pada *data splitting*, akan memisahkan dataset menjadi dua bagian yaitu data latih dan data uji (Amardita, Adiwijaya, & Purbolaksono, 2022). Pada penelitian ini *split data* menggunakan *train/test split*. *Data training* adalah data yang akan dilatih dengan metode Naïve Bayes sedangkan *data testing* adalah data yang akan dilakukan untuk prediksi berdasarkan pelatihan data yang sudah ditraining. Pada pembagian data ini selain menggunakan *split* bisa juga dilakukan manual, artinya pembagian data langsung dipisah antara data training dan data testing. Walaupun dengan dataset yang sama, jika berbeda cara pembagiannya maka akan mempengaruhi hasilnya. Jika melakukan pembagian manual, record data yang dibagi sudah pasti, tetapi jika dilakukan pembagian dengan split data maka sistem akan membagi secara acak (Astuti, Wulandari, Putra, & Kharomadhona, 2022). Rasio yang umum digunakan adalah 80:20, yang berarti 80% data untuk pelatihan dan 20% untuk pengujian. Rasio lain seperti 70:30, 60:40, dan bahkan 50:50 juga digunakan dalam praktik (Amardita et al., 2022).

Pembagian data live chat yang digunakan pada proses ini adalah rasio 80:20 yang merupakan 80% data akan digunakan sebagai data latih dan 20% sisanya akan digunakan sebagai data uji. Rasio pembagian data latih dan data uji ini dirujuk dari penelitian-penelitian sebelumnya yang juga menggunakan rasio 80:20. Jumlah data yang digunakan dalam dataset penelitian ini berjumlah 19877 data sehingga kemudian akan dibagi dengan rasio tersebut diatas menjadi 15901 data latih dan 3976 data uji.

Rumus untuk TF:

$$TF(t,d) = \frac{Jumlah\ kemunculan\ t\ dalam\ d}{Total\ jumlah\ kata\ dalam\ d}$$

Keterangan rumus *TF*:

- *t* = Term yang dihitung frekuensinya
- *d* = Dokumen yang berisi term *t*

Rumus untuk IDF:

$$IDF(t) = \log\left(\frac{N}{1 + DF(t)}\right) + 1$$

Keterangan rumus *IDF*:

- *t* = Term yang dihitung IDF-nya
- *N* = Total jumlah dokumen dalam korpus
- *DF(t)* = Jumlah dokumen yang mengandung term *t*

Rumus untuk TF-IDF:

$$TF\ IDF(t,d) = TF(t,d).IDF(t)$$

Keterangan rumus TF-IDF:

- TF(*t*,*d*) = *Term Frequency* dari term t dalam dokumen *d*
- IDF(*t*) = *Inverse Document Frequency* dari term *t*

Metode ini dipilih karena efisien, mudah serta memiliki hasil yang akurat. Sederhananya, metode ini digunakan untuk menghitung frekuensi munculnya suatu kata atau term dalam sebuah dokumen. *Term Frequency* adalah frekuensi kemunculuan sebuah kata atau term dalam dokumen. Semakin tinggi TF atau semakin besar jumlah kata atau *term* yang muncul maka semakin besar bobot yang menunjukkan pentingnya kata tersebut dalam dokumen. Ini akan membantu dalam mengekstrak informasi penting dari teks dan meningkatkan kerja algoritma *Machine Learning* dalam tugas klasifikasi yang akan dilakukan. Contoh nilai tf-idf yang dihasilkan dapat dilihat pada Tabel 3.8.

Tabel 3. 8 Contoh nilai TF-IDF

garansi onic juara	
garansi	0.858158127
onic	0.2751677
juara	0.433898391

Diketahui dalam korpus data latih yang memiliki dokumen berjumlah 15901, terdapat sebuah dokumen yang berisikan term (garansi onic juara). Dokumen tersebut akan digunakan sebagai contoh formulasi pembobotan TF-IDF. Berikut merupakan contoh proses formulasi pembobotan TF-IDF yang dilakukan pada dokumen tersebut.

Dari data yang ada, dapat dicari dan diketahui bahwa :

- Dokumen *d* : “garansi onic juara”

- Jumlah korpus data latih : 15901
- *Term* garansi muncul pada 1 dokumen dalam korpus
- *Term* onic muncul pada 1776 dokumen dalam korpus
- *Term* juara muncul pada 277 dokumen dalam korpus

Mencari TF:

$$\mathrm{T}F(garansi, d) = \frac{Jumlah\ kemunculan\ garansi\ dalam\ d}{Total\ jumlah\ kata\ dalam\ d} = \frac{1}{3}$$

$$\mathrm{T}F(onic, d) = \frac{Jumlah\ kemunculan\ onic\ dalam\ d}{Total\ jumlah\ kata\ dalam\ d} = \frac{1}{3}$$

$$\mathrm{T}F(juara, d) = \frac{Jumlah\ kemunculan\ juara\ dalam\ d}{Total\ jumlah\ kata\ dalam\ d} = \frac{1}{3}$$

Mencari IDF:

$$IDF(garansi) = \log\left(\frac{15901}{1+1}\right) + 1$$
$$IDF(garansi) = \log\left(\frac{15901}{2}\right) + 1$$
$$IDF(garansi) = \log(7950) + 1$$
$$IDF(garansi) = 8.975 + 1$$
$$IDF(garansi) = 9.975$$

$$IDF(onic) = \log\left(\frac{15901}{1+1776}\right) + 1$$
$$IDF(onic) = \log\left(\frac{15901}{1777}\right) + 1$$
$$IDF(onic) = \log(8.95) + 1$$
$$IDF(onic) = 2.20 + 1$$
$$IDF(onic) = 3.20$$

$$IDF(juara) = \log\left(\frac{15901}{1+277}\right) + 1$$
$$IDF(juara) = \log\left(\frac{15901}{278}\right) + 1$$
$$IDF(juara) = \log(57.4) + 1$$
$$IDF(juara) = 4.04 + 1$$
$$IDF(juara) = 5.04$$

Penghitungan TF-IDF:

$TFIDF(garansi, d) = \frac{1}{3}.9.975 = 3.325$

$TFIDF(onic, d) = \frac{1}{3}.3.2 = 1.066$

$TFIDF(garansi, d) = \frac{1}{3}.5.04 = 1.68$

Normalisasi TF-IDF: membagi setiap nilai TF-IDF dengan nilai norma L2 yang akan dihitung menggunakan rumus sebagai berikut:

$$Norm = \sqrt{\left\{\sum\{\mathrm{t}\}\left(\{\mathrm{TF}-\mathrm{IDF}\}_{\{\mathrm{t}\}}\right)^2\right\}}$$

Mencari norma:

$Norm = \sqrt{(3.325)^2+(1.066)^2 + (1.680)^2}$

$Norm = \sqrt{11.055625 + 1.136356 + 2.8224} = 3.874$

Penghitungan TF-IDF yang dinormalisasikan dengan membagi setiap nilai TF-IDF dengan norma L2 yang sudah dihitung diatas.

$TFIDFgaransi(norm) = \frac{3.325}{3.874} = 0.858$

$TFIDFonic(norm) = \frac{1.066}{3.874} = 0.275$

$TFIDFjuara(norm) = \frac{1.680}{3.874} = 0.433$

3.5 Pelatihan Model

Ada berbagai cara melakukan klasifikasi teks salah satunya menggunakan Algoritma Naïve Bayes. Algoritma Naïve Bayes merupakan sebuah algoritma klasifikasi probabilistic yang menggunakan teorema Bayes, dengan menghitung probabilitas dari penjumlahan frekuensi dan kombinasi nilai dari sekumpulan data/dokumen yang ada. Algoritma ini mengasumsikan bahwa semua atribut yang diperoleh dari nilai dalam variabel kelas tidak memiliki hubungan atau keterikatan (independen) (B. G. Putra & Rochmawati, 2021). Naïve Bayes juga diasumsikan sebagai metode *classifier* yang diusulkan oleh seorang ilmuwan Inggris bernama Thomas Bayes. Pada penemuannya yang dikenal dengan Teorema Bayes, dia memprediksi peluang masa yang akan datang dengan menggunakan pengalaman masa lampau.

Sehingga, Naïve Bayes dianggap sebagai metode klasifikasi menggunakan probabilitas dan ilmu statistic (Rochmawati & Wibawa, 2018).

Pada proses pelatihan model ini, model dilatih menggunakan data latih yang disiapkan dan akan dievaluasi menggunakan data uji untuk mengukur performanya. algoritma yang digunakan adalah *Naïve Bayes Classifier* dengan menggunakan model *Gaussian Naïve Bayes, Multinomial Naïve Bayes, serta Bernoulli Naïve Bayes.*

Data yang digunakan dalam proses klasifikasi ini berasal dari hasil pemroresan data di tahap pre-processing dan ekstraksi fitur yang menggunakan *TF-IDF* dalam pembobotannya. Output yang dihasilkan dari proses ini adalah model *machine learning* yang dapat melakukan pengelompokkan sentimen terhadap data *live chat*. Kumpulan data *live chat* ini dimasukkan sebagai input dan akan diklasifikasikan berdasarkan tiga kelas sentiment, yaitu sentimen positif, sentimen negatif, dan sentimen netral.

Dalam pelatihan 3 model Naive Bayes terdapat perbedaan penghitungan yang dilakukan. *Gaussian Naive Bayes* melakukan penghitungan terhadap *theta, variance, dan class log prior*. Sedangkan *Multinomial* dan *Bernoulli* melakukan penghitungan terhadap *class log prior dan feature log probability* yang mana ada sedikit perbedaan dalam cara menghitung *feature log probability*-nya.

Pelatihan model *Gaussian Naive Bayes* melakukan penghitungan untuk mendapatkan tiga nilai utama yang berupa *theta (mean), variance*, serta *class prior*. *Theta (mean)* merupakan rata-rata dari nilai *TF-IDF* untuk tiap fitur pada tiap kelas label sentimen. Sedangkan *variance* merupakan nilai untuk mengukur seberapa jauh nilai-nilai fitur menyebar di sekitar rata-rata *(mean)* dalam kelas tertentu. *Class prior* merupakan probabilitas awal dari suatu kelas sebelum mempertimbangkan fitur-fitur dari data. Berikut merupakan contoh penghitungan yang dilakukan pada saat pelatihan model *Gaussian Naive Bayes*:

Rumus yang digunakan *Gaussian Naive Bayes* untuk menghitung *class prior* adalah sebagai berikut:

$$P(y_k) = \frac{n_k}{N}$$

Keterangan:

- $P(y_k)$ adalah probabilitas awal kelas y_k atau prior kelas.
- n_k adalah jumlah dokumen dalam kelas y_k.
- N adalah jumlah total dokumen.

Diketahui:

- Jumlah dokumen dalam label negatif adalah 4116
- Jumlah dokumen dalam label netral adalah 7049
- Jumlah dokumen dalam label positif adalah 4736
- Jumlah total dokumen adalah 15902

$$P_{(negatif)} = \frac{4116}{15901} = 0.258851645$$
$$P_{(netral)} = \frac{7049}{15901} = 0.443305452$$
$$P_{(positif)} = \frac{4736}{15901} = 0.297842903$$

Rumus yang digunakan *Gaussian Naive Bayes* dalam menghitung *theta* adalah sebagai berikut:

$$\theta_{\{k,i\}} = \frac{1}{n_k} \sum_{\{j=1\}}^{\{n_k\}} x_{\{i,j\}}$$

Keterangan:

- $\theta k, i$ adalah mean (rata-rata) dari fitur xix_ixi dalam kelas y_k.
- n_k adalah jumlah total dokumen dalam kelas y_k.
- $x_{i,j}$ adalah nilai fitur x_i dalam dokumen ke-j di kelas y_k.

Rumus tersebut dapat disamakan menjadi rumus berikut:

$$\theta_{\{k,i\}} = \frac{Total\ nilai\ TFIDF\ dari\ fitur\ i\ dalam\ kelas\ k}{Jumlah\ total\ dokumen\ dalam\ kelas\ k}$$

Contoh dokumen yang akan digunakan untuk penghitungan adalah dokumen yang berisikan pesan ”kelas onic”.

Diketahui:

- Nilai total TF-IDF “kelas” dalam label negatif adalah 3.008730522
- Nilai total TF-IDF “kelas” dalam label netral adalah 6.436369714
- Nilai total TF-IDF “kelas” dalam label positif adalah 23.56560446
- Nilai total TF-IDF “onic” dalam label negatif adalah 40.12061797
- Nilai total TF-IDF “onic” dalam label netral adalah 120.171938
- Nilai total TF-IDF “onic” dalam label positif adalah 311.5690802

Dengan data yang diketahui diatas, penghitungan *theta (mean)* dapat dilakukan sebagai berikut:

$$\theta_{\{negatif,kelas\}} = \frac{3.008730522}{4116} = 0.000730984$$

$$\theta_{\{netral,kelas\}} = \frac{6.436369714}{7049} = 0.00091309$$

$$\theta_{\{positif,kelas\}} = \frac{23.56560446}{4736} = 0.004975846$$

$$\theta_{\{negatif,onic\}} = \frac{40.12061797}{4116} = 0.009747478$$

$$\theta_{\{netral,onic\}} = \frac{120.171938}{7049} = 0.017048083$$

$$\theta_{\{positif,onic\}} = \frac{311.5690802}{4736} = 0.06578739$$

Rumus yang digunakan *Gaussian Naive Bayes* dalam menghitung *variance* adalah sebagai berikut:

$$\sigma_{k,i}^2 = \frac{1}{n_k - 1} \sum_{\{j=1\}}^{\{n_k\}} \left(x_{i,j} - \theta_{k,i}\right\}^2$$

Keterangan:

- $\sigma_{k,i}^2$ adalah *variance* dari fitur x_i dalam kelas y_k.
- n_k adalah jumlah total dokumen dalam kelas y_k.
- $x_{i,j}$ adalah nilai fitur xix_ixi dalam dokumen ke-j di kelas y_k
- $\theta_{k,i}$ adalah rata-rata (mean) dari fitur x_i dalam kelas y_k.

Sebagai contoh, diketahui bahwa Nilai *TF-IDF* dari *term* "onic" pada dokumen berlabel negatif adalah sebagai berikut:

- Dokumen 1 = 0.217017874
- Dokumen 2 = 0.104003029
- Dokumen 3 = 0.208380134
- *Theta (Mean)* untuk "onic" dalam kelas berlabel negatif adalah 0.009747478

Langkah pertama adalah menghitung selisih antara *Theta (Mean)* dari "onic" dalam kelas berlabel "negatif" dan nilai *TF-IDF* dari "onic"

$$Dokumen1 = 0.217017874 - 0.009747478 = 0.207270396$$

$$Dokumen2 = 0.104003029 - 0.009747478 = 0.094255551$$

$$Dokumen3 = 0.208380134 - 0.009747478 = 0.198632656$$

Langkah kedua adalah mengkuadratkan hasil selisih dari *Theta* dan nilai *TF-IDF* yang telah didapatkan

$Dokumen1 = (0.207270396)^2 = 0.042961017$

$Dokumen2 = (0.094255551)^2 = 0.008884101$

$Dokumen3 = (0.198632656)^2 = 0.03945493$

Langkah ketiga adalah menjumlahkan semua kuadrat selisih yang telah didapatkan

$0.042961017 + 0.008884101 + 0.03945493 = 0.091300048$

Langkah terakhir adalah membagi jumlah semua kuadrat dengan jumlah nilai $n_k - 1$. Karena ada 3 dokumen maka nilai $n_k - 1$ adalah 2

$\sigma^2_{onic,negatif} = \frac{0.091300048}{2} = 0.045650024$

Pelatihan model *Multinomial Naive Bayes* melakukan penghitungan untuk mendapatkan dua nilai utama yang berupa *class log prior* dan *feature log probability*. *Class log prior* merupakan logaritma dari probabilitas awal suatu kelas sentimen. Nilai *class log prior* menunjukkan seberapa sering kelas tersebut muncul dalam data pelatihan. *Class log prior* memberikan bobot awal kepada setiap kelas sentimen sebelum informasi mengenai fitur dipertimbangkan. Sedangkan *feature log probability* merupakan logaritma dari probabilitas kemunculan suatu fitur dalam kelas sentimen tertentu. *Feature log probability* digunakan untuk menghitung kemungkinan dokumen masuk ke dalam kelas sentimen tertentu. Berikut merupakan contoh penghitungan yang dilakukan pada saat pelatihan model *Multinomial Naive Bayes*:

Rumus yang digunakan *Multinomial Naive Bayes* untuk menghitung *class prior* adalah sebagai berikut:

$$log\ P(y_k) = \log(\frac{n_k}{N})$$

Keterangan:

- $P(y_k)$ adalah prior kelas y_k
- n_k adalah jumlah dokumen dalam kelas y_k.
- N adalah jumlah total dokumen

Diketahui bahwa:

- Jumlah dokumen dalam label negatif adalah 4116
- Jumlah dokumen dalam label netral adalah 7049
- Jumlah dokumen dalam label positif adalah 4736
- Jumlah total dokumen adalah 15901

Berikut merupakan penghitungan *class log prior* untuk masing-masing kelas negatif, positif, dan netral:

$$\log P_{(negatif)} = log\,\frac{4116}{15901} = \log 0.258851645 = -1.3515$$

$$\log P_{(netral)} = log\,\frac{7049}{15901} = \log 0.443305452 = -0.8135$$

$$\log P_{(positif)} = log\,\frac{4736}{15901} = \log 0.297842903 = -1.21119$$

Rumus yang digunakan *Multinomial Naïve Bayes* untuk menghitung *feature log probability* adalah sebagai berikut:

$$\log P(x_i|y_k) = \ln\left(\frac{Total\ nilai\ TFIDF\ dari\ x_i\ dalam\ kelas\ y_k + \alpha}{Total\ nilai\ TFIDF\ dari\ semua\ fitur\ dalam\ kelas\ y_k + \alpha V}\right)$$

Keterangan:

- x_i adalah fitur atau *term* atau kata yang sedang dihitung probabilitasnya
- y_k adalah kelas sentimen (label) dari data yang sedang dianalisis
- α adalah parameter *smoothing* yang digunakan untuk mencegah probabilitas 0. Nilai α yang digunakan adalah 1. *Smoothing* digunakan pada setiap fitur untuk memastikan bahwa fitur yang tidak muncul dalam kelas tertentu tidak membuat probabilitas menjadi nol
- V adalah jumlah total fitur unik yang terdapat dalam korpus.

