BAB II
TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Tahapan utama yang dilakukan yaitu eksplorasi literatur yang membahas penggunaan normalisasi kata *slang* pada langkah awal *text preprocessing.* Berikut ini beberapa penelitian terdahulu yang membahas penerapan normasilasi kata *slang* dalam klasifikasi sentimen.

Penelitian [5] bertujuan untuk mengimplementasikan proses normalisasi terhadap kata *slang* pada tahap *preprocessing* menggunakan dataset dari komentar youtube terhadap aturan keringanan pembayaran listrik selama pandemi COVID-19. Hasil akurasi tertinggi didapatkan apabila diterapkan tahapan *preprocessing* secara lengkap yaitu proses *preprocessing* yang menjadi standar, penghapusan *stop words*, normalisasi dengan pengubahan kata *slang* menjadi kata baku, dan penghapusan kaya yang berbentuk subjek atau objek. Akurasi tertinggi sebesar 88% didapatkan dengan menggunaka ekstraksi fitur *Count – Vectorizer* dengan uni-gram dan menggunakan algoritma *ensemble* yaitu *Extra Trees classifier.*

Penelitian [3] juga melakukan penelitian untuk membandingkan hasil akurasi model klasifikasi dengan normalisasi dan tanpa normalisasi. Penelitian ini menggunakan data *tweet* tentang *gadget* dengan menerapkan ekstraksi fitur *word2vec*. Berbeda dari penelitian [5] , algoritma yang digunakan yaitu algoritma *naive bayes* dengan hasil akurasi yang didapat 91 % setelah dilakukan normalisasi sedangkan tanpa normalisasi akurasi sebesar 88%.

Selain penelitian[3], [5] penelitian [9] juga membahas penerapan normalisasi kata *slang.* Penelitian ini mengambil studi kasus terkait tanggapan public terhadap layanan teman bus. Dalam mengoreksi ejaan kata yang tidak formal, penelitian tersebut memanfaatkan jarak *Levenshtein* dan diterapkan setelah proses *tokenizing* dan *stemming.* Hasil akurasi terbaik tercapai setelah melakukan normalisasi dengan memanfaatkan jarak *Levenshtein* untuk mengoreksi ejaan kata dan untuk konversi kata *slang* menggunakan *min-count word2vec* yaitu masing – masing 0,9 dan 10.

Penggunaan *levenshtein distance* dalam normalisasi kata baku juga dilakukan oleh Penelitian [2]. Dataset yang digunakan merupakan data *tweet* tentang PT KAI yang terdiri dari 450 data latih dan 100 data uji dan algoritma yang digunakan untuk analisis sentimen ini yaitu *Naïve Bayes*. Hasil akurai terbaik dalam analisis sentimen Twitter Pt KAI didapatkan setelah menerapkan normalisasi *Levenshtein Distance* untuk perbaikan kata tidak baku didapatkan akurasi sebesar 67,05 %. Kelemahan dari penelitian tersebut akurasi yang didapatkan rendah walaupun sudah dilakukan normalisasi kata baku.

Penelitian [10] lain yang memfokuskan untuk meningkatkan akurasi analisis sentimen dengan melakukan *preprocessing* yang melibatkan perluasan akronim, terjemahan kata *slang*, dan translasi emoji pernah dilakukan. Hasil yang diperoleh menunjukkan bahwa kombinasi dari ketiga metode *preprocessing* berhasil meningkatkan akurasi model secara signifikan, dengan skor F1 mencapai 83,362%.

Selain metode normalisasi, urutan dalam tahap *preprocessing* pernah dibahas dalam penelitian [11]. Penelitian ini menggunakan data *tweet* tentang gangguan mental, algoritma *naïve bayes*. Kamus *slang* yang digunakan dibuat secara manual dan diperoleh sebanyak 250 kata *slang.* Permasalan yang ada dalam penelitian tersebut yaitu bagaiman urutan *preprocesing* pada proses normalisasi, *stop word*, dan *stemming* sebelum ekstraksi fitur. Hasil akurasi terbaik didapatkan apabila melakukan tahapan *preprocessing* secara full dengan urutan melakukan normalisasi terlebih dahulu sebelum tahapan *stemming* yaitu sebesar 89,2 %.

