ANALISA PERBANDINGAN ALGORITMA MULTINOMIAL NAÏVE BAYES DAN ADABOOST DALAM MENGKLASIFIKASIKAN SENTIMEN MASYARAKAT TERKAIT PINJAMAN ONLINE

SKRIPSI

DISUSUN OLEH

YOGA PANGESTU

2009010068

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2024

ANALISA PERBANDINGAN ALGORITMA MULTINOMIAL NAÏVE BAYES DAN ADABOOST DALAM MENGKLASIFIKASIKAN SENTIMEN MASYARAKAT TERKAIT PINJAMAN ONLINE

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) dalam Program Studi Sistem Informasi pada Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara

YOGA PANGESTU

NPM. 2009010068

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2024

LEMBAR PENGESAHAN

Judul Skripsi	:	Analisa Perbandingan Algoritma Multinomial Naïve Bayes dan Adaboost Dalam Mengklasifikasikan Sentiment Masyarakat Terkait Pinjaman Online.
Nama Mahasiswa	:	YOGA PANGESTU
NPM	:	2009010068
Program Studi	:	SISTEM INFORMASI

Menyetujui
Komisi Pembimbing

(Mhd. Basri, S.Si, M.Kom)
NIDN. 0111078802

Ketua Program Studi	Dekan
(Martiano S.Pd, S.Kom., M.Kom)	**(Dr. Al-Khowarizmi, S.Kom., M.Kom.)**
NIDN. 0128029302	**NIDN. 0127099201**

PERNYATAAN ORISINALITAS

ANALISA PERBANDINGAN ALGORITMA MULTINOMIAL NAÏVE BAYES DAN ADABOOST DALAM MENGKLASIFIKASIKAN SENTIMEN MASYARAKAT TERKAIT PINJAMAN ONLINE

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 11 Juni 2024

Yang membuat pernyataan

Materai
10000

Yoga Pangestu

NPM. 2009010068

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama	: Yoga Pangestu
NPM	: 2009010068
Program Studi	: Sistem Informasi
Karya Ilmiah	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bebas Royalti Non-Eksekutif *(Non-Exclusive Royalty free Right)* atas penelitian skripsi saya yang berjudul:

"ANALISA PERBANDINGAN ALGORITMA MULTINOMIAL NAÏVE BAYES DAN ADABOOST DALAM MENGKLASIFIKASIKAN SENTIMEN MASYARAKAT TERKAIT PINJAMAN ONLINE"

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 11 Juni 2024

Yang membuat pernyataan

Yoga Pangestu

NPM. 2009010068

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Yoga Pangestu
Tempat dan Tanggal Lahir	: Pagar Bosi, 21 November 2001
Alamat Rumah	: Huta II Bandar Selamat Nagori Pagar Bosi
Telepon/Faks/HP	: 082277546435
E-mail	: yogapangestuuu23@gmail.com
Instansi Tempat Kerja	: -
Alamat Kantor	: -

DATA PENDIDIKAN

SD	: MIS Nurul Hikmah Afd. VII Tinjowan	TAMAT: 2013
SMP	: MTS Nurul Hikmah Aek Ger-Ger	TAMAT: 2016
SMA	: SMA Negeri 1 Ujung Padang	TAMAT: 2019

KATA PENGANTAR

Assalamu'alaikum Warahmatullaah Wabarakaatuh

Alhamdulillah hirobbil alamin puji syukur penulis panjatkan kepada Allah SWT, karena atas nikmat dan rahmat-Nya lah penulis dapat menyelesaikan penelitian skripsi ini yang berjudul **"Analisa Perbandingan Algoritma Multinomial Naïve Bayes dan Adaboost Dalam Mengklasifikasikan Sentimen Masyarakat Terkait Pinjaman Online"** sebagai syarat untuk mendapat gelar Sarjana Komputer Prodi Sistem Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara.

Selama penyusunan skripsi ini tentunya penulis mendapatkan banyak bantuan saran, motivasi, dorongan hingga bimbingan dari berbagai pihak. Maka dalam kesempatan ini dengan segala hormat dan kerendahan hati penulis tentunya berterima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Bapak Martiano, S.Pd., M.Kom sebagai Ketua Program Studi Sistem Informasi Fakultas Ilmu Komputer dan Teknologi Informasi.
4. Ibu Yosida Sari, S.Kom., M.Kom sebagai Sekretaris Program Studi Sistem Informasi Fakultas Ilmu Komputer dan Teknologi Informasi.
5. Pembimbing Bapak Mhd. Basri, M.Kom yang telah membantu dan membimbing sehingga penulis mampu menyelesaikan skripsi ini dengan baik.
6. Terutama dan yang paling utama kedua orang tua, Bapak Bambang Hermanto dan Ibu Parwati yang selalu memberikan doa dukungan dan semangat dalam segala hal, materil maupun moril yang tanpa adanya mereka penulis mungkin tidak bisa apa-apa hingga saat ini.

7. Keluarga dan adik-adik yang selalu memberikan dukungan dan doa hingga saat ini.
8. Organisasi IMM FIKTI UMSU tempat dimana penulis banyak belajar serta menemukan hal baru dan telah penulis anggap sebagai rumah kedua selama masa kuliah. Terimakasih atas kesempatan yang diberikan kepada penulis sebagai Ketua Umum PA 2022-2023 semoga apa yang penulis kerjakan dan lakukan berarti untuk kedepannya.
9. Kepada partner dalam berorganisasi, Nurul Sastia Diningsih dan Nur Aini, selaku sekretaris dan bendahara yang senantiasa menemani perjalanan organisasi hingga selesai. Semoga hubungan kekeluargaan kita akan terjalin selamanya.
10. Medan Book Party sebagai komunitas tempat penulis menemukan hal-hal baru serta cukup menjadi tempat penulis untuk refreshing dikala jenuhnya proses penulisan skripsi.
11. Terkhusus kepada diri sendiri, terimakasih sudah berjalan sejauh ini kamu hebat dan akan selalu menjadi yang terhebat.
12. Semua pihak yang terlibat langsung maupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini.

"Emas akan tetap menjadi emas, sekalipun dibuang ke tong sampah"

ANALISA PERBANDINGAN ALGORITMA MULTINOMIAL NAÏVE BAYES DAN ADABOOST DALAM MENGKLASIFIKASIKAN SENTIMEN MASYARAKAT TERKAIT PINJAMAN ONLINE

ABSTRAK

Penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja algoritma Multinomial Naive Bayes (MNB) dan AdaBoost dalam mengklasifikasikan sentimen masyarakat terkait pinjaman online. Data yang digunakan dalam penelitian ini adalah komentar pengguna di media sosial twitter terkait pinjaman online yang diambil dengan metode scraping pada periode waktu 01-12-2023 hingga 31-01-2024. Metode yang digunakan untuk mengolah data adalah SEMMA (Sample, explore, modify, model, acces), tahapan tersebut mencakup tahap preprocessing data pemodelan dan evaluasi. Sentiment diklasifikasikan kedalam kelas positif, negatif dan netral dengan menggunakan kamus lexicon base bahasa indonesia. Model yang dibangun menggunakan Algoritma Multinomilan Naïve Bayes dan AdaBoost untuk dibandingkan performanya dengan mengukur nilai akurasi, presisi, recall dan f1 score pada masing-masing algoritma. Hasil evaluasi model menunjukan menunjukkan bahwa algoritma AdaBoost memiliki kinerja yang lebih baik dibandingkan dengan algoritma Multinomial Naïve Bayes dalam mengklasifikasikan sentimen masyarakat terkait pinjaman online. Hal ini dibuktikan dengan nilai akurasi dari algoritma AdaBoost sebesar 76%, sedangkan akurasi dari algoritma MNB sebesar 71%.

Kata Kunci: *Analisis sentimen, Pinjaman online, Multinomial Naive Bayes, AdaBoost, klasifikasi, Data Mining.*

COMPARATIVE ANALYSIS OF MULTINOMIAL NAÏVE BAYES AND ADABOOST ALGORITHMS IN CLASSIFYING SENTIMENT OF COMMUNITY RELATED TO ONLINE LOANS

ABSTRACT

This study aims to analyze and compare the performance of the Multinomial Naive Bayes (MNB) and AdaBoost algorithms in classifying public sentiment related to online loans. The data used in this study are user comments on Twitter social media related to online loans taken using the scraping method in the period 01-12-2023 to 31-01-2024. The method used to process the data is SEMMA (Sample, explore, modify, model, access), these stages include the data preprocessing, modeling and evaluation stages. Sentiment is classified into positive, negative and neutral classes using the Indonesian lexicon base dictionary. The model built using the Multinomial Naïve Bayes and AdaBoost Algorithms to compare its performance by measuring the accuracy, precision, recall and f1 score values of each algorithm. The results of the model evaluation show that the AdaBoost algorithm has better performance than the Multinomial Naïve Bayes algorithm in classifying public sentiment related to online loans. This is proven by the accuracy value of the AdaBoost algorithm of 76%, while the accuracy of the MNB algorithm is 71%.

Keywords: *Sentiment analysis, online loans, Multinomial Naive Bayes, AdaBoost, classification.*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Sekarang ini manusia tidak bisa lepas dari perkembangan teknologi dalam setiap aktivitasnya. Perkembangan tersebut tentu dipengaruhi dengan kecepatan mengakses informasi dan komunikasi. Hal tersebut juga didukung dengan keberadaan media sosial yang mewadahi manusia untuk mengakses informasi secara instan. Kemudahan dalam mengakses informasi berdampak pada sifat manusia yang lebih konsumtif dalam memenuhi kebutuhanya. Keberadaan media sosial secara tidak langsung juga telah mempengaruhi gaya hidup manusia, dari apa yang dilihat di media sosial membuat manusia menginginkan sesuatu yang lebih dari apa yang dibutuhkan dan tidak pernah puas akan hal tersebut (*Hedonisme*). Sehingga perkembangan inovasi di bidang ekonomi keuangan menjadi salah satu isu yang terus dikembangkan. Salah satu yang cukup populer akhir-akhir ini yaitu pinjaman online.

Pinjaman online merupakan salah satu bentuk perkembangan teknologi dari dari sektor ekonomi yang mana konsep peminjaman uang kini bisa melalui online untuk memudahkan transaksi. Pinjaman online telah menjadi salah satu solusi finansial yang populer bagi masyarakat diera digital saat ini. Pinjaman online menawarkan kemudahan dan kecepatan dalam mendapatkan pinjaman. Kemudahan dalam segi persyaratan dan proses pengajuan untuk mendapatkan pinjaman menjadi salah satu nilai tambah bagi masyarakat untuk mengambil pinjaman.

Meskipun memberikan akses cepat dan mudah untuk mendapatkan dana, masih banyak penyedia layanan pinjaman online yang belum memiliki regulasi yang memadai sehingga potensi risiko bagi para peminjam menjadi lebih tinggi. Fenomena ini juga menimbulkan perhatian atas beragam isu terkait, termasuk kesulitan pembayaran, suku bunga yang tinggi, dan praktik peminjaman yang tidak bertanggung jawab. Hal ini diperparah dengan maraknya kasus penipuan dan tindakan yang tidak etis oleh beberapa pemberi pinjaman online. Oleh karena itu timbulah berbagai macam persepsi masyarakat terhadap penggunaan pinjaman online.

Adanya berbagai macam persepsi atau opini masyarakat yang beredar luas tentu akan memberikan gambaran atau pandangan terhadap fenomena pinjaman online. Oleh karenanya untuk mengetahui apakah keberadaan pinjaman online berdampak baik atau buruk bagi masyarakat perlu adanya analisis sentiment. Analisa sentimen digunakan untuk mengidentifikasi pola dari opini dan persepsi yang beredar sehingga diketahui fenomena tersebut berdampak positif atau negatif bagi masyarakat.

Maraknya stigma masyarakat terkait keberadaan pinjaman online bisa dilihat melalui media massa, berita, website dan media sosial. Salah satu media sosial yang sering digunakan masyarakat untuk menyampaikan keluhan dan tanggapan adalah *twitter. Twitter* adalah sebuah situs jejaring media sosial yang sedang berkembang pesat saat ini yang mana pengguna dapat berinteraksi dengan pengguna lainnya secara bebas melalui penyampaian opini atau keluhan dan ditanggapi. Penggunaan dari media sosial yang mudah karena dapat diakses melalui perangkat mobile atau komputer dari manapun dan kapanpun. Menurut laporan We

Are Social, ada 564,1 juta pengguna *twitter* di seluruh dunia per Juli 2023. Sehinggan *twitter* sangat cocok untuk melihat sentimen masyarakat terkait suatu isu yang ada.

Untuk menganalisa sentimen tersebut digunakan algoritma *Multinomial Naïve Bayes* dan *Adaboost.* Kedua algoritma tersebut nantinya akan dibandingkan performanya dengan melihat nilai akurasi, *precision, recall* dan *f1-score* dari masing-masing model algoritma. Algoritma *Multinomial Naïve Bayes Classifier* adalah modifikasi dari algoritma *Naïve Bayes Classifier* yang terbukti efektif dalam klasifikasi sentimen dan memiliki kinerja lebih baik dari algoritma klasifikasi lainnya (Sentia, 2023). Pada penelitian yang dilakukan oleh Zaki Hariansyah (2022), menghasilkan skor akurasi sebesar 93%, *precision* 90%, *recall* 93%, dan *f1-score 91*% dengan rata-rata skor sebesar 92%. Hal tersebut diperkuat dengan penelitian yang dilakukan oleh Zelin Gaa Ngilo & Nuryuliani Nuryuliani (2023), dengan menggunakan algoritma *Multinomial Naïve Bayes* diperoleh tingkat akurasi pelatihan sebesar 91.50% dan tingkat akurasi validasi sebesar 85.35% dengan menggunakan 2211 data uji dan 553 data validasi yang bersumber dari *twitter*. Dari hasil penelitian terdahulu yang dipaparkan cukup untuk menunjukan bahwa algoritma *Multinomial Naïve Bayes* diklaim baik dalam melakukan klasifikasi.

Adaboost adalah singkatan dari *"Adaptive Boosting"* merupakan algoritma yang cukup populer untuk melakukan klasifikasi. Algoritma *Adaboost* memiliki performa yang lebih baik jika dibandingkan dengan algoritma *Boosting* lainnya seperti *Gradien Boosting* dan *XGBoost* (Latief et al., 2021). Dari penelitian yang dilakukan (Syah et al., 2023) dengan menggunakan algoritma *AdaBoost* didapat nilai *Accuracy* 80.21, *Precision 85*.01, *Recall* 73.36 dan AUC 0.861. Penelitian

dengan menggunakan algoritma Adaboost juga dilakukan oleh (Wahyu et al., 2023) dengan 560 data tweet yang dibagi menjadi 500 untuk data latih dan 60 untuk data uji, menghasilkan bahwa algoritma AdaBoost memiliki kemampuan untuk mengkategorikan *tweet* kenaikan BBM Pertamina ke dalam kelas bersentimen positif atau negatif dengan akurasi sebesar 86,8%.

Atas dasar maraknya stigma masyarakat terkait isu pinjaman online maka dilakukan penelitian dengan cara melakukan analisa sentimen. Analisis sentimen atas kasus pinjaman online dapat menjadi kunci untuk memahami persepsi, pendapat, serta perasaan yang berkembang di kalangan masyarakat terkait masalah ini. Dari fenomena tersebut maka penulis mengangkat topik penelitian yang berjudul "ANALISA PERBANDINGAN ALGORITMA MULTINOMIAL NAIVE BAYES DAN ADABOOST DALAM MENGKLASIFIKASIKAN SENTIMEN MASYARAKAT TERKAIT PINJAMAN ONLINE" Dalam konteks ini, penelitian yang berfokus pada analisis sentimen terhadap kasus pinjaman online memiliki potensi untuk memberikan wawasan yang berharga bagi pemangku kepentingan terkait, mulai dari regulator, penyedia layanan, hingga konsumen, untuk meningkatkan pemahaman, pengelolaan risiko, serta pelayanan yang lebih baik di industri pinjaman online.

1.2. Rumusan Masalah

Berdasarkan uraian dari latar belakang masalah, maka rumusan masalah yang menjadi fokus pada penelitian ini adalah bagaimana perbandingan performa algoritma *Multinomial Naive Bayes* dan *Adaboost* dalam mengklasifikasikan

sentiment masyarakat terkait pinjaman online dengan mengukur nilai akurasi, presisi, *recall* dan *f1 score* pada masing-masing algoritma.

1.3. Batasan Masalah

Adapun batasan masalah dalam melakukan penelitian ini adalah:

1. Data yang digunakan bersumber dari *twitter* (X).
2. Analisa dilakukan kepada *tweet* yang berkaitan dengan isu pinjaman online dan berbahasa Indonesia.
3. Hasil analisis berupa klasifikasi positif, negatif dan netral.

1.4. Tujuan Penelitian

Dari rumusan masalah yang telah diberikan maka tujuan dari penelitian ini adalah:

1. Untuk mengetahui apakah keberadaan pinjaman online berdampak baik atau buruk bagi masyarakat.
2. Melihat dan menganalisa perbandingan performa dari algoritma *Multinomial Naive Bayes* dan *Adaboost* dalam melakukan klasifikasi data sentimen berbasis *text mining.*

1.5. Manfaat Penelitian

Manfaat dari hasil penelitian ini dibagi menjadi dua, yaitu berdasarkan manfaat secara teoritis dan manfaat secara praktis.

1. Manfaat teoritis :

a. Dapat mengetahui tingkat performa dari algoritma *Multinomial Naive Bayes* dan *Adaboost* dalam melakukan klasifikasi data pada sentimen masyarakat terkait isu pinjaman online.
b. Sebagai acuan, rujukan atau bahan referensi dan pengembangan pada penelitian-penelitian selanjutnya yang berhubungan dengan penelitian sentimen analisis.

2. Manfaat praktis:

a. Bagi Masyarakat
   - Dapat mengetahui dampak dari keberadaan instansi pinjaman online, apakah baik atau buruk bagi masyarakat.

b. Bagi Instansi
   - Dapat menjadi bahan pertimbangan bagi instansi dengan mengetahui sentimen publik mengenai isu pinjaman online.
   - Memberikan wawasan yang berharga bagi pemangku kepentingan untuk meningkatkan pemahaman, pengelolaan risiko, serta pelayanan yang lebih baik di industri pinjaman online.