Sebagai contoh, dokumen yang digunakan dalam penghitungan feature log probability adalah dokumen yang berisikan pesan “kelas onic”. Oleh karena itu perlu diketahui bahwa:

- Nilai total TF-IDF “kelas” dalam label negatif adalah 3.008730522
- Nilai total TF-IDF “kelas” dalam label netral adalah 6.436369714
- Nilai total TF-IDF “kelas” dalam label positif adalah 23.56560446
- Nilai total TF-IDF “onic” dalam label negatif adalah 40.12061797
- Nilai total TF-IDF “onic” dalam label netral adalah 120.171938
- Nilai total TF-IDF “onic” dalam label positif adalah 311.5690802
- Total kata unik dalam korpus adalah 9950

Berikut merupakan contoh penghitungan untuk *term* “kelas” dan term “onic” pada masing-masing kelas:

$$\log P(kelas|negatif) = \ln\left(\frac{3.008730522+1}{6961.922697+9950}\right) = ln\frac{4.0087305224}{16{,}911.922697} = \ln(0.000237) = -8.347299524$$

$$\log P(kelas|netral) = \ln\left(\frac{6.436369714+1}{12266.01247+9950}\right) = ln\frac{7.436369714}{22{,}216.01247} = \ln(0.000347) = -8.002185802$$

$$\log P(kelas|positif) = \ln\left(\frac{23.56560446+1}{8135.531197+9950}\right) = ln\frac{24.56560446}{18{,}085.531197} = \ln(0.001358) = -6.601520244$$

$$\log P(onic|negatif) = \ln\left(\frac{40.12061797+1}{6961.922697+9950}\right) = ln\frac{41.12061797}{16{,}911.922697} = \ln(0.002431) = -6.019264488$$

$$\log P(onic|netral) = \ln\left(\frac{120.171938+1}{12266.01247+9950}\right) = ln\frac{121.171938}{22{,}216.01247} = \ln(0.005454) = -5.211358078$$

$$\log P(onic|positif) = \ln\left(\frac{311.5690802+1}{8135.531197+9950}\right) = ln\frac{312.5690802}{18{,}085.531197} = \ln(0.017282) = -4.058042015$$

Pelatihan model *Bernoulli Naive Bayes* melakukan penghitungan untuk mendapatkan dua nilai utama yang berupa *class log prior* dan *feature log probability*. *Class log prior* merupakan logaritma dari probabilitas awal suatu kelas sentimen. Nilai *class log prior* menunjukkan seberapa sering kelas tersebut muncul dalam data pelatihan. *Class log prior* memberikan bobot awal kepada setiap kelas sentimen sebelum informasi mengenai fitur dipertimbangkan. Sedangkan *feature log probability* merupakan logaritma dari probabilitas kemunculan suatu fitur dalam kelas sentimen tertentu. *Feature log probability* digunakan untuk menghitung kemungkinan dokumen masuk ke dalam kelas sentimen tertentu. Tetapi sedikit ada perbedaan mengenai penghitungan *feature log probability* yang dilakukan oleh *Bernoulli* dengan *Multinomial*. Pada *Multinomial Naive Bayes*, yang digunakan adalah nilai *TF-IDF*. Sedangkan pada *Bernoulli Naive Bayes* yang digunakan untuk menghitung *feature log probability*-nya adalah ada dan tidaknya kemunculan suatu term pada dokumen tertentu. Berikut merupakan contoh penghitungan yang dilakukan pada saat pelatihan model Bernoulli Naive Bayes:

Rumus yang digunakan Bernoulli Naive Bayes untuk menghitung *class prior* adalah sebagai berikut:

$$log\ P(y_k) = \log(\frac{n_k}{N})$$

Keterangan:

- $P(y_k)$ adalah prior kelas y_k
- n_k adalah jumlah dokumen dalam kelas y_k.
- N adalah jumlah total dokumen

Diketahui bahwa:

- Jumlah dokumen dalam label negatif adalah 4116
- Jumlah dokumen dalam label netral adalah 7049
- Jumlah dokumen dalam label positif adalah 4736
- Jumlah total dokumen adalah 15901

Berikut merupakan penghitungan *class log prior* untuk masing-masing kelas negatif, positif, dan netral:

$$\log P_{(negatif)} = log\ \frac{4116}{15901} = \log 0.258851645 = -1.3515$$

$$\log P_{(netral)} = log\ \frac{7049}{15901} = \log 0.443305452 = -0.8135$$

$$\log P_{(positif)} = log\ \frac{4736}{15901} = \log 0.297842903 = -1.21119$$

Rumus yang digunakan *Bernoulli Naïve Bayes* untuk menghitung *feature log probability* adalah sebagai berikut:

$$\log P(x_i = 1|y_k) = \log\left(\frac{count(x_i = 1\ dalam\ kelas\ y_k) + \alpha}{N_k + \alpha.2}\right)$$

Keterangan:

- x_i adalah fitur yang sedang dihitung
- y_k adalah label kelas sentimen tertentu
- $count(x_i = 1\ dalam\ kelas\ y_k)$ adalah jumlah dokumen dimana fitur x_i muncul (bernilai 1) dalam kelas y_k.
- N_k adalah jumlah total dokumen dalam kelas y_k.
- α adalah parameter smoothing, bernilai 1, digunakan untuk menghindari probabilitas kosong.
- $\alpha.2$ digunakan karena Bernoulli menganggap bahwa setiap fitur dapat bernilai 0 atau 1, sehingga ada dua kemungkinan total untuk setiap fitur.

Diketahui bahwa:

- Jumlah dokumen dalam label negatif adalah 4116
- Jumlah dokumen dalam label netral adalah 7049
- Jumlah dokumen dalam label positif adalah 4736
- Jumlah total dokumen adalah 15901
- Jumlah kemunculan "kelas" dalam label negatif adalah 7
- Jumlah kemunculan "kelas" dalam label netral adalah 12
- Jumlah kemunculan "kelas" dalam label positif adalah 38
- Jumlah kemunculan "onic" dalam label negatif adalah 168
- Jumlah kemunculan "onic" dalam label netral adalah 466
- Jumlah kemunculan "onic" dalam label positif adalah 1142

Berikut merupakan contoh penghitungan fitur atau *term* "kelas" dan "onic"

$\log P(kelas = 1|negatif) = \log\left(\frac{7+1}{4116+2}\right) = \log\left(\frac{8}{4118}\right) = \log 0.001942 = -6.243681346$

$\log P(kelas = 1|netral) = \log\left(\frac{12+1}{7049+2}\right) = \log\left(\frac{13}{7051}\right) = \log 0.001843 = -6.295975372$

$\log P(kelas = 1|positif) = \log\left(\frac{38+1}{4736+2}\right) = \log\left(\frac{39}{4738}\right) = \log 0.008231 = -4.799808739$

$\log P(onic = 1|negatif) = \log\left(\frac{168+1}{4116+2}\right) = \log\left(\frac{169}{4118}\right) = \log 0.041039 = -3.193224173$

$\log P(onic = 1|netral) = \log\left(\frac{466+1}{7049+2}\right) = \log\left(\frac{467}{7051}\right) = \log 0.066231 = -2.714595472$

$\log P(onic = 1|positif) = \log\left(\frac{1142+1}{4736+2}\right) = \log\left(\frac{1143}{4738}\right) = \log 0.241241 = -1.421958721$

3.6 Evaluasi Model

Dalam proses evaluasi model, perhitungan dilakukan pada model yang telah dibuat dan dilatih pada tahapan sebelumnya untuk dapat dilakukan penilaian. Evaluasi model dilakukan untuk memastikan bahwa model yang digunakan atau dibangun dapat membuat prediksi yang

akurat. *Performance evaluation measures* merupakan metrik atau ukuran yang digunakan untuk mengukur kinerja dan akurasi dari model Machine Learning dalam melakukan prediksi atau klasifikasi pada data yang belum pernah dilihat sebelumnya. *Performance Evaluation Measure* biasanya digambarkan dengan *confusion matrix* (Jiao & Du, 2016). Confussion Matrix yang merupakan tabel yang digunakan untuk menggambarkan kinerja model klasifikasi pada data uji yang nilai sebenarnya diketahui. Confussion matrix membantu dalam memvisualisasikan kinerja model dengan memberikan gambaran tentang seberapa baik model memprediksi kelas target (Jiao & Du, 2016). Terdapat lima aspek yang dijadikan sebagai objek penilaian, yaitu *Accuracy, Precision, Recall, dan F1-score*. Aspek yang pertama ialah *accuracy* yang merupakan perhitungan yang mengukur ketepatan dalam melakukan prediksi atau klasifikasi terhadap data. *Precision* merupakan pengukuran yang dilakukan untuk mengukur tingkat kepastian dalam melakukan prediksi atau klasifikasi data. *Recall* mengukur seberapa baik model dapat memprediksi atau mengklasifikasikan data. *Recall* digunakan untuk memberikan gambaran tentang seberapa banyak data positif yang diprediksi atau diklasifikasikan dengan benar. Sedangkan *F1-Score* merupakan rataan *harmonic* dari *precision* dan *recall*. *F1-Score* berguna ketika terdapat ketidakseimbangan antara *precision* dan *recall*.

Untuk membandingkan *Gaussian, Multinomial, dan Bernoulli Naive Bayes* didasarkan pada perbedaan karakteristik dari masing-masing algoritma yang cocok untuk tipe data yang berbeda. Perbandingan ini membantu memahami bagaimana berbagai jenis Naive Bayes bekerja pada dataset tertentu dan memilih model terbaik berdasarkan kinerja dan karakteristik data.

Multinomial Naïve Bayes

Cocok untuk data diskrit, terutama dalam penghitungan frekuensi kata dalam teks. Ideal untuk tugas klasifikasi teks seperti analisis sentimen dan pengelompokan dokumen. Multinomial Naive Bayes mengasumsikan bahwa data adalah penghitungan atau frekuensi dari variabel diskrit. Lebih unggul dalam tugas klasifikasi teks dimana frekuensi kata penting, Pada Naïve Bayes, nilai peluang dokumen d pada kelas c bisa dihitung misalnya persamaan :

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

Keterangan :

c : kelas c

d : dokumen

P : peluang

Dengan memperhitungkan term nilai peluang dalam dokumen d pada kelas c bisa dihitung menggunakan persamaan :

$P(c \mid d) = P\ (c) \prod_{1 \leq k \leq nd} P(fk|c)$

Keterangan :

P(c|d) : peluang kategori c pada dokumen d

P(c) : nilai peluang kategori c

P(fk|c) : peluang term fk yang ada pada kelas c

Parameter P(fk|c) di hitung dengan persamaan :

$$\mathrm{P(fk|c)} = \frac{\mathrm{Tct} + 1}{\sum \mathrm{t}' \in \mathrm{VTc}t' + BF}$$

Tct merupakan peristiwa *term* t dalam sebuah dokumen menurut kelas c ∑t'ϵVTct' merupakan jumlah *term* yg masih ada dalam seluruh dokumen pada kelas c B merupakan jumlah variasi kata (*vocabulary*) yg masih ada pada data latih.

Perhitungan Multinomial Naïve Bayes dalam melakukan prediksi. Pada dokumen uji, terdapat sebuah dokumen yang berisikan “lag”, diketahui pada dokumen tersebut terdapat nilai-nilai sebagai berikut:

TF-IDF = 1

Log Probabilitas Fitur “lag” :

- Log Probabilitas Fitur “lag” untuk kelas negatif = -7.180019595
- Log Probabilitas Fitur “lag” untuk kelas netral = -9.015102163
- Log Probabilitas Fitur “lag” untuk kelas positif = -9.802867516

Class Prior :

- Class Prior untuk kelas negatif = 0.258851645
- Class Prior untuk kelas netral = 0.443305452
- Class Prior untuk kelas positif = 0.297842903

Sehingga, Class Log Priornya :

- Log Prior untuk kelas negatif = -1.351500182
- Log Prior untuk kelas netral = -0.813496238
- Log Prior untuk kelas positif = -1.211189103

Multinomial melakukan prediksi menggunakan menjumlahkan *Log Prior* dan *Log Likelihood* dengan menggunakan rumus sebagai berikut:

$$JLL(c) = log\ P(c)\ +\ log\ P(x\ \mid c)$$

Log Likelihood dihitung berdasarkan kontribusi setiap fitur terhadap kelas menggunakan rumus :

$$\log P(\ x\ \mid c\ ) = \sum_{i=1}^{n} x_i\ \cdot\ log\ P(w_i\ \mid c\ )$$

Pada kelas negatif, nilai log likelihood yang didapatkan adalah = -7.180019595

Pada kelas netral, nilai log likelihood yang didapatkan adalah = -9.015102163

Pada kelas positif, nilai log likelihood yang didapatkan adalah = -9.802867516

Jumlah dari *Log Prior* dan *Log Likelihood* dihitung sebagai *Joint Log Likelihood* yang dihitung dengan rumus dan memiliki nilai sebagai berikut :

$$JLL(c) = log\ P(c)\ +\ log\ P(x\ \mid c)$$

Kelas negatif = -1.351500182 + -7.180019595 = -8.531519776

Kelas netral = -0.813496238 + -9.015102163 = -9.828598402

Kelas positf = -1.211189103 + -9.802867516 = -11.01405662

Berdasarkan hasil perhitungan, nilai *Joint Log Likelihood* (JLL) tertinggi dimiliki oleh kelas negatif dengan nilai -8.531519776, sehingga prediksi yang dihasilkan oleh Multinomial Naïve Bayes untuk dokumen ini adalah negatif.

Bernoulli Naïve Bayes

Pada model Bernoulli, setiap data uji dihitung nilai probabilitas setiap kata untuk setiap kelas. Perbedaannya dengan contoh pertama adalah perhitungan probabilitas menggunakan jumlah data yang mengandung kata t (Tct), bukan frekuensi kemunculan kata t. Selain itu, jika contoh Multinomial menggunakan jumlah kata dan jumlah kosakata, model Bernoulli menggunakan jumlah data pelatihan per kelas (Tc) dan jumlah kelas (Σc). Dapat digunakan dalam klasifikasi dokumen pendek atau review produk dimana keberadaan kata tertentu lebih informatif daripada frekuensi kemunculannya. Probabilitas setiap kata dihitung dengan menggunakan persamaan :

$$\mathrm{P(fk|c)} = \frac{\mathrm{Tct}\ +\ 1}{\mathrm{Tc}\ +\ \sum s}$$

Setelah diperoleh probabilitas setiap kata, maka dihitung nilai perkalian invers untuk probabilitas setiap kata pada data latih berbeda dengan kata pada data uji Nilai probabilitas kalimat untuk suatu kelas dihitung sebesar dengan persamaan :

$$P(c|d) = P(c) \prod_{i=1}^{N} P(fk|c)\ x\ \prod_{i+1}^{M} (1 - P(fk'|c))$$

Perhitungan Bernoulli Naïve Bayes dalam melakukan prediksi. Pada dokumen uji, terdapat sebuah dokumen yang berisikan “lag”, diketahui pada dokumen tersebut terdapat nilai-nilai sebagai berikut:

TF-IDF = 1

Log Probabilitas Fitur “lag” :

- Log Probabilitas Fitur “lag” untuk kelas negatif = -5.327390614
- Log Probabilitas Fitur “lag” untuk kelas netral = -7.251486817
- Log Probabilitas Fitur “lag” untuk kelas positif = -8.463370385

Class Prior :

- Class Prior untuk kelas negatif = 0.258851645
- Class Prior untuk kelas netral = 0.443305452
- Class Prior untuk kelas positif = 0.297842903

Sehingga, Class Log Priornya :

- Log Prior untuk kelas negatif = -1.351500182
- Log Prior untuk kelas netral = -0.813496238
- Log Prior untuk kelas positif = -1.211189103

Bernoulli melakukan prediksi menggunakan menjumlahkan *Log Prior* dan *Log Likelihood* dengan menggunakan rumus sebagai berikut:

$$JLL(c) = log\ P(c)\ +\ log\ P(x\ |\ c)$$

Log Likelihood dihitung berdasarkan kontribusi setiap fitur terhadap kelas menggunakan rumus :

$$\log P(\ x\ |\ c\ ) = \sum_{i=1}^{n} x_i\ \cdot\ log\ P(w_i\ |\ c\ )$$

Pada kelas negatif, nilai *log likelihood* yang didapatkan adalah = -5.327390614

Pada kelas netral, nilai *log likelihood* yang didapatkan adalah = -7.251486817

Pada kelas positif, nilai *log likelihood* yang didapatkan adalah = -8.463370385

Jumlah dari *Log Prior* dan *Log Likelihood* dihitung sebagai *Joint Log Likelihood* yang dihitung dengan rumus dan memiliki nilai sebagai berikut :

$$JLL(c) = log\ P(c)\ +\ log\ P(x\ |\ c)$$

Kelas negatif = -1.351500182 + -5.327390614 = -6.678890796

Kelas netral = -0.813496238 + -7.251486817 = -8.064983055

Kelas positf = -1.211189103 + -8.463370385 = -9.674559487

Berdasarkan hasil perhitungan, nilai *Joint Log Likelihood* (JLL) tertinggi dimiliki oleh kelas negatif dengan nilai -6.678890796, sehingga prediksi yang dihasilkan oleh Bernoulli Naïve Bayes untuk dokumen ini adalah negatif.

Gausian Naïve Bayes

Dirancang untuk menangani data kontinu yang diasumsikan berdistribusi normal. Sangat cocok untuk fitur-fitur yang berupa nilai kontinu (misalnya, panjang, berat, suhu). Gaussian Naïve Bayes sangat cocok untuk membuat prediksi berdasarkan karakteristik distribusi normal. Gaussian Naive Bayes mungkin lebih baik pada dataset dengan fitur kontinu. Gaussian memperkirakan probabilitas suatu fitur terhadap suatu kelas menggunakan fungsi distribusi norrmal sebagai berikut :

$$\log P\ (\ x\ |\ c\ ) = \ -\frac{1}{2}\log(2\pi\sigma^2) - \frac{(x-\mu)^2}{2\sigma^2}$$

Keterangan :

xi : *log likelihood*

σ : variance

μ : mean

Berikut merupakan contoh perhitungan Gaussian Naïve Bayes dalam melakukan prediksi:

Pada dokumen uji, terdapat sebuah dokumen yang berisikan “lag”, diketahui pada dokumen tersebut terdapat nilai-nilai sebagai berikut:

TF-IDF (x) = 1

Mean :

- *Mean* (μ) untuk kelas negatif = 0.002886544

- *Mean* (μ) untuk kelas netral = 0.000241251
- *Mean* (μ) untuk kelas positif = 0

Variance :

- *Variance* (σ^2) untuk kelas negatif = 0.001847833
- *Variance* (σ^2) untuk kelas netral = 0.000105902
- *Variance* (σ^2) untuk kelas positif = 0

Class Prior :

- *Class Prior* untuk kelas negatif = 0.258851645
- *Class Prior* untuk kelas netral = 0.443305452
- *Class Prior* untuk kelas positif = 0.297842903

Sehingga, *Class Log Prior* nya adalah :

- *Log Prior* untuk kelas negatif = -1.351500182
- *Log Prior* untuk kelas netral = -0.813496238
- *Log Prior* untuk kelas positif = -1.211189103

Gaussian melakukan prediksi menggunakan penjumlahan *Log Likelihood* dengan *Log Prior* dengan rumus sebagai berikut :

$$JLL(c) = log\ P(c)\ +\ log\ P(x\ |\ c)$$

Log Likelihood dihitung menggunakan rumus :

$$\log P\ (\ x\ |\ c\ ) =\ -\frac{1}{2}\log(2\pi\sigma^2) - \frac{(x-\mu)^2}{2\sigma^2}$$

Pada kelas negatif, nilai *log likelihood* yang didapatkan adalah = -266.7994214

Pada kelas netral, nilai *log likelihood* yang didapatkan adalah = -4715.410806

Pada kelas positif, nilai *log likelihood* yang didapatkan adalah = -499999990.6

Jumlah dari *Log Prior* dan *Log Likelihood* dihitung sebagai *Joint Log Likelihood* yang memiliki nilai sebagai berikut :

Kelas negatif = -1.351500182 + -266.7994214 = -268.1509216

Kelas netral = -0.813496238 + -4715.410806 = -4716.224303

Kelas positif = -1.211189103 + 499999990.6 = -499999991.8

Berdasasrkan nilai di atas, nilai *joint log likelihood* yang terbesar dimiliki oleh kelas negatif, sehingga prediksi yang dihasilkan oleh Gaussian Naïve Bayes untuk dokumen ini adalah negatif.

Performance Evaluation Measure menggunakan *Confusion Matrix*. Dengan menggunakan confusion matrix, nilai *accuracy*, *precision*, *recall*, serta *f1-score* dihasilkan menggunakan rumus-rumus berikut ini :

Rumus untuk menghitung nilai *accuracy* adalah sebagai berikut:

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Total\ Prediksi} = \frac{TP + TN}{TP + TN + FP + FN}$$

Rumus untuk menghitung nilai *precision* adalah sebagai berikut:

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

Rumus untuk menghitung nilai *recall* adalah sebagai berikut:

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

Rumus untuk menghitung nilai *f1-score* adalah sebagai berikut:

$$f1score = \frac{2(Presisi.Recall)}{Presisi + Recall}$$

Cross validation merupakan teknik validasi model yang umum digunakan dalam machine learning untuk mengevaluasi kinerja model secara lebih objektif dan akurat (Mardiana, Kusnandar, & Satyahadewi, 2022). Teknik ini melibatkan pembagian data menjadi beberapa subset, kemudian model dilatih dan diuji pada subset yang berbeda secara berulang. Hal ini memungkinkan untuk mendapatkan perkiraan yang lebih baik tentang kinerja model pada data yang belum dilihat sebelumnya (Hafid, 2023).