Hasil analisis yang berbeda didapatkan pada penelitian [12] dalam hal penanganan noise. Dalam penelitian ini akurasi terbaik didapatkan pada skenario dengan menerapkan internet secara keseluruhan yaitu sebesar 47.65 % sebelum normalisasi dan 47.15 % setelah dilakukan normlisasi. Penurunan tingkat akurasi disebabkan oleh kesalahan dalam proses normalisasi yang belum optimal. Penelitian ini melakukan tiga skenario dalam mendapatkan akurasi yaitu dengan memanfaatkan daftar lengkap *internet slang* keseluruhan, *internet slang* bahasa Indonesia yang bersifat umum dan *internet slang* dengan *domain kosmetik*.

Penelitian lain yang menghadapi tantangan dalam menangani sentimen, terutama teks yang mengandung sarkasme juga pernah dilakukan [13]. Metode

yang diterapkan meliputi pengambilan fitur dengan menggunakan *unigram* dan empat *set fitur Boazizi*, dengan deteksi *sarkasme* menggunakan algoritma *Random Forest*. Selain itu, proses analisis sentimen juga melibatkan ekstraksi fitur TF-IDF dan penerapan algoritma klasifikasi *Naïve Bayes*. Hasil penelitian menunjukkan bahwa dengan menerapkan deteksi sarkasme, termasuk normalisasi kata *slang*, akurasi analisis sentimen meningkat sebesar 5,49%, mencapai tingkat akurasi tertinggi sebesar 80,4%.

Penelitian [14] memiliki hasil pengujian kinerja data yang tidak melalui normalisasi dan kinerja data dengan normalisasi tidak memiliki perbedaan yang signifikan berdasarkan *confution matriknya*. Data yang digunakan sebanyak delapan yang bersumber dari *tweet* dengan *keyword* vaksin dan Indonesia, Selain itu juga dari skripsi dan github dan algoritma *naïve bayes classifier model multivariate*. Penelitian ini menemukan bahwa dalam hasil normalisasi, kata-kata yang lebih sering muncul adalah kata-kata *stopword*, sehingga kehadiran kata-kata ini tidak memiliki dampak yang signifikan terhadap akurasi hasil.

Penelitian lain yang bertujuan untuk membuat kamus *slang word* dilakukan pada tahun 2022 [15]. Studi ini akan mengembangkan Kamus *Slang* untuk mengnormalisasi teks menggunakan model *Pre-Trained FastText*. Penelitian ini didasari oleh kesulitan dalam menganalisis sentimen teks Ketika kata – kata tersebut tidak baku. Word embedding menggunakan *pre-trained FastText* dengan model CBOW*,* dihasilkan akurasi sebesar 0.65 didapatkan dengan Threshold value 0.05 untuk 100 kata *slang* dan normal, sementara threshold 0.1 sebanyak 99 kata *slang* dan normal, dan 0.2. sebanyak 79 kata *slang* dan normal.

Berdasarkan penelitian terdahulu, didapatkan bahwa normalisasi dengan penambahan kamus *slang* dalam tahap preprocessing dapat meningkatkan akurasi dalam klasifikasi sentimen dalam beberapa penelitian di atas, Sedangkan penelitian yang mempunyai hasil yang berbanding terbalik dikarenakan tahap normalisasi yang belum tepat. Penelitian kali ini akan memfokuskan membangun kamus *slang word.* Perbedaan penelitian ini yaitu dari topik dataset yang berbeda dimana data yang akan digunakan bersumber dari *github*, *Kaggle* dan algoritma yang digunakan yaitu *Naïve Bayes Classifier* dan *Decision Tree Classifier.*

Tabel 2. 1 Penelitian Terdahulu

No	Penulis & Tahun	Judul Penelitian	Masalah Penelitian	Tujuan Penelitian	Metode	Data	Hasil Penelitian	Perbedaan Penelitian
1.	Siti Khomsah, Agus Sasmito Ariwibowo (2021)	Model Text-*preprocessing* Komentar Youtube Dalam Bahasa Indonesia	Bagaimana hasil akurasi jika model *preprocessing* diterapkan semua?	Menerapkan berbagai model *preprocessing* dan membandingkan hasil akurasi analisis sentimen	Algoritma *Extra Trees Clasifier*	Komentar Youtube pada video yang membahas kebijakan pemerintah Indonesia terkait penyediaan listrik gratis selama masa *COVID-19*	Hasil akurasi terbaik sebesar 88 % diperoleh dengan menerapkan *preprocessing yang menjadi* standar, menghapus *stop words*, pengubahan kata *slang* ke dalam kata tidak formal , dan menghapus kata dalam bentuk subjek dan objek	Data *review* Lazada & Tokopedia, Algoritma *Naive Bayes dan Decision Tree*
2.	Zhafirah Rizqy, Ade Romadhony, Erwin Budi Setiawan (2022)	Analisis Pengaruh Normalisasi Teks pada Klasifikasi Sentimen Ulasan Produk Kecantikan	Pengaruh hasil normalisasi teks terhadap keakuratan hasil sentimen	Menganalisis pengaruh hasil normalisasi dengan penggunaan *internet slang* dan *speling correction* terhadap hasil akurasi	Algoritma *Multinomial Naïve Bayes*	Ulasan produk kecantikan	Hasil Akurasi sebelum normalisasi teks sebsar 47.69 % lebih tinggi dibandingkan dengan dilakukan normalisasi sebesar 47.15 %.	Data *review* Lazada & Tokopedia, Algoritma *Decision Tree*