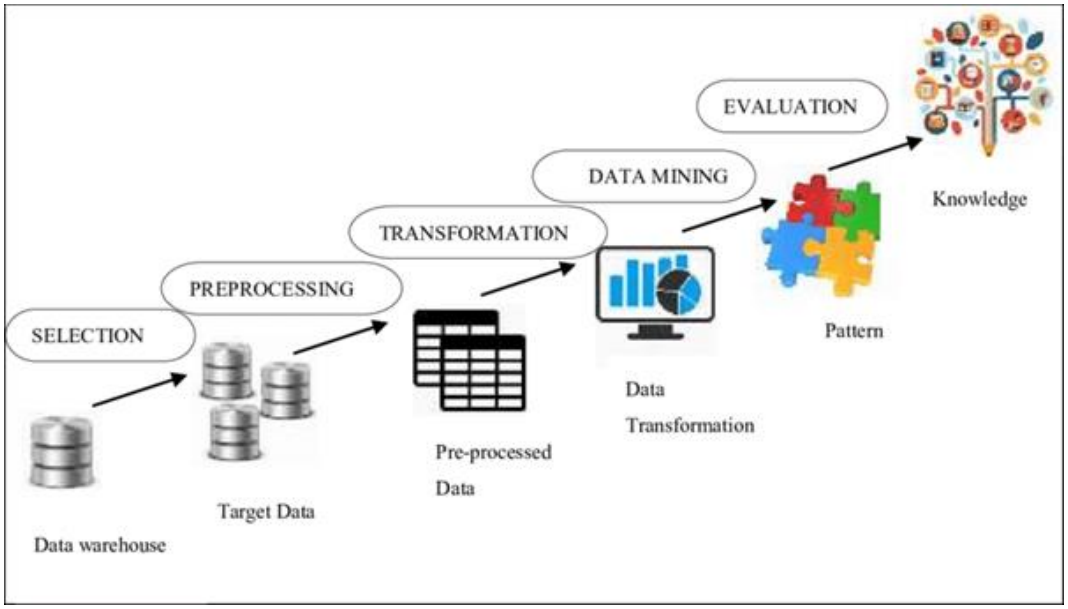
BAB II

LANDASAN TEORI

2.1. Data Mining

Data mining adalah proses pengolahan data untuk mendapatkan informasi yang berguna dari basis data yang besar dan perlu diekstraksi agar menjadi informasi baru sehingga membantu dalam pengambilan keputusan. *Data mining* adalah proses menganalisis data dari berbagai sumber dan menggabungkannya untuk menciptakan informasi, pengetahuan, atau pola yang penting untuk meningkatkan keuntungan, mengurangi biaya, atau keduanya. (Suntoro, 2019). Sedangkan menurut Menurut Werdiningsih (2020:17), *Data mining* merupakan disiplin ilmu yang digunakan untuk menangani proses pengambilan informasi dari database dengan menggunakan teknik dari statistik, pembelajaran mesin, visualisasi data, pengenalan pola, dan database.

Data mining erat kaitannya dengan KDD (*Knowledge Discovery In Database*), yaitu proses tahapan mencari dan meneliti sejumlah besar himpunan atau data yang dibantu oleh kinerja komputer untuk mengekstrak informasi dan pengetahuan yang berguna. Adanya *data mining* bertujuan untuk memberikan wawasan dari data yang dikelolah, memberikan informasi yang relevan dan menguji kebenaran gagasan atau hipotesis. Sehingga akan sangat bermanfaat dalam proses pengambilan keputusan. Dalam proses kerjanya, *data mining* digambarkan sebagai berikut:



Gambar 2. 1 Proses *Data Mining*

Proses data mining diawali dengan tahapan *selection* atau pemilihan data. Tahapan ini juga meliputi pencarian data dari berbagai sumber serta penggabungan data yang telah diperoleh (*data integration*). Tahap selanjutnya adalah *preprocessing* atau tahapan persiapan data, pada tahap ini dilakukan tindakan *data cleaning* atau pembersihan data dari kesalahan, inkonsistensi, dan ketidaksesuaian. Proses ini mencakup identifikasi dan perbaikan nilai-nilai yang hilang, duplikat, atau tidak valid, serta penanganan *outlier* dan data yang tidak lengkap.

Pada tahapan data *transformation* data dilakukan pemilihan fitur dan target data dengan cara *filtering* data yang tidak diperlukan. Selanjutnya proses *data mining*, proses dimana dilakukan berbagai teknik dan metode untuk mengetahui pola-pola potensial sehingga menghasilkan data yang berguna. Setelah tahapan *data mining* perlu adanya tahapan evaluasi, evaluasi digunakan untuk mengukur ketepatan atau kesesuaian pada model yang sudah dibangun sehingga diketahui seberapa optimal model dalam memecahkan masalah.

2.1.1. Metode Data Mining

Dalam pembelajaran *data mining* terdapat beberapa metode yang digunakan untuk mengolah data, adapun beberapa metode yang digunakan *data mining* adalah sebagai berikut:

1. Prediksi

   Proses menemukan pola dari data dengan menggunakan beberapa variabel acuan untuk memprediksikan variabel lain yang tidak diketahui jenis atau nilainya. Teknik ini juga digunakan untuk menemukan nilai atau memperkirakan peristiwa dimasa depan dengan data dari masa lalu sehingga biasanya datanya bersifat time series.

2. Klasifikasi

   Klasifikasi merupakan suatu proses untuk menemukan model atau fungsi yang menggambarkan class atau hubungan dari suatu data. Klasifikasi juga diartikan proses pemberian label baru pada tiap kelompok data yang sudah teridentifikasi dari data yang sebelumnya. Selain itu juga klasifikasi bertujuan membedakan kelas atau kelompok data untuk mengidentifikasi pola dari objek yang data tidak diketahui.

3. *Clustering*

   Adalah proses mengidentifikasi tiap titik data untuk selanjutnya dikelompokan berdasarkan karakteristik dari data tersebut. Data yang dikelompokan dalam satu *cluster* memiliki karakteristik yang sama.

4. Asosiasi

   Asosiasi adalah Proses ini digunakan untuk menemukan suatu hubungan dari variabel yang terdapat pada nilai atribut dari sekumpulan data.

Asosiasi digunakan untuk menemukan korelasi serta menjelaskan pola data. Asosiasi dalam *data mining*, juga dikenal sebagai analisis keranjang belanja atau market basket analysis, adalah teknik penelitian yang bertujuan untuk menemukan hubungan atau korelasi antara elemen dalam database.

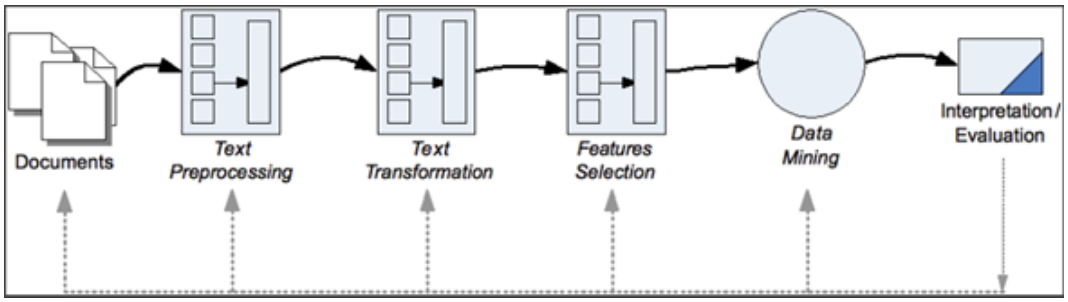
2.2. Analisa Sentimen

Analisis sentiment adalah bidang *data mining* yang biasa digunakan untuk menganalisis data teks berupa opini dengan cara menemukan polarisasi dari data tersebut dan kemudian menghasilkan informasi yang bernilai positif, negatif, atau netral. Metode analisis sentimen merupakan suatu metode untuk menganalisis informasi yang diperoleh dari internet dengan sedemikian rupa sehingga memungkinkan untuk mengetahui polaritas informasi tersebut. Analisis sentimen dapat digunakan untuk mengumpulkan polaritas opini yang ada untuk digunakan dalam memprediksi sentimen publik. (Que et al., 2020).

Sedangkan menurut Ghozali & Sugiharto (2022) Analisis sentimen adalah proses menentukan polaritas teks dokumen atau kalimat dan mengelompokkan sentimen sehingga dapat ditentukan kategori sentimen positif, negatif, atau netral. Jajak pendapat juga dapat dibandingkan dengan penelitian opini karena berfokus pada opini positif atau negatif. (Laurensz & Sediyono, 2021). Analisis sentimen melibatkan penambangan data, yang menganalisis, memproses, dan mengekstrak data tekstual dari entitas seperti layanan, produk, manusia, fenomena, dan topik. Proses analisis dapat mencakup teks ulasan, forum, *tweet*, blog, dll. *Text mining* adalah proses menganalisis data dalam bentuk teks.

2.3. Text Mining

Text mining adalah metode yang digunakan untuk menganalisis data tidak terstruktur dalam format teks. Ada dua fase utama dalam analisis *text mining*. Salah satunya adalah *preprocessing* atau persiapan data dan integrasi data yang tidak terstruktur, proses tersebut dilakukan untuk analisis statistik dari data yang telah diproses sebelumnya untuk mengekstraksi konten atau informasi dari teks. Dengan kata lain *text mining* mengubah data tak terstruktur menjadi data terstruktur (Zaki Hariansyah, 2022). *Text mining* mengacu pada ekstraksi informasi dan pola yang implisit, sebelumnya tidak diketahui, dan berpotensi berharga secara otomatis atau semi-otomatis dari data tekstual tidak terstruktur yang sangat besar, seperti teks bahasa alami (Hassani et al., 2020). Berikut ini adalah gambaran proses perjalanan text mining (Firdaus & Firdaus, 2021)



Gambar 2. 2 Alur Proses Text Mining

Alur proses data mining diawali dengan pengumpulan dokumen atau data yang nantinya akan dikelolah. Data berupa text akan melewati tahap preprocessing sebelum ke tahap pemodelan, pada bagian ini dilakukan persiapan data yaitu pengubahan data mentah menjadi format data yang lebih mudah dipahami. Pengubahan struktur data dan pemilihan fitur juga dilakukan pada alur tahapan text mining sebelum masuk tahapan pemodelan data.

Data mining dan *text mining* berbeda berdasarkan jenis data yang ditanganinya. *Data mining* menangani data terstruktur yang berasal dari sistem, seperti *database*, *text mining* menangani data tidak terstruktur yang ditemukan di dokumen, email, media sosial, dan web. Dengan demikian, perbedaan antara *data mining* dan *text mining* adalah bahwa dalam *text mining* polanya diekstraksi dari teks bahasa alami bukan dari database yang terstruktur (Hassani et al., 2020).

2.3.1. Text Klasifikasi

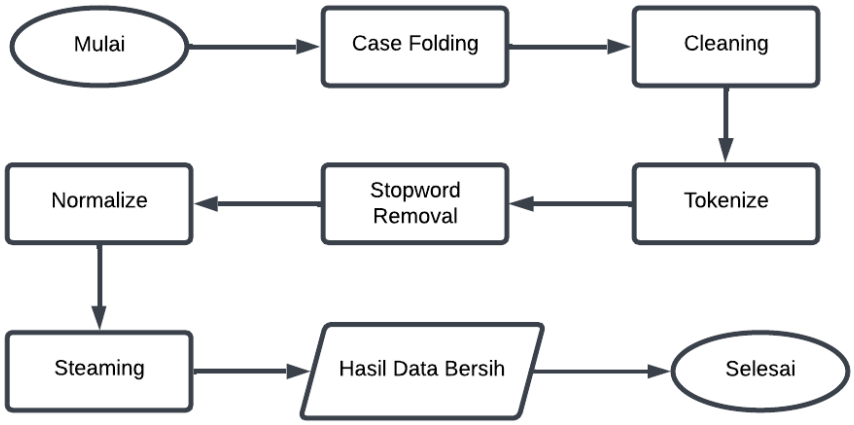
Di Dalam *text mining*, sistem klasifikasi teks merupakan proses pengelompokan text berdasarkan acuan tertentu. Sistem klasifikasi teks adalah sebuah proses penentuan untuk pemisahan dan pengelompokan teks secara langsung atau otomatis, sesuai dengan acuan klasifikasi konten teks yang diberikan sistem. Klasifikasi teks adalah teknik pembelajaran mesin yang menetapkan serangkaian kategori yang telah ditentukan sebelumnya ke data teks. Klasifikasi teks dapat menganalisis teks secara otomatis dan kemudian menetapkan serangkaian tag atau kategori yang telah ditentukan sebelumnya berdasarkan konteksnya.

Teks klasifikasi digunakan untuk mengelompokkan atau mengkategorikan informasi ke dalam kelas-kelas atau kategori-kategori tertentu berdasarkan ciri-ciri atau karakteristik yang dimiliki. Teks klasifikasi nantinya akan mengelompokan kata atau kalimat kedalam kelas positif atau negatif dari sentiment yang dianalisa.

2.3.2. Text Preprocessing

Text processing adalah metode yang digunakan untuk mengelolah suatu text yang tidak terstruktur menjadi lebih terstruktur dengan menemukan

pola bertujuan untuk teks tersebut dapat diolah kembali untuk mendapatkan informasi dan pola teks pada suatu dokumen tertentu (Ashari et al., 2020). *Text processing* adalah tahapan dalam text mining yang mana proses didalamnya dilakukan untuk persiapan data text. Proses ini dibagi menjadi beberapa tahap supaya kumpulan data yang telah diperoleh dapat digunakan untuk model yang dibuat. Tahapan dari text preprocessing adalah:



Gambar 2. 3 Alur Proses Text Preprocessing

1. *Case Folding*

   Case Folding adalah proses pengubahan semua kata yang ada pada data text menjadi huruf kecil.

2. *Cleaning*

   Cleaning adalah pembersihan data text atau dokumen dari kata-kata yang tidak diperlukan seperti adanya emoticon, url, link dan hastag yang ada pada data text.

3. *Tokenize*

   Tokenizing adalah tahap untuk memisahkan atau memecah teks atau kalimat menjadi bagian–bagian kata yang disebut token.

4. Normalisasi

   Merupakan proses untuk mengubah teks pada dari sebuat kata-kata yang tidak tepat atau singkatan bahasa menjadi kata atau kalimat yang memiliki arti.

5. *Stopword Removal*

   Stopwords Removal adalah tahap untuk menghapus kata-kata yang tidak penting atau tidak memiliki bobot makna dari hasil normalisasi.

6. *Steaming*

   Stemming adalah proses pengolahan teks dengan membuang imbuhan dari setiap kata dan membuat kata tersebut menjadi kata dasar.

2.4. *Lexicon Base*

Lexicon Based adalah sebuah metode yang didasari oleh kamus yang bertujuan untuk mendapatkan bobot dari suatu kalimat di dalam data set sehingga dapat diketahui sentimen dari kalimat tersebut (Hendriyadi et al., 2023). Metode *lexicon base* adalah pengklasifikasian sentiment yang didasarkan pada polaritas potongan text yang diperoleh dari polarisasi kata-kata yang menyusunya. *Lexicon base* mempunyai beberapa kelebihan salah satunya adalah metode ini dapat melakukan pelabelan secara otomatis pada suatu kalimat sehingga terjadi penghematan waktu dalam proses pengolahan dataset yang berjumlah banyak atau besar. Kemudian dengan menggunakan metode ini, sentimen didalam dataset

digunakan untuk menghindari kalimat bias dari opini pribadi seseorang (Laurensz & Sediyono, 2021).

Pada lexicon based juga dikenal istilah kamus *lexicon*, kamus menjadi acuan dalam menentukan nilai sentiment dari tiap text. Menggunakan kamus untuk menentukan nilat dari kata-kata sentiment adalah metode yang jelas dan efektif, caranya adalah dengan menggunakan beberapa kata sentimen benih sebagai referensi dan mencocokkannya berdasarkan sinonim dan struktur antonimnya dalam kamus. Secara khusus, metode ini berfungsi sebagai acuan yang berupa sekumpulan kata sentimen dengan orientasi bobot nilai positif atau negatif yang diketahui yang kemudian dikumpulkan secara manual.

2.5. Pembobotan Kata

Pembobotan kata adalah teknik untuk memberikan nilai pada kata dalam dokumen. Tujuannya adalah untuk mengetahui seberapa penting makna kata tersebut. Pembobotan kata adalah proses pengubahan kata menjadi nilai numerik yang mampu dibaca oleh sistem (Wati et al., 2023). Salah satu metode yang digunakan dalam proses pembobotan kata adalah TF-IDF atau *Term Frequency-Inverse Document Frequency.* TF-IDF adalah metode yang berfungsi dalam menghitung nilai bobot frekuensi kemunculan setiap kata yang ada pada dokumen (Ashari et al., 2020). *Term Frequency* adalah jumlah frekuensi kemunculan sebuah kata dalam suatu dokumen. *Inverse Document Frequency* adalah perhitungan untuk mengetahui seberapa banyak persebaran kata yang dimaksud dalam sebuah dokumen yang dikelolah. Frekuensi dokumen menunjukan seberapa umum kata

tersebut muncul pada sebuah dokumen, sehingga nilai bobot antara suatu kata dan dokumen menjadi tinggi bila kemunculan kata tersebut tinggi pada suatu dokumen.

TF-IDF akan berfungsi sebagai metode ekstraksi fitur dari dokumen yang sudah diproses setelah pemrosesan teks. Fungsinya adalah untuk menentukan seberapa relevan kata-kata dalam sebuah dokumen terhadap sekelompok dokumen. TF-IDF menerapkan proses ini dengan mengaitkan perhitungan bobot kata dalam dokumen atau korpus. Bobot ini adalah nilai penting dari sebuah kata dalam dokumen, dan semakin tinggi nilainya, semakin penting peran kata dalam dokumen tersebut. Berikut perhitungan TF-IDF dalam persamaan (Devita et al., 2018):

$$TF(dt) = tfd \times f(dt) \tag{2.1}$$

$$IDF\ (t) = 1 + log\frac{nd}{df(t)} \tag{2.2}$$

Keterangan :

1. tfd = Jumlah munculnya kata t pada dokumen d
2. $f(dt)$ = Jumlah dokumen yang mengandung kata t
3. nd = Jumlah seluruh dokumen
4. $df(t)$ = Jumlah dokumen yang terdapat term t

Dengan persamaan tersebut dapat digunakan persamaan berikut untuk menemukan nilai TF-IDF

$$TF - IDF = TF(dt).IDF(t) \tag{2.3}$$

Metode TF-IDF menggabungkan dua konsep yaitu frekuensi suatu kata dalam suatu dokumen dan frekuensi kebalikan dari dokumen yang mengandung kata tersebut. Saat menghitung bobot dengan TF-IDF, hitung terlebih dahulu nilai

TF per kata sehingga bobot setiap kata adalah 1. Sedangkan nilai IDF dirumuskan pada persamaan di atas.

2.6. Multinomial *Naïve Bayes*

Berdasarkan penelitian yang dilakukan oleh Sentia (2023) algoritma *Multinomial Naïve Bayes* terbukti efektif dalam klasifikasi sentimen dan memiliki kinerja lebih baik dari algoritma klasifikasi lainnya. *Multinomial Naïve Bayes* merupakan percabangan dari algoritma *Naïve Bayes* yang digunakan untuk klasifikasi. Metode Naïve Bayes terbukti dapat memberikan hasil yang cukup memuaskan ketika digunakan untuk klasifikasi teks (Sabrani et al., 2020). Metode ini mengasumsikan bahwa tidak ada ketergantungan dari setiap atribut terhadap atribut lain. Pada keadaan yang sebenarnya, asumsi bahwa setiap kata tidak bergantung satu sama lain lain berlawanan dengan pemahaman yang sebenarnya. Hal ini dikarenakan suatu dokumen atau teks perlu memiliki kata yang saling berhubungan agar dokumen tersebut memiliki makna. Akan tetapi, metode ini terbukti mampu memberikan hasil yang cukup memuaskan apabila diterapkan di bidang klasifikasi teks (Sabrani et al., 2020).