Pada *k-Fold cross validation*, dataset dibagi menjadi k bagian (*fold*) yang seukuran. Secara umum, langkah-langkahnya adalah sebagai berikut (Fushiki, 2011):

1. Pembagian Dataset

   Dataset dibagi menjadi k bagian dengan ukuran yang sama.
2. Iterasi Training dan Evaluasi

   Model dilatih dan dievaluasi sebanyak k kali. Pada setiap iterasi, salah satu *fold* dijadikan data uji, sementara *k-1 fold* lainnya digunakan sebagai data latih. Model dilatih pada data latih dan dievaluasi pada data uji.
3. Pengukuran Performa

Performa model diukur menggunakan metrik evaluasi seperti akurasi, presisi, *recall*, atau *F1-score*.

4. Perhitungan Rata-rata Performa

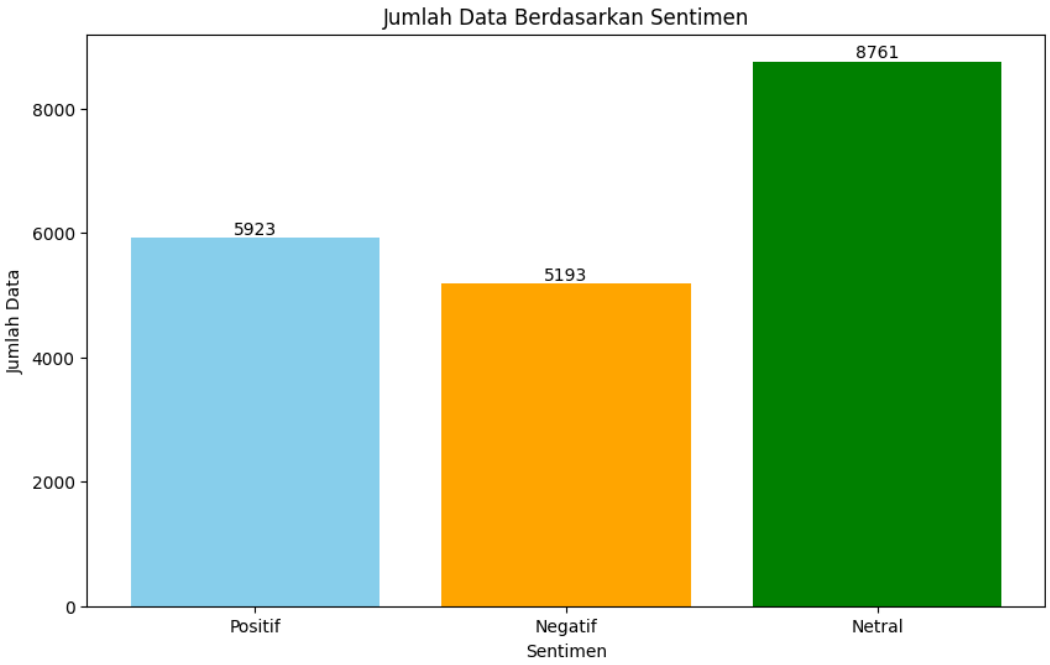
   Nilai performa yang dihasilkan dari setiap iterasi dapat dijumlahkan dan dirata-ratakan untuk mendapatkan estimasi performa keseluruhan model.

3.7 Visualisasi Data

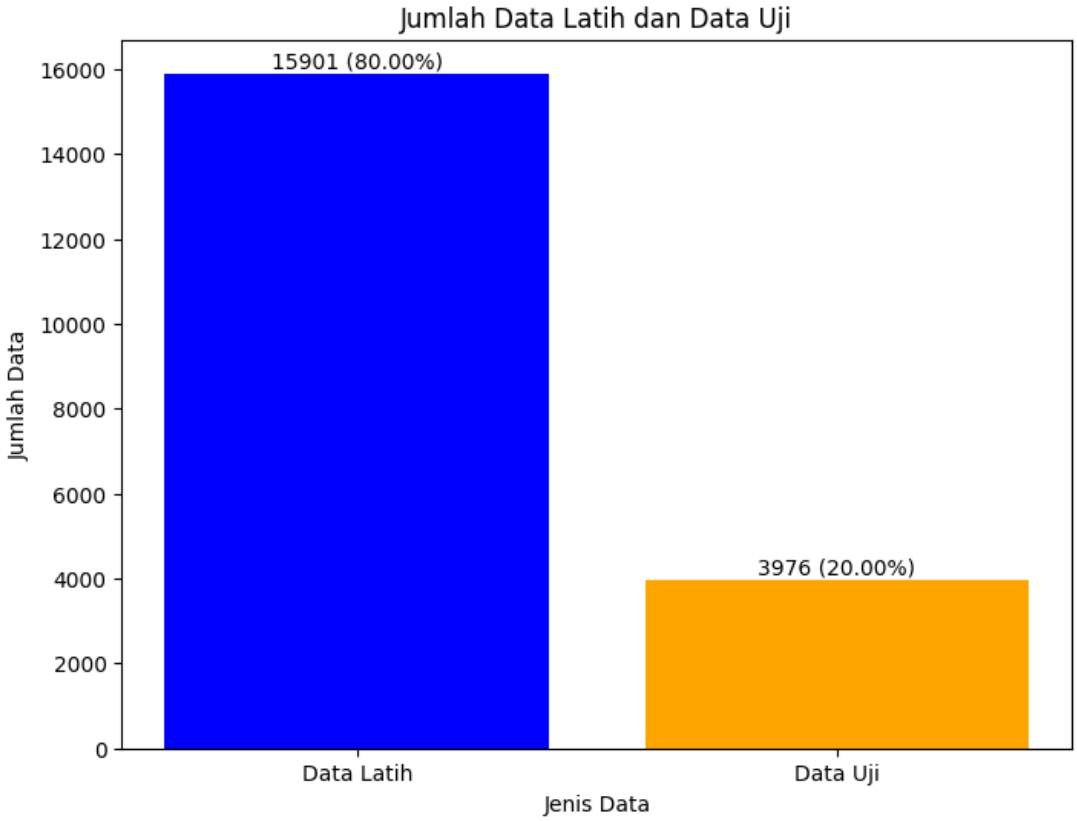
Menurut penemu aplikasi Tableau, menjelaskan visualisasi data atau data visualization adalah tampilan berupa grafis atau visual dari informasi dan data. Dengan kata lain, data visualization mengubah kumpulan data menjadi hal lebih sederhana untuk ditampilkan. Teknik visualisasi data yang efektif akan memberikan dampak signifikan pada keberhasilan komunikasi dengan target audiens. Tidak hanya mempermudah pengenalan pola pada data, visualisasi data juga dapat memperkuat pesan yang ingin disampaikan (Johannes Kurniawan, 2023).

Dua teknik yang umum digunakan dalam visualisasi data ialah dengan menggunakan *Bar Chart* dan *WordCloud*. *Bar Chart* yang merupakan diagram batang digunakan untuk membandingkan kuantitas atau frekuensi dari kategori atau variabel yang berbeda. Setiap kategori direpresentasikan oleh batang dengan tinggi yang proporsional terhadap nilai atau frekuensi yang terkait. Sedangkan *WordCloud* merupakan representasi visual dari frekuensi kemunculan kata-kata dalam data. Kata-kata yang lebih sering muncul akan ditampilkan dengan ukuran yang lebih besar, menciptakan gambaran yang cepat dan menarik tentang topik yang dominan dalam data. Dengan kombinasi diagram batang dan *WordCloud*, data yang ingin divisualisasikan diharapkan akan dapat dipahami dengan mudah.

Contoh visualisasi jumlah data menggunakan diagram batang dan visualisasi menggunakan WordCloud disajikan pada Gambar 3.5. Gambar 3.6, dan Gambar 3.7.



Gambar 3. 6 Jumlah dataset berdasarkan kelas sentimen



Gambar 3. 5 Jumlah data latih dan data uji



Gambar 3. 7 *WordCloud* kata yang paling sering muncul

BAB IV
HASIL DAN PEMBAHASAN

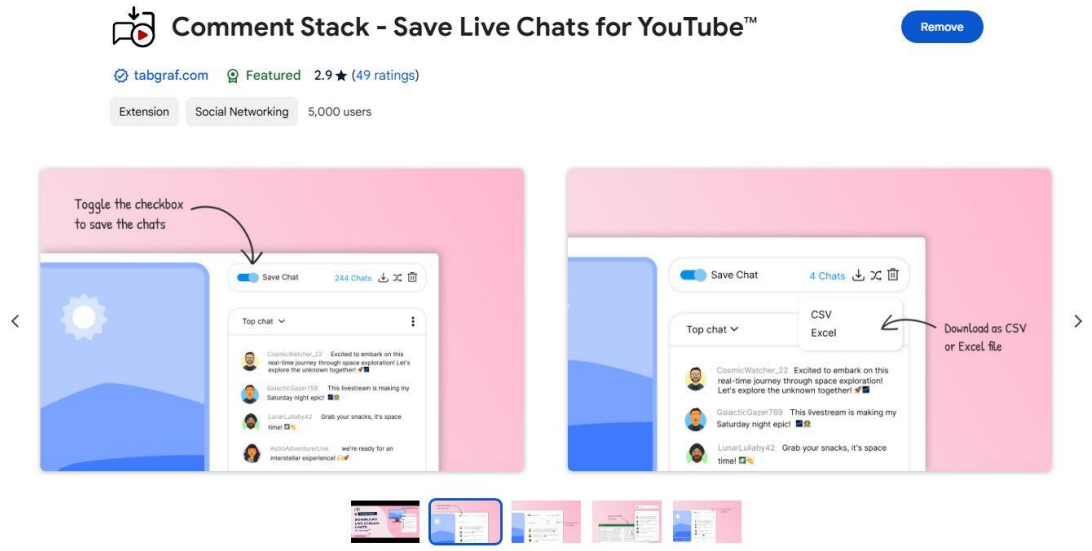
4.1 Eksplorasi data

Data yang digunakan dalam penelitian ini merupakan data yang diambil secara langsung dari rekaman siaran langsung pertandingan Grand Final MSC 2023 yang berjudul “Bahasa Indonesia | Grand Final MSC | MSC 2023” pada kanal Youtube Mobile Legends: Bang Bang, yang mempertemukan antara tim Onic Indonesia dengan tim Blacklist International pada tanggal 18 Juni 2023.

4.2 Pengumpulan Data

Data pesan yang dikumpulkan untuk dapat dipergunakan dalam penelitian ini diambil dengan cara melakukan perekaman *live chat* pada kolom *live chat* YouTube dengan memanfaatkan bantuan dari *extension* CommentStack. Output dari proses ini ialah sebuah file .csv yang kemudian dapat dilakukan pembersihan dan pelabelan data.

Ekstensi yang digunakan untuk mengumpulkan data dari *live chat* YouTube ditunjukkan pada Gambar 4.1.



Gambar 4. 1 Ekstensi *Comment Stack*

Tampilan siaran Grand Final MSC 2023 pada YouTube, dari mana data live chat diambil, dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Siaran Grand Final MSC 2023

Langkah pertama yang dilakukan yaitu mengunduh dan menambahkan *extension* CommentStack di browser. Setelah CommentStack terpasang pada browser, buka rekaman siaran langsung pada platform YouTube. Dalam penelitian ini, data diambil dari rekaman siaran langsung pertandingan Grand Final MSC 2023 pada kanal Youtube Mobile Legends: Bang Bang. Langkah selanjutnya adalah merekam *live chat* menggunakan CommentStack, kemudian menyimpan *live chat* tersebut dan mengekspornya ke dalam bentuk file .csv. Hasil ekspor data *live chat* disajikan pada Gambar 4.3.

	A	B	C	D	E	F	G	H	I	J	K
1	Author	Date	Author Name	Role	Message	Video Time	ID				
2	https://yt4.ggpht	2023-08-25	SYAIFUS SHOLIHIN OFFICIAL		fans evos 100% dukung blacklist	01:36:50	ChwKGkNNaXZ5X0hXeIA4Q0ZWREN3Z1FkM2pzSVpR				
3	https://yt4.ggpht	2023-08-25	Hizkia effendy		everyone know kairi and coach yeb from philipines	01:36:50	ChwKGkNNbi0zdkhXeIA4Q0ZkSEN3Z1FkUTZVTHB3				
4	https://yt4.ggpht	2023-08-25	Fendi Khan_chiL		text-green-game-overtext-green-game-overtext-green-game-overtext-green-game-overtext-green-game	01:36:51	ChwKGkNOelIqZkxXeIA4Q0ZhN0R3Z1Fkb09vRlB3				
5	https://yt4.ggpht	2023-08-25	NIAA`03		kebanyakan bocil rasisss ??????	01:36:51	ChwKGkNNQ0FsdkxXeIA4Q0ZWSEJ3Z1FkRlZRSjRn				
6	https://yt4.ggpht	2023-08-25	Haerul suruga		jasjus enak	01:36:51	ChwKGkNJeXdtZkxXeIA4Q0ZTY19yUVlkc0NzSTJR				
7	https://yt4.ggpht	2023-08-25	Alfin Mahendra putra		intro nya lama bngt	01:36:51	ChwKGkNOYmpvZkxXeIA4Q0ZhVEF3Z1FkN3pzSU1B				
8	https://yt4.ggpht	2023-08-25	Radenmas Yosi		?????????????????????	01:36:51	ChwKGkNOYmNvX0xXeIA4Q0ZiTEN3Z1FkczRjQUxn				
9	https://yt4.ggpht	2023-08-25	marco		YANG SARA LANGSUNG BLOKIR AJA YA ADIK",,MEREKA CUMA SAMPAH MASYARAKAT KURANG PERHATIAN C	01:36:51	ChwKGkNNSGZvX0xXeIA4Q0ZVOF9yUVlkM2UwUHF3				
10	https://yt4.ggpht	2023-08-25	Akang Muhamadrizal		kapan mulai nya ini	01:36:51	ChwKGkNLdWZxX0xXeIA4Q0ZiUEN3Z1FkTEUwTjdB				
11	https://yt4.ggpht	2023-08-25	Rakha Agustira		pada rasis banget orang indo	01:36:51	ChwKGkNPbVJydkxXeIA4Q0ZVVVpmUW9kakk0RmNn				
12	https://yt4.ggpht	2023-08-25	Gabriel Yuda		jam berapa sih mainnya?	01:36:51	ChwKGkNLX2NzUExXeIA4Q0ZiUEN3Z1FkTEUwTjdB				
13	https://yt4.ggpht	2023-08-25	Indra99		kingdom??????	01:36:52	ChwKGkNOT2x4dkxXeIA4Q0ZWREN3Z1FkM2pzSVpR				
14	https://yt4.ggpht	2023-08-25	Tegar Tegar		face-turquoise-covering-eyesface-turquoise-covering-eyesface-turquoise-covering-eyesface-turquoise-c	01:36:52	ChwKGkNKRzl6UExXeIA4Q0ZiTEN3Z1FkczRjQUxn				
15	https://yt4.ggpht	2023-08-25	Lukman Bae		kpn main	01:36:52	ChwKGkNKN0wydkxXeIA4Q0ZZekN3Z1FkdkJ3TVNR				
16	https://yt4.ggpht	2023-08-25	Yusof Zaky		ban estes digi	01:36:53	ChwKGkNNajI2UExXeIA4Q0ZVOF9yUVlkM2UwUHF3				
17	https://yt4.ggpht	2023-08-25	Randi Muhammad		kata tepos mau bantai tim di masc hmmm,kok dia gaada sih wkwk	01:36:53	ChwKGkNQdW1pX1BXeIA4Q0ZVVVpmUW9kakk0RmN				
18	https://yt4.ggpht	2023-08-25	why		anjayyy	01:36:53	ChwKGkNOdmppX1BXeIA4Q0ZWSEJ3Z1FkRlZRSjRn				
19	https://yt4.ggpht	2023-08-25	Freddy Cia		lama x jink	01:36:53	ChwKGkNOM2ZsdlBXeIA4Q0ZVVVpmUW9kakk0RmNr				
20	https://yt4.ggpht	2023-08-25	AAr_FDSk		YANG MAIN ONIK YANG D HUJAT EVOS HADEH BOCIL	01:36:54	ChwKGkNNNmxzZlBXeIA4Q0ZkSEN3Z1FkUTZVTHB3				
21	https://yt4.ggpht	2023-08-25	Breathe Coil		Pen?uri	01:36:54	ChwKGkNJTGF1UFBXeIA4Q0ZkSEN3Z1FkUTZVTHB3				
22	https://yt4.ggpht	2023-08-25	Rusiana Ludivar		I wish onic wins so that the indonisian people becomes happily.	01:36:54	ChwKGkNKei11UFBXeIA4Q0ZiUEN3Z1FkTEUwTjdB				
23	https://yt4.ggpht	2023-08-25	gembel ratan		face-fuchsia-wide-eyesface-fuchsia-wide-eyesface-fuchsia-wide-eyesface-fuchsia-wide-eyesface-fuchsia	01:36:54	ChwKGkNJbmd2X1BXeIA4Q0ZiM0R3Z1FkRlBFTnRR				
24	https://yt4.ggpht	2023-08-25	DeOpandi		#gobct	01:36:54	ChwKGkNKTHF5UFBXeIA4Q0ZhVEF3Z1FkN3pzSU1B				
25	https://yt4.ggpht	2023-08-25	Kirin 2019		go evos go	01:36:55	ChwKGkNQYU40X1BXeIA4Q0ZVVVpmUW9kakk0RmNr				
26	https://yt4.ggpht	2023-08-25	Novelia Maharani		ONICCC	01:36:55	ChwKGkNJVEk4X1BXeIA4Q0ZTY19yUVlkc0NzSTJR				
27	https://yt4.ggpht	2023-08-25	ahmad maulana		MSC CUMA HAHA HIHI ->TEVOS	01:36:55	ChwKGkNPV1AtX1BXeIA4Q0ZWREN3Z1FkM2pzSVpR				
28	https://yt4.ggpht	2023-08-25	Radenmas Yosi		???????????????????	01:36:55	ChwKGkNKQy1tUFRXeIA4Q0ZhN0R3Z1Fkb09vRlB3				
29	https://yt4.ggpht	2023-08-25	Kuya Edan Gaming		turu nanti kau ph	01:36:55	ChwKGkNJVDBvUFRXeIA4Q0ZjN0J3Z1FkUUdZS0R3				
30	https://yt4.ggpht	2023-08-25	Suci Nhy		helow yang typingnya toxic, gw sumpah in hp Lo rusak	01:36:56	ChwKGkNPS3NxX1RXeIA4Q0ZhN0R3Z1Fkb09vRlB3				
31	https://yt4.ggpht	2023-08-25	GunBARBAR		face-red-droopy-eyesface-red-droopy-eyesface-red-droopy-eyesface-red-droopy-eyesface-red-droopy-e	01:36:56	ChwKGkNOYS1yZlRXeIA4Q0ZiM0R3Z1FkRlBFTnRR				
32	https://yt4.ggpht	2023-08-25	Mute Ono		AE REKT IS REAL KAH MANIS	01:36:56	ChwKGkNLRzVyX1RXeIA4Q0ZhTER3Z1FkNHhRUERR				
33	https://yt4.ggpht	2023-08-25	Labib Asyam		sok asik lu semua	01:36:56	ChwKGkNMYV92dlRXeIA4Q0ZZekN3Z1FkdkJ3TVNR				
34	https://yt4.ggpht	2023-08-25	Cusmin Speed		GW YAKIN NI YG BACOT KALAU DI KUMPULIN GK ADA NYALI	01:36:56	ChwKGkNOV255dlRXeIA4Q0ZmTEJ3Z1FkaXJvRjhR				
35	https://yt4.ggpht	2023-08-25	Dery		elbowcoughelbowcoughelbowcoughelbowcoughelbowcoughelbowcoughelbowcoughelbowcoughelbowco	01:36:57	ChwKGkNPYno3dlRXeIA4Q0ZiTEN3Z1FkczRjQUxn				
36	https://yt4.ggpht	2023-08-25	Gilang Saputra		????	01:36:57	ChwKGkNQXy03UFRXeIA4Q0ZiTEN3Z1FkczRjQUxn				
37	https://yt4.ggpht	2023-08-25	Irvan Arif		lepas estes	01:36:57	ChwKGkNOVFI5ZlRXeIA4Q0ZiUEN3Z1FkTEUwTjdB				
38	https://yt4.ggpht	2023-08-25	tasya indah		lama	01:36:57	ChwKGkNQek1fZlRXeIA4Q0ZjWER3Z1FkdkhRSW9n				
39	https://yt4.ggpht	2023-08-25	Ryan Oxy		onic kiboy siap siap di sepong bunda venus Ya gaes ya , ALL IN BLACKLIST	01:36:57	ChwKGkNQeTdpUFhXeIA4Q0ZZekN3Z1FkdkJ3TVNR				

Gambar 4. 3 Hasil ekspor data *live chat*

Data yang terkumpul adalah sebanyak 45.335 pesan pada *live chat* yang kemudian akan diolah dengan teknik *pre-processing* data.