No	Penulis & Tahun	Judul Penelitian	Masalah Penelitian	Tujuan Penelitian	Metode	Data	Hasil Penelitian	Perbedaan
3.	Riri Riyaddulloh, Ade Romadhony (2021)	Normalisasi Teks Bahasa Indonesia Berbasis Kamus *Slang* (Studi Kasus : *Tweet* Produk *Gadget* Pada Twitter)	Bagaimana melakukan normalisasi kata non baku untuk meningkatkan akurasi model	Melakukan normalisasi kata pada dataset dan menganalisa pengaruhnya terhadap akurasi model klasifikasi	Algoritma *Naïve Bayes* dengan normalisai *Word2vec*	Data Twitter konteks produk *gadget*	Hasil akurasi dengan normalisasi lebih tinggi sebesr 91 % dibandingkan tanpa normalisasi sebesar 88 %	Vectorisasi TF-IDF, Data *review* Lazada & Tokopedia, Algoritma *Decision Tree*
4.	Sheila Sheviraa, I Made Agus Dwi Suarjaya, Putu Wira Buana (2022)	Pengaruh Kombinasi dan Urutan *Pre-Processing* pada *Tweets* Bahasa Indonesia	Bagaimana urutan dan kombinasi *preprocessing* yang tepat untuk mendapatkan akurasi terbaik	Melihat pengaruh urutan dan kombinasi tahap *preprocessing* sebelum ekstraksi fitur dalam meningkatkan akurasi model klasifikasi.	Algoritma *Naïve Bayes*	Data Twitter konteks gangguan mental	Akurasi lebih tinggi dilakukan dengan full *prepcessing* dengan urutan normalisasi terlebih dahulu sebelum stemming yaitu sebesar 89,2 %. Normalisasi dilakukan membuat kamus *slang* manual sebanyak 250 kata.	Data *review* Lazada & Tokopedia, Algoritma *Decision Tree*

No	Penulis & Tahun	Judul Penelitian	Masalah Penelitian	Tujuan Penelitian	Metode	Data	Hasil Penelitian	Perbedaan
5.	M.Adnan Nur, Nurilmiyanti Wardhani (2022)	Optimasi Normalisasi Kata Pada Data Twitter Untuk Meningkatkan Akurasi Analisis Sentimen (Studi Kasu Respon Masyarakat Terhadap Layanan Teman Bus)	Pengaruh penerapan normalisasi kata dengan metode *Levenshtein distance dan* konversi *Slang word* dalam meningkat akurasi model klasifikasi	Menerapkan normalisasi kata dalam tahap preprocessing dalam meningkatkan akurasi	Algoritma *Levenshtein distance* untuk koreksi kesalahaan ejaan kata dan konversi *slang* word menggunakan *word2vec.* Algoritma *Naïve Bayes* untuk klasifikasi	Data Twitter konteks teman bus	Akurasi tertinggi sebesar 0,776 didapatkan setelah menerapkan normalisasi kaya dengan ratio jarak *Levenshtein* sebesar 0,9 dan *word2vec* sebesar 10	Normalisasi kata dengan kamus *slang,* Data *review* Lazada & Tokopedia, Algoritma *Decision Tree*
6.	Imam Fahrur Rozi, Rizky Ardiansyah, Naomi Rebeka (2019)	Penerapan Normalisasi Kata Tidak Baku Menggunakan Levenshtein Distance pada Analisa Sentimen Layanan PT. KAI di Twitter	Bagaimana penerapan normalisasi kata dengan *Levenshtein distance* dalam tahap preprocessing dapat meningkatkan akurasi	Menganilisis hasil akurasi setelah diterapkan normalisasi kata dengan *Levenshtein distance* pada tahap *preprocessing*	Algoritma *Levenshtein distance* untuk koreksi kesalahaan ejaan kata. Algoritma *Naïve Bayes* untuk klasifikasi	Data tweet masyarakat terkait layanagn Pt. KAI	Hasil akurasi terbaik didapatkan setelah menerapkan normalisasi dengan *Levenshtein distance* dan algoritma klasifikasi *Naïve Bayes* sebesar 67,05 %	Normalisasi kata dengan kamus *slang,* Data *review* Lazada & Tokopedia, Algoritma *Decision Tree*

