**Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)** DOI: 10.25126/jtiik.2025129079
Vol. 12, No. 2, April 2025, hlm. 275-282 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

# PENGARUH TEKNIK PENANGANAN NEGASI DALAM ANALISIS SENTIMEN

**I Nengah Oka Darmayasa[1], Ngurah Agus Sanjaya ER[*2], I Gusti Agung Gede Arya Kadyanan[3], Anak Agung Istri Ngurah Eka Karyawati[4]**

[1,2,3,4]Universitas Udayana, Denpasar
Email: [1]okadarmayasa10@gmail.com, [2]agus_sanjaya@unud.ac.id, [3]gungde@unud.ac.id,
[4]eka.karyawati@unud.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 23 Desember 2025, diterima untuk diterbitkan: 10 April 2025)

**Abstrak**

"*Garbage in, garbage out*" merupakan sebuah ungkapan klasik dalam *data science* yang menyatakan bahwa kualitas keluaran suatu sistem bergantung pada kualitas data yang dimasukkan. Dalam klasifikasi sentimen, negasi memainkan peran penting dalam menentukan polaritas sentimen kalimat, tetapi sering kali dihapus pada tahap *preprocessing* sebagai *stopword*, yang dapat menghilangkan konteks negasi tersebut. Penelitian ini mengevaluasi dampak dua teknik penanganan negasi *Next Word Negation* dan penggantian antonim terhadap performa *Naïve Bayes Classifier* dan *Support Vector Machine Classifier*. Teknik *Next Word Negation* menggabungkan kata penanda negasi dengan kata setelahnya seperti "tidak cepat" menjadi "tidak_cepat". Sementara itu, teknik penggantian antonim mengganti kata penanda negasi dan kata setelahnya dengan antonim dari kata setelahnya, misalnya "tidak cepat" menjadi "lambat". Hasil penelitian menunjukkan bahwa teknik penanganan negasi meningkatkan akurasi *Naïve Bayes* dari 82,94% tanpa penanganan negasi menjadi 85,88% dengan *Next Word Negation* dan 87,64% dengan penggantian antonim. Untuk *Support Vector Machine*, akurasi meningkat dari 84,70% tanpa penanganan negasi menjadi 89,41% dengan penggantian antonim dan 88,23% dengan *Next Word Negation.*

**Kata kunci**: *penanganan negasi, next word negation, antonim sebagai penanganan negasi, analisis sentimen, naïve bayes classifier, support vector machine classifier.*

# *THE INFLUENCE OF NEGATION HANDLING TECHNIQUE IN SENTIMENT ANALYSIS*

***Abstract***

*"Garbage in, garbage out" is a classic expression in data science that states the quality of a system's output depends on the quality of the input data. In sentiment classification, negation plays a crucial role in determining the sentiment polarity of a sentence but is often removed during the preprocessing stage as a stopword, potentially eliminating the context of negation. This study evaluates the impact of two negation-handling techniques, Next Word Negation and antonym replacement, on the performance of Naïve Bayes Classifier and Support Vector Machine Classifier. The Next Word Negation technique combines the negation marker with the following word, for example, "*tidak cepat*" becomes "*tidak_cepat*". Meanwhile, the antonym replacement technique replaces the negation marker and the following word with the antonym of the following word, for example, "*tidak cepat*" becomes "*lambat"*. The results of the study show that negation-handling techniques improve the accuracy of Naïve Bayes from 82.94% without negation handling to 85.88% with Next Word Negation and 87.64% with antonym replacement. For the Support Vector Machine, accuracy increases from 84.70% without negation handling to 89.41% with antonym replacement and 88.23% with Next Word Negation.*

**Keywords**: *negation handling, next word negation, antonym as negation handling, sentiment analysis, naïve bayes classifier, support vector machine classifier.*

## 1. PENDAHULUAN

Salah satu komponen dalam suatu *online marketplace* adalah adanya ulasan atau *review* dari pelanggan yang telah membeli suatu produk. Berdasarkan penelitian (Hariyanto & Trisunarno, 2020), *customer review* memiliki pengaruh yang positif dan signifikan terhadap kepercayaan calon pembeli untuk membeli suatu produk. Dalam dunia *data mining*, terdapat suatu teknik yang bernama

analisis sentimen yang dapat digunakan untuk mengenali dan mengekstraksi informasi dari suatu teks untuk memahami sentimen atau pendapat yang ada dalam teks tersebut (Nama *et al*., 2022). Dengan melakukan analisis sentimen pada ulasan produk yang ada di suatu *online marketplace* dapat memberikan *insight* bagi penjual produk yang dapat digunakan sebagai tolok ukur bagaimana impresi orang-orang terhadap produk yang dijualnya.