4.3 Implementasi Pra-pemrosesan Data

Data rekaman *live chat* yang telah didapatkan belum dapat digunakan karena masih merupakan data mentah yang belum diolah. Agar dapat digunakan, data rekaman *live chat* tersebut harus melalui pra-pemrosesan terlebih dahulu. Pra-pemrosesan pada penelitian ini dilakukan menggunakan platform Google Colab untuk menjalankan kode Python. Langkah yang pertama dilakukan adalah dengan menginstall *package* serta mengimpor *library* yang akan digunakan.

```
    !pip install Sastrawi
    !pip install nltk
    !pip install neattext
```

Gambar 4. 4 Kode program untuk menginstal *package*

Gambar 4.4 merupakan kode perintah yang digunakan dalam Google Colab untuk menginstal paket-paket Python menggunakan ‘pip’, yang merupakan *package manager* untuk Python. Perintah-perintah diatas dijalankan untuk menginstal paket yang diperlukan untuk analisis teks dan pemrosesan bahasa alami atau biasa dikenal dengan NLP (*Natural Language Processing*). Paket pertama yang diinstal adalah Sastrawi yang merupakan sebuah pustaka Python untuk stemming dan stopword removal dalam bahasa Indonesia. Paket kedua adalah NLTK (*Natural Language Toolkit*) yang juga merupakan pustaka Python yang menyediakan alat-alat untuk bekerja dengan teks dalam pengolahan bahasa alami.

```
    import pandas as pd
    import string
    import nltk
    nltk.download('punkt')
    from nltk.tokenize import word_tokenize
    from        Sastrawi.StopWordRemover.StopWordRemoverFactory        import
StopWordRemoverFactory
    from google.colab import drive
    from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

Gambar 4. 5 Kode program untuk mengimpor *library*

Gambar 4.5 merupakan kode yang berfungsi sebagai rangkaian perintah untuk mengimpor library. Kode pada baris yang pertama berfungsi untuk mengimpor library ‘pandas’, yang digunakan untuk manipulasi data dan analisis dalam bentuk DataFrame.

‘pandas’ sangat berguna untuk menangani data dalam format tabel seperti CSV. Kode pada baris kedua berfungsi untuk mengimpor module string, yang menyediakan konstanta dan fungsi untuk operasi string, seperti daftar karakter alfabet dan simbol . Kode pada baris yang ketiga berfungsi untuk mengimpor library nltk (*Natural Language Toolkit*), yang digunakan untuk pemrosesan bahasa alami (NLP). nltk menyediakan alat untuk tokenisasi, *stemming*. Kode pada baris yang keempat berfungsi untuk mengunduh paket 'punkt' dari nltk, yang berisi *tokenizer*. *Tokenizer* ini digunakan untuk memecah teks menjadi kalimat atau kata. Kode pada baris kelima berfungsi untuk mengimpor fungsi word_tokenize dari nltk.tokenize. Fungsi ini digunakan untuk memecah teks menjadi token (kata). Kode pada baris keenam berfungsi untuk mengimpor StopWordRemoverFactory dari library Sastrawi. Sastrawi adalah library untuk pemrosesan bahasa Indonesia yang menyediakan alat untuk menghilangkan kata-kata umum (stopwords) dan melakukan stemming. StopWordRemoverFactory digunakan untuk membuat objek yang dapat menghapus stopwords dari teks bahasa Indonesia. Kode pada baris ketujuh berfungsi untuk mengimpor module drive dari google.colab. Ini memungkinkan akses ke Google Drive dari lingkungan Google Colab. Anda dapat menggunakan ini untuk membaca dan menyimpan file di Google Drive, memudahkan integrasi dengan file yang disimpan di cloud. Kode pada baris terakhir berfungsi untuk mengimpor StemmerFactory dari Sastrawi. StemmerFactory digunakan untuk melakukan stemming, yaitu proses mengubah kata menjadi bentuk dasarnya dalam bahasa Indonesia. Stemming membantu mengurangi variasi kata sehingga analisis teks menjadi lebih konsisten.

Langkah selanjutnya adalah memasukkan dataset ke dalam Google Colab dengan menggunakan kode yang disajikan pada Gambar 4.6.

```
drive.mount('/content/gdrive')
df=pd.read_csv('gdrive/My Drive/DATASET/DATASET.csv')
```

Gambar 4. 6 Kode program untuk mengakses dataset

Kode pada baris pertama berfungsi untuk mengakses Google Drive dari Google Colab. Kode pada baris kedua berfungsi untuk membaca file CSV dari Google Drive dan memuatnya ke dalam DataFrame ‘pandas’. Tampilan dataset saat diakses pada Google Colab dapat dilihat pada Gambar 4.7.



	Author	Date	Author Name	Role	Message	Video Time	ID
0	https://yt4.ggpht.com/AIQaVBOHKRvkEPwah5VzNZkS...	2023-08-25T18:36:00.000Z	SYAIFUS SHOLIHIN OFFICIAL	NaN	fans evos 100% dukung blacklist	01:36:50	ChwKGkNNaXZ5X0hXelA4Q0ZWREN3Z1FkM2pzSVpR
1	https://yt4.ggpht.com/ytc/AOPolaSoeC3InbKXHrVj...	2023-08-25T18:36:00.000Z	Hizkia effendy	NaN	everyone know kairi and coach yeb from philipines	01:36:50	ChwKGkNNbi0zdkhXelA4Q0ZkSEN3Z1FkUTZVTHB3
2	https://yt4.ggpht.com/ytc/AOPolaTY8QxglXNwqc31...	2023-08-25T18:36:00.000Z	Fendi Khan_chiL	NaN	text-green-game-overtext-green-game-overtext-g...	01:36:51	ChwKGkNOellqZkxXelA4Q0ZhN0R3Z1Fkb09vRlB3
3	https://yt4.ggpht.com/SKKXFRAu0DTzJhumxImvJqsg...	2023-08-25T18:36:00.000Z	NIAA`03	NaN	kebanyakan bocil rasisss ??????	01:36:51	ChwKGkNNQ0FsdkxXelA4Q0ZWSEJ3Z1FkRlZRSjRn
4	https://yt4.ggpht.com/ytc/AOPolaRnvXQzUubrd6dO...	2023-08-25T18:36:00.000Z	Haerul suruga	NaN	jasjus enak	01:36:51	ChwKGkNJeXdtZkxXelA4Q0ZTY19yUVlkc0NzSTJR
...	...	...	...	...	...	...	...
45330	https://yt4.ggpht.com/ytc/AOPolaRX6YwlpftFt3B9...	2023-08-25T23:06:00.000Z	Calvin Chandra	NaN	ZEYS MENYESAL, LBH MILIH DREAMS DRPD KIBOY	06:06:02	ChwKGkNOLXYzWWVUemY4Q0ZVVpmUW9kakk0RmNn
45331	https://yt4.ggpht.com/oU-nBMH0b_lfcpjWff6rmzi6...	2023-08-25T23:06:00.000Z	Luthfi	NaN	MVP ROAMER NI BOS. BUKAN KALENG2 KAYAK IESF. P...	06:06:02	ChwKGkNOeTk0SWVUemY4Q0ZhN0R3Z1Fkb09vRlB3
45332	https://yt4.ggpht.com/13BH9rb2A9cyy-S0GGHKEERB...	2023-08-25T23:06:00.000Z	Mj Camanga	NaN	Congrats Onic on behalf of Filipino Supporter ...	06:06:03	ChwKGkNNNmE3b2VUemY4Q0ZkSEN3Z1FkUTZVTHB3
45333	https://yt4.ggpht.com/ytc/AOPolaTlDJhHdMEela-v...	2023-08-25T23:06:00.000Z	Bachoi gaming	NaN	FRENGKY .ANA	06:06:03	ChwKGkNLaVJfNGVUemY4Q0ZiM0R3Z1FkRlBFTnRR
45334	https://yt4.ggpht.com/Ba7vbzMhJ8ry6WbQdLIPRaNO...	2023-08-25T23:06:00.000Z	rainbowstar	NaN	gg	06:06:03	ChwKGkNOemtwSWlUemY4Q0ZhVEF3Z1FkN3pzSU1B

45335 rows × 7 columns

Gambar 4. 7 Contoh dataset saat diakses pada Google Colab

Setelah memasukkan data ke dalam lingkungan Google Colab, langkah-langkah yang kemudian dilakukan adalah *cleaning*, *case folding*, *tokenizing*, *stopword removal*, *stemming*, serta pembersihan data dan pelabelan data.

4.3.1 *Cleaning*

Cleaning atau pembersihan data yang dilakukan adalah dengan menghapus kolom-kolom yang tidak diperlukan, menghapus kolom dengan pesan yang sama atau terduplikasi, serta menghapus tanda baca, Terdapat beberapa kolom yang tidak diperlukan yang berupa kolom *author*, *date*, *author name*, *role*, *video time*, dan kolom ID.

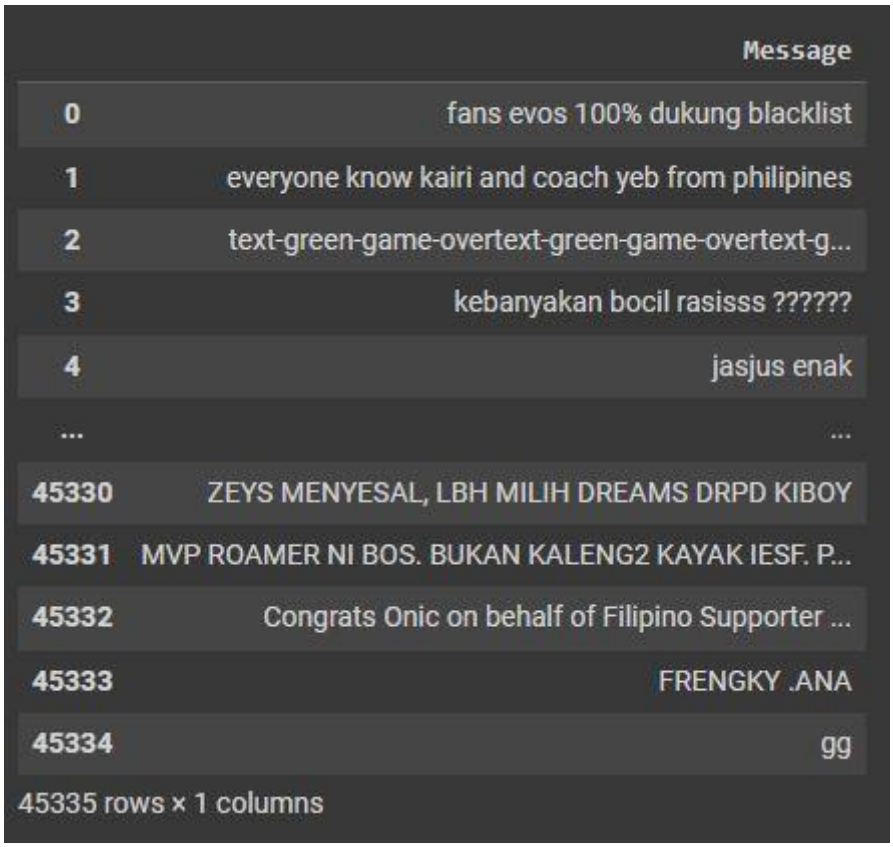
Penghapusan kolom yang tidak diperlukan dilakukan menggunakan kode pada Gambar 4.8 berikut ini pada lingkungan Google Colab.

```
# Menghapus kolom yang tidak diperlukan
df = df.drop('Author', axis=1)
df = df.drop('Date', axis=1)
df = df.drop('Author Name', axis=1)
df = df.drop('Role', axis=1)
df = df.drop('Video Time', axis=1)
df = df.drop('ID', axis=1)
```

Gambar 4. 8 Kode program untuk menghapus kolom yang tidak diperlukan

Kode di atas merupakan perintah yang berfungsi untuk menghapus kolom yang bernama ‘Author’, ‘Date’, ‘Author Name’, ‘Role’, ‘Video Time’ dan kolom yang bernama ‘ID’.

Gambar 4.9 berikut merupakan tampilan dataset setelah penghapusan kolom menggunakan kode diatas pada Google Colab:



	Message
0	fans evos 100% dukung blacklist
1	everyone know kairi and coach yeb from philipines
2	text-green-game-overtext-green-game-overtext-g...
3	kebanyakan bocil rasisss ?????
4	jasjus enak
...	...
45330	ZEYS MENYESAL, LBH MILIH DREAMS DRPD KIBOY
45331	MVP ROAMER NI BOS. BUKAN KALENG2 KAYAK IESF. P...
45332	Congrats Onic on behalf of Filipino Supporter ...
45333	FRENGKY .ANA
45334	gg

45335 rows × 1 columns

Gambar 4. 9 Hasil setelah penghapusan kolom yang tidak diperlukan

Penghapusan kolom yang berisi pesan yang sama atau terduplikasi dilakukan menggunakan kode yang disajikan pada Gambar 4.10 berikut ini pada lingkungan Google Colab:

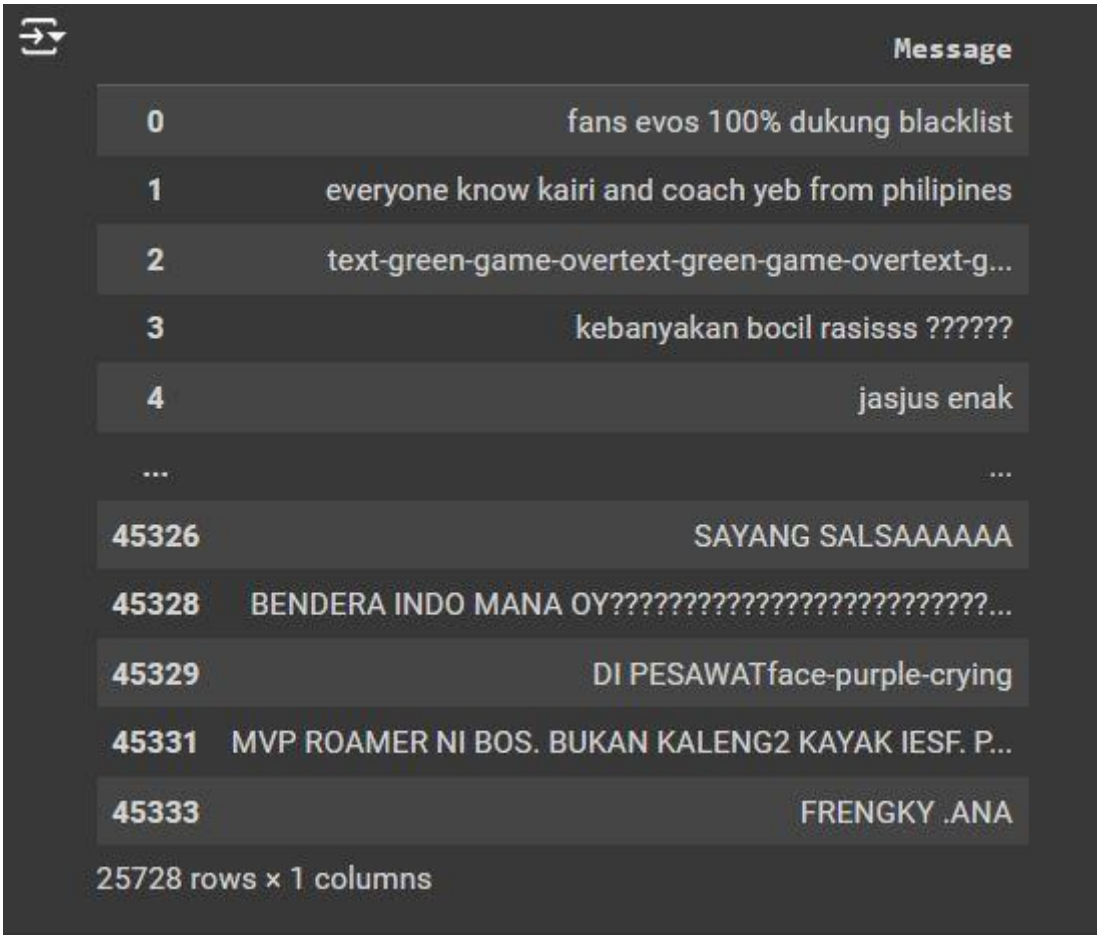
```
df = df.drop_duplicates(subset='Message', keep='first')
```

Gambar 4. 10 Kode program untuk menghapus pesan yang sama

Kode di atas merupakan perintah yang berfungsi untuk menghapus duplikat dari dari dataframe dan hanya mempertimbangkan nilai atau pesan pada kolom ‘Message’ untuk memeriksa duplikasinya dan hanya akan mempertahankan baris pertama.

Pada proses penghapusan kolom dengan isi pesan yang sama atau terduplikasi, telah dilakukan penghapusan pada sekitar 19.607 pesan yang terduplikasi. Dari dataset atau dataframe awal yang berjumlah 45.335, setelah dilakukan penghapusan kolom yang terduplikasi, menyisakan 25.728 pesan.

Hasil dan contoh dari penghapusan kolom dengan pesan yang sama atau terduplikasi dapat dilihat pada Gambar 4.11 dan Tabel 4.1.



	Message
0	fans evos 100% dukung blacklist
1	everyone know kairi and coach yeb from philipines
2	text-green-game-overtext-green-game-overtext-g...
3	kebanyakan bocil rasisss ??????
4	jasjus enak
...	...
45326	SAYANG SALSAAAAAA
45328	BENDERA INDO MANA OY?????????????????????????...
45329	DI PESAWATface-purple-crying
45331	MVP ROAMER NI BOS. BUKAN KALENG2 KAYAK IESF. P...
45333	FRENGKY .ANA

25728 rows × 1 columns

Gambar 4. 11 Hasil setelah penghapusan kolom dengan pesan yang sama

Tabel 4. 1 Sampel pesan yang terduplikasi

Sampel pesan yang terduplikasi
gg onic
kelas
bismillah
onic win
lama
lag
burik
indopride
lemon gay
blacklist

Proses penghapusan tanda baca dilakukan menggunakan kode berikut:

```
    # Fungsi untuk Menghapus tanda baca pada kalimat
    def hapus_tanda_baca(text):
        tanda_baca = string.punctuation
        tanpa_tanda_baca = ''.join(char for char in str(text) if char not
in tanda_baca)
        return tanpa_tanda_baca