No	Penulis & Tahun	Judul Penelitian	Masalah Penelitian	Tujuan Penelitian	Metode	Data	Hasil Penelitian	Perbedaan
7.	Firnanda Zuhad, Nori Wilantika(2022)	Perbandingan Penggunaan Kamus Normalisasi dalam Analisis Sentimen Berbahasa Indonesia	Bagaimana hasil akurasi analisis sentimen jika diterapkan proses normalisasi kata tidak baku	Membandingkan performa data dalam anaisis sentimen dengan normalisasi dan tanpa normalisasi	Algoritma *naïve bayes classifier model multivariate*	Data *Tweet* dengan keyword vaksin dan Indonesia, Selain itu juga data yang dgunakan juga dari skripsi dan github	Dari total delapan dataset yang digunakan, hasil kinerja data yang tidak dilakukan normalisasi dan kinerja data dengan normalisasi tidak memiliki perbedaan yang signifikan berdasarkan *convution matriknya*	Data *review* Lazada & Tokopedia, Algoritma *Decision Tree*
8.	Fatihah Rahmadayana, Yuliant Sibaron (2021)	Sentiment Analysis of Work from Home Activity using SVM with Randomized Search Optimization	Bagaimana normalisasi kata *slang* mempengaruhi akurasi Evaluasi pendapat masyarakat terhadap kebijakan *Work from Home* di platform media sosial *Twitter*.	Mengevaluasi pengaruh normalisasi kata *slang* dalam *preprocessing* terhadap peningkatan akurasi analisis sentimen menggunakan Support Vector Machine.	Metode *preprocessing*, yaitu perluasan akronim, terjemahan kata *slang*, dan translasi emoji dengan model *Support Vector Machine*	Data *Tweet* opini kebijakan *Work from Home*	Penerapan ketiga metode *preprocessing* menghasilkan peningkatan yang signifikan dengan diperolehnya nilai Skor F1 terbaik sebesar 83,362%.	Data *review* Lazada & Tokopedia, Algoritma *Decision Tree*

No	Penulis & Tahun	Judul Penelitian	Masalah Penelitian	Tujuan Penelitian	Metode	Data	Hasil Penelitian	Perbedaan
9.	Yessi Yunitasari, Aina Musdholifah, and Anny Kartika Sari (2019)	*Sarcasm Detection for Sentiment Analysis in Indonesian Tweets*	Kesulitan dalam menentukan sentimen yang akurat dari *tweet* berbahasa Indonesia yang mengandung sarkasme	Meningkatkan keakuratan analisis sentimen dengan mengintegrasikan deteksi sarkasme, termasuk normalisasi kata *slang* dalam proses *preprocessing*.	Deteksi sarkasme dijalankan menggunakan algoritma *Random Forest*, dan klasifikasi menggunakan algoritma *Naïve Bayes*	Data *Tweet* dengan *hashtag* yang mengandung sarkasme.	Penerapan deteksi sarkasme, termasuk normalisasi *slang*, keakuratan analisis sentimen meningkat sebesar 5,49%, mencapai akurasi maksimal 80,4%.	Normalisasi kata dengan kamus *slang,* Data *review* Lazada & Tokopedia, Algoritma *Decision Tree*
10.	Lavenia Situmorang, Enjelin Hutahaean, Ruth Angeli Sibarani Junita Amalia (2022)	Membangun *Slang* Dictionary Untuk Normalisasi Teks Menggunakan Pre-Trained Fasttext Model	Kata *slang* yang terdapat pada data opini yang menyulitkan dalam klasifikasi teks sehingga perlu dilakukan representasi ulang menjadi kata normal	Membuat kamus *slang* untuk menormalisasi teks menggunakan model *FastText* yang telah dilatih sebelumnya.	Word embedding menggunakan *pre-trained FastText* dengan model CBOW	Data Komentar Youtube dari 10 akun *subscriber* tebanyak	Hasil akurasi sebesar 0.65 didapatkan dengan Threshold value 0.05 dihasilkan sebanyak 100 kata *slang* dan normal, dengan threshold 0.1 sebanyak 99 kata *slang* dan normal, dan 0.2. sebanyak 79 kata *slang* dan normal.	Membuat kamus *slang* dengan kombinasi kamus yang sudah ada dan kata *slang* yang terdapat pada data