Untuk melakukan analisis sentimen memerlukan data yang berupa data tekstual. Akan tetapi, data yang berupa teks tersebut tidak bisa digunakan secara langsung untuk melakukan analisis sentimen. Data tersebut harus diubah menjadi bentuk numerik agar dapat diproses oleh model-model analisis sentimen. Proses pengubahan ini akan bergantung dari data teks yang digunakan.

Untuk mempersiapkan data tekstual yang akan diubah menjadi data numerikal perlu dilakukan proses pembersihan data (Marco *et al*, 2022), seperti mengubah kapitalisasi kata, menghapus tanda baca, *stemming*, menghapus *stopword*, dan tokenisasi. Pada tahap menghapus *stopword*, kata-kata yang sering muncul (*stopword*) akan dihapus. Salah satunya adalah kata penanda negasi, seperti "tidak", "bukan", dan "jangan". Yang menjadi masalah di sini adalah apabila ingin melakukan analisis sentimen dengan menghapus kata-kata penanda negasi tersebut, hal ini akan menyebabkan data yang digunakan kehilangan konteks negasinya. Contohnya terdapat kata "… tidak cepat", pada tahap menghapus *stopword*, kata "tidak" akan dihapus, hal ini akan menyebabkan kalimat tersebut akan kehilangan konteks negasinya.

Untuk menangani hal ini, (Das & Chakraborty, 2018) mengusulkan sebuah cara dengan menggabungkan kata penanda negasi dengan kata setelahnya dengan tanda "_". Contohnya, kata "… tidak cepat …" akan diubah menjadi "… tidak_cepat …". Dengan demikian, saat dilakukan penghapusan *stopword*, kata "tidak_cepat" ini tidak akan dihapus. Hal ini akan menyebabkan kalimat tersebut masih memiliki konteks negasinya. Penelitian (Tarecha *et al*, 2022) menangani negasi dengan melakukan inversi dan reduksi skor berdasarkan *negation scope* dari sebuah kata negasi, yaitu apakah kata negasi tersebut akan mempengaruhi satu kata, beberapa kata, atau sebuah kalimat. Penelitian (Tarecha *et al*, 2022) ini membandingkan hasil analisis sentimen tanpa penanganan negasi, dengan penanganan negasi menggunakan *Fixed Word Length* (FWL) dengan *length* 3 dengan penanda negasi biasa, dan dengan penanganan negasi menggunakan FWL *length* 3 dengan tambahan penanda negasi reduktif. Metode penentuan negasi yang digunakan adalah dengan menjumlahkan skor sentimen masing-masing kata berdasarkan *scoring system* AFINN-111.

Penelitian ini bertujuan untuk mengetahui apakah metode penanganan negasi dengan mengganti kata penanda negasi dan kata setelahnya menjadi antonim dari kata setelah penanda negasi tersebut akan memiliki pengaruh terhadap performa model analisis sentimen. Untuk model analisis sentimen yang akan digunakan adalah *Naïve Bayes* dan SVM karena kemudahan dan performanya yang baik (Waworundeng *et al*, 2022).

## 2. METODE PENELITIAN

Penelitian ini bertujuan untuk mencari bagaimana pengaruh metode penanganan negasi dengan menggunakan *Next Word Negation* dan penggantian antonim terhadap performa model analisis sentimen seperti *Naïve Bayes* dan SVM.