    # Menghapus tanda baca
    df['Message'] = df['Message'].apply(hapus_tanda_baca)
```

Gambar 4. 12 Kode program untuk menghapus tanda baca

Kode di atas merupakan perintah yang berfungsi mendefinisikan fungsi hapus_tanda_baca yang akan menghapus tanda baca standar (seperti !, ", #, $, %, &, ', (,), *, +, ,, -, ., /, :, ;, <, =, >, ?, @, [, \,], ^, _, `, {, |, }, ~). dan kemudian menerapkannya pada setiap elemen dalam kolom pesan atau 'Message' dari dataset atau DataFrame 'df'.

Contoh penghapusan tanda baca ditunjukkan pada Tabel 4.2.

Tabel 4. 2 Contoh hasil penghapusan tanda baca

Pesan	Seteleah Proses Penghapusan Tanda Baca
Go Indo, udah bosan PH pegang terus. From Malaysia	Go Indo udah bosan PH pegang terus From Malaysia
#ONICWIN #GOONIC	ONICWIN GOONIC
ONIC TAKE OUR ENERGY!!!??????	ONIC TAKE OUR ENERGY
KASIH PAHAM BOY ??????????????	KASIH PAHAM BOY
GO ONIC GO ONIC GOO..... DARI FANS LEMON INDONESIA	GO ONIC GO ONIC GOO DARI FANS LEMON INDONESIA

4.3.2 *Case Folding*

Case Folding yang dilakukan merupakan proses untuk mengubah semua huruf dalam teks menjadi bentuk yang sama yaitu huruf kecil semua, bukan huruf kapital.

Proses *case folding* atau mengubah huruf menjadi bentuk yang sama dilakukan menggunakan kode berikut ini pada lingkungan Google Colab:

```
df['Message'] = df['Message'].str.lower()
```

Gambar 4. 13 Kode program untuk mengubah menjadi huruf kecil

Gambar 4.1 di atas merupakan tampilan kode yang berfungsi untuk mengakses kolom 'Message' dari DataFrame 'df' dan kemudian akan mengubah huruf pada kolom tersebut menjadi huruf kecil tanpa mengubah teks asli dari pesan string pada kolom tersebut. Contoh hasil dari proses *case folding* dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Contoh hasil proses *case folding*

Pesan	Setelah Proses Case Folding
KASIH PAHAM BOY	kasih paham boy
GO ONIC GO ONIC GOO	go onic go onic goo
Support dari malaysia	support dari malaysia
HARUS CURI POINT game 1	harus curi point game 1

4.3.3 *Tokenizing*

Proses *tokenizing* atau tokenisasi dilakukan untuk memisahkan atau memecah dokumen menjadi kumpulan kata. Kumpulan kata yang sudah terpisah atau terpecah disebut token. Tujuan dari *tokenizing* atau tokenisasi itu sendiri adalah untuk mempermudah proses yang akan dilakukan berikutnya seperti penghapusan stopword dan stemming atau pengubahan kata menjadi kata dasarnya.

Proses tokenizing dilakukan menggunakan kode berikut ini:

```
# Fungsi untuk memisah setiap kata pada DataFrame
def tokenize_text(x):
  return word_tokenize(x)

# Menginisialisasi fungsi tokenisasi
df['Message'] = df['Message'].apply(tokenize_text)
df
```

Gambar 4. 14 Kode program untuk melakukan *tokenizing*

Gamber 4.14 menunjukkan kode yang berfungsi untuk mendefinisikan fungsi tokenize_text yang kemudian akan memanggil fungsi word_tokenize yang berasal dari library nltk. Fungsi ini digunakan untuk membagi teks menjadi token. Fungsi ini memisahkan kata-kata individu berdasarkan spasi. Contoh hasil dari proses tokenizing dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Contoh hasil *tokenizing*

Pesan	Setelah Proses Tokenizing
fans evos 100 dukung blacklist	[fans, evos, 100, dukung, blacklist]
bendera indo mana oy	[bendera, indo, mana, oy]
ban estes digi	[ban, estes, digi]
go onic go	[go, onic, go]
bismilah onic	[bismilah, onic]

4.3.4 Stopword Removal

Stopword removal merupakan penghapusan stopword yang berupa kata-kata umum yang biasanya muncul dalam jumlah banyak dan dianggap tidak memiliki makna yang penting.

```
        # Inisialisasi StopWordRemover Sastrawi
        def create_stopword_remover():
            stopword_factory = StopWordRemoverFactory()
            stopword_remover = stopword_factory.create_stop_word_remover()
            return stopword_remover
        # Fungsi untuk menghapus stop words dari teks
        def remove_stopwords(text):
            stopword_remover = create_stopword_remover()
            return stopword_remover.remove(text)
        df['Message']        =        df['Message'].apply(lambda        tokens:
[remove_stopwords(token) for token in tokens])
```

Gambar 4. 15 Kode program untuk menghapus *stopword*

Gambar 4.15 menunjukkan kode yang berfungsi untuk melakukan penghapusan stopword berdasarkan *library stopword* dari Sastrawi yang berisi kata seperti “dan”, “atau”, “dari”, “itu”, “yang”, ”untuk”, “pada”, “namun”, “menurut”, “setelah” dan sebagainya. Contoh hasil proses penghapusan *stopword* dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Contoh hasil proses penghapusan *stopword*

Pesan	Setelah Penghapusan Stopword
ada pak pulung dan ranger emas pertandingan semakin seru	pak pulung ranger emas pertandingan semakin seru
menang atau kalah itu wajar	menang kalah wajar
dari jam 4 sampai sekarang	jam 4 sekarang
seperti biasa meta pause	biasa meta pause
pada sara isi komen	sara isi komen

4.3.5 Stemming

Stemming merupakan proses normalisasi teks dengan mengubah atau mengembalikan kata menjadi kata dasarnya. Pada tahap *stemming* ini *library* yang digunakan adalah *library* sastrawi.

```
# Inisialisasi Stemmer Sastrawi
def create_stemmer():
    stemmer_factory = StemmerFactory()
    stemmer = stemmer_factory.create_stemmer()
    return stemmer
# Fungsi untuk melakukan stemming pada teks
def stem_text(text):
    stemmer = create_stemmer()
    return [stemmer.stem(word) for word in text]
# Melakukan stemming pada kolom 'Message'
df['Message'] = df['Message'].apply(stem_text)
```

Gambar 4. 16 Kode program untuk melakukan *stemming*

Gamber 4.16 menunjukkan kode yang dijalankan pada saat melakukan proses *stemming* pada teks di kolom ‘Message’ dari dataframe ‘df’. Proses *stemming* ini berguna untuk membantu proses-proses berikutnya dengan mengurangi variasi kata yang berbeda namun memiliki makna yang sama. Contoh hasil proses *stemming* dapat dilihat pada Tabel 4.6.

Tabel 4. 6 Contoh hasil proses *stemming*

Pesan	Setelah Proses Stemming
udah menang unggul kebiasaan dah	udah menang unggul biasa dah
pertandingan semakin seru	tanding makin seru
berguna buat game	guna buat game
rrq kemana	rrq mana
kalahkan negara pelangi	kalah negara pelangi

4.3.6 Pembersihan Data dan Pelabelan Data

Tahap terakhir dari *pre-processing* adalah pembersihan data secara manual yang kemudian akan dilanjutkan dengan proses pelabelan data. Pembersihan data manual yang dilakukan adalah dengan menghapus emoji yang tidak terhapus dalam proses *pre-processing* sebelumnya dikarenakan emoji yang tersedia pada platform youtube sangatlah banyak dan kebanyakan berupa teks apabila dikonversi atau diekspor menjadi file CSV. Contoh hasil penghapusan emoji dapat dilihat pada Tabel 4.7.

Tabel 4. 7 Contoh hasil penghapusan emoji yang terkonversi menjadi teks

Emoji	Emoji yang terkonversi menjadi teks
🎮	textgreengameover
🤜🤛	elbowbump
🔄	buffering
😭	eyespurplecrying
🐶	chillwdog
😊	facebluesmiling

Pelabelan data dilakukan dengan membagi dataset yang sudah melewati tahap pre-processing ke dalam 3 kelas sentimen, yaitu kelas positif, kelas negatif, dan kelas netral. Contoh hasil pelabelan data dapat dilihat pada Tabel 4.8, Tabel 4.9, dan Tabel 4.10.

Tabel 4. 8 Contoh hasil pelabelan data dengan label positif

Pesan dengan Label Positif
bismillah onic juara msc amin
optimis juara
respect buat kiboy
gg onic gg blacklist
indopride
caster favorit nih kb ranger emas pak pulung mantap

Tabel 4. 9 Contoh hasil pelabelan data dengan label negatif

Pesan dengan Label Negatif
lag banget
evos rrq mode badut
lawak kairi
cw beban
nova mulu anj
lama banget co

Tabel 4. 10 Contoh hasil pelabelan data dengan label netral

Pesan dengan Label Netral
skor berapa bang
info skor bang serius nanya
terlalu sengit
onic kurang damage
jangan pernah pick wanwan
m5 kapan bang
jam berapa mulai

Setelah melalui seluruh proses *pre-processing* atau pra-pemrosesan data, jumlah data pesan dalam dataset tentu berkurang. Dataset yang awalnya berisi 45.335 pesan, kini menjadi 19.877, seperti yang ditunjukkan pada Gambar 4.17.



	Label	Message
0	Negatif	fans evos 100 dukung blacklist
1	Netral	everyone know kairi and coach yeb from philipines
2	Netral	banyak bocil rasisss
3	Negatif	intro nya lama bngt
4	Negatif	sara langsung blokir aja adikmereka cuma sam...
...	...	...
19872	Netral	skin atlas gratiiiis
19873	Netral	coach yeb to
19874	Negatif	peran adi apa ups
19875	Netral	bendera indo mana oy
19876	Negatif	mvp roamer ni bos bukan kaleng2 kayak iesf pia...

19877 rows × 2 columns

Gambar 4. 17 Dataset setelah proses *pre-processing*

4.4 Implementasi Pembagian Data dan Pembototan TF-IDF

Setelah melewati tahap pra-pemrosesan data, data yang digunakan harus melalui tahap pembobotan agar data yang masih berupa kata-kata dapat diolah dan dihitung. Data yang belum berupa angka diubah menjadi vektor dan kemudian diberi nilai dan bobot untuk setiap kata atau term-nya. Algoritma yang digunakan dalam pembobotan adalah *Term Frequency – Inverse Document Frequency* (TF-IDF).

Sebelum dilakukan pembobotan, data yang digunakan akan melalui proses splitting data atau pembagian data untuk memisahkan dataset menjadi dua bagian yaitu data latih dan data uji. Data latih merupakan data yang akan dilatihkan kepada model dari Naïve Bayes, sedangkan data uji merupakan data yang akan digunakan untuk menguji model Naïve Bayes yang telah dilatih menggunakan data latih. Rasio pembagian data yang digunakan dalam penelitian ini adalah 80:20. 80% dari dataset digunakan sebagai data latih dan 20% sisanya digunakan menjadi data uji.

Langkah pertama yang dilakukan adalah mengimpor pustaka Python yang digunakan dalam pengolahan data, ekstraksi fitur, pembuatan model, pelatihan model, evaluasi model, dan visualisasi data. Kode yang digunakan adalah sebagai berikut:

```
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder
from sklearn.pipeline import make_pipeline
from sklearn import metrics
from sklearn.model_selection import cross_val_score
import seaborn as sns
```

Gambar 4. 18 Kode program untuk mengimpor *library* yang digunakan

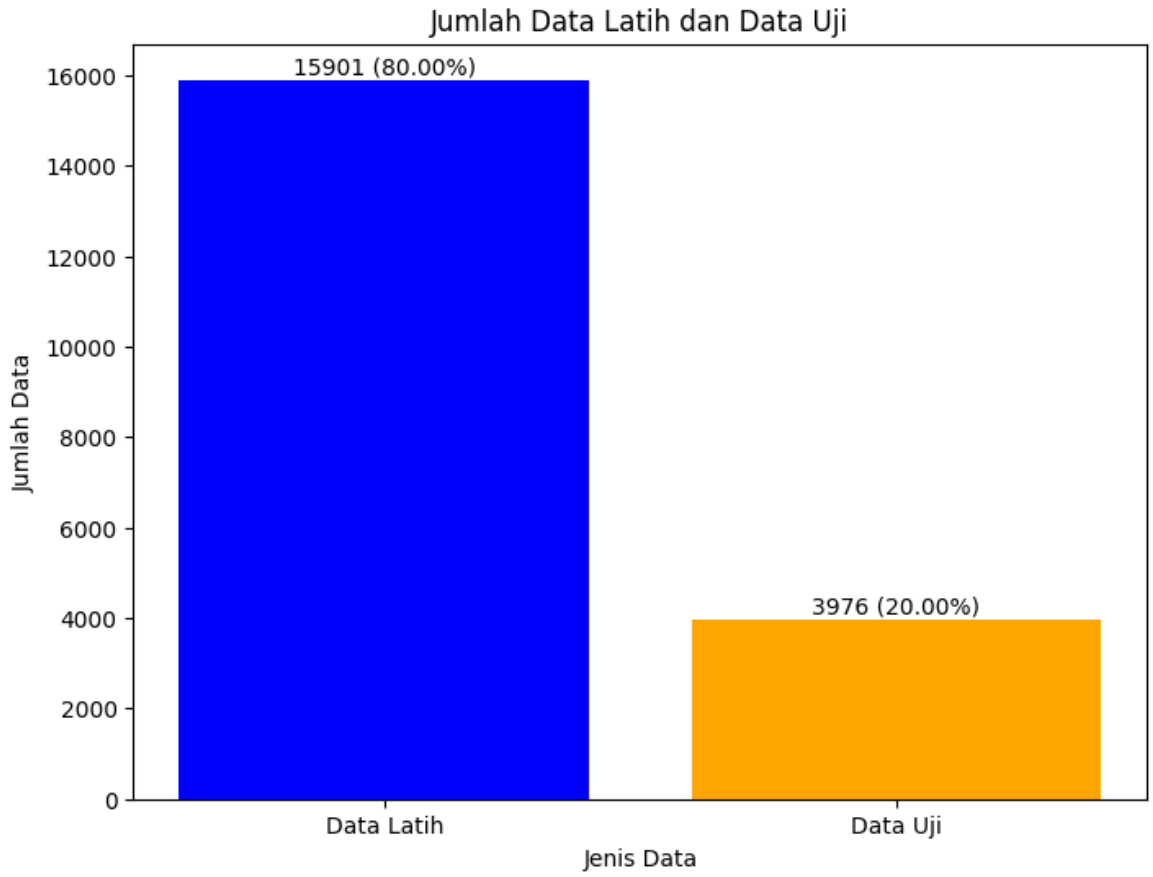
Gambar 4.18 merupakan kode yang digunakan untuk mengimpor pustaka-pustaka yang digunakan seperti TfidfVectorizer, Gaussian Naïve Bayes, Multinomial Naïve Bayes, Bernoulli Naïve Bayes, accuracy_score dan classification_report yang digunakan untuk mengukur kinerja model klasifikasi, metrics yang menyediakan berbagai fungsi untuk mengevaluasi model, cross_val_score yang digunakan untuk mengevaluasi model menggunakan teknik *cross-validation*, dan yang terakhir adalah mengimpor seaborn yang merupakan pustaka visualisasi data yang digunakan untuk pembuatan grafik statistik yang informatif. Kode di atas digunakan untuk menyediakan alat dan metode yang digunakan untuk mengolah data teks dan mengubahnya menjadi fitur berbobot, membuat dan melatih tiga model Naïve Bayes untuk klasifikasi teks, mengukur dan melaporkan kinerja model menggunakan metrik evaluasi, mengubah label kategori menjadi label numerik, membuat pipeline untuk alur kerja pemodelan yang lebih efisien, melakukan *cross-validation* untuk mengevaluasi stabilitas dan kinerja model, serta alat untuk membuat visualisasi data untuk analisis yang lebih mendalam dan informatif.

```
from sklearn.model_selection import train_test_split
# Bagi data menjadi data pelatihan dan pengujian
X_train, X_test, y_train, y_test = train_test_split(data['Message'],
data['Label'], test_size=0.2, random_state=42)
```

Gambar 4. 19 Kode program untuk membagi dataset menjadi data latih dan data uji

Gambar 4.19 merupakan kode yang digunakan untuk membagi dataset menjadi data latih dan data uji dengan rasio 80:20. Dengan rasio tersebut proporsi untuk data uji adalah 20% dari dataset dan proporsi untuk data latih adalah 80% dari dataset. Kolom pesan ‘Message’ digunakan sebagai fitur untuk pelatihan model. Kolom label ‘Label’ digunakan sebagai target

dalam pelatihan model. Hasil pembagian data latih dan data uji dengan rasio 80:20 dalam bentuk diagram dapat dilihat pada Gambar 4.20



Gambar 4. 20 Hasil pembagian data latih dan data uji

Setelah dilakukan pembagian data dari dataset yang berisikan 19877 pesan, terdapat data latih dengan jumlah 15901 data/pesan dan data uji yang berjumlah 3976 data/pesan.

Setelah pembagian data, proses berikutnya adalah pembobotan menggunakan algortima *Term Frequency – Inverse Document Frequency* (TF-IDF). Pembobotan TF-IDF dilakukan dengan menjalankan kode yang ditampilkan pada Gambar 4.21 berikut pada Google Colab:

```
# Mengganti NaN dengan string kosong pada data latih dan uji
X_train = X_train.fillna('')
X_test = X_test.fillna('')

# Membuat TfidfVectorizer
vectorizer = TfidfVectorizer()

# Melakukan vektorisasi pada data latih dan uji
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)
```

Gambar 4. 21 Kode program untuk melakukan pembobotan data

Kode diatas berfungsi untuk memastikan bahwa tidak ada nilai dalam data atau teks sebelum pembobotan, membuat alat untuk mengubah teks menjadi fitur numerik dengan menghitung nilai TF-IDF, dan mengubah data atau teks dalam data latih dan data uji menjadi representasi numerik berbasis TF-IDF yang kemudian akan digunakan sebagai input untuk machine learning.

Tabel 4.11 berikut merupakan beberapa contoh hasil dari pembobotan TF-IDF yang telah dilakukan.

Tabel 4. 11 Contoh hasil pembobotan dengan TF-IDF

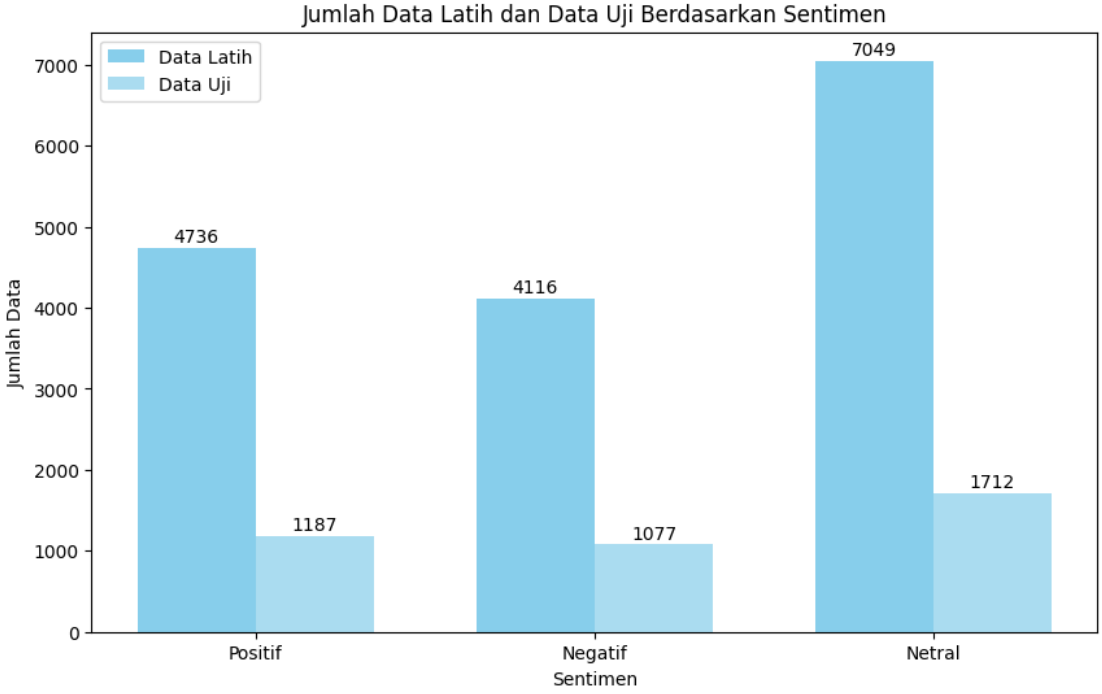
meta pause	
meta	0.720798
pause	0.693145
meta pause kalah	
meta	0.589048
pause	0.566449
kalah	0.576331
kelas gg onic	
kelas	0.751738
gg	0.550725
onic	0.362757
kelas onic	
kelas	0.900623
onic	0.434602

4.5 Pelatihan Model Naïve Bayes

Pada proses pelatihan model ini, model dilatih menggunakan data latih yang telah disiapkan dan akan dievaluasi menggunakan data uji untuk mengukur performanya. Algoritma yang digunakan adalah Naïve Bayes Classifier dengan menggunakan model Gaussian Naïve Bayes, Multinomial Naïve Bayes, dan Bernoulli Naïve Bayes.

Data yang digunakan dalam pelatihan model merupakan data latih yang berjumlah 15.901 data dengan rincian untuk masing-masing kelas positif, negatif dan netral adalah

berjumlah 4.736 data, 4.116 data, dan 7.049 data. Sebaran data latih dan data uji digambarkan pada Gambar 4.22.



Gambar 4. 22 Diagram batang data latih dan data uji berdasarkan label sentimen

Adapun kode yang digunakan untuk melatih ketiga model Naïve Bayes tersebut diatas ditunjukkan pada Gambar 4.23.