T

2.2 Landasan Teori

2.2.1 Kata *Slang*

Kata *slang* merupakan kata informal yang sering digunakan dalam percakapan sehari – hari. Bentuk kata *slang* terus berkembang seiring dengan berkembangnya kehidupan budaya dan sosial. Menurut penelitian [16], variasi bahasa *slang* adalah bentuk komunikasi yang dikembangkan oleh kelompok tertentu, dan hanya dimengerti serta digunakan oleh anggota kelompok tersebut. Penggunaan bahasa *slang* tidak terbatas pada percakapan langsung, tetapi juga lazim ditemukan dalam komunikasi di media sosial. Penggunaan kata *slang* atau bahasa gaul lebih dominan digunakan oleh para remaja di media sosial. Hal ini karena bahasa gaul dipengaruhi oleh tren dan popularitas. Ragam bahasa *slang* yang digunakan dalam berkomunikasi di media sosial mencakup penggunaa kata – kata yang disingkat, akronim, plesetan (kata baru), perubaan letak huruf kata umum dan berbagai bentuk lainnya [16]. Berikut contoh penggunaan kata *slang* yang terdapat pada media sosial.

1. Kata singkatan:
   ‘bgt”: singkatan kata banget
   “bgs”: singkatan kata bagus
   “mantul”: singkatan kata mantap betul
2. Plesetan

   Pada kalimat *review* “produknya lucu beud” terdapat kata *slang* dalam bentuk plesetan yaitu kata “beud”. Kata *slang* tersebut memiliki arti “banget”.
3. Perubahan letak huruf

   Kata “asik” menjadi ‘sika” dengan pertukaran huruf a dan i sebagai bentuk variasi dalam penggunaan kata *slang*.
4. Pengulangan karakter

   Kalimat *review* “basonya enak sekali mantappppp”. Pada kalaimat tersebut terdapat kata *slang* dengan pengulangan karakter “p” pada kata”mantap”.

2.2.2 Natural Language Processing

Natural Language Processing atau pemrosesan bahasa alami merupakan cabang ilmu komputer dan kecerdasan buatan yang berfokus pada pemahaman dan analisis bahasa manusia oleh komputer. Penggunaan pemrosesan bahasa alami (NLP) semakin populer dan berkembang pesat dalam berbagai bidang, mulai dari yang terkait erat seperti semantik (makna dalam bahasa) dan linguistik (bahasa manusia), hingga bidang yang lebih jauh seperti *biobliometri*, keamanan *Cyber*, mekanika kuantum, studi gender, kimia, dan *ortodonti* [17]. Dalam buku [18] peran utama analisis teks dan pemrosesan bahasa alami (NLP) yaitu mengekstraksi informasi dari data bahasa yang ada. Ini dicapai melalui pendekatan seperti segmentasi kalimat atau pemberian label bagi bagian-bagian kalimat berdasarkan peran gramatikalnya.

2.2.3 Analisis Sentimen

Analisis sentimen merupakan aspek dari data mining yang mengambil informasi dari data teks yang berupa pendapat, evaluasi, sikap, emosi, seseorang terhadap suatu objek[19]. Analisis sentimen biasanya menghasilkan dua kategori kelas yaitu positif dan negatif. Beberapa penelitian juga mengadopsi tiga kategori kelas yaitu kelas negatif, positif, dan netral.

2.2.4 *Exploratory Data Analysis* (EDA)

Analisis data exploratori merupakan tahap awal yang digunakan untuk menggali struktur dan karakteristik dari data tersebut. *EDA* ini memiliki tujuan untuk menemukan wawasan dan informasi penting yang terdapat dalam data sebelum melakukan analisis lebih lanjut atau membangun model. Pentingnya *EDA* terletak pada kemampuannya membantu peneliti atau ilmuwan data merumuskan pertanyaan yang relevan, memastikan kualitas data, dan mengambil keputusan tentang pendekatan analisis yang akan dilakukan selanjutnya. Untuk mengekplorasi data yang belum dikenal maka para ilmuwan data secara aktif melakukan serangkaian operasi analisis seperti *filtering*, agregasi dan *filtering* [20].