Berikut adalah persamaan algoritma *Naïve Bayes:*

$$P(y \mid x1, x2, \ldots, xn) = \frac{P(x1, x2, \ldots, xn)}{1nP(xi \mid y)} \qquad (2.4)$$

- $P(y \mid x1, x2, \ldots, xn)$ adalah probabilitas kelas y yang diberikan $x1, x2, \ldots, xn$
- $P(y)$ adalah probabilitas prior dari kelas y
- $P(xi|y)$ adalah probabilitas *likelihood* dari fitur *xi* diberikan kelas *y*

- $P(x1,x2,...,xn)$ adalah probabilitas evidence atau bukti dari fitur $x1,x2,...,xn$

Algoritma Naive Bayes adalah metode klasifikasi yang menggunakan teorema Bayes dengan asumsi independensi antar fitur. Persamaan ini menggambarkan bagaimana algoritma Naive Bayes menghitung probabilitas kelas berdasarkan fitur-fitur yang diamati. *Multinomial Naïve Bayes Classifier* adalah salah satu teknik klasifikasi teks yang paling umum digunakan dalam sentimen analisis. Teknik ini didasarkan pada asumsi bahwa setiap kata dalam sebuah dokumen independen dari kata-kata lainnya (Sentia, 2023). Berikut penjelasan mengenai langkah-langkah penyelesaian masalah algoritma *Multinomial Naïve Bayes Classifier* yaitu sebagai berikut:

1. Persiapan data testing
2. Pembuatan tabel persebaran frekuensi
3. Hitung nilai probabilitas
4. Hitung nilai probabilitas dari kemunculan kata
5. Hitung nilai akhir
6. Penarikan kesimpulan berdasarkan hasil akhir

Algoritma Naive Bayes multinomial merupakan salah satu metode khusus dari algoritma Naive Bayes sebagai metode proses text mining pada proses klasifikasi teks, dimana metode tersebut menggunakan probabilitas kelas pada dokumen. Prosesnya dimulai dengan menginput data latih yang digunakan untuk pembelajaran kemudian menghitung nilai probabilitas menggunakan persamaan:

$$P(c|d) \propto P(c)\, \Pi 1 \leq k \leq nd\, P(tk|c) \qquad (2.5)$$

Keterangan:

- P(c), adalah *prior probability* dari sebuah dokumen yang terdapat dalam kelas c. Bila *term* dari sebuah dokumen tidak memberikan petunjuk yang jelas untuk satu kelas dibandingkan dengan kelas lainnya, maka dipilih satu kelas yang memiliki *prior probability* yang tertinggi.
- <t1,t2, ..., tnd >, adalah kumpulan token dalam dokumen d yang merupakan bagian dari *vocabulary* yang digunakan untuk mengklasifikasi dan nd adalah jumlah token tersebut di dalam dokumen d. Contoh, <t1,t2, ..., tnd> untuk dokumen dengan satu kalimat "Bunga pinjaman terlalu tinggi" menjadi <bunga, pinjaman, tinggi >, dengan nd = 4, jika *term* and dan the dianggap sebagai *stop words*.

Peluang untuk memperkirakan *prior probability* P(c) atau peluang kemunculan suatu kelas pada data latih dilakukan menggunakan persamaan (Hadaina & Budiyanto, 2022).

$$P(c)=\frac{Nc}{Ndoc} \qquad (2.6)$$

Keterangan:

c : Kategori atau kelas

doc : Dokumen

Nc : Banyaknya kategori c pada dokumen latih

$Ndoc$: Banyaknya keseluruhan dokumen latih yang digunakan

Perhitungan selanjutnya dari probabilitas bahwa setiap kata termasuk dalam kategori atau kelas tertentu dapat dilakukan dengan menggunakan persamaan:

$$P(wi, c) = \frac{count(wi, c) + 1}{\Sigma w\ count(w, c) + |V|} \quad (2.7)$$

Keterangan:

wi : Kata ke-i dalam seluruh dokumen yang berkategorikan c

$(wi,)$: Jumlah kata tertentu yang muncul dalam suatu kategori atau kelas

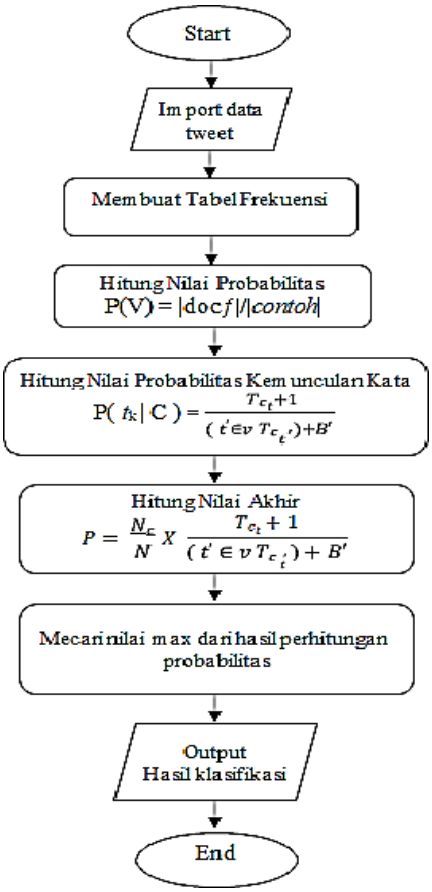
$\Sigma w\ (w,)$: Jumlah seluruh kata pada kelas

$|V|$: merupakan jumlah seluruh kata unik pada kelas

Untuk mendapatkan nilai akhir probabilitas pada dokumen data yang diuji, apakah dokumen uji tersebut termasuk dalam kelas positif atau negatif digunakan persamaan:

$$P(c) = \frac{Nc}{Ndoc} \times \frac{count(wi, c) + 1}{\Sigma w\ count(w, c) + |V|} \quad (2.8)$$

Persamaan untuk mendapatkan nilai akhir probabilitas data yang diuji adalah dengan mengalikan nilai *prior probability* atau banyaknya dokumen c pada data latih dibagi dengan seluruh dokumen atau data latih pada dokumen c dikali dengan probabilitas bahwa setiap kata termasuk dalam kategori atau kelas tertentu atau yang ada pada persamaan dua.Berikut ini adalah *Flowchart* algoritma *Multinomial Naïve Bayes Classifier* yaitu sebagai berikut:



Gambar 2. 4 *Flowchart Multinomial Naive Bayes*

2.7. Adaboost

Adaboost atau kepanjangan dari *Adaptive Boosting* merupakan percabangan dari algoritma *Boosting* diantaranya adalah *XGBoost, Gradien Boost* dan *Adaboost.* Algoritma *Adaboost (Adaptive Boosting)* adalah algoritma pembelajaran mesin yang digunakan untuk meningkatkan kinerja model prediksi dengan menggabungkan beberapa model prediksi sederhana menjadi satu model yang lebih kompleks. Algoritma ini bekerja dengan memberikan bobot yang berbeda pada setiap sampel data, sehingga sampel yang sulit diprediksi akan diberikan bobot yang lebih tinggi. Kemudian, model prediksi sederhana akan dibuat untuk setiap iterasi dengan memperhatikan bobot pada setiap sampel data. Model-model

tersebut kemudian digabungkan menjadi satu model yang lebih kompleks dengan menggunakan teknik voting (Wahyu et al., 2023). *AdaBoost* adalah metode boosting yang mampu menyeimbangkan kelas dengan cara memberikan bobot pada tingkat error klasifikasi dan kelasnya.

Algoritma *AdaBoost* menggunakan metode boosting untuk membangun klasifikasi gabungan. Pada algoritma ini, pentingnya dasar pengklasifikasian tergantung dari error rate. Pada dasarnya, Algoritma *AdaBoost* memiliki 3 tahap. Tahap pertama memulai pendistribusian bobot data pelatihan. Jika memiliki nilai sampel N, pada awalnya setiap sampel pelatihan akan diberikan bobot yang sama. Tahap kedua yaitu melatih pengelompokan dasar. Dalam proses yang spesifik, jika sebuah sampel telah diklasifikasikan dengan akurat, maka bobot akan dikurangi pada set yang akan dibangun berikutnya. Jika sampel diklasifikasikan belum akurat, maka bobot akan ditingkatkan pada set yang akan dibangun berikutnya. Set berikutnya digunakan untuk melatih pengelompokan berikutnya. Pada tahap terakhir yaitu menggabungkan *weak classifiers* yang diperoleh dari setiap pelatihan menjadi strong classifier. Setelah melatih tiap golongan yang lemah, meningkatkan bobot pada weak classifier dengan klasifikasi error rate yang kecil akan berperan lebih baik pada klasifikasi akhir. Dengan kata lain, weak classifier dengan nilai error rate yang kecil akan memiliki bobot yang besar pada pengklasifikasian akhir, begitu pula sebaliknya (Rabbani et al., 2021).

Secara garis besar proses yang dilakukan dalam *Adaboost* adalah membangun sejumlah weak learners yang tidak memiliki korelasi satu sama lain, lalu kemudian menggabungkan prediksinya (Sinaga et al., 2022). Dalam penerapannya *Adaboost*

dikombinasikan dengan algoritma lain dengan tujuan untuk mengoptimalisasi performa yang dihasilkan [5]. *Adaboost* (x) didefinisikan sebagai:

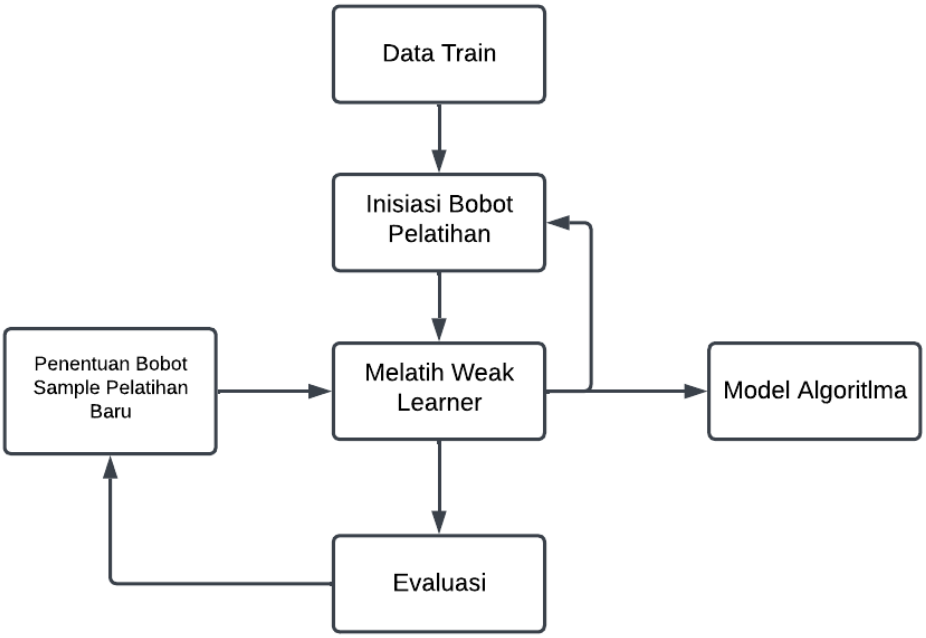
$$Hk(x) = \sum_{t=1}^{T} \left(\frac{log\ log\ 1}{\beta t}\right) ht\ (x) \tag{2.9}$$

Dimana $h(x)$ merupakan weak learners yang memiliki nilai error terendah , sedangkan βt merupakan bobot dari weak learners tersebut. Premis akhir dalam *Adaboost* dihasilkan dari kombinasi *weak learners* yang memiliki nilai suara tertinggi.

Algoritma *adaboost* dapat dideskripsikan secara high-level dari langkah dasarnya, langkah-langkahnya:

1. Inisialisasi bobot sample pelatihan
2. Melatih *weak learner*
3. Hasilkan set baru dengan pengambilan sampel dengan penggantian, sesuai dengan bobotnya.
4. Evaluasi performa *weak learner*
5. Memperbaharui bobot sample pelatihan
6. Sesuaikan bobot, tambah bobot instance pada weak learner untuk iterasi selanjutnya.
7. Ulangi langkah 2-6 sesuai jumlah iterasi yang ditentukan
8. *Weak learners* digabungkan dengan *voting. Vote pada* setiap *learner* diberi bobot sesuai dengan tingkat kesalahannya untuk menghasilkan model yang kuat.

Seluruh proses digambarkan dalam gambar berikut:



Gambar 2. 5 Langkah-Langkah Algoritma *Adaboost*

2.8. Pinjaman Online

Pinjaman online adalah layanan peminjaman uang yang disediakan oleh penyedia jasa keuangan online. Penyedia pinjaman online ini biasa dikenal dengan istilah fintech. Pinjaman Internet atau yang disebut dengan Layanan Pinjaman Uang dan Pinjaman Berbasis Teknologi Informasi (LPMUBTI) merupakan inovasi di bidang jasa keuangan yang menggunakan teknologi yang memungkinkan pemberi pinjaman dan peminjam untuk bertransaksi tanpa bertemu langsung melalui sistem pinjaman fintech, atau aplikasi yang terorganisir. atau situs web. Penyedia ini merupakan lembaga jasa keuangan yang beroperasi secara online melalui teknologi informasi. (Nihayah et al., 2023).

Dalam Pasal 1 ayat 3 Peraturan Otoritas Jasa Keuangan Nomor 77 Tahun 2016, menjelaskan bahwa layanan pinjam meminjam uang berbasis teknologi

informasi adalah penyelenggaraan layanan keuangan dengan mempertemukan pemberi pinjaman dengan penerima pinjaman secara online melalui platform dalam rangka melakukan perjanjian pinjam meminjam dalam mata uang Rupiah yang disediakan melalui sistem elektronik dan media dengan menggunakan jaringan internet. Dilansir dari situs pajak online, pinjaman online merupakan fasilitas pinjaman uang yang diselenggarakan oleh penyedia jasa layanan keuangan berbasis online.

Pendapat lain, menurut Supriyanto & Ismawati (2019). Berpendapat bahwa teknologi pengajuan pinjaman uang online merupakan model keuangan berbasis *financial technology*, yaitu solusi keuangan yang menggunakan teknologi pinjaman yang efektif dan efisien tanpa harus dibatasi ruang dan waktu, jika peralatan yang digunakan, misalnya ponsel pintar dan komputer, dapat terhubung ke Internet. Menurut kamus Tokopedia, pinjaman online adalah perjanjian pinjaman dari lembaga online. Cukup mengajukan permohonan atau melalui website, permohonan akan diproses tanpa harus pergi ke lembaga keuangan.

2.9. *Twitter*

Twitter adalah layanan jejaring sosial berbasis teks yang memungkinkan pengguna untuk mengunggah suatu informasi hingga 280 karakter yang dikenal dengan sebutan kicauan atau *tweet. Twitter* diciptakan oleh Jack Dorsey, Noah Glass, Biz Stone, dan Evan Williams pada bulan Maret 2006 dan diluncurkan pada bulan Juli tahun itu. Perusahaan induk sebelumnya, Twitter, Inc., berbasis di San Francisco, California dan memiliki lebih dari 25 kantor di seluruh dunia. Pada Oktober 2022, Elon Musk membeli kepemilikan Twitter senilai US$44 miliar atau

Rp683 triliun dan melakukan rebranding dengan mengubah logo serta nama yang semualnya *Twitter* diubah menjadi X hingga sekarang. (Wikipedia)

Sebagai media sosial berbasis teks *twitter* menjadi sumber informasi yang cepat. Di Indonesia sendiri pengguna twitter sebanyak 25,25 juta pengguna per Juli 2023. (Databoks) sebagai media yang umum digunakan oleh banyak orang tentu twitter juga digunakan untuk menuangkan aspirasi, keluhan atau opini yang berkaitan dengan suatu isu. *Twitter* sebagai salah satu jejaring sosial interaktif memungkinkan penggunanya untuk mengkritik suatu isu atau fasilitas layanan secara *real time*. *Twitter* berfungsi sebagai sumber dalam mengumpulkan pemikiran masyarakat dan seringkali *twitter* memberikan kontribusi yang sangat tinggi di banyak bidang (Sentia, 2023). Berikut adalah beberapa fitur yang ada pada *twitter:*

1. *Trending topic* adalah fitur yang menampilkan topik atau pembahasan teratas berupa *hastag* yang banyak dibicarakan pengguna *twitter*.
2. *Hashtag* adalah fitur yang dapat mengelompokkan *tweet* atau pesan.
3. *Retweet* adalah fitur untuk membagikan *tweet* dari pengguna lain.
4. *Following* adalah fitur untuk menghubungkan antar pengguna atau sering disebut teman.

Data twitter dapat diambil menggunakan aplikasi atau API yang disediakan oleh *twitter*. Jika dibandingkan dengan media sosial lainnya, tidak mudah untuk mengumpulkan data secara terbuka. Media sosial lainya tidak mengizinkan data akses karena kebijakan keamanan yang berbeda-beda. Selain itu, twitter juga

mempunyai beberapa kecocokan dengan data mining, sebagai berikut (Wandani, 2021):

1. Format data *twitter* yang cocok dan nyaman bagi peneliti untuk dianalisis
2. Peraturan *twitter* untuk data relatif fleksibel jika dibandingkan dengan API lainnya.
3. *Twitter* mempunyai desain yang *user friendly* atau mudah diakses bagi penggunanya.

2.10. *Google Colab*

Google Collaboratory atau disebut juga dengan *Collab* merupakan perangkat komputasi awan (*cloud computing*) atau tools yang dibuat oleh Google dengan tujuan untuk memudahkan penggunaan untuk pembelajaran dan pengolahan data dengan mudah seperti menggunakan tampilan berbasis *Jupyter Notebook* atau *iPython* (*interactive Python*). Google Collab menyediakan sebuah layanan platform komputasi gratis berupa *software* virtual untuk setiap penggunanya yang dilengkapi dengan kemampuan pengolahan data yang memadai. *Google Collab* merupakan produk yang dihasilkan oleh *Google Research*.

Collaboratory menyediakan *runtime Python* 2 dan 3 yang telah dikonfigurasi sebelumnya dengan *library machine learning* yang penting seperti *Scikit-learn Matplotlib* dan *Seaborn*. Dengan memanfaatkan *Google Collab*, pengguna tidak perlu melakukan instalasi atau pengaturan yang rumit untuk keperluan pengolahan data dengan menggunakan bahasa *Python*. Kelebihan lain yang ditawarkan oleh *Google Collab,* menurut Bonner (2019) adalah sebagai berikut:

1. *Library* bawaan dari *machine learning* yang lengkap
2. Data yang ditulis pada *Google Collaboratory* dapat diakses dan diedit dengan mudah
3. Mempermudah proses kolaborasi antar tim sehingga menjadi lebih fleksibel
4. Memiliki fitur GPU dan TPU yang dapat dimanfaatkan secara gratis

2.11. Penelitian Terdahulu

Berikut adalah tabel penelitian terdahulu yang mendukung kerangka teoritis pada penelitian ini.