```
# Membuat dan melatih model Gaussian Naive Bayes
gnb = GaussianNB()
gnb.fit(X_train_vectorized.toarray(), y_train)

# Membuat dan melatih model Multinomial Naive Bayes
mnb = MultinomialNB()
mnb.fit(X_train_vectorized, y_train)

# Membuat dan melatih model Bernoulli Naive Bayes
bnb = BernoulliNB()
bnb.fit(X_train_vectorized, y_train)
```

Gambar 4. 23 Kode program untuk membuat dan melatih model

Dalam kode di atas, tiga model Naive Bayes yang berbeda dibuat dan dilatih untuk menganalisis data teks. Model pertama adalah Gaussian Naive Bayes (GaussianNB), yang digunakan untuk data yang diasumsikan mengikuti distribusi Gaussian. Karena GaussianNB memerlukan data dalam bentuk array densa, data vektorisasi (X_train_vectorized) diubah menjadi array menggunakan metode .toarray(). Model kedua adalah Multinomial Naive Bayes (MultinomialNB), yang cocok untuk data berbentuk frekuensi fitur, seperti data teks di mana fitur dianggap sebagai jumlah kemunculan atau frekuensi. Dalam hal ini, X_train_vectorized yang merupakan matriks sparse digunakan langsung karena MultinomialNB dapat menangani

data dalam format tersebut. Model ketiga adalah Bernoulli Naive Bayes (BernoulliNB), yang dirancang untuk data biner, di mana fitur dianggap ada atau tidak ada. Sama seperti MultinomialNB, BernoulliNB juga bekerja dengan matriks sparse. Ketiga model ini dilatih menggunakan data pelatihan (X_train_vectorized dan y_train), yang kemudian dapat dievaluasi untuk menilai kinerja mereka dalam menangani data yang tidak terlihat.

4.6 Evaluasi Model Naïve Bayes

Penghitungan dilakukan pada model yang telah dilatih pada tahap pelatihan model untuk dapat dilakukan penilaian terhadap model-model tersebut. Evaluasi model dilakukan untuk memastikan apakah model yang telah dilatih dapat membuat prediksi dengan benar. Evaluasi model yang dilakukan adalah menggunakan *Performance Evaluation Measure* dengan *confusion matrix* dan juga *cross validation*. Terdapat beberapa aspek dalam penilaian, seperti *accuracy, precision, recall*, serta *f1-score*. *Accuracy* sendiri adalah ketepatan model dalam melakukan prediksi terhadap data. *Precision* adalah tingkat kepastian dalam melakukan prediksi atau klasifikasi data. *Recall* digunakan untuk mengukur seberapa baik model dapat memprediksi atau melakukan klasifikasi data. Sedangkan *f1-score* merupakan rataan harmonic dari precision dan recall.

Data yang digunakan dalam evaluasi model merupakan data uji yang berjumlah 3.986 data dengan rincian untuk masing-masing kelas positif, negatif dan netral adalah berjumlah 1.187 data, 1.077 data, dan 1.712 data.

```
        from sklearn.metrics import confusion_matrix

        # Evaluasi model
        y_pred_gnb = gnb.predict(X_test_vectorized.toarray())
        y_pred_mnb = mnb.predict(X_test_vectorized)
        y_pred_bnb = bnb.predict(X_test_vectorized)

        # Lakukan cross-validation dengan 5 folds
        scores_gnb   =   cross_val_score(gnb,   X_train_vectorized.toarray(),
y_train, cv=5)
        scores_mnb = cross_val_score(mnb, X_train_vectorized, y_train, cv=5)
        scores_bnb = cross_val_score(bnb, X_train_vectorized, y_train, cv=5)

        # Menampilkan confusion matrix untuk setiap model
        print("Gaussian Naive Bayes Confusion Matrix:")
        print(confusion_matrix(y_test, y_pred_gnb))

        # Cetak akurasi rata-rata dari cross-validation
        print("Mean    cross-validation    accuracy    for    Gaussian    NB:",
scores_gnb.mean())

        print("\nMultinomial Naive Bayes Confusion Matrix:")
        print(confusion_matrix(y_test, y_pred_mnb))

        # Cetak akurasi rata-rata dari cross-validation
        print("Mean   cross-validation   accuracy   for   Multinomial   NB:",
scores_mnb.mean())

        print("\nBernoulli Naive Bayes Confusion Matrix:")
        print(confusion_matrix(y_test, y_pred_bnb))