2.2.5 IndoBERT

Pelabelan data sentimen bisa dilakukan dengan otomatis yaitu dengan *IndoBERT. IndoBERT* merupakan teknik *pre-trained model berdasarkan bert* yang bertujuan untuk menciptakan representasi teks Indonesia yang tidak hanya akurat tapi juga mendalam secara linguistik. *Bidirectional Encoder Representations from Transformer (*BERT) adalah model yang telah dilatih sebelumnya (*pre-trained*) dari data teks yang tidak berlabel. Model ini, berdasarkan arsitektur *BERT*, dilatih menggunakan korpus besar dari Indonesia, dikenal sebagai korpus Indo4B. Struktur dasar *IndoBERT* mirip dengan *BERT*, yang meliputi rangkaian *enkoder Transformer* [21]. Proses pelatihan *BERT* dilakukan secara bidireksional dengan menggabungkan konteks dari kedua arah (kiri dan kanan) dalam berbagai lapisan (layer) model[22]. Metode ini memanfaatkan model yang sudah mengalami latihan sebelumnya dan hanya perlu sedikit penyesuaian tambahan untuk mencapai performa optimal pada tugas yang baru. Model tersebut sudah dijadikan bahan latihan menggunakan data sebanyak 4 miliar kata yang mencakup sekitar 250 juta kalimat formal dan sehari-hari dalam Bahasa Indonesia [23].

2.2.6 Text Preprocessing

Penelitian analisis yang menggunakan data harus memgikuti proses analisis pada data yang biasa disebut dengan tahapan prepocessing data. *Prepocessing* data dapat mempermudah isi teks dalam dokumen yang tidak terstruktur, memiliki banyak gangguan, dan memiliki struktur teks yang kurang optimal. Dalam Analisis sentiment *preprocessing* data memiliki beberapa tahapan untuk membersihkan atau cleaning data diantara yaitu *Case Folding, Filtering, Tokenizing, Stemming*[24]. Tahapan *preprocessing* teks yang umum digunakan meliputi *case folding, tokenizing, penghapusan stop words*, dan *stemming* [5] dan penambahan tapana Normalisasi juga dilakukan dalam penelitian tersebut.

1. *Case Folding* merupakan tahapan untuk mengkonversi huruf *uppercase* menjadi huruf *lowercase.* Tujuan dari case folding adalah untuk Menyamakan tampilan antara huruf kapital dan huruf kecil dalam teks., sehingga kata yang

sebenarnya memiliki makna yang sama akan dianggap sama meskipun ditulis dalam bentuk huruf besar atau huruf kecil

2. *Filtering* merupakan proses pembersihan kata – kata dari noise yang dapat mengganggu dalam klasifikasi sentimen. Beberapa *noise* yang harus *filtering* yait*u emoticon, url, hastag, punction, numerical* dan lain – lain. Dengan melakukan *filtering*, teks menjadi lebih bersih dan lebih mudah diolah oleh model analisis sentimen, sehingga menghasilkan hasil analisis yang lebih akurat dan informatif.
3. *Tokenizing* merupakan proses untuk memisah kalimat menjadi unit terkecil. Dalam analisis sentimen sebuah kalimat bila dilakukan tokenizing menjadi unit tercilnya yaitu kata. Tujuan utama dari tokenizing adalah untuk membagi teks menjadi unit-unit yang lebih kecil agar dapat diolah lebih lanjut dalam analisis teks, pemrosesan bahasa alami, atau pembentukan model bahasa.
4. *Stopword Removal* merupakan proses untuk langkah untuk menghilangkan kata-kata umum yang tidak memiliki emosi atau sentimen seperti kata dan, di, untuk dan lainnya.
5. Normalisasi adalah proses mengubah kata-kata non-baku atau tidak konvensional menjadi bentuk yang sesuai dengan ketentuan bahasa dalam Kamus Besar Bahasa Indonesia (KBBI). Pada tahap ini, gaya penulisan setiap kata disesuaikan dengan standar yang ditetapkan oleh KBBI. Proses ini melibatkan pemadanan setiap kata dalam teks dengan *entri* kata-kata yang terdapat dalam kamus.