Tabel 2. 1 Tabel Penelitian Terdahulu

No	Referensi	Object	Metode	Hasil Penelitian
1	Analisis Sentimen Terhadap Dampak Inflasi di Indonesia Menggunakan Metode Multinomial Naïve Bayes.Tri Wijaya, Aldi Hermawan, Arief (2023)	Data *twitter* tentang inflasi di Indonesia	Multinomial Naïve Bayes	Hasil klasifikasi proses penelitian diuji menggunakan beberapa skenario pembagian data yaitu 90:10, 80:20, 70:30, untuk tingkat akurasi terbaik dihasilkan menggunakan skenario 90% data train dan 10% data test mendapatkan hasil akurasi 75,5%, precision 75%, f1-score 75% dan recall 74%.
2	MULTINOMIAL NAÏVE BAYES *CLASSIFIER* UNTUK ANALISIS SENTIMEN *TWITTER*	Analisa sentiment data *twitter*	Klasifikasi menggunakan algoritma *Multinominla Naïve Bayes*	Algoritma *Multinomial Naïve Bayes Classifier* memiliki kinerja yang efektif dalam klasifikasi sentimen dan banyak digunakan karena kecepatan,

	Ayuni Sentia (2023)			kesederhanan, dan mudahnya diimplementasikan dalam klasifikasi teks.
3	Analisa Sentimen Terhadap *Twitter IndihomeCare* Menggunakan Perbandingan Algoritma *Smote, Support Vector Machine, AdaBoost* dan *Particle Swarm Optimization.* Syah, Ferdian Fajrin, Hanif Afif, Abiyyu Nur Saeputra, Muhamad Rafi Mirranty, Dinda Saputra, Dedi Dwi (2023)	Data *twitter* tentang *IndihomeCare*	Algoritma *Smote, Support Vector Machine, AdaBoost* dan *Particle Swarm Optimization*	Hasil dengan menggunakan metode SVM Accuracy 80.01, Precision 82.29, Recall 71.75 dan AUC 0.907. Sedangkan untuk metode *AdaBoost* didapat nilai Accuracy 80.21, Precision 85.01, Recall 73.36 dan AUC 0.861. Terakhir, hasil dari metode SPW Accuracy 76.59, Precision 76.57, Recall 80.35 dan AUC 0.868. Berdasarkan hasil penelitian, metode *Adaboost* mempunyai hasil terbesar dan dinilai efektif dengan dataset yang ada
4	Prediksi Tingkat Pelanggan Churn Pada Perusahaan Telekomunikasi Dengan Algoritma *Adaboost*. Latief, Iqbal Muhammad Subekti, Agus Gata, Windu (2021)	Data karakteristik pelanggan pada perusahaan telkom	*Adaboost*	Algoritma divalidasi melalui data latih dan data uji dengan perbandingan 80:20. Dari hasil yang kami dapatkan dengan menggunakan *python tools*, ditemukan bahwa algoritma *adaboost* mempunyai akurasi sebesar 80%
5	Penerapan Metode Adaptive Boosting Pada Analisis Sentimen Kenaikan BBM Pertamina Faizi &Nugroho (2023)	Data *twitter* tentang isu kenaikan BBM	Klasifikasi sentiment dengan algoritma *Adaboost*	Studi ini menghasilkan algoritma AdaBoost memiliki kemampuan untuk mengkategorikan tweet kenaikan BBM Pertamina ke dalam

				kelas bersentimen positif atau negatif dengan akurasi sebesar 86,8%.
6	Implementasi Metode Multinomial Naïve Bayes Untuk Sentiment Analysis Terhadap Data Ulasan Produk *Colearn* Pada *Google Play Store* Fadhlan Hadaina & Utomo Budiyanto (2023)	Ulasan produk *Colearn* pada *Google Play Store*	Klasifikasi dengan algoritma *Multinominal Naïve Bayes*	Hasil pengujian menghasilkan analisa pengujian akurasi bernilai 88,89% dengan 536 dataset, 439 data positif dan 97 data negatif. Hal itu menunjukan bahwa metode Multinomial Naïve Bayes dapat melakukan analisa sentiment tentang data ulasan dengan baik.
7	ANALISIS SENTIMEN OPINI PENGGUNA TWITTER PADA APLIKASI BIBIT MENGGUNAKAN MULTINOMIAL NAÏVE BAYES Zelin Gaa Ngiloa & Nuryulianib	Data *twitter* tentang opini penggunaan aplikasi bibit.	Pengujian kinerja algoritma *Multinomial Naïve Bayes* dalam mengolah data teks	Hasil pengujian menunjukkan tingkat akurasi yang diperoleh sebesar 88%.
8	KLASIFIKASI DATA DETEKSI JATUH MENGGUNAKAN MACHINE LEARNING DENGAN ALGORITMA ADAPTIVE BOOSTING (ADABOOST) Reza Rabbani, Ida Wahidah & Iman Hedi Santoso	Data gambar tentang keadaan ketika dan saat terjatuh	Klasifikasi keadaan jatuh berdasarkan gambar	Didapatkan hasil Adaboost sebagai model baik dengan nilai akurasi tertinggi di 4 rasio yaitu 97,5% pada rasio 20%:80%, 98,7% pada rasio 30%:70%, 99,3% pada rasio 40%:60% dan 100% pada rasio 50%:50%.

BAB III

METODOLOGI PENELITIAN

3.1. Pendekatan Penelitian

Penelitian ini dilakukan untuk mencari tahu hasil analisis sentiment pengguna *twitter* terkait fenomena pinjaman online yang diklasifikasikan berdasarkan sentimen positif, negatif dan netral. Sehingga dari hal tersebut dapat diketahui apakah keberadaan pinjaman online berdampak baik atau buruk bagi masyarakat. Penggunaan algoritma *Multinomial Naïve Bayes* dan *Adaboost* yang nantinya akan dibandingkan kinerjanya untuk mengetahui algoritma mana yang terbaik dalam mengklasifikasikan sentiment masyarakat terkait isu pinjaman online.

3.2. Metode Pengumpulan Data

3.2.1. Data Sekunder

Data sekunder berupa sentiment masyarakat pengguna *twitter* terkait isu pinjaman online. Data diambil dengan menggunakan metode *scraping* data memakai tools *Google Colab* dan *library pandas*. Data yang diambil berupa *tweet* dari 2023-12-01 sampai dengan 2024-01-31, dan diperoleh data sebanyak 3.260 data. Data yang diambil dalam periode waktu dua bulan, dimaksudkan untuk mendapatkan sentimen yang terbaru dari masyarakat terkait isu pinjaman online sehingga hasil klasifikasi yang didapat nantinya merupakan hasil yang relevan untuk digunakan sebagai acuan kedepan. Data tersebut masih berupa data mentah yang selanjutnya akan dilakukan tahap *preprocessing* data.

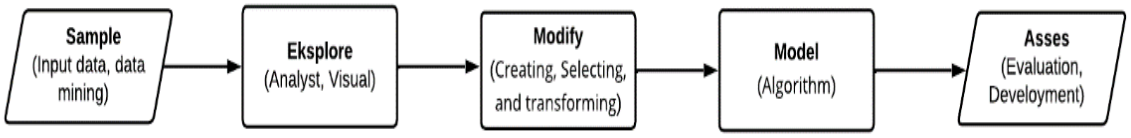
3.2.2. Studi Pustaka

Studi pustaka pada penelitian ini dilakukan untuk menunjang penelitian ini dengan cara membaca, mempelajari dan memahami literatur terkait masalah dari penelitian yang dilakukan sebelumnya. Studi pustaka juga dilakukan dengan mencari referensi seperti dari penelitian sejenis, baik berbentuk fisik maupun elektronik, jurnal artikel, skripsi maupun tesis dalam upaya untuk pemecahan masalah.

3.3. Metode SEMMA

Penelitian ini mengacu kepada metode SEMMA *Data Mining Process* untuk metodologi penelitian. Fokus dalam metode SEMMA ini adalah modifikasi, pemodelan, dan penambangan data yang dirancang untuk membantu pengguna SAS *enterprise miner*. Metode SEMMA terdiri dari *sample, explore, modify, model* dan *asses.*

Metode SEMMA berfokus pada proses modifikasi, pengumpulan data, dan pembangunan model yang dirancang untuk membantu pengguna software SAS enterprise miner (Akuntansi & Primakara, 2022). Metode SEMMA jika dibandingkan dengan metode *data mining* yang lain seperti CRISP-DM dan KDD, metode SEMMA memiliki tahapan atau proses yang lebih sederhana sehingga memberikan keluasan dan kemudahan pada penerapannya (Sitorus et al., 2020). Kelebihan dari metode SEMMA adalah tahapannya mudah dipahami dan dapat dijadikan pengembangan serta kemudahan dalam pemeliharaan proyek data mining yang terstruktur (Komputer et al., 2022). Berikut tahapan metode SEMMA (Alizah et al., 2020):



Gambar 3. 1 Alur Metode SEMMA

3.3.1. *Sample*

Pada tahap ini dilakukan sampling data dengan mengekstraksi data yang telah ditetapkan untuk menampung informasi yang signifikan. Pengumpulan data bersumber dari media sosial *twitter* dengan metode *scraping* atau *crowling* dengan memanfaatkan API *twitter* dan *library Python* untuk scraping data. *Scraping* data dilakukan dengan memasukan *query* atau kata kunci "pinjol" dengan menggunakan *tools Google Colab* data tersebut disimpan di dalam file csv.

3.3.2. *Eksplore*

Pada tahap ini data dideskripsikan dan dieksplorasi untuk tren yang tak terduga dan anomali dalam rangka untuk mendapatkan pemahaman data dan ide-ide. Pada tahapan eksplore juga dilakukan penyeleksian atribut yang tidak digunakan dalam analisa data (data *cleaning*), hal ini berguna supaya modem bekerja lebih maksimal. Pada tahap ini juga dilihat kata-kata yang sering muncul dalam data untuk melihat hasil kata yang sering muncul sebelum dan sesudah dilakukan tahapan *text preprocessing.*

3.3.3. *Modify*

Pada tahap *modify* dilakukan dengan membuat, memilih, dan mengubah variabel untuk memusatkan pemilihan model, dan informasi atau variabel tambahan untuk membuat keluaran informasi menjadi signifikan.

tahapan ini juga dilakukan persiapan data atau pre-processing seperti, *case folding, cleaning, tokenize, stopword removal, normalize* dan *stemming.* Selanjutnya melakukan visualisasi kata yang sering muncul dalam bentuk *wordcloud*.

3.3.4. *Modeling*

Pada tahap ini dilakukan pelabelan data berdasarkan kelasnya untuk menentukan opini positif, negatif dan netral menggunakan pelabelan manual yang mengacu pada kamus sentimen positif dan negatif atau yang lebih dikenal dengan metode *Lexicon Base*. Setelah itu, data set yang berupa text diubah kedalam bentuk vektor yang disebut *feature extraction* sehingga data menjadi bentuk numerik untuk selanjutnya bisa diolah oleh mesin. Kemudian data diolah berdasarkan kedua metode yaitu *Multinomial Naïve bayes* dan *Adaboost.*

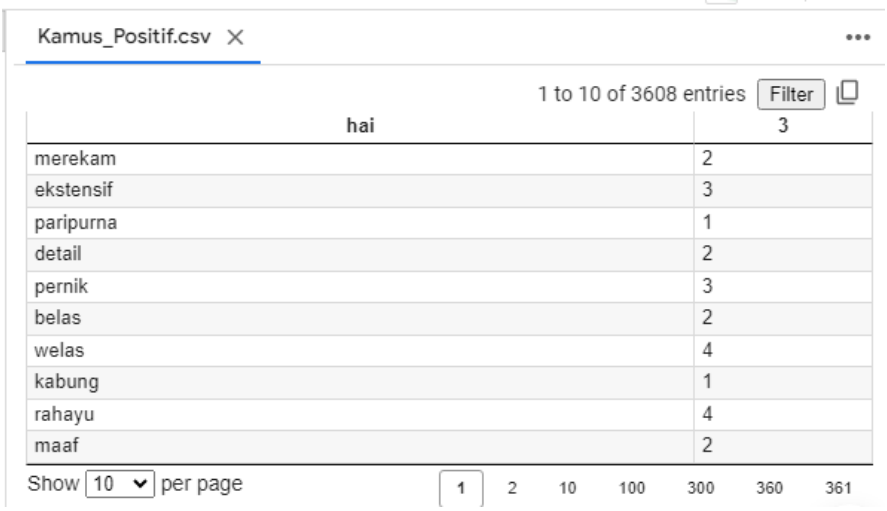
3.3.5. *Asses*

Pada tahap ini dilakukan evaluasi dari model dengan mengukur seberapa baik kinerjanya. Pada metode *Multinomial Naïve Bayes* dan *Adaboost* digunakan *cross validation* untuk mengetahui akurasi, presisi, *recall dan f1-score* dari model. Pada tahapan ini juga dilihat nilai dari *confusion matrix* pada tiap tiap kelasnya, sehingga perhitungan akurasi, presisi, *recall* dan *f1-score* dapat divalidasi dengan benar. Jika model data valid, model tersebut akan berfungsi dengan baik pada sampel yang dicadangkan dan sampel yang dibuat.

3.4. *Labeling* dengan *Lexicon Base*

Labeling data adalah tahapan memberikan label sentimen positif, negatif atau netral pada data teks. Label tersebut dipengaruhi oleh kata-kata yang ada dalam kalimat dalam teks tersebut, sementara untuk mengidentifikasi setiap kata tersebut masuk kedalam kata-kata positif atau negatif adalah dengan menggunakan metode *lexicon base. Lexicon base* bekerja dengan mengoreksi setiap kata dengan rumus kata positif dan negatif. Kamus lexicon tersebut didapat dari penelitian yang dilakukan oleh Koto & Rahmaningtyas, (2017) yang mengumpulkan kata-kata positif dan negatif berbahasa Indonesia yang ada pada blog dan memberinya bobot nilai plus dan minus. Kamus tersebut juga telah diuji dalam jurnal yang ditulis oleh Azhar, (2018) tentang identifikasi opini pada tweet berbahasa Indonesia dan hasilnya akurat.

Berikut contoh kamus yang digunakan:


Kamus_Positif.csv

1 to 10 of 3608 entries Filter

hai	3
merekam	2
ekstensif	3
paripurna	1
detail	2
pernik	3
belas	2
welas	4
kabung	1
rahayu	4
maaf	2

Show 10 per page 1 2 10 100 300 360 361

Gambar 3. 2 Kamus Kata Positif

Kamus_Negatif.csv

1 to 10 of 6608 entries Filter

putus tali gantung	-2
gelebah	-2
gobar hati	-2
tersentuh (perasaan)	-1
isak	-5
larat hati	-3
nelangsa	-3
remuk redam	-5
tidak segan	-2
gemar	-1
tak segan	-1

Show 10 per page 1 2 10 100 600 660 661

Gambar 3. 3 Kamus Kata Negatif

Orientasi sentimen kalimat ditentukan dengan menjumlahkan nilai orientasi semua kata sentimen dalam kalimat. Kata positif diberi nilai sentimen plus (+) dan kata negatif diberi nilai sentimen minus (-). Kata-kata negasi dan kata-kata yang berlawanan juga dipertimbangkan. Terdapat 3 langkah dalam menentukan suatu orientasi sentimen berdasarkan pendekatan lexicon, yaitu:

1. Identifikasi kata yang mengandung sentimen : Untuk setiap kalimat yang memiliki lebih dari satu sentiment caranya dengan memilih semua kata atau frasa yang ada pada kalimat tersebut. Setiap kata positif diberi skor sentimen +1 dan setiap kata negatif diberi skor sentimen -1. Contoh "Barang ini kualitasnya kurang bagus [+1], tapi dayanya tahan lama [+1]". Dari contoh ini kata bagus bernilai +1 dan tahan lama juga bernilai +1 karena merupakan kata yang positif.
2. Pengubah sentimen (sentimen shifter) adalah kata dan frasa yang dapat mengubah arah sentimen. Ada beberapa jenis pengonversi kata negatif (shifter negasi), seperti tidak, tidak pernah, dan tidak ada yang merupakan jenis yang paling umum. Dengan demikian, kalimatnya menjadi “Kualitas

dari barang ini tidak bagus [-1], tetapi kekuatannya tahan lama [+1]" karena kata negasinya "Tidak.".

3. Agregasi: Pada langkah ini, fungsi agregasi opini diterapkan pada skor sentimen yang dihasilkan untuk menentukan arah akhir sentimen.

Kamus lexicon pernah digunakan dalam penelitian tentang implementasi kamus lexicon untuk melakukan analisis sentimen tentang komentar masyarakat terkait daerah wisata di kota Yogyakarta, dan didapatkan hasil yang baik dalam menentukan label komentar masyarakat. (Ismail & Hakim, 2023). Karenanya metode ini dipilih pada penelitian ini untuk lebaling data.

3.5. Feature Extraction

Tahapan *feature extraction* adalah pemilihan fitur yang paling berpengaruh terhadap hasil data actual sehingga ketika saat melakukan pengolahan oleh model algoritma sistem akan memberikan hasil yang akurat dan dapat meningkatkan nilai akurasi dari model algoritma tersebut. Pada analisa data text tahapan *feature extraction* adalah dengan mengubah token setiap kata menjadi bentuk numerik untuk bisa dibaca oleh algoritma. Pada tahap ini tahapan *feature extraction* menggunakan metode TF-IDF.

Metode *feature extraction* dengan TF-IDF adalah dengan menghitung frekuensi kemunculan kata yang ada pada sebuah dokumen dan menghitung banyaknya dokumen yang memiliki kata tersebut dibagi dengan banyaknya dokumen yang ada. Nilai dari setiap kata yang dihasilkan menunjukan seberapa berpengaruh kata tersebut dalam dokumen, dalam kasus klasifikasi sentiment hal ini tentu akan memudahkan model algoritma untuk menentukan kelas sentiment

dengan mengutamakan pengolahan data pada fitur kata yang bernilai tinggi. Berikut adalah tahapan *feature extraction* dengan TF-IDF.

1. *Trem Frequency (TF)*

   TF adalah dengan menghitung term frequency kata, yaitu menghitung frekuensi kemunculan kata dalam suatu dokumen, digambarkan dengan persamaan :

   $$TF(dt) = tfd \times f(dt) \quad (3.1)$$

   Yang mana *tfd* adalah banyaknya kata di kali banyaknya dokumen yang ada atau *f(dt).*

2. *Inverse Document Frequency (IDF)*

   Tahapan ini menghitung logaritma dari dokumen yang memiliki kata tersebut dengan jumlah keseluruhan dokumen hal ini untuk mengukur seberapa umum suatu kata tertentu diluruh dokumen, digambarkan dengan persamaan:

   $$IDF\ (t) = 1 + log\frac{nd}{df(t)} \quad (3.2)$$

   Yang mana *nd* adalah dokumen yang memiliki kata tersebut dibagi dengan *df(t)* adalah jumlah seluruh dokumen.

3. *Term Frequency Inverse Document Frequency (TF-IDF)*

   $$TF - IDF = TF(dt).IDF(t) \quad (3.3)$$

   Metode TF-IDF bekerja dengan menghitung skor TF-IDF untuk setiap kata dalam sebuah dokumen. Skor TF-IDF dihitung dengan mengalikan frekuensi kemunculan kata tersebut dalam dokumen (TF) dengan logaritma terbalik dari jumlah dokumen yang berisi kata tersebut (IDF).

3.6. Waktu dan Tempat Penelitian

3.6.1. Waktu Penelitian

Tabel 3. 1 Tabel Waktu Penelitian

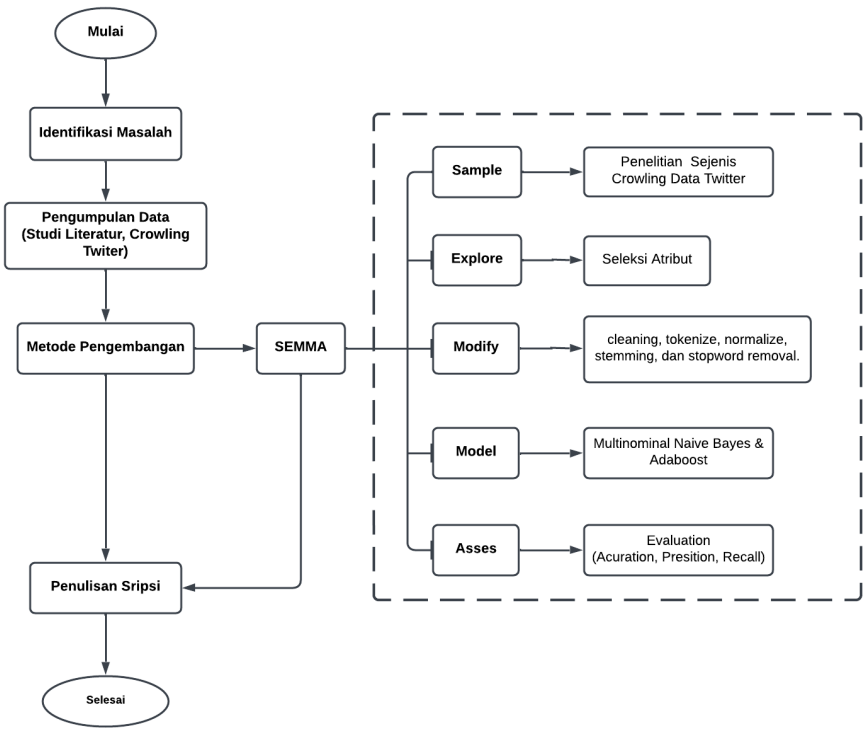
No.	Tahapan	Januari 2024	Februari 2024	Maret 2024	April 2024	Mei 2024
1.	Pengajuan Judul					
2.	Pengumpulan data					
3.	Penyusunan Proposal					
4.	Seminar Proposal					
5.	Analisis Data dan Penelitian					
6.	Penyusunan Skripsi					
7.	Sidang Meja Hijau					
8.	Penyempurnaan Skripsi dan Penulisan Artikel Jurnal					

3.6.2. Tempat Penelitian

Penelitian ini dilakukan pada sentimen yang ada pada media sosial *twitter* tentang isu pinjaman online. Pengambilan dan pengolahan data menggunakan *tools Google Colab* dengan dibantu *library Python Pandas* serta API *twitter* untuk scraping data.