        # Cetak akurasi rata-rata dari cross-validation
        print("Mean    cross-validation    accuracy    for    Bernoulli    NB:",
scores_bnb.mean())
```

Gambar 4. 24 Kode program untuk mengevaluasi performansi model

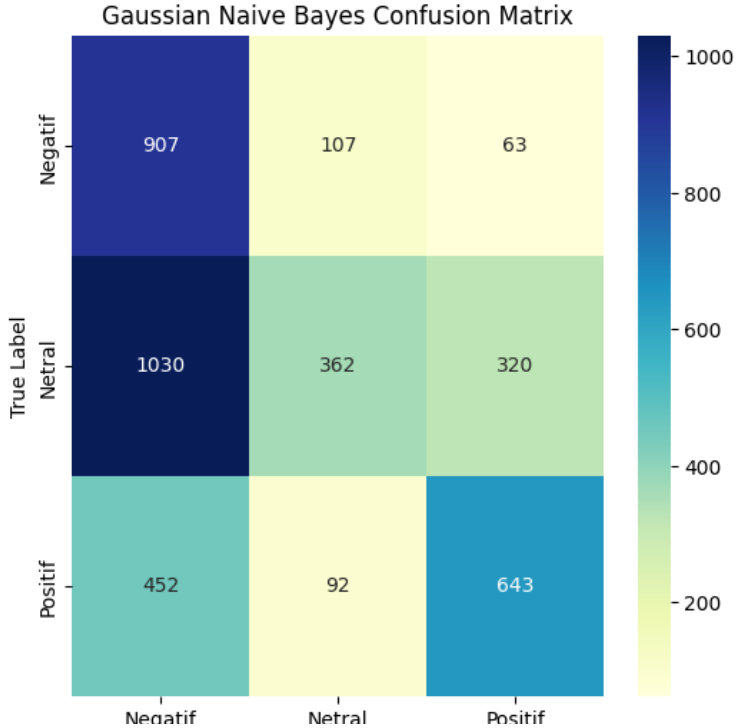
Gambar 4.24 merupakan kode yang digunakan untuk mengevaluasi kinerja tiga model Naive Bayes yaitu Gaussian Naive Bayes, Multinomial Naive Bayes, dan Bernoulli Naive Bayes dalam klasifikasi data. Pertama, kode ini menghitung dan menampilkan *confusion matrix* untuk masing-masing model, yang menunjukkan seberapa baik setiap model dalam memprediksi kelas yang benar dan salah pada data uji. *Confusion matrix* ini memberikan gambaran tentang performa model dalam mengklasifikasikan data ke dalam kategori yang benar. Selanjutnya, kode ini melakukan *cross-validation* dengan 5 *folds* pada data latih untuk setiap model, memberikan akurasi rata-rata dari proses pelatihan model. *Cross-validation* ini membantu dalam mengevaluasi stabilitas dan generalisasi model dengan mengukur seberapa konsisten hasil akurasi pada berbagai subset data latih. Output dari kode ini akan memberikan wawasan tentang kekuatan dan kelemahan masing-masing model dalam tugas klasifikasi yang dihadapi.

Proses *cross-validation* yang dilakukan adalah dengan membagi data menjadi 5 folds atau subset yang kemudian akan digunakan sebagai data latih dan data uji. Evaluasi yang dilakukan dengan *cross-validation* menggunakan salah 1 dari 5 folds atau subset dan dilakukan

pengulangan atau iterasi sampai setiap fold-nya menjadi data uji untuk evaluasi. Dari hasil evaluasi menggunakan cross-validation dengan 5 *folds*, rata-rata akurasi dari model Gaussian Naïve Bayes memiliki nilai 0.484, sedangkan model Multinomial Naïve Bayes memiliki nilai rata-rata akurasi sebesar 0.7225, untuk model Bernoulli Naïve Bayes memiliki nilai rata-rata akurasi sebesar 0.7256.

Confusion matrix didapatkan pada saat melakukan evaluasi model. *Confusion matrix* memberikan gambaran tentang bagaimana model melakukan prediksi dengan mengklasifikasikan hasil menjadi empat kategori yaitu *True Positif* (TP), *False Positif* (FP), *True Negatif* (TN), dan *False Negatif* (FN). Dari *confusion matrix* tersebut, nilai akurasi, presisi, *recall*, serta *f1-score* dapat dihitung. Akurasi mengukur proporsi prediksi benar dari total prediksi, memberikan gambaran umum mengenai kinerja model. Presisi mengukur keakuratan prediksi positif atau benar. *Recall*, atau sensitivitas, mengukur kemampuan model dalam mendeteksi semua sampel positif yang sebenarnya. Sedangkan *f1-score* merupakan rata-rata harmonis dari presisi dan *recall*. Dengan menggunakan *confusion matrix*, performa model dapat dievaluasi dan dipahami secara komprehensif.

Berikut merupakan confusion matrix dari evaluasi 3 model Naïve Bayes yang ditunjukkan pada Gambar 4.25, Gambar 4.26, dan Gambar 4.27.



Gambar 4. 25 Confusion matrix dari model Gaussian

Dari nilai confusion matrix model Gaussian Naïve Bayes diatas, hasil penghitungan untuk *accuracy*, *precision*, *recall*, serta *f1-score* adalah sebagai berikut:

$$Akurasi = \frac{907+362+643}{907 + 107 + 63 + 1030 + 362 + 320 + 452 + 92 + 643} = \frac{1912}{3976} = 0.481$$

$$Presisi(negatif) = \frac{TP}{TP+FP} = \frac{907}{907+(1030+452)} = \frac{907}{2389} = 0.38$$

$$Presisi(netral) = \frac{TP}{TP+FP} = \frac{362}{362+(107+92)} = \frac{362}{561} = 0.65$$

$$Presisi(positif) = \frac{TP}{TP+FP} = \frac{643}{643+(63+320)} = \frac{643}{1026} = 0.63$$

$$Presisi(rata-rata) = \frac{0.38+0.65+0.63}{3} = \frac{1.66}{3} = 0.55$$

$$Recall(negatif) = \frac{TP}{TP+FN} = \frac{907}{907+(107+63)} = \frac{907}{1077} = 0.84$$

$$Recall(netral) = \frac{TP}{TP+FN} = \frac{362}{362+(1030+320)} = \frac{362}{1712} = 0.21$$

$$Recall(positif) = \frac{TP}{TP+FN} = \frac{643}{643+(452+92)} = \frac{643}{1187} = 0.54$$

$$Recall(average) = \frac{0.842+0.211+0.54}{3} = \frac{1.59}{3} = 0.53$$

$$f1score(negatif) = \frac{2(Presisi\ .\ Recall)}{Presisi+Recall} = \frac{2(0.38\ .0.84)}{0.38+0.84} = \frac{0.6384}{1.22} = 0.52$$

$$f1score(netral) = \frac{2(Presisi\ .\ Recall)}{Presisi+Recall} = \frac{2(0.65\ .0.21)}{0.65+0.21} = \frac{0.273}{0.86} = 0.32$$

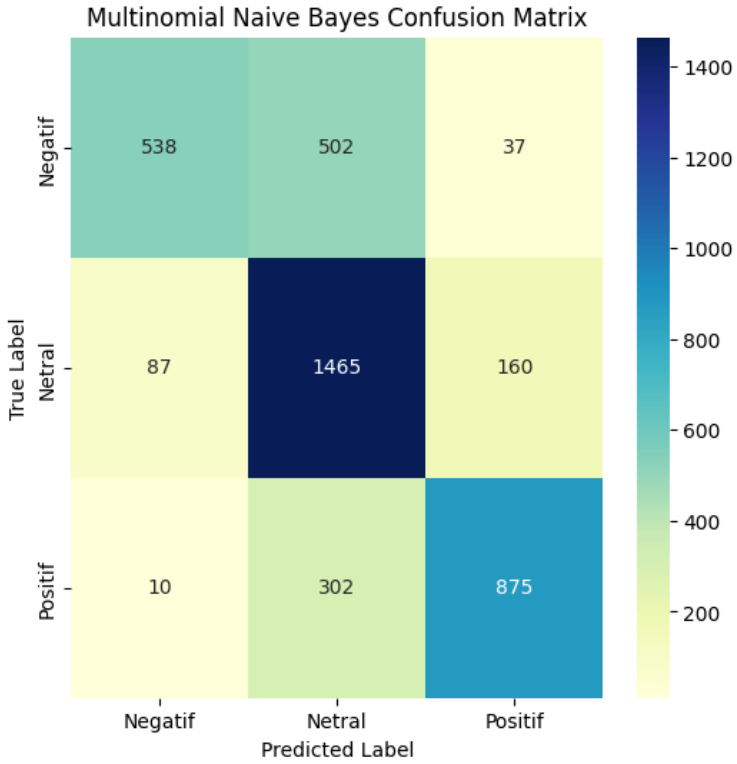
$$f1score(positif) = \frac{2(Presisi\ .\ Recall)}{Presisi+Recall} = \frac{2(0.63\ .0.54)}{0.63+0.54} = \frac{0.6804}{1.17} = 0.58$$

$$f1score(average) = \frac{0.52+0.32+0.58}{3} = \frac{1.42}{3} = 0.47$$

Hasil evaluasi dari model Gaussian disajikan pada Tabel 4.12.

Tabel 4. 12 Hasil evaluasi dari model Gaussian

Gaussian Naïve Bayes	
Accuracy	0.48
Precision	0.55
Recall	0.53
F1-Score	0.47



Gambar 4. 26 Confusion matrix dari model Multinomial

Dari nilai confusion matrix model Multinomial Naïve Bayes diatas, hasil penghitungan untuk accuracy, precision, recall, serta f1 score adalah sebagai berikut:

$$Akurasi = \frac{538+1465+875}{538+502+37+87+1465+160+10+302+875} = \frac{2878}{3976} = 0.7238$$

$$Presisi(negatif) = \frac{TP}{TP+FP} = \frac{538}{538+(87+10)} = \frac{538}{635} = 0.85$$

$$Presisi(netral) = \frac{TP}{TP+FP} = \frac{1465}{1465+(502+302)} = \frac{1465}{2269} = 0.65$$

$$Presisi(positif) = \frac{TP}{TP+FP} = \frac{875}{875+(37+160)} = \frac{875}{1072} = 0.82$$

$$Presisi(rata-rata) = \frac{0.85+0.65+0.82}{3} = \frac{2.32}{3} = 0.77$$

$$Recall(negatif) = \frac{TP}{TP+FN} = \frac{538}{538+(502+37)} = \frac{538}{1077} = 0.50$$

$$Recall(netral) = \frac{TP}{TP+FN} = \frac{1465}{1465+(87+160)} = \frac{1465}{1712} = 0.86$$

$$Recall(positif) = \frac{TP}{TP+FN} = \frac{875}{875+(10+302)} = \frac{875}{1187} = 0.74$$

$$Recall(average) = \frac{0.50+0.86+0.74}{3} = \frac{2.1}{3} = 0.70$$

$$f1score(negatif) = \frac{2(Presisi\ .\ Recall)}{Presisi+Recall} = \frac{2(0.85\ .0.50)}{0.85+0.50} = \frac{0.85}{1.35} = 0.63$$

$$f1score(netral) = \frac{2(Presisi\ .\ Recall)}{Presisi+Recall} = \frac{2(0.65\ .0.86)}{0.65+0.86} = \frac{1.118}{1.51} = 0.74$$

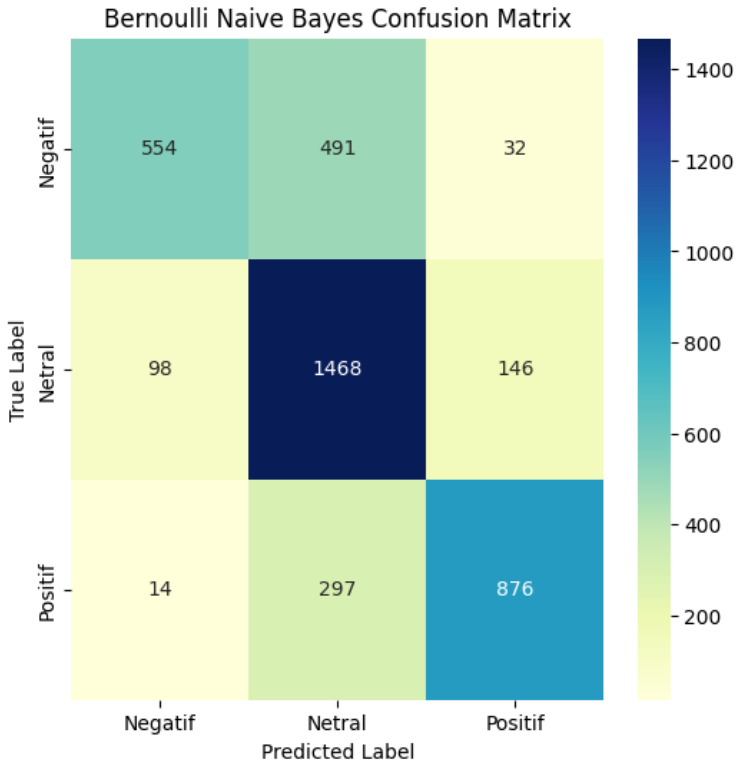
$$f1score(positif) = \frac{2(Presisi\ .\ Recall)}{Presisi+Recall} = \frac{2(0.82\ .0.74)}{0.82+0.74} = \frac{1.2136}{1.56} = 0.77$$

$$f1score(average) = \frac{0.63+0.74+0.77}{3} = \frac{2.14}{3} = 0.71$$

Tabel 4.13 menampilkan hasil evaluasi dari model Multinomial.

Tabel 4. 13 Hasil evaluasi dari model Multinomial

Multinomial Naïve Bayes	
Accuracy	0.72
Precision	0.77
Recall	0.70
F1-Score	0.71



Gambar 4. 27 Confusion matrix dari model Bernoulli

Dari nilai confusion matrix model Bernoulli Naïve Bayes diatas, hasil penghitungan untuk accuracy, precision, recall, serta f1 score adalah sebagai berikut:

$$Akurasi = \frac{554+1468+876}{554+491+32+98+1468+146+14+297+876} = \frac{2898}{3976} = 0.729$$

$$Presisi(negatif) = \frac{TP}{TP+FP} = \frac{554}{554+(98+14)} = \frac{554}{666} = 0.83$$
$$Presisi(netral) = \frac{TP}{TP+FP} = \frac{1468}{1468+(491+297)} = \frac{1468}{2256} = 0.65$$
$$Presisi(positif) = \frac{TP}{TP+FP} = \frac{876}{876+(32+146)} = \frac{876}{1054} = 0.83$$
$$Presisi(rata-rata) = \frac{0.83+0.65+0.83}{3} = \frac{2.31}{3} = 0.77$$

$$Recall(negatif) = \frac{TP}{TP+FN} = \frac{554}{554+(491+32)} = \frac{554}{1077} = 0.51$$
$$Recall(netral) = \frac{TP}{TP+FN} = \frac{1468}{1468+(98+146)} = \frac{1468}{1712} = 0.86$$
$$Recall(positif) = \frac{TP}{TP+FN} = \frac{876}{876+(14+297)} = \frac{876}{1187} = 0.74$$
$$Recall(average) = \frac{0.51+0.86+0.74}{3} = \frac{2.11}{3} = 0.70$$

$$f1score(negatif) = \frac{2(Presisi\ .\ Recall)}{Presisi+Recall} = \frac{2(0.83\ .0.51)}{0.83+0.51} = \frac{0.8466}{1.34} = 0.63$$
$$f1score(netral) = \frac{2(Presisi\ .\ Recall)}{Presisi+Recall} = \frac{2(0.65\ .0.86)}{0.65+0.86} = \frac{1.118}{1.51} = 0.74$$
$$f1score(positif) = \frac{2(Presisi\ .\ Recall)}{Presisi+Recall} = \frac{2(0.83\ .0.74)}{0.83+0.74} = \frac{1.2284}{1.57} = 0.78$$
$$f1score(average) = \frac{0.63+0.74+0.78}{3} = \frac{2.14}{3} = 0.716$$

Hasil evaluasi kinerja yang dicapai oleh model Bernoulli dapat dilihat pada Tabel 4.14

Tabel 4. 14 Hasil evaluasi dari model Bernoulli

Bernoulli Naïve Bayes	
Accuracy	0.73
Precision	0.77
Recall	0.70
F1-Score	0.72

Dalam penelitian ini, tiga model Naïve Bayes, yaitu Gaussian Naïve Bayes, Multinomial Naïve Bayes, dan Bernoulli Naïve Bayes, digunakan untuk melakukan analisis sentimen. Hasil evaluasi menunjukkan bahwa kinerja setiap model bervariasi berdasarkan *confusion matrix*-nya yang menghasilkan metrics akurasi, presisi, *recall*, dan *f1-score*.

Model Gaussian Naïve bayes menunjukkan kinerja dengan hasil yang paling rendah dengan akurasi sebesar 0.48. Presisi model Gaussian Naïve Bayes ini adalah 0.55, yang menunjukkan bahwa dari semua prediksi positif atau benar yang dihasilkan, hanya 55% di antaranya yang benar-benar positif. *Recall* dari model Gaussian Naïve Bayes ini adalah 0.53, yang berarti model ini berhasil mendeteksi 53% dari semua contoh positif atau benar yang ada. *F1-score* yang merupakan rata-rata harmonis dari presisi dan *recall*, pada model Gaussian Naïve Bayes ini berada pada nilai 0.47, mencerminkan keseimbangan antara presisi dan recall yang cukup rendah.

Sebaliknya, model Multinomial Naïve Bayes menunjukkan kinerja yang lebih baik dengan akurasi sebesar 0.72. Presisi model Multinomial Naïve Bayes ini adalah 0.77 , menunjukkan bahwa 77% dari prediksi positif atau benar adalah prediksi yang benar. *Recall* dari model Multinomial Naïve Bayes ini adalah 0.70, yang berarti model ini dapat mendeteksi 70% dari semua contoh positif atau benar yang ada. *F1-score* dari model Multinomial Naïve Bayes ini adalah 0.71, menunjukkan keseimbangan yang cukup baik antara presisi dan *recall*.

Model Bernoulli Naïve Bayes memiliki kinerja yang sedikit lebih baik dibandingkan model Multinomial Naïve Bayes, dengan akurasi sebesar 0.73. Presisi dari model Bernoulli Naïve Bayes ini adalah 0.77, menunjukkan bahwa 77% dari prediksi positif atau benar adalah prediksi yang benar. *Recall* dari model Bernoulli Naïve Bayes ini adalah 0.70, yang berarti model ini dapat mendeteksi 70% dari semua contoh positif atau benar yang ada. *F1-score* pada model Bernoulli Naïve Bayes ini memiliki nilai sebesar 0.71, menunjukkan performa yang seimbang dan sedikit lebih baik dibandingkan dengan Multinomial Naïve Bayes.

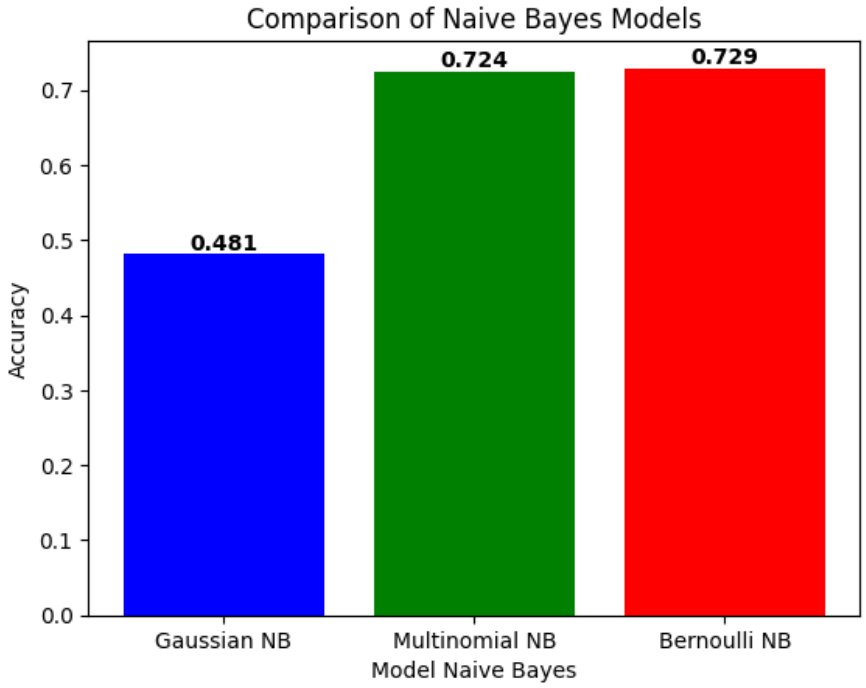
Selain itu hasil *cross-validation* menunjukkan mean *cross-validation accuracy* untuk setuap model. Nilai mean *cross validation accuracy* untuk model Gaussian Naïve Bayes adalah sebesar 0.484, untuk model Multinomial Naïve Bayes adalah 0.723, dan untuk Bernoulli Naïve Bayes adalah 0.726. hasil ini menunjukkan konsistensi kinerja model pada dataset yang berbeda-beda selama proses *cross-validation*, dengan Bernoulli dan Multinomial Naïve Bayes menunjukkan performa yang lebih baik dan stabil dibandingkan dengan model Gaussian Naïve Bayes.

Berdasarkan nilai metrik-metrik evaluasi seperti akurasi, presisi, *recall*, serta *f1-score* yang dihitung dari *confusion matrix*, nilai akurasi tertinggi didapatkan oleh model Bernoulli dengan nilai 0.73, nilai presisi tertinggi didapatkan oleh model Bernoulli dan Multinomial yang masing-masing mendapatkan nilai 0.77, nilai recall tertinggi didapatkan kembali oleh model Bernoulli dan Multinomial dengan nilai 0.70, sedangkan untuk nilai f1-score tertinggi dicatatkan oleh model Bernoulli dengan nilai 0.72.

Performa tiap model dalam mengklasifikasikan sentimen tertentu seperti sentimen negatif, netral, maupun sentimen positif dapat dinilai dari nilai *f1-score* nya. Model Gaussian Naïve Bayes, pada kelas negatif mendapatkan nilai *f1-score* sebesar 0.52, sedangkan pada kelas netral mendapatkan nilai sebesar 0.32, dan pada kelas positif mendapatkan nilai sebesar 0.58. Model Multinomial Naïve Bayes, pada kelas negatif mendapatkan nilai *f1-score* sebesar 0.63, sedangkan pada kelas netral mendapatkan nilai sebesar 0.74, dan pada kelas positif mendapatkan nilai sebesar 0.77. Model Bernoulli Naïve Bayes, pada kelas negatif mendapatkan nilai *f1-score* sebesar 0.63, sedangkan pada kelas netral mendapatkan nilai sebesar 0.74, dan pada kelas positif mendapatkan nilai sebesar 0.78. Hasil tersebut dapat menunjukkan bahwa model-model Naïve Bayes cenderung lebih efektif dalam mengklasifikasikan sentimen positif dan negatif dibandingkan dengan sentimen netral.

Secara keseluruhan, model Bernoulli dan Multinomial Naïve Bayes menunjukkan performa yang superior dalam analisis sentimen ini, dengan kinerja yang lebih baik dalam hal akurasi, presisi, *recall*, dan *f1-score* dibandingkan dengan model Gaussian Naïve Bayes. Hasil ini mengindikasikan bahwa Bernoulli dan Multinomial Naïve Bayes lebih efektif dalam menangkap pola dan sentimen kata dalam data teks yang digunakan dalam penelitian ini.

Gambar 4.28 berikut merupakan visualisasi nilai akurasi dari ketiga model Naïve Bayes yang digunakan dalam penelitian ini:



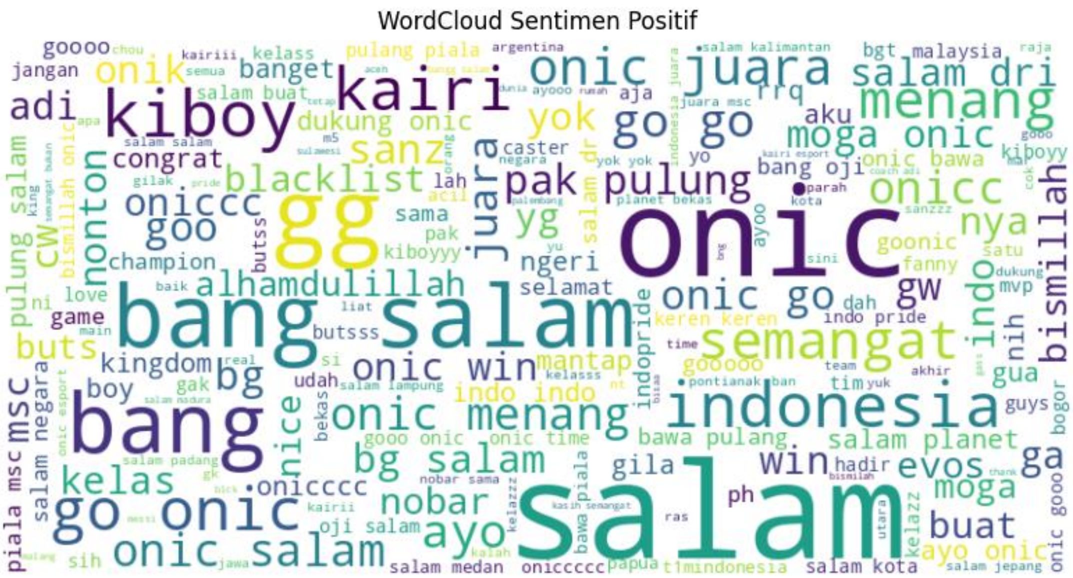
Gambar 4. 28 Hasil evaluasi 3 model Naive Bayes

4.7 Visualisasi Data

Visualisasi data dilakukan terhadap keseluruhan data. Pada *wordcloud* positif hanya terdapat kata yang paling sering muncul dalam kelas berlabel positif, pada wordcloud negatif hanya terdapat kata yang paling sering muncul dalam kelas berlabel negatif, sedangkan pada *wordcloud* netral hanya terdapat kata yang paling sering muncul dalam kelas berlabel netral. Dalam penelitian ini, *wordcloud* digunakan untuk memberikan gambaran visual tentang kata-kata yang paling sering muncul dalam data yang dianalisis. Ukuran setiap kata dalam *wordcloud* berbanding lurus dengan frekuensi kemunculannya, sehingga kata-kata yang lebih sering muncul akan ditampilkan lebih besar dan lebih menonjol. Teknik ini tidak hanya memudahkan dalam mengidentifikasi kata-kata kunci atau tema utama dalam data, tetapi juga membantu dalam memahami distribusi frekuensi kata secara keseluruhan. *Wordcloud* dapat menjadi alat yang sangat berguna dalam eksplorasi data teks, karena mampu menyajikan informasi secara ringkas dan langsung.

Dengan membandingkan wordcloud dari masing-masing kelas, peneliti dapat memperoleh wawasan mendalam tentang bagaimana kata-kata tertentu digunakan dalam konteks sentimen yang berbeda, serta bagaimana perbedaan dalam frekuensi kata dapat mempengaruhi analisis sentimen secara keseluruhan.

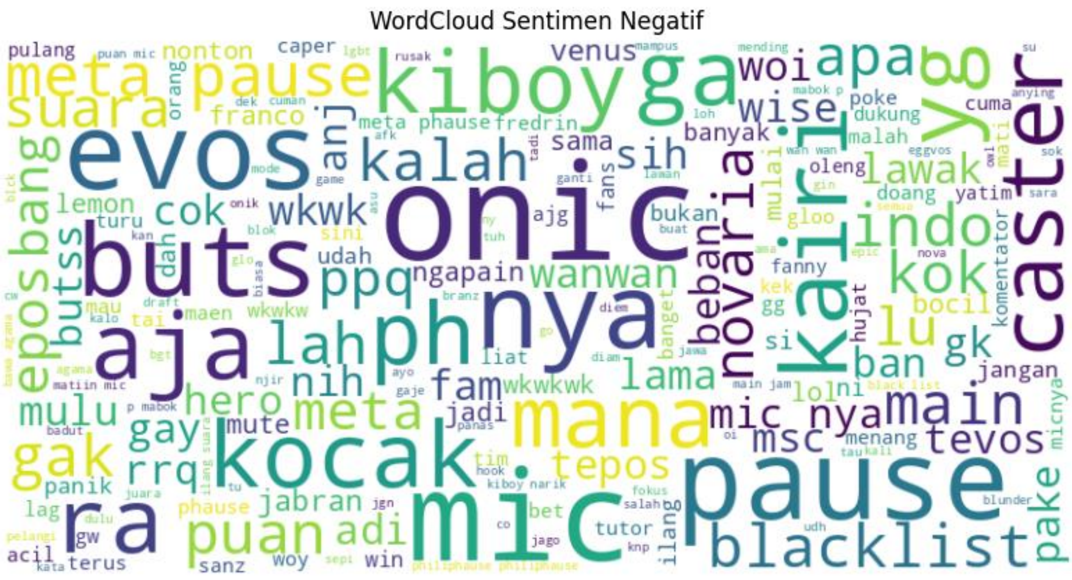
WordCloud sentimen positif, WordCloud sentiment negatif, WordCloud sentimen netral, dan WordCloud keseluruhan data disajikan pada Gambar 4.29, Gambar 4.30, Gambar 4.31, dan Gambar 4.32.



Gambar 4. 29 *WordCloud* pesan dengan label positif

Wordcloud untuk kelas positif menampilkan kata-kata yang sering muncul dalam teks dengan sentimen positif. Ini membantu mengidentifikasi kata-kata yang berkaitan dengan perasaan atau reaksi positif dari pengguna. Visualisasi data menggunakan *wordcloud* ini bertujuan untuk memperoleh informasi secara menyeluruh mengenai apa yang sering dibicarakan terkait topik yang dianalisis yang merupakan pesan atau live chat pada kolom live chat siaran Grand Final MSC 2023 pada kanal Youtube Mobile Legends: Bang Bang.

Kata-kata yang paling sering masuk ke dalam kelas dengan sentimen berlabel positif pada kolom *live chat* siaran Grand Final MSC 2023 pada kanal Youtube Mobile Legends: Bang Bang adalah kata “gg”, “onic”, “salam”, “semangat”, “go onic”, “juara”, dan lain-lain sesuai dengan *wordcloud* di atas. Ukuran setiap kata dalam *wordcloud* berbanding lurus dengan frekuensi kemunculannya, sehingga kata-kata yang lebih sering muncul akan ditampilkan lebih besar dan lebih menonjol. Hal itu berarti bahwa penonton sering menggunakan kata-kata tersebut sebagai topik yang sering dibicarakan pada kolom *live chat* siaran Grand Final MSC 2023 pada kanal Youtube Mobile Legends: Bang Bang.



Gambar 4. 30 *WordCloud* pesan dengan label negatif

Wordcloud untuk kelas negatif menampilkan kata-kata yang sering muncul dalam teks dengan sentimen negatif. Ini membantu mengidentifikasi kata-kata yang berkaitan dengan perasaan atau reaksi negatif dari pengguna. *Wordcloud* untuk kelas negatif menyoroti kata-kata yang sering digunakan dalam teks dengan sentimen negatif, memberikan wawasan tentang aspek atau isu yang mungkin menjadi sumber ketidakpuasan atau kritik. Visualisasi data

menggunakan *wordcloud* ini bertujuan untuk memperoleh informasi secara menyeluruh mengenai apa yang sering dibicarakan terkait topik yang dianalisis yang merupakan pesan atau *live chat* pada kolom *live chat* siaran Grand Final MSC 2023 pada kanal Youtube Mobile Legends: Bang Bang.

Kata-kata yang paling sering masuk ke dalam kelas dengan sentimen berlabel positif pada kolom *live chat* siaran Grand Final MSC 2023 pada kanal Youtube Mobile Legends: Bang Bang adalah kata “kocak”, “onic”, “caster”, “pause”, “kalah”, “evos”, dan lain-lain sesuai dengan *wordcloud* di atas. Ukuran setiap kata dalam *wordcloud* berbanding lurus dengan frekuensi kemunculannya, sehingga kata-kata yang lebih sering muncul akan ditampilkan lebih besar dan lebih menonjol. Hal itu berarti bahwa penonton sering menggunakan kata-kata tersebut sebagai topik yang sering dibicarakan pada kolom *live chat* siaran Grand Final MSC 2023 pada kanal Youtube Mobile Legends: Bang Bang.



Gambar 4. 31 *WordCloud* pesan dengan label netral

Wordcloud untuk kelas netral menampilkan kata-kata yang sering muncul dalam teks dengan sentimen netral. Ini membantu mengidentifikasi kata-kata yang berkaitan dengan perasaan atau reaksi negatif dari pengguna. *Wordcloud* untuk kelas netral menampilkan kata-kata yang sering muncul dalam teks dengan sentimen netral, yang sering kali mencerminkan informasi faktual atau diskusi tanpa emosi yang kuat.. Visualisasi data menggunakan wordcloud ini bertujuan untuk memperoleh informasi secara menyeluruh mengenai apa yang sering dibicarakan terkait topik yang dianalisis yang merupakan pesan atau *live chat* pada