2.2.7 Feature Extraction

Proses *feature extraction* adalah proses mengubah data asli menjadi representasi fitur numerik yang lebih ringkas dan berarti. Ini merupakan langkah kunci dalam pemrosesan bahasa alami untuk menghasilkan vektor angka dari teks. *TF-IDF (Term Frequency-Inverse Document Frequency)* adalah sebuah teknik pembobotan yang mengkombinasikan frekuensi kemunculan kata dalam suatu dokumen *(term frequency)* dengan *inversi* frekuensi kemunculan kata tersebut dalam seluruh kumpulan dokumen (*inverse document frequency*) [25]. Metode TF-IDF digunakan untuk melakukan seleksi fitur sebagai hasil ringkasan, dengan

menerapkan pembobotan kata untuk menentukan bobot relatif dari kata-kata dalam dokumen. Rumus TF -IDF dapat dilihat pada persamaan (2.1)

$$tf_{i,j} = \frac{n_{i,j}}{\sum n_{i,j}} \quad (2.1)$$

Dimana:

$tf_{i,j}$: Jumlah frekuensi *term*

$n_{i,j}$: Jumlah *term* i dalam dokumn j

IDF adalah ukuran kemampuan term untuk membedakan kategori. Nilai IDF dilihat pada persamaan (2.2).

$$idf_i = log \frac{N}{df_i} \quad (2.2)$$

Dimana:

N : Jumlah dokumen

df_i : Jumlah dokumen yang mengandung termfi

Bobot TF-IDF merupakan hasil dari perkalian antara bobot frekuensi term (*TF - Term Frequency*) dengan invers frekuensi dokumen term (*IDF - Inverse Document Frequency*). Dalam pendekatan TF-IDF, bobot TF mengindikasikan seberapa sering suatu term muncul dalam suatu dokumen tertentu, sedangkan IDF mengevaluasi tingkat signifikansi term tersebut dalam keseluruhan kumpulan dokumen. Bobot TF-IDF dapat dilihat pada persamaan (2.3)

$$W_{i,j} = tf_{i,j} \; x \; idf_i \quad (2.3)$$

Dimana:

$W_{i,j}$: Bobot TF-IDF

$tf_{i,j}$: Frekuensi term

idf_i: *Inverse Document Frequency*

2.2.8 Naïve Bayes Classifier

Ilmuwan Inggris bernama Thomas Bayes memperkenalkan *Naïve Bayes*, sebuah algoritma klasifikasi yang menggunakan prinsip probabilitas dan statistik dengan menerapkan *teorema bayes* untuk menggabungkan informasi sebelumnya dengan informasi baru. *Naïve Bayes Classifier* ditandai dengan asumsi yang sangat

kuat dari masing-masing kondisi atau kejadian. [26]. Metode *Naïve Bayes* untuk pengklasifikasian analisis sentimen memiliki beberapa kelebihannya termasuk kesimpelan, kecepatan, dan tingkat akurasi yang tinggi Berikut bentuk Umum dari *Naïve Bayes*

$$P(H|X) = \frac{(P(X|H)\ x\ P(H))}{P(X)} \quad (2.4)$$

Keterangan:

$P(H|X)$ = Probabilitas hipotesis H berdasar kondisi X

$P(H)$ = Probabilitas hipotesis H

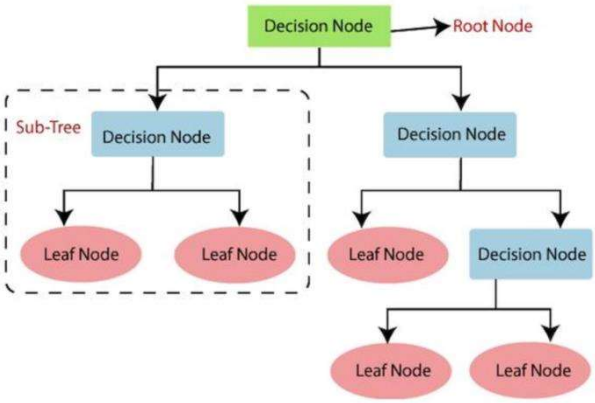
$P(X|H)$ = Probabilitas X berdasar kondisi pada hipotesis H

$P(X)$ = Probabilitas dari X

Dengan metode *Naïve Bayes* proses klasifikasi memerlukan sejumlah penunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis.