3.7. Prosedur Penelitian

Prosedur penelitian merupakan langkah-langkah dan tahapan yang digunakan sebagai kerangka pemikiran untuk menyelesaikan masalah penelitian. Dalam prosedur penelitian terdapat alat-alat untuk pengumpulan dan pengolahan data, tahapan penyelesaian masalah, dan penulisan. Prosedur penelitian digambarkan sebagai berikut:



Gambar 3. 4 Prosedur Penelitian

3.8. Perangkat Penelitian

Perangkat penelitian adalah alat-alat yang digunakan untuk mengumpulkan dan mengolah data dalam sebuah kegiatan penelitian. Contoh perangkat penelitian tersebut meliputi berupa perangkat keras (*Hardware*) ataupun perangkat lunak (*Software*) yang digunakan dalam penelitian, ini dipilih sesuai dengan kebutuhan penelitian. Perangkat keras (Hardware) yang digunakan pada penelitian ini adalah satu unit laptop dengan spesifikasi yang cukup untuk menjalankan perangkat lunak (Software) yang digunakan dalam penelitian. Keduanya digunakan berdasarkan kapasitas dan kemampuan dari tiap perangkat. Perangkat keras dan perangkat lunak digunakan dengan maksimal sehingga penelitian ini dapat berjalan dengan baik. Berikut adalah deskripsi tiap perangkat:

Tabel 3. 2 Tabel Kebutuhan Perangkat Keras

No	Nama Perangkat	Deskripsi
1	Laptop	Asus X441Ba
2	Processor	*Intel Celeron*
3	RAM	4GB
4	Penyimpanan	500GB HDD

Tabel 3. 3 Tabel Kebutuhan Perangkat Lunak

No	Nama	Deskripsi
1	*Windows 10 64-bit*	*Operating System*
2	*Google Colab*	Tools yang digunakan untuk membangun sistem dan pengolahan data serta membuat dan melatih model yang akan digunakan.
3	*Python 3.10.11 dan*	Bahasa pemrograman yang digunakan untuk membangun system.

4	*Pandas*	Pustaka *Python* untuk memanipulasi dan menganalisis data.
5	*Scikit-learn*	Pustaka *Python* yang menyediakan algoritma dan fungsi untuk analisis data, termasuk pengelompokan data (*klasifikasi*).
6	*Matplotlib dan Seaborn*	Pustaka Python untuk visualisasi data.
7	*Microsoft Office* 2016	*Tools* untuk membuat laporan penelitian

BAB 4

HASIL DAN PEMBAHASAN

4.1. *Sample*

4.1.1. Penelitian Sejenis

Penelitian sejenis merupakan tahapan untuk mencari materi yang berkaitan dengan analisa sentimen dan metode *Multinomial Naïve Bayes* serta *Adaboots* sebagai bahan rujukan atau acuan dari penelitian ini. Tahapan ini juga mencari informasi dari penelitian lain yang relevan dengan isu yang dibahas dari penelitian ini.

4.1.2. Scraping Data

Scraping data dengan mengambil data tweet pengguna media sosial twitter (X) dengan memanfaatkan API *twitter* dan autentifikasi token akun pengguna. Data yang diambil adalah data tweet pengguna twitter terkait isu pinjaman online dengan menggunakan kata kunci atau *query* "pinjol". *Scraping* data menggunakan tools Google Collaboratory dengan bahasa pemrograman *Python* dan dibantu dengan *library Python Pandas.*

Data yang diambil adalah data tweet dalam kurun waktu dua bulan yaitu dari mulai tanggal 01-12-2023 sampai 31-01-2024. Didapatkan sebanyak 2.896 data dengan 12 kolom, diantaranya *created_at, id_str, full_text, quote_count, reply_count, retweet_count, favorite_count, lang, user_id_str, conversation_id_str, username dan tweet_url,* namun data yang akan digunakan hanyalah data yang ada pada kolom *full_text,* sehingga kolom lianya akan dibersihkan pada proses *preprocessing.*

```
!pip install pandas
!curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
!sudo apt-get install -y nodejs

Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.23.5)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
2024-02-10 03:56:13 - Installing pre-requisites
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:2 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease [3,626 B]
Get:3 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  InRelease [1,581 B]
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:6 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  Packages [673 kB]
Hit:7 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u4.0+/ubuntu jammy InRelease
Hit:8 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1,440 kB]
Hit:10 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy InRelease
Get:11 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Hit:12 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Get:13 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1,065 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1,751 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,728 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1,810 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,335 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [28.1 kB]
```

Gambar 4. 1 *Instalasi Pandas*

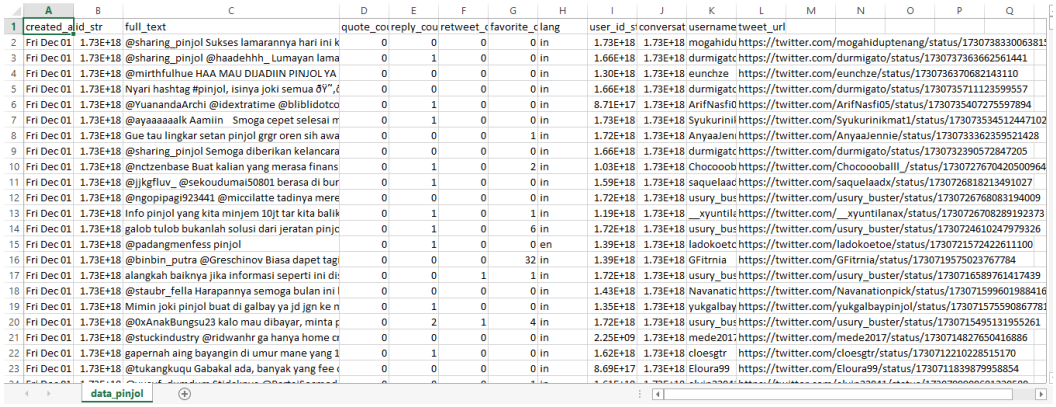
Tahapan *Scraping* dimulai dengan melakukan instalasi *Pandas* dan *instal nodejs*.

```
data = 'data_pinjol.csv'
search_keyword = 'pinjol until:2023-12-01 since:2024-01-31'
limit = 3000

!npx --yes tweet-harvest@2.2.8 -o "{data}" -s "{search_keyword}" -l {limit} --token ""
```

Gambar 4. 2 *Code Python Scraping* Data

Scraping data dilakukan dengan menggunakan kode diatas, yang mana membuat variabel untuk menampung data dalam bentuk csv. Memasukan kata kunci pencarian dengan memasukan *query* “pinjol” dan memberikan batas waktu dari data yang ingin diambil. Kemudian memasukan limit data yang diinginkan, saat kode dijalankan sistem akan meminta memasukan token autentikasi akun pengguna *twitter* yang didapatkan dari API *twitter*. Maka selanjutnya sistem akan mengumpulkan data sesuai dengan perintah yang dimasukan dan secara otomatis akan tersimpan ke dalam file csv.

created_at	id_str	full_text	quote_count	reply_count	retweet_count	favorite_count	lang	user_id_str	conversation_id_str	username	
Fri Dec 01 23:59:08 +0000 2023	1.730000e+18	@sharing_pinjol Sukses lamarannya hari ini ka....	0.0	0.0	0	0.0	in	1.73E+18	1.73E+18	mogahiduptenang	https://twi
Fri Dec 01 23:55:17 +0000 2023	1.730000e+18	@sharing_pinjol @haadehhh_ Lumayan lama ya. Ru...	0.0	1.0	0	0.0	in	1.66E+18	1.73E+18	durmigato	https://tw
Fri Dec 01 23:51:20 +0000 2023	1.730000e+18	@mirthfulhue HAA MAU DIJADIIN PINJOL YA	0.0	0.0	0	0.0	in	1.30E+18	1.73E+18	eunchze	https://twi
Fri Dec 01 23:48:43 +0000 2023	1.730000e+18	Nyari hashtag #pinjol, isinya joki semua ð□□□ð□□□	0.0	0.0	0	0.0	in	1.66E+18	1.73E+18	durmigato	https://tw
Fri Dec 01 23:47:31 +0000 2023	1.730000e+18	@YuanandaArchi @idextratime @bliblidotcom Kesi...	0.0	1.0	0	0.0	in	8.71E+17	1.73E+18	ArifNasfi05	https://t



Gambar 4. 3 Data *Hasil Scraping*

4.2. *Explore*

Tahapan *explore* adalah tahapan melakukan eksplorasi data untuk menemukan informasi sementara yang ada pada data. Sebelumnya data yang telah dikumpulkan dengan query “pinjol” melalui tahapan scraping juga menghasilkan data yang tidak berbahasa Indonesia, maka dilakukan penghapusan data yang tidak berbahasa Indonesia, tahapan tersebut dilakukan secara manual dengan menggunakan tools Microsoft excel. Data hasil scraping didapatkan sebanyak 3.260 dan setelah dilakukan proses penghapusan data yang tidak berbahasa Indonesia didapatkan data sebanyak 2.896 data. Tahapan *explore* dimulai dengan melihat info dari dataset untuk mengetahui jumlah baris dan kolom, tipe data tiap

kolom, baris kosong yang ada pada data dan memfilter data yang diperlukan saja yaitu kolom *full_text.*

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2896 entries, 0 to 2895
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   created_at          2896 non-null   object 
 1   id_str              2896 non-null   float64
 2   full_text           2896 non-null   object 
 3   quote_count         2895 non-null   float64
 4   reply_count         2895 non-null   float64
 5   retweet_count       2895 non-null   object 
 6   favorite_count      2895 non-null   float64
 7   lang                2895 non-null   object 
 8   user_id_str         2895 non-null   object 
 9   conversation_id_str 2895 non-null   object 
 10  username            2894 non-null   object 
 11  tweet_url           2894 non-null   object 
dtypes: float64(4), object(8)
memory usage: 271.6+ KB
```

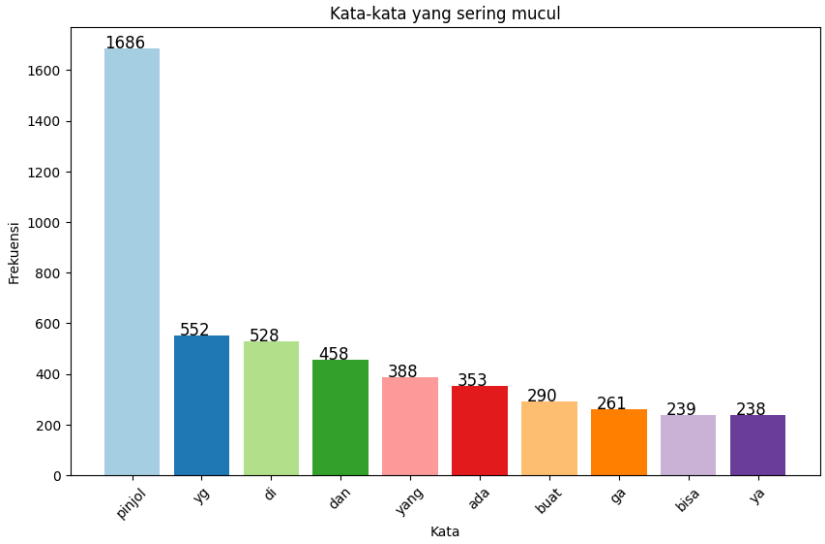
```
[ ] df.shape
```

```
(2896, 12)
```

```
#Memfilter kolom yang diperlukan
df = df.filter(['full_text'])
df.head()
```

Gambar 4. 4 *Ekslpore* Data

Tahapan selanjutnya yaitu melihat frekuensi kata-kata yang sering muncul dari data awal yang belum melalui tahapan *preprocessing*. Hal ini dilakukan untuk nantinya dibandingkan dengan data yang telah melalui tahapan *preprocessing* dan dilihan apakan ada perbedaan dari kedua hal tersebut. Tentu hal ini akan memunculkan informasi yang berguna untuk tahapan-tahapan selanjutnya.



Gambar 4. 5 Frekuensi Kemunculan Kata

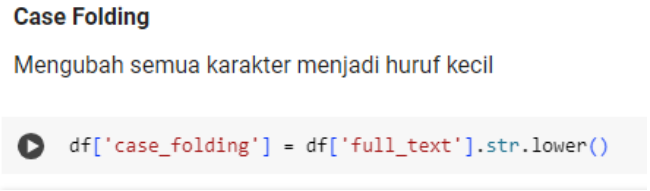
Dihasilkan bahwa kemunculan kata terbanyak dari data yang belum melalui tahapan *preprocessing* adalah "pinjol". Kata-kata lainnya adalah kata bantu atau kata-kata yang kurang memiliki makna atau kurang berpengaruh pada suatu kalimat, hal ini menunjukan bahwa data masih kotor dan perlu adanya tahapan *preprocessing* yang benar untuk mengekstraksi kata yang memang memiliki makna dan berpengaruh dalam suatu kalimat.

4.3. *Modify*

Tahap ini merupakan tahap dimana dataset dilakukan *text preprocessing* untuk dimodifikasi dengan tujuan untuk menjadikan data set lebih terstruktur dan dapat dibaca atau dikenali oleh sistem untuk tahap selanjutnya. Tahapan *text preprocessing* dimulai dari *case folding, cleaning, tokenize, stopword removal, normalize* dan *stemming.* Semua tahapan tersebut menggunakan *tools Google Collaboratory* dengan bahasa pemrograman *Python.* Tahapan *preprocessing* data sangat penting dilakukan karena keadaan data yang kotor dan tidak stabil sehingga akan sulit untuk nantinya dibaca oleh model algoritma.

4.3.1. *Case Folding*

Langkah pertama pada tahapan *modify* atau *text preprocessing* pada penelitian ini adalah *case folding*. Tahap *case folding* adalah merubah semua karakter kata dalam dataset menjadi huruf kecil.



Gambar 4. 6 *Source Code Case Folding*

Setelah dilakukan *case folding* maka hasil data teks akan berubah menjadi seperti berikut:

Tabel 4. 1 Hasil *Case Folding*

Full_text	***Case folding***
@YuanandaArchi @idextratime @bliblidotcom Kesian bro... Ntar lg bnyak tagihan pinjol si botak... Kyaknya bntr lg kebayar dgn azab mnggadaikan kehormatan bgsa dgn jd mangsa pinjol...	@yuanandaarchi @idextratime @bliblidotcom kesian bro... ntar lg bnyak tagihan pinjol si botak... kyaknya bntr lg kebayar dgn azab mnggadaikan kehormatan bgsa dgn jd mangsa pinjol...

4.3.2. Cleaning

Tahapan *cleaning* dilakukan untuk membersihkan data dari adanya tanda baca, *emoticon*, url link, *hashtag*, *white space* serta angka yang tidak berguna serta menghapus dari atribut atribut yang tidak penting.

```
import string
import re
def cleaning(komentar):
    #remove ascii
    komentar = komentar.encode('ascii', 'replace').decode('ascii')

    #remove angka
    komentar = re.sub('[0-9]+', '', komentar)

    #remove mention, link, hashtag
    komentar = re.sub('<br.*?>(.+?)</br>',' ',komentar)
    komentar = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)"," ", komentar).split())
    komentar = re.sub('@[^\s]+','',komentar)

    #remove url
    komentar = re.sub(r'\w+:\/{2}[\d\w-]+(\.[\d\w-]+)*(?:(?:\/[^\s/]*))*', '', komentar)

    #remove tanda baca
    komentar = re.sub(r'[^\w\d\s]+', '', komentar)

    #remove whitespace
    komentar = re.sub('\s+',' ',komentar)

    #remove line baru
    komentar = re.sub('\n',' ',komentar)

    #remove garis bawah
    komentar = re.sub('_',' ',komentar)

    return komentar
df['cleaning'] = df['case_folding'].apply(cleaning)
# menghapus data duplikat
df.drop_duplicates(subset ="cleaning", keep = 'first', inplace = True)
df.head()
```

Gambar 4. 7 *Source Code Cleaning* Data

Tabel 4. 2 Hasil *Cleaning* Data

Full_text	***Cleaning***
@YuanandaArchi @idextratime @bliblidotcom Kesian bro... Ntar lg bnyak tagihan pinjol si botak... Kyaknya bntr lg kebayar dgn azab mnggadaikan kehormatan bgsa dgn jd mangsa pinjol...	kesian bro tar lg bnyak tagihan pinjol si botak kyaknya bntr lg kebayar dgn azab mnggadaikan kehormatan bgsa dgn jd mangsa pinjol

Proses *cleaning* dibantu dengan *library string* dan *ragex*. Kemudian dilakukan dengan membuat *function* dengan parameter komentar dan memasukan perintah pada parameter tersebut. Selanjutnya memanggil kembali parameter komentar tersebut. Data yang dibersihkan adalah data hasil dari *case folding* sehingga menghasilkan data yang terstruktur.

4.3.3. *Tokenize*

Selanjutnya proses tokenisasi adalah proses memecahkan kalimat menjadi potongan kata-kata menjadi token untuk mempermudah mengetahui kata dasar dari kata tersebut.