kolom *live chat* siaran Grand Final MSC 2023 pada kanal Youtube Mobile Legends: Bang Bang.

Kata-kata yang paling sering masuk ke dalam kelas dengan sentimen berlabel netral pada kolom *live chat* siaran Grand Final MSC 2023 pada kanal Youtube Mobile Legends: Bang Bang adalah kata “ban”, “kalah”, “menang”, “main”, “mana”, “msc”, dan lain-lain sesuai dengan *wordcloud* di atas. Ukuran setiap kata dalam *wordcloud* berbanding lurus dengan frekuensi kemunculannya, sehingga kata-kata yang lebih sering muncul akan ditampilkan lebih besar dan lebih menonjol. Hal itu berarti bahwa penonton sering menggunakan kata-kata tersebut sebagai topik yang sering dibicarakan pada kolom *live chat* siaran Grand Final MSC 2023 pada kanal Youtube Mobile Legends: Bang Bang.

Berikut merupakan visualisasi data dalam seluruh kelas sentimen menggunakan *wordcloud*:



Gambar 4. 32 WordCloud keseluruhan data

Kata-kata yang paling sering masuk atau muncul pada kolom *live chat* siaran Grand Final MSC 2023 pada kanal Youtube Mobile Legends: Bang Bang adalah kata “salam”, “kairi”, “gg”, “onic”, “bang”, “kiboy”, “indonesia”, dan lain-lain sesuai dengan *wordcloud* di atas. Ukuran setiap kata dalam *wordcloud* berbanding lurus dengan frekuensi kemunculannya, sehingga kata-kata yang lebih sering muncul akan ditampilkan lebih besar dan lebih menonjol. Hal itu berarti bahwa penonton sering menggunakan kata-kata tersebut sebagai topik yang sering dibicarakan pada kolom *live chat* siaran Grand Final MSC 2023 pada kanal Youtube Mobile Legends: Bang Bang.

Wordcloud keseluruhan memberikan pandangan umum tentang kata-kata yang dominan dalam seluruh korpus data. Kata-kata yang sering muncul dalam data keseluruhan akan tampil

lebih besar dan lebih menonjol, sehingga memudahkan dalam mengidentifikasi topik atau tema utama.

Berdasarkan evaluasi dengan menilai keberagaman sentimen-sentimen yang diberikan oleh penonton, pemahaman dan pengembangan strategi yang perlu dilakukan dan dikembangkan adalah mempertahankan hal-hal yang positif seperti penampilan performa tim yang bertanding, sportifitas yang ditunjukkan, serta semangat dan nasionalisme yang selalu dijunjung tinggi. Hal yang perlu ditingkatkan adalah menjaga kualitas stabilitas penyiaran agar tidak terjadi lag atau hilangnya suara seperti yang terjadi saat pertandingan final MSC tersebut, meminimalisir kesalahan atau masalah teknis pada saat pertandingan berlangsung sehingga tidak mengganggu jalannya pertandingan dengan adanya jeda atau pause, dan yang paling penting adalah dengan selalu mengedukasi semua orang untuk tetap menjaga kesopanan, menghargai perbedaan, serta menjaga ucapan atau ketikan.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil membandingkan performansi dari tiga varian model Naïve Bayes dalam menganalisis sentimen mengenai esport Mobile Legends. Hasil analisis menunjukkan bahwa model Bernoulli Naïve Bayes menunjukkan performa terbaik, sedikit lebih unggul dari Multinomial Naïve Bayes. Hasil ini menunjukkan bahwa Bernoulli lebih sesuai dalam menangani pola sentimen pada data teks singkat yang umum ditemukan pada kolom *live chat*. Penelitian ini juga memberikan kontribusi sebagai referensi tambahan mengenai penerapan model Naïve Bayes dalam analisis sentimen, khususnya dalam bidang atau industri esport serta memberikan wawasan yang bermanfaat bagi penyelenggara acara dalam memahami reaksi komunitas atau audiens melalui analisis sentimen ini. Temuan ini juga diharapkan dapat mendukung pengambilan keputusan yang lebih baik dalam pengelolaan komunitas dan industri esport secara umum, dan Mobile Legends khususnya.

Berdasarkan penelitian terhadap kolom *live chat* rekaman siaran langsung Grand Final MSC 2023, didapatkan kesimpulan yaitu Gaussian, Multinomial, dan Bernoulli Naïve Bayes dapat mengklasifikasikan komentar ke dalam tiga kelas label sentimen dengan hasil performansi yang bervariasi. Berdasarkan hasil evaluasi, model Bernoulli dan Multinomial Naïve Bayes terbukti memberikan performa yang lebih baik dibandingkan Gaussian Naïve Bayes dengan akurasi masing-masing sebesar 0,73 dan 0,72 sementara Gaussian hanya mendapatkan akurasi sebesar 0,48. Bernoulli dan Multinomial Naïve Bayes juga menunjukkan keseimbangan antara presisi dan *recall* yang baik, dengan *f1-score* sebesar 0,71. *Cross-validation* juga mengonfirmasi konsistensi performa Multinomial dan Bernoulli, dengan *mean cross-validation accuracy* masing-masing sebesar 0,723 dan 0,726, sedangkan Gaussian hanya mencapai 0,484. Secara keseluruhan, model Bernoulli Naïve Bayes memperlihatkan kinerja terbaik dengan performa sedikit lebih tinggi daripada Multinomial Naïve Bayes. Hasil ini menunjukkan bahwa Bernoulli lebih sesuai dalam menangani pola sentimen pada data teks pendek seperti komentar *live chat*, sehingga dapat menjadi opsi utama dalam analisis sentimen untuk jenis data serupa. Berdasarkan evaluasi sentimen penonton, strategi yang perlu dikembangkan dalam mengelola komunitas esport dan Mobile Legends adalah mempertahankan hal-hal positif seperti performa tim serta sportivitas. Peningkatan diperlukan dalam menjaga kualitas penyiaran untuk menghindari masalah-masalah teknis. Edukasi juga perlu dilakukan untuk menjaga kondusifitas ekosistem industri esports dan Mobile Legends ini.

5.2 Saran

Pada penelitian ini, masih banyak keterbatasan dan kekurangan, oleh karena itu penulis memberikan saran untuk penelitian selanjutnya, agar proses pelabelan data dilakukan dengan mempertimbangkan konteks kalimat secara lebih mendalam, sehingga makna sentimen dapat terinterpretasi dengan lebih akurat, terutama dalam situasi ambigu. Peneliti menyadari bahwa asumsi independensi fitur dalam Naïve Bayes sering kali tidak realistis untuk data teks, yang dapat mempengaruhi akurasi model. Oleh karena itu, penelitian selanjutnya bisa mengeksplorasi pendekatan yang lebih dapat menangani ketergantungan antar fitur dalam teks. Proses prapemrosesan juga perlu ditingkatkan, misalnya dengan memperkaya pustaka (*library*) prapemrosesan untuk mengenali lebih banyak kata slang, istilah khusus, atau ekspresi informal yang sering muncul pada *live chat*. Untuk studi di masa mendatang, disarankan untuk mengimplementasikan pendekatan yang mampu menangani data multibahasa atau melakukan translasi data terlebih dahulu sebelum proses prapemrosesan. Selain itu, verifikasi oleh ahli bahasa juga diperlukan untuk memastikan bahwa data telah melalui tahap pelabelan dan prapemrosesan dengan benar, serta menghindari kesalahan interpretasi kata yang mungkin mengubah makna sentimen. Untuk interpretasi yang lebih mendalam, selain confusion matrix dan word cloud, visualisasi seperti *Precision-Recall Curve*, *ROC Curve*, dan *heatmap* juga dapat digunakan untuk evaluasi performa yang lebih menyeluruh. Saran ini diharapkan dapat memperbaiki akurasi dan efektivitas model serta interpretasinya dalam menganalisis sentimen data teks pada penelitian selanjutnya.

DAFTAR PUSTAKA

Amardita, R. S., Adiwijaya, A., & Purbolaksono, M. D. (2022). Analisis Sentimen terhadap Ulasan Paris Van Java Resort Lifestyle Place di Kota Bandung Menggunakan Algoritma KNN. *JURIKOM (Jurnal Riset Komputer)*, *9*(1), 62. https://doi.org/10.30865/jurikom.v9i1.3793

Astuti, Y., Wulandari, I. R., Putra, A. R., & Kharomadhona, N. (2022). Naïve Bayes untuk Prediksi Tingkat Pemahaman Kuliah Online Terhadap Mata Kuliah Algoritma Struktur Data. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *8*(1), 28. https://doi.org/10.26418/jp.v8i1.48848

Bayes, T. (1968). Naïve Bayes classifier & Evaluation framework Naïve Bayes classifier. *Naïve Bayes Classifier Thoma*.

Berrar, D. (2018). Bayes' theorem and naive bayes classifier. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, *1–3*, 403–412. https://doi.org/10.1016/B978-0-12-809633-8.20473-1

Chedia Dhaoui, Cynthia M. Webster, L. P. T. (2017). *Article information : SOCIAL MEDIA SENTIMENT ANALYSIS: LEXICON VERSUS MACHINE LEARNING*.

Cortes, C., & Vapnik, V. (2009). Support-Vector Networks. *Chemical Biology and Drug Design*, *74*(2), 142–147. https://doi.org/10.1111/j.1747-0285.2009.00840.x

Domingos, P., & Pazzani, M. (1997). On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning*, *29*(2–3), 103–130. https://doi.org/10.1023/a:1007413511361

Dwi Putra, H., Khairani, L., Hastari, D., Studi Sistem Informasi, P., Sains dan Teknologi, F., & Author, C. (2023). SENTIMAS: Seminar Nasional Penelitian dan Pengabdian Masyarakat Comparison of Naive Bayes Classifier and Support Vector Machine Algorithms for Classifying Student Mental Health Data Perbandingan Algoritma Naive Bayes Classifier dan Support Vector Machine . *SENTIMAS: Seminar Nasional Penelitian Dan Pengabdian Masyarakat*, 120–125. Retrieved from https://journal.irpi.or.id/index.php/sentimas

Fakhruddin, A., Kom, J. S., & Ridwan, A. (2023). *Performance Measurement of Confusion Matrix Accuracy in Sentiment Analysis with Decision Trees , Naïve Bayes , K-Nearest Neighbor Methods Using Rapidminer*. *8*(4), 123–127.

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, *30*(1), 27–38.

https://doi.org/10.1016/j.patrec.2008.08.010

Fushiki, T. (2011). Estimation of prediction error by using K-fold cross-validation. *Statistics and Computing*, *21*(2), 137–146. https://doi.org/10.1007/s11222-009-9153-8

Gunawan, B., Sastypratiwi, H., & Pratama, E. E. (2018). Sistem Analisis Sentimen pada Ulasan Produk Menggunakan Metode Naive Bayes. *JEPIN (Jurnal Edukasi Dan Penelitian Informatika)*, *4*(2), 113–118.

Hafid, H. (2023). Penerapan K-Fold Cross Validation untuk Menganalisis Kinerja Algoritma K-Nearest Neighbor pada Data Kasus Covid-19 di Indonesia. *Journal of Mathematics*, *6*(2), 161–168. Retrieved from http://www.ojs.unm.ac.id/jmathcos

Herbert, K. G., & Wang, J. T. L. (2007). Biological data cleaning: A case study. *International Journal of Information Quality*, *1*(1), 60–82. https://doi.org/10.1504/IJIQ.2007.013376

Hua, W., Cuiqin, M., & Lijuan, Z. (2009). Proceedings - 2009 International Conference on Information Engineering and Computer Science, ICIECS 2009. *Proceedings - 2009 International Conference on Information Engineering and Computer Science, ICIECS 2009*.

Id, I. D. (2021). *Machine Learning: Teori, Studi Kasus dan Implementasi Menggunakan Python (Vol. 1)*.

Jannah, Y. A. N., & Prasetyo, R. B. (2022). Analisis Sentimen dan Emosi Publik pada Awal Pandemi COVID-19 Berdasarkan Data Twitter dengan Pendekatan Berbasis Leksikon (Analysis of Public Sentiment and Emotion at the Beginning of COVID-19 Pandemic Based on Twitter Data with Lexicon Based Approach). *Seminar Nasional Official Statistics 2022*, 597–607.

Jiao, Y., & Du, P. (2016). Performance measures in evaluating machine learning based bioinformatics predictors for classifications. *Quantitative Biology*, *4*(4), 320–330. https://doi.org/10.1007/s40484-016-0081-2

Johannes Kurniawan. (2023). Analisis Dan Visualisasi Data. In *Angewandte Chemie International Edition, 6(11), 951–952.* (Vol. 3). Retrieved from https://medium.com/@arifwicaksanaa/pengertian-use-case-a7e576e1b6bf

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260. https://doi.org/10.1126/science.aaa8415

Kohavi, R., & Edu, S. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, *2*, 1137–1143.

Kolchyna, O., Souza, T. T. P., Treleaven, P., & Aste, T. (2015). *Twitter Sentiment Analysis:*

Lexicon Method, Machine Learning Method and Their Combination. Retrieved from http://arxiv.org/abs/1507.00955

Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: A review of classification and combining techniques. *Artificial Intelligence Review*, *26*(3), 159–190. https://doi.org/10.1007/s10462-007-9052-3

Kusnanda, D., & Permana, A. (2023). Implementation of Naive Bayes Classifier (NBC) for Sentiment Analysis on Twitter in Mobile Legends. *International Journal of Science, Technology & Management*, *4*(5), 1132–1138. https://doi.org/10.46729/ijstm.v4i5.935

Liu, B. (2020). Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Second Edition. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Second Edition*, (May), 1–432. https://doi.org/10.1017/9781108639286

Liu, Y., & Liao, S. (2017). Granularity selection for cross-validation of SVM. *Information Sciences*, *378*, 475–483. https://doi.org/10.1016/j.ins.2016.06.051

Lowd, D., & Domingos, P. (2005). Naive Bayes models for probability estimation. *ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning*, 529–536. https://doi.org/10.1145/1102351.1102418

Maimon, O. Z., & Rokach, L. (2014). *Data mining with decision trees: theory and applications, 2nd Edition*. *1*(22 Jan), 1–17.

Mardiana, L., Kusnandar, D., & Satyahadewi, N. (2022). Analisis Diskriminan Dengan K Fold Cross Validation Untuk Klasifikasi Kualitas Air Di Kota Pontianak. *Buletin Ilmiah Mat. Stat. Dan Terapannya (Bimaster)*, *11*(1), 97–102.

Musfiroh, D., Khaira, U., Utomo, P. E. P., & Suratno, T. (2021). Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *1*(1), 24–33. https://doi.org/10.57152/malcom.v1i1.20

Ningsih, W., Alfianda, B., Rahmaddeni, R., & Wulandari, D. (2024). Perbandingan Algoritma SVM dan Naïve Bayes dalam Analisis Sentimen Twitter pada Penggunaan Mobil Listrik di Indonesia. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *4*(2), 556–562. https://doi.org/10.57152/malcom.v4i2.1253

Nurjannah, M., & Fitri Astuti, I. (2013). PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK TEXT MINING Mahasiswa S1 Program Studi Ilmu Komputer FMIPA Universitas Mulawarman Dosen Program Studi Ilmu Komputer FMIPA Universitas Mulawarman. *Jurnal Informatika Mulawarman*, *8*(3), 110–113.

Nurul A'ayunnisa, Salim, Y., & Azis, H. (2022). Analisis Performa Metode Gaussian Naïve Bayes untuk Klasifikasi Citra Tulisan Tangan Karakter Arab. *Indonesian Journal of Data and Science*, *3*(3), 115–121. https://doi.org/10.56705/ijodas.v3i3.54

Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, *1*(2), 91–233. https://doi.org/10.1561/1500000001

Putra, B. G., & Rochmawati, N. (2021). Klasifikasi Berdasarkan Question dalam Stack Overflow Menggunakan Algoritma Naïve Bayes. *Journal of Informatics and Computer Science (JINACS)*, *2*(04), 259–267. https://doi.org/10.26740/jinacs.v2n04.p259-267

Putra, E. P., Safitri, A., Kurniawan, M. A., Vadilah, M. I., Prasetyo, M. A., Wijaya, M. R., … Fansyuri, M. (2023). Merubah Perspektif Masyarakat Terhadap Game Online. *Abdi Jurnal Publikasi*, *1*(6), 485–488.

Ramezan, C. A., Warner, T. A., & Maxwell, A. E. (2019). Evaluation of sampling and cross-validation tuning strategies for regional-scale machine learning classification. *Remote Sensing*, *11*(2). https://doi.org/10.3390/rs11020185

Ravi, K., & Ravi, V. (2015). A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. In *Knowledge-Based Systems* (Vol. 89). Elsevier B.V. https://doi.org/10.1016/j.knosys.2015.06.015

Rochmawati, N., & Wibawa, S. C. (2018). Opinion Analysis on Rohingya using Twitter Data. *IOP Conference Series: Materials Science and Engineering*, *336*(1). https://doi.org/10.1088/1757-899X/336/1/012013

Roy, S., Sharma, P., Nath, K., Bhattacharyya, D. K., & Kalita, J. K. (2018). Pre-processing: A data preparation step. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, *1–3*, 463–471. https://doi.org/10.1016/B978-0-12-809633-8.20457-3

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*, *10*(3), 1–21. https://doi.org/10.1371/journal.pone.0118432

Saputra. (2015). Analisis Sentimen Data Presiden Jokowi Dengan Preprocessing Normalisasi Dan Stemming Menggunakan Metode Naive Bayes Dan Svm. *Jurnal Dinamika Informatika*, *5*(November), 1–13.

Sari, F. V., & Wibowo, A. (2019). Analisis Sentimen Pelanggan Toko Online Jd.Id Menggunakan Metode Naïve Bayes Classifier Berbasis Konversi Ikon Emosi. *Jurnal SIMETRIS*, *10*(2), 681–686.

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2*(3), 1–21. https://doi.org/10.1007/s42979-021-00592-

x

Sinaga, D., & Jatmoko, C. (2020). Analisis Sentimen Untuk Mengetahui Kesan Player Game Mobile Legends Menggunakan Naïve Bayes Classifier. *Semnas LPPM UMP*, *V*, 540–547.

Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, *27*(2), 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044

Soofi, A. A., & Awan, A. (2017). Classification Techniques in Machine Learning: Applications and Issues. *Journal of Basic & Applied Sciences*, *13*, 459–465.

Surya, D. P. (2023). MENGINTEGRASIKAN TEKNIK NLP DAN MACHINE LEARNING UNTUK MENINGKATKAN ANALISIS SENTIMEN. *Jurnal Teknologi Pintar*, *3*(11).

Syahril Dwi Prasetyo, Shofa Shofiah Hilabi, & Fitri Nurapriani. (2023). Analisis Sentimen Relokasi Ibukota Nusantara Menggunakan Algoritma Naïve Bayes dan KNN. *Jurnal KomtekInfo*, *10*, 1–7. https://doi.org/10.35134/komtekinfo.v10i1.330

Widyasanti, N. K., Darma Putra, I. K. G., & Dwi Rusjayanthi, N. K. (2018). Seleksi Fitur Bobot Kata dengan Metode TFIDF untuk Ringkasan Bahasa Indonesia. *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, *6*(2), 119. https://doi.org/10.24843/jim.2018.v06.i02.p06

Yodi Prayoga, A., Id Hadiana, A., & Rakhmat Umbara, F. (2021). Deteksi Hoax pada Berita Online Bahasa Inggris Menggunakan Bernoulli Naïve Bayes dengan Ekstraksi Fitur Tf-Idf. *Jurnal Syntax Admiration*, *2*(10), 1808–1823. https://doi.org/10.46799/jsa.v2i10.327

Yuyun, Nurul Hidayah, & Supriadi Sahibu. (2021). Algoritma Multinomial Naïve Bayes Untuk Klasifikasi Sentimen Pemerintah Terhadap Penanganan Covid-19 Menggunakan Data Twitter. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(4), 820–826. https://doi.org/10.29207/resti.v5i4.3146