2.2.9 Decision Tree

Decision tree adalah teknik dalam machine learning yang digunakan untuk melakukan prediksi dan klasifikasi. Algoritma *decision tree* mengolah data dengan memanfaatkan aturan dan keputusan., membentuk struktur pohon yang mudah dipahami. Pada metode *Decision tree*, dilakukan pengubahan data ke dalam bentuk struktur pohon keputusan, kemudian mengubah pohon menjadi aturan dan menyederhanakan aturan tersebut [27]. Struktur pada *decision tree* seperti sebuah pohon, dimana terdapat *root node* (node akar), *child node* (node anak), dan *leaf*(daun) sebagai tempat untuk menampilkan label suatu data. Tampilan pohon keputusan dapat dilihat pada Gambar 2.1 [28]



Gambar 2. 1 Decision Tree

Dalam memilih atribut sebagai akar, algoritma akan berfokus pada atribut yang memberikan nilai gain tertinggi di antara atribut-atribut yang ada.

$$Gain\ (S, A)\ =\ Entropy\ (S) - \sum_{i=1}^{n} (|\frac{Si}{s}|) * Entropy\ (Si) \quad (2.5)$$

Dengan:

S : Himpunan Kasus

A : Atribut

|Si|: Jumlah kasus pada partisipasi ke i

|S|: Jumlah kasus dalam s

Semakin rendah nilai *Entropy,* semakin baik untuk digunakan dalam ekstraksi kelas.

$$Entropy\ (S) - \sum_{i=1}^{n} - pi * log2\ pi \quad (2.6)$$

Dengan:

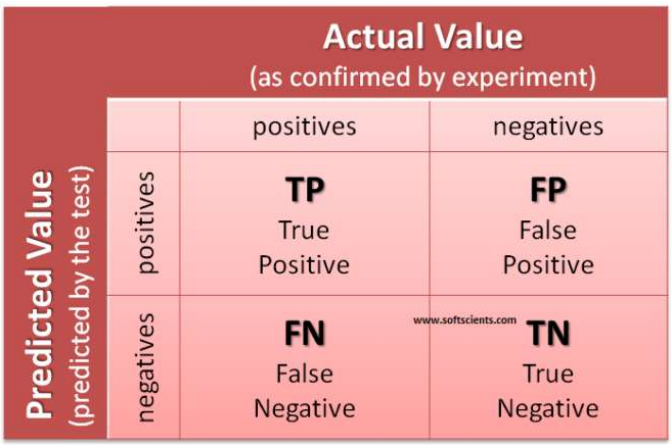
S : Himpunan kasus

n : Jumlah partisi S

p : Proporsi dari Si terhadap S

2.2.10 Confusion Matrik

Tahapan pengujian model klasifikasi digunakan menggunakan pendekatan *confusion matrik. Confusion matrik* digunanakan dengan melakukan perbandingan antara prediksi model dengan nilai aktual pada dataset pengujian. Secara umum bentuk *Confusion Matrix* dapat diamati pada Gambar 2.2 [29]



Gambar 2. 2 Confusion Matrix

1. Akurasi

Berdasarkan gambar diatas maka nilai akurasi dapat diamati dalam persamaan berikut.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (2.7)$$

Dimana:

TP (*True Positif*): Sistem memprediksi bahwa suatu *review* merupakan positif dan *review* tersebut memang benar memiliki sentimen positif.

TN (*True Negatif*): Sistem meprediksi bahwa suatu *review* merupakan negatif dan *review* tersebut memang benar memiliki sentimen negatif.

FP (*False Positif*): Sistem memprediksi bahwa suatu *review* merupakan positif akan tetapi *review* tersebut tentanya memiliki sentimen negatif.

FN (*False Negatif*): Sistem memprediksi bahwa suatu *review* merupakan negatif akan tetapi *review* tersebut ternyata memiliki sentimen positif.

2. Precision

Precision merupakan metrik evaluasi yang digunakan untuk mengukur ketepatan model dalam memberikan hasil yang relevan atau benar-benar positif. *Precision* menunjukkan seberapa banyak dari prediksi positif yang benar-benar akurat.

$$Precision = \frac{TP}{TP + FP} \qquad (2.8)$$

3. Recall

Recall adalah metrik evaluasi yang digunakan untuk mengukur kemampuan model dalam menemukan atau mengidentifikasi semua kasus positif yang ada. *Recall* menunjukkan seberapa baik model dalam menemukan semua contoh positif yang ada dalam data.

$$Recall = \frac{TP}{TP + FN} \qquad (2.9)$$

4. F1-Score

F1 Score adalah metrik evaluasi yang mengkombinasikan *precision* dan *recall*. *F1 Score* dihitung sebagai harmoni rata-rata dari *precision* dan *recall*, sehingga memperhitungkan kedua metrik tersebut secara seimbang.

$$F1 - score = 2\,x\,\frac{Precision\ x\ Recall}{Precision + Recall} \qquad (2.8)$$