```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
from nltk.tokenize import word_tokenize
def word_tokenize_wrapper(cleaning):
  return word_tokenize(cleaning)
df['tokenize'] = df['cleaning'].apply(word_tokenize_wrapper)
df.head()
```

Gambar 4. 8 *Source Code Tokenize*

Tokenisasi menggunakan *library nltk python* dengan mengimportkan fungsi *word_tokenize.* Selanjutnya data yang diambil adalah data dari kolom *cleaning* atau data yang sudah bersih sehingga hasil *tokenize* hanyalah berupa kata-kata yang ada pada kalimat tersebut.

Tabel 4. 3 Hasil *Tokenize*

Cleaning	***Tokenize***
kesian bro tar lg bnyak tagihan pinjol si botak kyaknya bntr lg kebayar dgn azab mnggadaikan kehormatan bgsa dgn jd mangsa pinjol	[kesian, bro, ntar, lg, bnyak, tagihan, pinjol, si, botak, kyaknya, bntar, lg, kebayar, dgn, azab, mnggadaikan, kehormatan, bgsa, dgn, jd, mangsa, pinjol]

4.3.4. *Stopword Removal*

Stopword removal adalah tahapan menghapus kata-kata umum yang banyak digunakan namun tidak memiliki makna dan tidak memberikan pengaruh sentimen pada suatu kalimat. Proses *stopword* yang digunakan adalah dengan memanfaatkan *library* dari *nltk* yang di dalamnya terdapat *corpus stopword* bahasa Indonesia.

Selain itu juga dilakukan penambahan *corpus stoword* secara manual untuk memaksimalkan hasil data. Kemudian data yang diambil adalah data dari kolom *tokenize*, ini lah gunanya tokenisasi kata, untuk memudahkan sistem membaca data satu persatu berdasarkan token. Berikut adalah *source code stopword removal.*

```
from nltk.corpus import stopwords
list_stopwords = stopwords.words('indonesian')
#tambahkan stopword manual
list_stopwords.extend(['tu', 'uf', 'deh','nak','amp','b','a','w','je','jd','x','sih',
                       'dos', 'haa', 'eh', 'tuh', 'hmm','grgr', 'nya','ufufuf', 'lho',
                       'rm', 'nya', 'ufcc', 'ppv','dok', 'je', 'gb', 'pa', 'e', 'br',
                       'ya', 'yg', 'di', 'ga','lu','noh','thx','gpp','hehehe','dll'])
sw = set(list_stopwords) - set (['jangan', 'jangankan', 'janganlah', 'tidak', 'tidakkah', 'tidaklah'])
def stopwords_removal(words):
  return [word for word in words if word not in sw]

df['stopword'] = df['tokenize'].apply(stopwords_removal)
df.head()
```

Gambar 4. 9 *Source Code Stopword Removal*

Tabel 4. 4 Hasil Stopword Removal

Tokenize	***Stopword Removal***
['aamiin', 'smoga', 'cepet', 'selesai', 'masalahnya', 'ya', 'kak', 'sy', 'jga', 'sama', 'kak', 'mumet', 'banget', 'ditagih', 'pinjol', 'drpada', 'galob', 'tulob', 'sy', 'putuskan', 'buat', 'pasang', 'badan', 'aja', 'mau', 'gimana', 'lagi', 'smoga', 'kita', 'smua', 'dlm', 'lindungannya', 'aamiin']	'aamiin', 'smoga', 'cepet', 'selesai', 'kak', 'sy', 'jga', 'kak', 'mumet', 'banget', 'ditagih', 'pinjol', 'drpada', 'galob', 'tulob', 'sy', 'putuskan', 'pasang', 'badan', 'aja', 'gimana', 'smoga', 'smua', 'dlm', 'lindungannya', 'aamiin'

4.3.5. *Normalize*

Tahap berikutnya adalah normalisasi kata atau *normalize*. tahap normalisasi dilakukan untuk menstandarisasi kata yang memiliki makna yang sama dengan melakukan perubahan penulisan kata yang disingkat dan atau tidak baku. Penelitian ini menggunakan sebuah kamus normalisasi yang didapatkan dari kamus NLP (*Neuro Linguistic Programming*) bahasa Indonesia Resource (Owen et al., 2020). Kamus kata tersebut tersedia *open source* di github.

	A	B
1	**tidak_baku**	**kata_baku**
2	woww	wow
3	aminn	amin
4	met	selamat
5	netaas	menetas
6	keberpa	keberapa
7	eeeehhhh	eh
8	kata2nyaaa	kata-katanya
9	hallo	halo
10	kaka	kakak
11	ka	kak
12	daah	dah
13	aaaaahhhh	ah
14	yaa	ya
15	smga	semoga
16	slalu	selalu
17	amiin	amin
18	kk	kakak
19	trus	terus
20	kk	kakak
21	sii	sih
22	nyenengin	menyenangkan
23	bgt	banget

Gambar 4. 10 Kamus Normalisasi Kata

```
normalisasi_kata = pd.read_excel("kamuskatabaku.xlsx")

normalisasi_kata_dict = {}
for index, row in normalisasi_kata.iterrows():
  if row[0] not in normalisasi_kata_dict:
    normalisasi_kata_dict[row[0]] = row[1]

def normalisasi_term(document):
  return [normalisasi_kata_dict[term]
  if term in normalisasi_kata_dict
          else term
          for term in document]

df['normalisasi'] = df['stopword'].apply(normalisasi_term)
df.head()
```

Gambar 4. 11 *Source Code* Normalisasi Kata

Sistem akan menganalisa setiap kata yang ada kemudian jika ditemukan kata yang tidak baku yang ada pada kamus kata maka sistem akan menggantinya dengan kata baku yang ada pada kamus kata. Berikut adalah hasil normalisasi kata :

Tabel 4. 5 Hasil Normalisasi Kata

Stopword Removal	**Normalisasi**
'aamiin', 'smoga', 'cepet', 'selesai', 'kak', 'sy', 'jga', 'kak', 'mumet', 'banget', 'ditagih', 'pinjol', 'drpada', 'galob', 'tulob', 'sy', 'putuskan', 'pasang', 'badan', 'aja', 'gimana', 'smoga', 'smua', 'dlm', 'lindungannya', 'aamiin'	'amin', 'semoga', 'cepat', 'selesai', 'kak', 'saya', 'juga', 'kak', 'mumet', 'banget', 'ditagih', 'pinjol', 'daripada', 'galob', 'tulob', 'saya', 'putuskan', 'pasang', 'badan', 'saja', 'bagaimana', 'semoga', 'semua', 'dalam', 'lindungannya', 'amin'

4.3.6. *Stemming*

Tahapan selanjutnya adalah *stemming*, yaitu pengubahan setiap kata menjadi kata dasar berdasarkan KBBI, tahapan ini berguna untuk menemukan makna yang sesuai dengan kata yang ada sehingga identifikasi

kelas nantinya akan mudah. Proses *stemming* menggunakan *library sastrawi* sebagai acuan *corpus* kamus besar bahasa Indonesia.

```
!pip install PySastrawi

import Sastrawi
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
```

```
Collecting PySastrawi
  Downloading PySastrawi-1.2.0-py2.py3-none-any.whl (210 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 210.6/210.6 kB 4.2 MB/s eta 0:00:00
Installing collected packages: PySastrawi
Successfully installed PySastrawi-1.2.0
```

Stemming

Menghapus kata-kata yang tidak memiliki makna berdasarkan library sastrawi

```
[10] factory = StemmerFactory()
     stemmer = factory.create_stemmer()
     def stemmed_wrapper(term):
       return stemmer.stem(term)
     df['stemming'] = df['normalisasi'].apply(lambda x: [stemmer.stem(y) for y in x])
     df.head()
```

Gambar 4. 12 *Source Code Steamming*

Proses stemming akan mengambil data dari hasil normalisasi untuk selanjutnya dideteksi keberadaan kata yang tidak sesuai KBBI dan diganti kata tersebut sesuai dengan kamus besar bahasa Indonesia. Berikut adalah hasil *stemming* data.

Tabel 4. 6 Hasil *Steamming* Data

Normalisasi	Stemming
'pinjol', 'semoga', 'kelancaran', 'kak', 'semoga', 'jalan', 'pinjamannya'	'pinjol', 'moga', 'lancar', 'kak', 'moga', 'jalan', 'pinjam'

4.3.7. *Clean*

Tahapan akhir dari proses text preprocessing adalah pembersihan data dari adanya tanda baca yang ada pada kata. Hal ini dilakukan karena tahapan labeling tidak bisa bekerja jika kalimat masih memiliki tanda baca.

```
import string
def remove_punct(text):
  text = " ".join([char for char in text if char not in string.punctuation])
  return text
df['clean'] = df['stemming'].apply(lambda x: remove_punct(x))
df.head()
```

Gambar 4. 13 *Source Code Cleaning*

Tabel 4. 7 Hasil *Clean* Data

Stemming	Clean
['pinjol', 'moga', 'lancar', 'kak', 'moga', 'jalan', 'pinjam']	pinjol moga lancar kak moga jalan pinjam

Cleaning data dilakukan dengan mengambil data dari kolom stemming. Menggunakan library string untuk membaca data string dan fungsi penghapusan tanda baca yang ada pada data.

4.4. Modeling

Tahapan pemodelan dimulai dengan melakukan *labeling* pada dataset untuk menentukan kelas sentiment pada data. *Labeling* dilakukan dengan menggunakan metode *lexicon based* dengan kamus kata positif dan negatif. Selanjutnya sebelum data diolah oleh model algoritma dilakukan tahapan *feature extraction*, tahapan ini mengubah data teks menjadi bentuk numerik agar bisa dibaca oleh model algoritma. Kemudian data akan diolah berdasarkan algoritma yang telah ditentukan yaitu *Multinomial Naïve Bayes* dan *Adaboost* untuk menemukan hasil prediksi klasifikasi kelas sentiment berdasarkan kedua model algoritma tersebut.

4.4.1. *Labeling* dengan *Lexicon Based*

Untuk melakukan klasifikasi digunakanlah metode *lexicon based*, dilakukan untuk memberikan label pada kelas dataset. Terdapat 3 kelas yaitu kelas positif, kelas negatif dan kelas netral di dalam suatu kalimat atau

dokumen pada dataset. Kata-kata dalam kumpulan dataset akan dibandingkan dengan dokumen kamus *lexicon*. Perbandingan kata tersebut nantinya diberikan nilai atau score jika kata pada kalimat atau dokumen yang ada pada data, terdapat kesamaan dengan kata yang ada di dalam kamus *lexicon.*

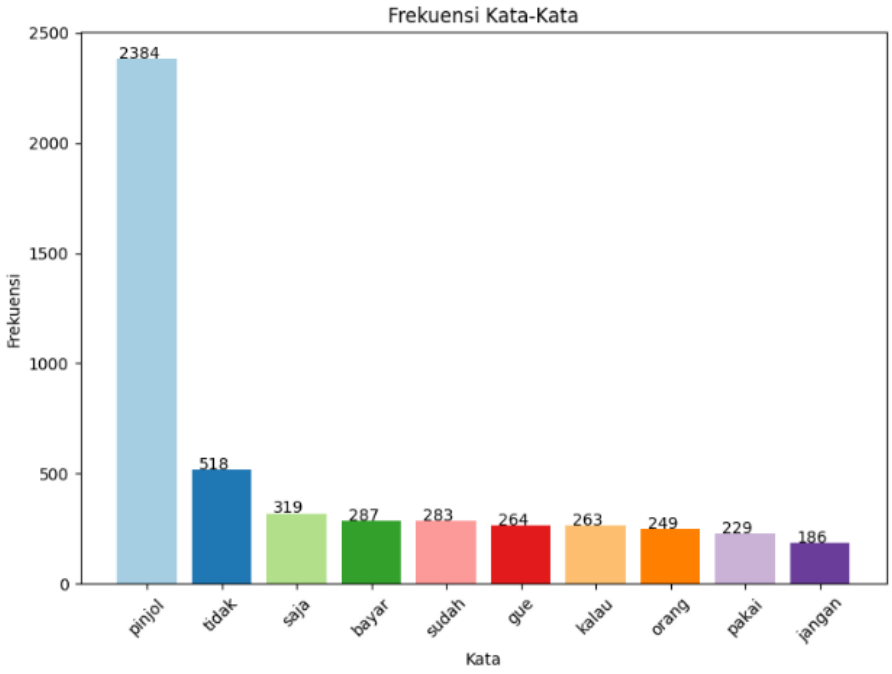
Jumlah nilai atau score ini akan menentukan komentar tersebut termasuk ke dalam label positif atau negatif, jika nilai score pada komentar tersebut bernilai 0 maka komentar tersebut tergolong kelas netral. Kamus lexicon yang digunakan bersumber dari yaitu *Indonesian Sentiment* (Inset) (Koto & Rahmaningtyas, 2017). Terdapat kurang lebih 10.250 kata yang diberi nilai dari -5 hingga +5.

```
# Membuat class untuk membaca kamus lexicon positif dan negatif
lexicon_positive = dict()
with open("/content/Kamus_Positif.csv", "r") as csvfile:
  reader = csv.reader(csvfile, delimiter=",")
  for row in reader:
    lexicon_positive[row[0]] = int(row[1])


lexicon_negative = dict()
with open("/content/Kamus_Negatif.csv", "r") as csvfile:
  reader = csv.reader(csvfile, delimiter=",")
  for row in reader:
    lexicon_negative[row[0]] = int(row[1])


# Membuat fungsi untuk mempolarisasi sentimen berdasarkan kamus lexicon yang ada
def sentiment_analysis(text):
  score = 0
  for word in text:
    if (word in lexicon_positive):
      score = score + lexicon_positive[word]
  for word in text:
    if (word in lexicon_negative):
      score = score + lexicon_negative[word]
  polarity = ''
  if (score > 0 ):
    polarity = "Positif"
  elif (score < 0 ):
    polarity = "Negatif"
  else :
    polarity = "Netral"
  return score, polarity
```

```
[ ]  result = df['stemming'].apply(sentiment_analysis)
     result = list(zip(*result))
     df["polarity_score"] = result[0]
     df["polarity"] = result[1]
     print(df["polarity"].value_counts())
     df.shape

     Negatif    1591
     Netral      509
     Positif     462
     Name: polarity, dtype: int64
     (2562, 10)
```

Gambar 4. 14 *Source Code Labeling* Data

Dari hasil *labeling* data dengan kamus *lexicon* didapatkan bahwa 1.591 data masuk kedalam kelas negatif, 509 data masuk kedalam kelas positif dan 462 masuk kedalam kelas netral. Kemudian pada plot frekuensi kemunculan kata, “pinjol” masih menjadi kata yang sering muncul, namun berbeda dengan kemunculan kata pada data sebelum dilakukan *preprocessing*, setelah dilakukan *preprocessing* data kata-kata yang sering muncul pada plot frekuensi kata memang banyak menunjukan kata yang berkonotasi negatif.



Gambar 4. 15 Frekuensi Kata Setelah *Preprocessing*

Gambar 4. 16 *Wordcloud*

Pada *Wordcloud* juga menampilkan hal yang sama. Digambarkan juga banyak kata lainnya yang memang mengarah pada konotasi yang negatif. Sehingga memang valid jika data lebih banyak yang berada pada kelas negatif.

4.4.2. *Feature Extraction*

Pada tahapan feature extraction pertama dilakukan pembagian data uji dan data latih dengan pembagian secara random dan didapatkan jumlah data latih sebanyak 2049 data Dan data jumlah data uji 513 data.

```
[ ] X_train, X_test, y_train, y_test = train_test_split(df['clean'], df['polarity'], test_size=0.2, random_state=34)

[ ] print(f'Jumlah Data Latih: {len(X_train)}')
    print(f'Jumlah Data Uji: {len(X_test)}')

    Jumlah Data Latih: 2049
    Jumlah Data Uji: 513
```

Gambar 4. 17 Pembagian Data Latih dan Uji

Selanjutnya adalah ekstraksi fitur dengan TF-IDF caranya dengan melakukan *import* dari *sklearn.feature_extraction.text* dan memanggil library *TfidfVectorizer*. Selanjutnya memasukan data uji dan data latih untuk diolah oleh library *TfidfVectorizer* dengan fungsi *fit_transform*. Sistem akan mengubah data menjadi bentuk numerik dengan perhitungan TF-IDF untuk data latih dan data uji.

```
[ ]  vectorizer = TfidfVectorizer(strip_accents='ascii')
     tf_idf_train = vectorizer.fit_transform(X_train)
     tf_idf_test = vectorizer.fit_transform(X_test)
```

Gambar 4. 18 TF-IDF *Feature Extraction*

Pada tahapan modeling dilakukan pembangunan model algoritma yang telah ditetapkan yaitu Multinomial *Naïve Bayes* dan *Adaboost*. Pembangunan model algoritma juga melakukan impor dari sklearn dan memanggil library algoritma *Multinomial Naïve Bayes* dan *Adaboost.* Caranya dengan membuat variabel model untuk menampung algoritma yang dipilih dan memasukan hasil ekstraksi fitur kedalam algoritma untuk evaluasi nantinya.

```
[ ]  from sklearn.naive_bayes import MultinomialNB
     from sklearn.ensemble import AdaBoostClassifier

[ ]  model = MultinomialNB()
     model.fit(tf_idf_train, y_train)

     ▾ MultinomialNB
     MultinomialNB()

[ ]  model_2 = AdaBoostClassifier()
     model_2.fit(tf_idf_train, y_train)

     ▾ AdaBoostClassifier
     AdaBoostClassifier()
```

Gambar 4. 19 Pembangunan Model Algoritma

4.5. *Asses*

Terakhir adalah tahapan *asses* yaitu tahapan untuk mengevaluasi kedua model algoritma untuk mengukur performa dari model yang telah dibuat. Hasil evaluasi yang digunakan pada kedua metode ini berupa nilai *confusion matrix* dengan ukuran 3x3, yang berisi nilai akurasi, presisi, *f1-score* dan *recall* dari

masing-masing kelas. Nilai tersebut didapatkan dari hasil pengolahan data *train*. Dalam menentukan nilai *confusion matrix*, penelitian ini menggunakan *cross validation* agar nilai yang dihasilkan maksimal.

Secara umum penggunaan *confusion matrix* biasanya banyak digunakan pada data yang hanya memiliki dua kelas dan digambarkan sebagai berikut:

Tabel 4. 8 *Confusion Matrix* 2x2

Aktual Data	Prediksi Data	
	Positif	Negatif
Positif	TP	FN
Negatif	FP	TN

Pada penelitian ini kelas yang dihasilkan memiliki tiga kelas sehingga *confusion matrixnya* juga *matrix* berukuran 3x3. Untuk *matriks* yang memiliki *multi class* tentu perhitunganya berbeda dengan matrik biasa, berikut tabel perhitungan matrix *multiclass:*

1. Kelas negatif

Tabel 4. 9 *Confusion Matrix* 3x3 Kelas Negatif

Aktual Data	Prediksi Data		
	Negatif	Netral	Positif
Negatif	TP	FN	FN
Netral	FP	TN	TN
Positif	FP	TN	TN

2. Kelas netral

Tabel 4. 10 *Confusion Matrix* 3x3 Kelas Netral

Aktual Data	Prediksi Data		
	Negatif	Netral	Positif
Negatif	TN	FP	TN
Netral	FN	TP	FN
Positif	TN	FP	TN

3. Kelas positif

Tabel 4. 11 *Confusion Matrix* 3x3 Kelas Positif

Aktual Data	Prediksi Data		
	Positif	Netral	Positif
Negatif	TN	TN	FP
Netral	TN	TN	FP
Positif	FN	FN	TP

4.5.1. *Multinomial Naïve Bayes*

Pada tahapan pembangunan model pembagian data uji dan data latih dilakukan secara random. Pada tahapan evaluasi digunakan data latih karena jumlah datanya lebih banyak. Pada model algoritma *Multinomial Naïve Bayes* dilakukan evaluasi model dan didapatkan hasil nilai akurasi sebesar 0.71 atau 71%.

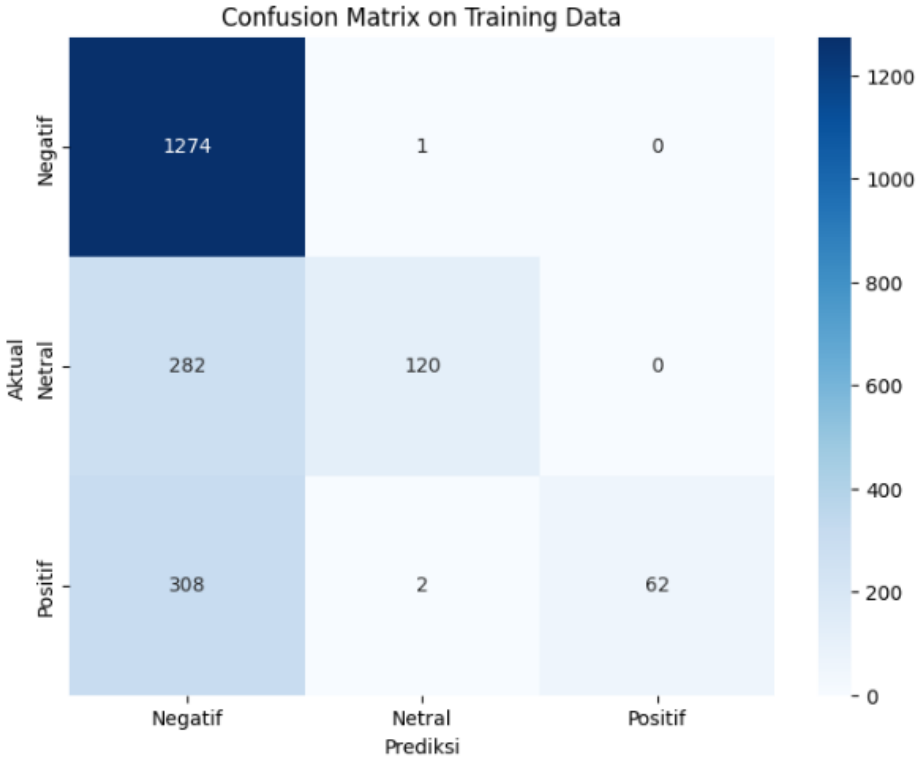
```
# Evaluasi Model menggunakan data latih
predictions_train = model.predict(tf_idf_train)

# Akurasi
accuracy_train = accuracy_score(y_train, predictions_train)
print(f'Accuracy on Training Data: {accuracy_train:.2f}')

# Report klasifikasi
print('\nClassification Report on Training Data:\n', classification_report(y_train, predictions_train))

conf_matrix_train = confusion_matrix(y_train, predictions_train)

# Plot confusion matrix untuk data latih
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_train, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Negatif', 'Netral', 'Positif'], yticklabels=['Negatif', 'Netral', 'Positif'])
plt.title('Confusion Matrix on Training Data')
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.show()
```

Gambar 4. 20 Evaluasi Model *Multinomial* NB

```
Accuracy on Training Data: 0.71

Classification Report on Training Data:
               precision    recall  f1-score   support

     Negatif       0.68      1.00      0.81      1275
      Netral       0.98      0.30      0.46       402
     Positif       1.00      0.17      0.29       372

    accuracy                           0.71      2049
   macro avg       0.89      0.49      0.52      2049
weighted avg       0.80      0.71      0.65      2049
```

Gambar 4. 21 Hasil Evaluasi Model *Multinomial* NB

Dari hasil evaluasi model algoritma *Multinomial Naïve Bayes* didapatkan nilai akurasi sebesar 71%, nilai presisi untuk kelas negatif 68%, *recall* sebesar 100% dan *f1-score* sebesar 81%. Untuk kelas positif nilai posisinya sebesar 100%, *recall* sebesar 17% dan *f1-score* sebesar 29%. Dan untuk kelas netral nilai posisinya sebesar 98%, *recall* sebesar 30% dan nilai *f1-score* sebesar 46%. Hasil dari evolusi model algoritma dipengaruhi oleh banyak hal karenanya nilai evaluasi yang tidak maksimal tidak menandakan bahwa algoritma tersebut tidak baik untuk mengolah data sejenis. Untuk lebih jelasnya perlu ditampilkan nilai dari *confusion matrix*, berikut adalah plot tampilan *confusion matrix:*



Gambar 4. 22 *Confusion Matrix Multinomial NB*

Berikut adalah perhitungan manual untuk mengukur kinerja algoritma Multinomial Naïve Bayes berdasarkan nilai dari *confusion matrix:*

Akurasi = $\frac{TP+TN}{TP+FN+FP+TN} \times 100\%$

$\frac{1274+(120+0+2+64)}{1274+1+(282+308)+(120+0+1+62)} \times 100\%$ $\frac{1457}{2048} \times 100\% = 71\%$

Presisi positif = $\frac{TP}{TP+FP} \times 100\%$ $\frac{62}{62+0} \times 100\% = 1$

Recall positif = $\frac{TP}{TP+FN} \times 100\%$ $\frac{62}{62+(308+2)} \times 100\% = 0{,}17$

F1-score positif = $\frac{presisi\ positif \times recall\ positif}{presisi\ positif + recall\ positif} \times 2$ $\frac{1 \times 0{,}17}{1+0{,}17} \times 2 = 0{,}29$

Presisi negatif = $\frac{TP}{TP+FP} \times 100\%$ $\frac{1274}{1274+(282+308)} \times 100\% = 0{,}68$

Recall negatif = $\frac{TP}{TP+FN} \times 100\%$ $\frac{1274}{1274+(1+0)} \times 100\% = 1$

F1-score negatif = $\frac{presisi\ negatif \times recall\ negatif}{presisi\ negatif + recall\ negatif} \times 2$ $\frac{0{,}68 \times 1}{0{,}68+1} \times 2 = 0{,}81$

Presisi netral = $\frac{TP}{TP+FP} \times 100\%$ $\frac{120}{120+(1+2)} \times 100\% = 0{,}98$

Recall netral = $\frac{TP}{TP+FN} \times 100\%$ $\frac{120}{120+(282+0)} \times 100\% = 0{,}30$

F1-score netral = $\frac{presisi\ netral \times recall\ netral}{presisi\ netral + recall\ netral} \times 2$ $\frac{0{,}98 \times 0{,}30}{0{,}98+0{,}30} \times 2 = 0{,}46$

4.5.2. *Adaboost*

Evaluasi model untuk algoritma *Adaboost* juga dilakukan dengan hal yang sama, yaitu dengan menghitung nilai akurasi, presisi, *recall* dan *f1-score*. Juga digunakan confusion matrix untuk melihat nilai dari kinerja algoritma. Dalam evaluasi model algoritma adaboost juga digunakan data latih.

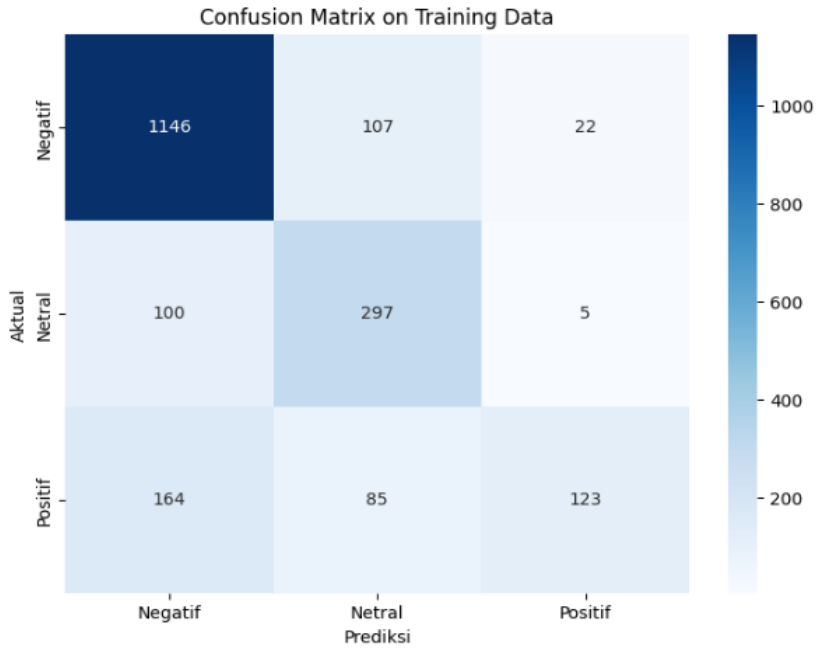
```
# Evaluasi Model menggunakan data latih
predictions_train = model_2.predict(tf_idf_train)

# Akurasi
accuracy_train = accuracy_score(y_train, predictions_train)
print(f'Accuracy on Training Data: {accuracy_train:.2f}')

# Report klasifikasi
print('\nClassification Report on Training Data:\n', classification_report(y_train, predictions_train))

conf_matrix_train = confusion_matrix(y_train, predictions_train)

# Plot confusion matrix untuk data latih
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_train, annot=True, fmt='d', cmap='Blues',
            xticklabels=['Negatif', 'Netral', 'Positif'], yticklabels=['Negatif', 'Netral', 'Positif'])
plt.title('Confusion Matrix on Training Data')
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.show()
```

Gambar 4. 23 Evaluasi Model *Adaboost*

```
Accuracy on Training Data: 0.76

Classification Report on Training Data:
              precision    recall  f1-score   support

     Negatif       0.81      0.90      0.85      1275
      Netral       0.61      0.74      0.67       402
     Positif       0.82      0.33      0.47       372

    accuracy                           0.76      2049
   macro avg       0.75      0.66      0.66      2049
weighted avg       0.77      0.76      0.75      2049
```

Gambar 4. 24 Hasil Evaluasi *Adaboost*

Sedangkan untuk hasil evaluasi model algoritma Adaboost didapatkan nilai akurasi sebesar 76%, nilai presisi untuk kelas negatif 81%, *recall* sebesar 90% dan *f1-score* sebesar 85%. Untuk kelas positif nilai presisi sebesar 82%, *recall* sebesar 33% dan *f1-score* sebesar 47%. Dan untuk kelas netral nilai presisinya sebesar 61%, *recall* sebesar 74% dan nilai *f1-score* sebesar 67%. Untuk lebih jelasnya perlu ditampilkan nilai dari *confusion matrix*, berikut adalah plot tampilan *confusion matrix:*



Gambar 4. 25 Confusion Matrix Adaboost

Akurasi = $\frac{TP+TN}{TP+FN+FP+TN} \times 100\%$

$\frac{1146+(297+5+85+123)}{1146+(107+22)+(100+164)+(297+5+85+123)} \times 100\%$

$\frac{1656}{1146+129+264+510} \times 100\%$ $\frac{1656}{2049} \times 100\% = 0,76$

Presisi positif = $\frac{TP}{TP+FP} \times 100\%$ $\frac{123}{123+(22+5)} \times 100\% = 0,82$

Recall positif = $\frac{TP}{TP+FN} \times 100\%$ $\frac{123}{123+(164+85)} \times 100\% = 0,33$

F1-score positif = $\frac{presisi\ positif \times recall\ positif}{presisi\ positif + recall\ positif} \times 2$ $\frac{0,82 \times 0,33}{0,82+0,33} \times 2 = 0,47$

Presisi negatif = $\frac{TP}{TP+FP} \times 100\%$ $\frac{1146}{1146+(100+164)} \times 100\% = 0,81$

Recall negatif = $\frac{TP}{TP+FN} \times 100\%$ $\frac{1274}{1274+(107+22)} \times 100\% = 0,90$

F1-score negatif = $\frac{presisi\ negatif \times recall\ negatif}{presisi\ negatif + recall\ negatif} \times 2$ $\frac{0,81 \times 0,90}{0,81+0,90} \times 2 = 0,85$

Presisi netral = $\frac{TP}{TP+FP} \times 100\%$ $\frac{297}{297+(107+85)} \times 100\% = 0,61$

Recall netral = $\frac{TP}{TP+FN} \times 100\%$ $\frac{297}{297+(100+5)} \times 100\% = 0,74$

F1-score netral = $\frac{presisi\ netral \times recall\ netral}{presisi\ netral + recall\ netral} \times 2$ $\frac{0,61 \times 0,74}{0,61+0,74} \times 2 = 0,67$

4.6. Interprestasi Hasil

Rincian hasil dari penelitian ini adalah proses pengambilan data menggunakan metode scraping data pada media sosial twitter dengan query atau kata kunci “pinjol” didapatkan data sebanyak 2.896 data. Pengambilan data komentar tersebut menggunakan tools Google Collaboratory dimana bahasa pemrograman yang dipakai adalah bahasa pemrograman python.

Tahapan tahapan yang dilakukan di dalam penelitian ini diantaranya adalah *web scraping*, *text pre- processing,* pelabelan dataset dengan menggunakan kamus lexicon, klasifikasi data menggunakan 2 metode klasifikasi yaitu dengan Naïve Bayes dan K-Nearest Neighbor, pengujian *confusion matrix* dan visualisasi dimana visualisasi menggunakan *wordcloud*.

Selanjutnya data yang sudah didapatkan dari teknik *scraping* tersebut dilakukan tahap *pre-processing*. Tahap *pre-processing* pada penelitian ini meliputi beberapa tahapan lagi yaitu seperti *case folding*, *cleaning*, *tokenize, normalize, dan stopword removal dan stemming,*. Dataset yang sudah dilakukan tahapan-tahapan *preprocessing* ini selanjutnya disebut dengan data bersih yang akhirnya berjumlah 2.562 data tweet bersih.

Data yang sudah dibersihkan dan terstruktur selanjutnya dilakukan proses pelabelan yaitu melabelkan data menjadi 3 kelas sentimen yaitu kelas positif, kelas negatif dan kelas netral. Pelabelan kelas didalam dataset ini menggunakan salah satu tahap pemodelan yaitu *lexicon based*. Kata-kata atau data komentar yang ada pada dataset dibandingkan dengan dokumen kamus *lexicon*. Data komentar tersebut akan dibandingkan dan diberikan nilai atau *score* sesuai dengan kamus *lexicon*. Jumlah dari *score* tersebut akan menentukan data komentar tersebut memiliki label

kelas positif atau kelas negatif. Kamus *lexicon* yang digunakan bersumber dari Koto (2017) *Indonesian Sentiment* (Inset). Hasil dari perbandingan data komentar dengan dokumen kamus *lexicon* ini terdapat 462 data berlabel positif, 1.591 data berlabel negatif, dan terdapat 509 data berlabel netral.

Tahap terakhir yaitu tahap *asses* pada penelitian ini. Tahap ini dilakukan evaluasi dari 2 model klasifikasi yaitu Multinomial Naïve Bayes dan Adaboost. Evaluasi tersebut berisikan nilai *confusion matrix*, nilai akurasi, nilai presisi, nilai *recall*, dan *f1-score* dari data uji pada masing masing model klasifikasi. Didapatkan bahwa metode Multinomial Naïve Bayes mendapatkan hasil nilai akurasi sebesar 71% dan metode Adaboost mendapatkan hasil nilai akurasi sebesar 76%.

Nilai presisi, recall dan f1-score pada masing-masing kelas juga ditampilkan pada perhitungan manual diatas. Dari hasil tersebut disimpulkan bahwa algoritma Adaboost bekerja lebih baik dalam mengolah data tweet terkait isu pinjaman online dibandingkan dengan algoritma Multinomial Naïve bayes. Dari hasil analisa juga didapatkan bahwa keberadaan pinjaman online lebih banyak mendapatkan komentar negatif di media sosial twitter dan hal ini mengindikasikan bahwa keberadaan pinjaman online berdampak buruk bagi masyarakat.

BAB V

PENUTUP

5.1. Kesimpulan

Dari penelitian ini kesimpulan yang dapat diambil adalah:

1. Nilai akurasi, presisi, *recall* dan *f1-score* pada algoritma *Adaboost* lebih tinggi jika dibandingkan dengan algoritma *Multinomial Naïve Bayes*, yang mana algoritma *Adaboost* mendapatkan nilai akurasi sebesar 76% sedangkan algoritma *Multinomial Naïve Bayes* mendapatkan nilai akurasi sebesar 71%. Terbukti bahwa algoritma *Adaboost* bekerja lebih baik dalam mengolah data sentiment twitter terkait isu pinjaman online.
2. Hasil klasifikasi sentiment dengan menggunakan kamus *lexicon* bahasa Indonesia didapatkan bahwa terdapat 462 data berlabel positif, 1.591 data berlabel negatif, dan terdapat 509 data berlabel netral. Hal ini membuktikan bahwa keberadaan pinjaman online dari komentar masyarakat di media sosial twitter berdampak buruk bagi masyarakat.

5.2. Saran

Berdasarkan hasil penelitian pada penelitian ini, peneliti memiliki beberapa saran yang dapat menjadi masukan dan bahan pertimbangan untuk penelitian selanjutnya disarankan untuk:

1. Penelitian selanjutnya dapat menggunakan media sosial lain sebagai object atau sumber data penelitian seperti instagram atau facebook.
2. Penelitian selanjutnya dapat menggunakan metode atau algoritma lain

3. Selain *Multinomial Naïve Bayes* dan *Adaboost* atau melakukan kombinasi algoritma untuk mendapatkan pemahaman baru dalam hal text mining.
4. Menggunakan metode pengklasifikasian yang lain selain dengan metode *lexicon* kamus bahasa indonesia
5. Menggunakan metode *feature extraction* yang lain selain TF-IDF untuk menghasilkan kinerja algoritma yang lebih baik, karena kinerja algoritma juga dipengaruhi oleh tahapan *feature extraction.*

DAFTAR PUSTAKA

Ashari, H., Arifianto, D., Azizah, H., & Faruq, A. (2020). Perbandingan Kinerja Algoritma Multinomial Naive Bayes (MNB, Multivariate Bernoulli dan Rocchio Algorithm Dalam Klasifikasi Konten Berita Hoax Berbahasa Indonesia Pada Media Sosial. *Http://Repository.Unmuhjember.Ac.Id*, 1–12.

Azhar, Y. (2018). Metode Lexicon-Learning Based Untuk Identifikasi Tweet Opini Berbahasa Indonesia. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, *6*(3), 237. https://doi.org/10.23887/janapati.v6i3.11739

Devita, R. N., Herwanto, H. W., & Wibawa, A. P. (2018). Perbandingan Kinerja Metode Naive Bayes dan K-Nearest Neighbor untuk Klasifikasi Artikel Berbahasa indonesia. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *5*(4), 427–434. https://doi.org/10.25126/jtiik.201854773

Hadaina, F., & Budiyanto, U. (2022). Implementasi Metode Multinomial Naïve Bayes Untuk Sentiment Analysis Terhadap Data Ulasan Produk Colearn Pada Google Play Store Implementation Of Multinomial Naive Bayes Method For Sentiment Analysis Of Colearn Product Review Data On Google Play Store. *Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI) Jakarta-Indonesia*, *September*, 660–666. https://senafti.budiluhur.ac.id/index.php

Hassani, H., Beneki, C., Unger, S., Mazinani, M. T., & Yeganegi, M. R. (2020). Text mining in big data analytics. *Big Data and Cognitive Computing*, *4*(1), 1–34. https://doi.org/10.3390/bdcc4010001

Ismail, A. R., & Hakim, R. B. F. (2023). Implementasi Lexicon Based Untuk Analisis Sentimen Dalam Mengetahui Trend Wisata Pantai Di DI Yogyakarta Berdasarkan Data Twitter. *Emerging Statistics and Data Science Journal*, *1*(1), 37–46.

Koto, F., & Rahmaningtyas, G. Y. (2017). Inset lexicon: Evaluation of a word list for Indonesian sentiment analysis in microblogs. *Proceedings of the 2017 International Conference on Asian Language Processing, IALP 2017*, *2018-January*(December), 391–394. https://doi.org/10.1109/IALP.2017.8300625

Latief, I. M., Subekti, A., & Gata, W. (2021). Prediksi Tingkat Pelanggan Churn Pada Perusahaan Telekomunikasi Dengan Algoritma Adaboost. *Jurnal*

Informatika, *21*(1), 34–43. https://doi.org/10.30873/ji.v21i1.2867

Laurensz, B., & Sediyono, E. (2021). Analisis Sentimen Masyarakat terhadap Tindakan Vaksinasi dalam Upaya Mengatasi Pandemi Covid-19 (Analysis of Public Sentiment on Vaccination in Efforts to Overcome the Covid-19 Pandemic). *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, *10*(2), 118–123.

Muhammad Imam Ghozali, Wibowo Harry Sugiharto, A. F. I. (2022). Analisis Sentimen Pinjaman Online Di Media Sosial Twitter Menggunakan Metode Naive Bayes. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, *33*(1), 1–12. https://doi.org/10.30865/klik.v3i6.936

Nihayah, A. Z., Kahrismasuci, I., Chamami, M. R., & Rifqi, L. H. (2023). Edukasi Keuangan Digital dalam Memanfaatkan Jasa Pinjaman Online. *Bubungan Tinggi: Jurnal Pengabdian Masyarakat*, *5*(1), 231. https://doi.org/10.20527/btjpm.v5i1.7325

Owen, D., Groom, Q., Hardisty, A., Leegwater, T., Livermore, L., van Walsum, M., Wijkamp, N., & Spasić, I. (2020). Towards a scientific workflow featuring Natural Language Processing for the digitisation of natural history collections. *Research Ideas and Outcomes*, *6*. https://doi.org/10.3897/rio.6.e58030

Que, V. K. S., Iriani, A., & Purnomo, H. D. (2020). Analisis Sentimen Transportasi Online Menggunakan Support Vector Machine Berbasis Particle Swarm Optimization. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, *9*(2), 162–170. https://doi.org/10.22146/jnteti.v9i2.102

Rabbani, R., Wahidah, I., & Santoso, I. H. (2021). Klasifikasi Data Deteksi Jatuh Menggunakan Machine Learning dengan Algoritma Adaptive Boosting (AdaBoost). *E-Proceeding of Engineering*, *8*(5), 5053–5063.

Sabrani, A., Gede Putu Wirarama Wedashwara, I. W., & Bimantoro, F. (2020). METODE MULTINOMIAL NAÏVE BAYES UNTUK KLASIFIKASI ARTIKEL ONLINE TENTANG GEMPA DI INDONESIA (Multinomial Naïve Bayes Method for Classification of Online Article About Earthquake in Indonesia). *Jtika*, *2*(1), 91–92. http://jtika.if.unram.ac.id/index.php/JTIKA/

Sentia, A. (2023). *MULTINOMIAL NAÏVE BAYES CLASSIFIER UNTUK ANALISIS SENTIMEN*. *December*, 0–8.

Sinaga, R. B., Widiyanto, D., & Wahyono, B. T. (2022). Deteksi Dini Penyakit Kanker Paru dengan Gabungan Algoritma Adaboost dan Random Forest. *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA)*, 1–10. https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer

Supriyanto, E., & Ismawati, N. (2019). Sistem Informasi Fintech Pinjaman Online Berbasis. *Jurnal Sistem Informasi, Teknologi Informasi Dan Komputer*, *9*(2), 100–107.

Syah, F., Fajrin, H., Afif, A. N., Saeputra, M. R., Mirranty, D., & Saputra, D. D. (2023). Analisa Sentimen Terhadap Twitter IndihomeCare Menggunakan Perbandingan Algoritma Smote, Support Vector Machine, AdaBoost dan Particle Swarm Optimization. *Jurnal JTIK (Jurnal Teknologi Informasi Dan Komunikasi)*, *7*(1), 53–58. https://doi.org/10.35870/jtik.v7i1.686

Wahyu, A., Faizi, N., & Nugroho, K. (2023). *Penerapan Metode Adaptive Boosting Pada Analisis Sentimen Kenaikan BBM Pertamina*. *08*(2016), 171–180.

Wati, R., Ernawati, S., & Rachmi, H. (2023). Pembobotan TF-IDF Menggunakan Naïve Bayes pada Sentimen Masyarakat Mengenai Isu Kenaikan BIPIH. *Jurnal Manajemen Informatika (JAMIKA)*, *13*(1), 84–93. https://doi.org/10.34010/jamika.v13i1.9424

Zaki Hariansyah, M. (2022). Implementasi Metode Multinomial Naive Bayes pada Analisis Sentimen Terhadap Layanan Aplikasi Livin by Mandiri Implementation of Naive Bayes Multinomial Method on Sentiment Analysis of Livin by Mandiri Application Services. *Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI) Jakarta-Indonesia*, *September*, 517–524. https://senafti.budiluhur.ac.id/index.php

Zelin Gaa Ngilo, & Nuryuliani Nuryuliani. (2023). Analisis Sentimen Opini Pengguna Twitter Pada Aplikasi Bibit Menggunakan Multinomial Naïve Bayes. *Jurnal Teknik Dan Science*, *2*(1), 08–15. https://doi.org/10.56127/jts.v2i1.521

LAMPIRAN

Lampiran 1 Surat Pengesahan Judul

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
https://fikti.umsu.ac.id fikti@umsu.ac.id umsumedan umsumedan umsumedan umsumedan

UMSU
Unggul | Cerdas | Terpercaya
Bila menjawab surat ini agar disebutkan nomor dan tanggalnya

PERSETUJUAN TOPIK/JUDUL PENELITIAN

Nomor Agenda	: —
Nama	: YOGA PANGESTU
NPM	: 2009010068
Tanggal Persetujuan	: 17 Januari 2024
Topik Yang Disetujui Program Studi	: Data Mining
Nama Dosen Pembimbing	: Mhd. Basri, S.Si., M.Kom
Judul Yang Disetujui Dosen Pembimbing	: Analisa Perbandingan Algoritma Multinomial Naïve Bayes dan Adaboost+ dalam mengklasifikasi Sentiment Masyoraicat terkait Pinjaman Online

Medan 17-9-2024

Disahkan oleh
Ketua Program Studi
Sistem Informasi

Marliono, S.Pd, M.Kom

Persetujuan
Dosen Pembimbing

Mhd Basri, S.Si, M.Kom

Lampiran 2 Surat Penetapan Dosen Pembimbing

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
https://fikti.umsu.ac.id fikti@umsu.ac.id umsumedan umsumedan umsumedan umsumedan

UMSU
Unggul | Cerdas | Terpercaya

PENETAPAN DOSEN PEMBIMBING
PROPOSAL/SKRIPSI MAHASISWA
NOMOR : 16/II.3-AU/UMSU-09/F/2024

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara, berdasarkan Persetujuan permohonan judul penelitian Proposal / Skripsi dari Ketua / Sekretaris.

Program Studi : **Sistem Informasi**
Pada tanggal : **05 Januari 2024**

Dengan ini menetapkan Dosen Pembimbing Proposal / Skripsi Mahasiswa.

Nama : **Yoga Pangestu**
NPM : **2009010068**
Semester : **VII (Tujuh)**
Program studi : **Sistem Informasi**
Judul Proposal / Skripsi : **Analisa Kinerja Algoritma Multinominal NB Dan Adaboost Dalam Menganalisa Sentimen Masyarakat Terkait Pinjaman Online**

Dosen Pembimbing : **Mhd. Basri., S.Si., M.Kom**

Dengan demikian di izinkan menulis Proposal / Skripsi dengan ketentuan

1. Penulisan berpedoman pada buku panduan penulisan Proposal / Skripsi Fakultas Ilmu Komputer dan Teknologi Informasi UMSU
2. Pelaksanaan Sidang Skripsi harus berjarak 3 bulan setelah dikeluarkannya Surat Penetapan Dosen Pembimbing Skripsi.
3. **Proyek Proposal / Skripsi** dinyatakan " **BATAL** " bila tidak selesai sebelum Masa Kadaluarsa tanggal : **05 Januari 2025**
4. Revisi judul……

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Ditetapkan di : Medan
Pada Tanggal : 23 Jumadil Akhir 1445 H
05 Januari 2023 M

Dekan

Dr. Al-Khowarizmi,S.Kom.,M.Kom
NIDN : 0127099201

Cc. File

Lampiran 3 Berita Acara Bimbingan

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
umsumedan umsumedan umsumedan umsumedan

Berita Acara Pembimbingan Proposal

Nama Mahasiswa : YOGA PANGESTU
NPM : 2009010068
Nama Dosen Pembimbing : Mhd. Basri, S.Si, M.Kom

Program Studi : Sistem Informasi
Konsentrasi : Data Mining
Judul Penelitian : Analisa Perbandingan Algoritma Multinominal NB dan Adaboost dalam Mengklasifikasikan Sentiment Masyarakat terkait Pinjaman online.

Tanggal Bimbingan	Hasil Evaluasi	Paraf Dosen
11/01/24	- Revisi Judul - Perbaikan Latar belakang.	
18/01/24	- Perbaikan BAB I. Latarbelakang, Tujuan dan manfaat. - Perbaikan BAB II. Perkaya bahan teori terkait tools dan penelitian terdahulu - Perbaikan BAB III. Tambahkan bagan (flowchart) dari metode yg digunakan.	
31/01/24	- Perbaikan daftar pustaka dan daftar gambar - BAB I. Paragraf 6 terkait kerapian paragraf.	
02/02/24	- BAB 3, Tambahan tentang flowchart Penelitian Kesimpulan dan tabel waktu penelitian - Typografi	
05/02/24	- Perbaikan Tabel waktu penelitian - Perbaikan Bagan Prosedur penelitian	
05/02/24	ACC Proposal	

Medan, 06 Feb 2024

Diketahui oleh :
Ketua Program Studi
Sistem Informasi

(Marlinda, S.Pd., M.Kom)

Disetujui oleh :
Dosen Pembimbing

(Mhd. Basri, S.Si., M.Kom)

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
https://fikti.umsu.ac.id fikti@umsu.ac.id umsumedan umsumedan umsumedan umsumedan
UMSU Unggul | Cerdas | Terpercaya

Berita Acara Pembimbingan Skripsi

Nama Mahasiswa : YOGA PANGESTU
NPM : 2008010060
Nama Dosen Pembimbing : Mhd. Basri, S.Si, M.Kom
Program Studi : Sistem Informasi
Konsentrasi : Data mining
Judul Penelitian : Analisa Perbandingan Algoritma Multinominal Nb dan Adaboost dalam mengklasifikasikan Sentimen Pinjol.

Item	Hasil Evaluasi	Tanggal	Paraf Dosen
	- Revisi Pasca Sempro - Perbaikan Bab III	19/03/24	
	- Bab IV - Desain sistem	25/03/24	
	- Revisi Bab IV dan Bab V - Perbaikan sistem	30/03/24	
	- Perhitungan manual - Bab V	10/04/24	
	- Perbaikan Typo dan penulisan	17/04/24	
	- Bimbingan Jurnal.	22/04/24	
	Acc Sidang	25/04/24	

Medan, 25 - 04 - 2024

Diketahui oleh :
Ketua Program Studi
Sistem Informasi

Marliano, S.Pd., M.Kom

Disetujui oleh :
Dosen Pembimbing

Mhd. Basri, S.Si, M.Kom

Lampiran 4 LoA Penerbitan Jurnal

Journal of Computer Engineering, System and Science
Jl. William Iskandar Ps. V Medan Estate - Sumatera Utara - Indonesia 20221
Email: journal_cess@unimed.ac.id
Home Page: https://jurnal.unimed.ac.id/2012/index.php/cess
ISSN: 2502-7131 (Print) | ISSN: 2502-714x (Online)

LETTER of ACCEPTANCE (LoA)

Kepada **Penulis (Author)**

Redaksi CESS (Journal of Computer Engineering, System and Science) menerangkan bahwa artikel dengan rincian berikut:

Judul	:	**Analisis Perbandingan Multinominal Naïve Bayes dan Adaboost dalam Mengklasifikasikan Sentimen Terkait Pinjaman Online**
Title	:	***Comparative Analysis of Multinominal Naïve Bayes and Adaboost in Classifying Sentiment Related to Online Loans***
Penulis	:	**1. Yoga Pangestu** **2. Muhammad Basri**
Institusi	:	**Universitas Muhammadiyah Sumatera Utara**

Dinyatakan **diterima** karena telah memenuhi kriteria publikasi di Jurnal CESS (Journal of Computer Engineering, System and Science) pada **Volume 9 Nomor 2 Juli 2024.**

Demikianlah surat ini disampaikan, atas partisipasinya kami ucapkan terimakasih.

Medan, 12 Juni 2024
Editor in Chief

Mohamad Ihwani

GARUDA Google scholar Dimensions sinta 4 doi Crossref

Lampiran 6 Source Code Sistem

```
import pandas as pd
df = pd.read_csv('/content/data_pinjol.csv', encoding='latin-
1')
df.head()
df.info()
#Memfilter kolom yang diperlukan
df = df.filter(['full_text'])
df.head()
#Melihat kata-kata yang sering muncul
import matplotlib.pyplot as plt
from collections import Counter

text = " ".join(df ["full_text"])

tokens = text.split()
word_counts = Counter(tokens)

top_words = word_counts.most_common(10)
word, count = zip(*top_words)

colors = plt.cm.Paired(range(len(word)))

plt.figure(figsize=(10,6))
bars = plt.bar(word, count, color=colors)
plt.xlabel("Kata")
plt.ylabel("Frekuensi")
plt.title("Kata-kata yang sering mucul")
plt.xticks(rotation=45)

for bar, num in zip(bars, count):
  plt.text(bar.get_x() + bar.get_width() / 2 - 0.1, num + 1,
           str(num), fontsize= 12, color='black', ha='center')
plt.show()
```

```
df['case_folding'] = df['full_text'].str.lower()
import string
import re
def cleaning(komentar):
    #remove ascii
    komentar = komentar.encode('ascii',
'replace').decode('ascii')

    #remove angka
    komentar = re.sub('[0-9]+', '', komentar)
```

```
    #remove mention, link, hashtag
    komentar = re.sub('<br.*?>(.+?)</br>',' ',komentar)
    komentar = ' '.join(re.sub("([@#][A-Za-z0-
9]+)|(\w+:\/\/\S+)"," ", komentar).split())
    komentar = re.sub('@[^\s]+','',komentar)

    #remove url
    komentar = re.sub(r'\w+:\/{2}[\d\w-]+(\.[\d\w-
]+)*(?:(?:\/[^\s/]*))*', '', komentar)

    #remove tanda baca
    komentar = re.sub(r'[^\w\d\s]+', '', komentar)

    #remove whitespace
    komentar = re.sub('\s+',' ',komentar)

    #remove line baru
    komentar = re.sub('\n',' ',komentar)

    #remove garis bawah
    komentar = re.sub('_',' ',komentar)

    return komentar
df['cleaning'] = df['case_folding'].apply(cleaning)
# menghapus data duplikat
df.drop_duplicates(subset ="cleaning", keep = 'first', inplace
= True)
df.head()
```

```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
from nltk.tokenize import word_tokenize
def word_tokenize_wrapper(cleaning):
  return word_tokenize(cleaning)
df['tokenize'] = df['cleaning'].apply(word_tokenize_wrapper)
df.head()
```

```
from nltk.corpus import stopwords
list_stopwords = stopwords.words('indonesian')
#tambahkan stopword manual
list_stopwords.extend(['tu', 'uf',
'deh','nak','amp','b','a','w','je','jd','x','sih',
```

```
                        'dos', 'haa', 'eh', 'tuh',
'hmm','grgr', 'nya','ufufuf', 'lho',
                        'rm', 'nya', 'ufcc', 'ppv','dok', 'je',
'gb', 'pa', 'e', 'br',
                        'ya', 'yg', 'di',
'ga','lu','noh','thx','gpp','hehehe','dll'])
sw = set(list_stopwords) - set (['jangan', 'jangankan',
'janganlah', 'tidak', 'tidakkah', 'tidaklah'])
def stopwords_removal(words):
  return [word for word in words if word not in sw]

df['stopword'] = df['tokenize'].apply(stopwords_removal)
df.head()
```

```
normalisasi_kata = pd.read_excel("kamuskatabaku.xlsx")

normalisasi_kata_dict = {}
for index, row in normalisasi_kata.iterrows():
  if row[0] not in normalisasi_kata_dict:
    normalisasi_kata_dict[row[0]] = row[1]

def normalisasi_term(document):
  return [normalisasi_kata_dict[term]
  if term in normalisasi_kata_dict
          else term
          for term in document]

df['normalisasi'] = df['stopword'].apply(normalisasi_term)
df.head()
```

```
!pip install PySastrawi

import Sastrawi
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()
def stemmed_wrapper(term):
  return stemmer.stem(term)
df['stemming'] = df['normalisasi'].apply(lambda x:
[stemmer.stem(y) for y in x])
df.head()
```

```
import string
def remove_punct(text):
  text = " ".join([char for char in text if char not in
string.punctuation])
  return text
df['clean'] = df['stemming'].apply(lambda x: remove_punct(x))
df.head()
```

```
# Membuat class untuk membaca kamus lexicon positif dan
negatif
lexicon_positive = dict()
with open("/content/Kamus_Positif.csv", "r") as csvfile:
  reader = csv.reader(csvfile, delimiter=",")
  for row in reader:
    lexicon_positive[row[0]] = int(row[1])


lexicon_negative = dict()
with open("/content/Kamus_Negatif.csv", "r") as csvfile:
  reader = csv.reader(csvfile, delimiter=",")
  for row in reader:
    lexicon_negative[row[0]] = int(row[1])


# Membuat fungsi untuk mempolarisasi sentimen berdasarkan
kamus lexicon yang ada
def sentiment_analysis(text):
  score = 0
  for word in text:
    if (word in lexicon_positive):
      score = score + lexicon_positive[word]
  for word in text:
    if (word in lexicon_negative):
      score = score + lexicon_negative[word]
  polarity = ''
  if (score > 0 ):
    polarity = "Positif"
  elif (score < 0 ):
    polarity = "Negatif"
  else :
    polarity = "Netral"
  return score, polarity
result = df['stemming'].apply(sentiment_analysis)
result = list(zip(*result))
df["polarity_score"] = result[0]
df["sentiment"] = result[1]
```

```
print(df["sentiment"].value_counts())
df.shape
```

```
df = df.filter(['clean', 'sentiment'])
df.head()
X = df['clean']
y = df['sentiment']
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import accuracy_score,
classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
X_train, X_test, y_train, y_test =
train_test_split(df['clean'], df['sentiment'], test_size=0.2,
random_state=34)
```

```
vectorizer = TfidfVectorizer(strip_accents='ascii')
tf_idf_train = vectorizer.fit_transform(X_train)
tf_idf_test = vectorizer.fit_transform(X_test)
```

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import AdaBoostClassifier
```

```
model_1 = MultinomialNB()
model_1.fit(tf_idf_train, y_train)
model_2 = AdaBoostClassifier()
model_2.fit(tf_idf_train, y_train)
```

```
# Evaluasi Model menggunakan data latih
predictions_train = model_1.predict(tf_idf_train)

# Akurasi
accuracy_train = accuracy_score(y_train, predictions_train)
print(f'Accuracy on Training Data: {accuracy_train:.2f}')

# Report klasifikasi
```

```
print('\nClassification Report on Training Data:\n',
classification_report(y_train, predictions_train))

conf_matrix_train = confusion_matrix(y_train,
predictions_train)

# Plot confusion matrix untuk data latih
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_train, annot=True, fmt='d',
cmap='Blues',
            xticklabels=['Positif', 'Netral', 'Negatif'],
yticklabels=['Positif', 'Netral', 'Negatif'])
plt.title('Confusion Matrix on Training Data')
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.show()
```

```
# Evaluasi Model menggunakan data latih
predictions_train = model_2.predict(tf_idf_train)

# Akurasi
accuracy_train = accuracy_score(y_train, predictions_train)
print(f'Accuracy on Training Data: {accuracy_train:.2f}')

# Report klasifikasi
print('\nClassification Report on Training Data:\n',
classification_report(y_train, predictions_train))

conf_matrix_train = confusion_matrix(y_train,
predictions_train)

# Plot confusion matrix untuk data latih
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_train, annot=True, fmt='d',
cmap='Blues',
            xticklabels=['Positif', 'Netral', 'Negatif'],
yticklabels=['Negatif', 'Netral', 'Positif'])
plt.title('Confusion Matrix on Training Data')
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.show()
```