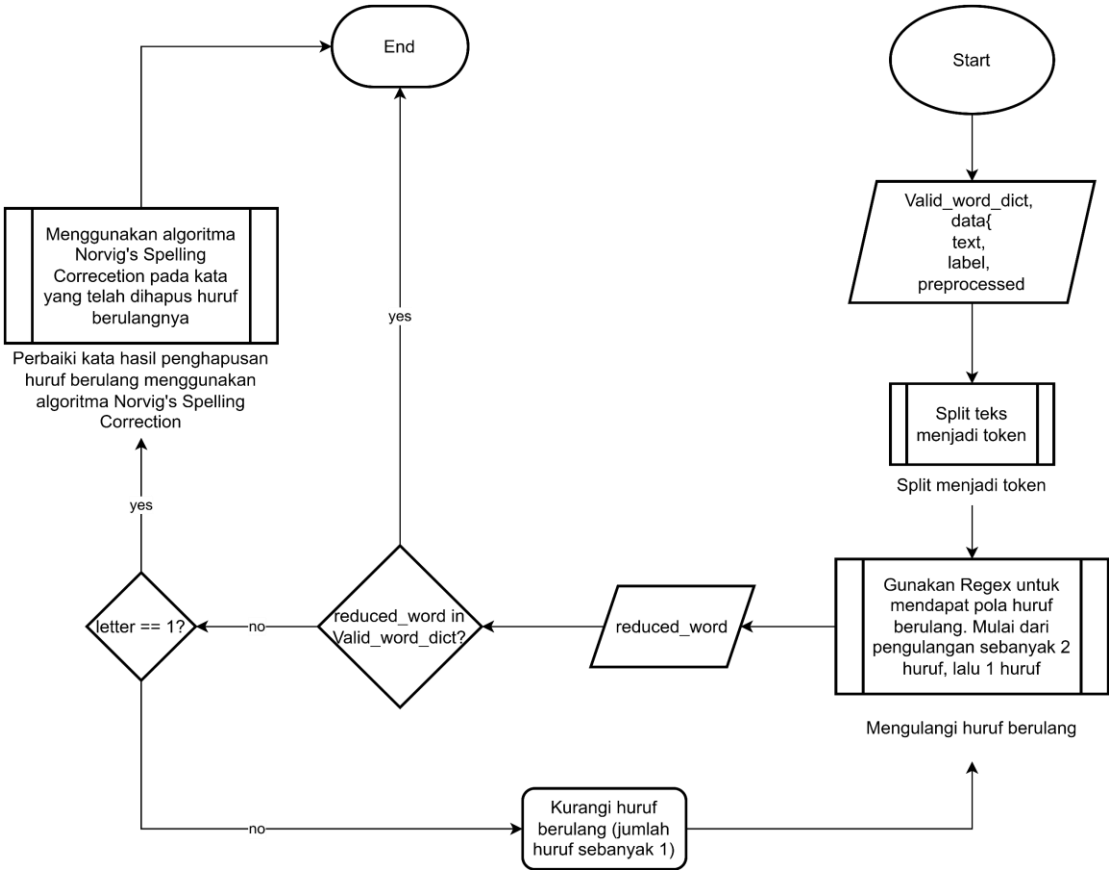
### 2.1. Data dan Persiapan Data

Data yang digunakan dalam pembuatan sistem ini berupa data sekunder yang didapat dari sebuah *website* yang bernama Kaggle berjudul "Tokopedia Product Reviews" dan "Lazada Indonesian Reviews". Data ini merupakan sebuah kumpulan ulasan pengguna aplikasi belanja *online*, Tokopedia dan Lazada, terhadap barang yang dijual di sana. Data mentah Tokopedia terdiri dari 9 buah kolom dan 40.607 baris data. Kesembilan kolom tersebut adalah sebagai berikut: 'id', 'text', 'rating', 'category', 'product_name', 'product_id', 'sold', 'shop_id', dan 'product_url'. Dari kesembilan kolom data ini, hanya 2 kolom data yang akan digunakan, yaitu kolom 'text' dan 'rating'. Kolom 'text' merupakan kolom dengan tipe data *string* yang berisi ulasan pengguna aplikasi dalam bentuk teks. Sedangkan kolom 'rating' merupakan kolom dengan tipe *integer* yang berisi *rating* produk yang diberikan oleh pengguna dari skor 1 sampai 5. Nantinya, kolom '*rating*' ini akan digunakan sebagai penanda label, yaitu nilai *rating* 1 sampai 2 akan diberikan label "negatif", *rating* bernilai 3 akan diberikan label "neutral", dan *rating* dengan nilai 4 sampai 5 akan diberikan label "positif". Sedangkan pada data mentah Lazada, terdiri dari 107.029 baris dan 15 kolom. Akan tetapi, sama seperti data Tokopedia, kolom yang digunakan hanya 2, yaitu 'reviewContent' yang berisi ulasan pengguna dan 'rating' yang berisi *rating* yang diberikan pengguna. Kedua dataset ini digunakan untuk mendapatkan sebuah dataset yang memiliki representasi kata bernegasi yang cukup. Selain itu, data yang tidak digunakan dapat dipakai sebagai data *testing* untuk menguji performa model yang dihasilkan.

Tabel 1. Persebaran Label pada 2 Dataset yang Digunakan

| | **Dataset Tokopedia** | **Dataset Lazada** | **Total** |
|---|---|---|---|
| Positif | 37.857 | 93.522 | 131.379 |
| Negatif | 925 | 9.135 | 10.060 |
| **Total** | 38.782 | 102.657 | 141.439 |

Setelah mendapatkan label dari semua baris data, yang selanjutnya perlu dilakukan adalah mengetahui jumlah data berlabel positif dan negatif.




Gambar 1. *Typo Correction* dengan Membersihkan Huruf Berulang

Hal ini perlu dilakukan untuk membuat dataset yang memiliki jumlah label yang seimbang. Berdasarkan Tabel 1, dapat diketahui bahwa jumlah data berlabel positif sangat jauh di atas data berlabel negatif, yaitu 37.857 berbanding 925 untuk dataset Tokopedia dan 93.522 berbanding 9.135 untuk data Lazada. Oleh karena itu, untuk membuat sebuah dataset yang memiliki jumlah label yang seimbang, maka perlu dilakukan *undersampling*, yaitu mengambil data pada masing-masing label dengan jumlah data sesuai dengan label yang memiliki jumlah terkecil, dalam hal ini adalah dengan mengambil 10.060 data berlabel positif dan 10.060 data berlabel negatif karena total data berlabel negatif (10.060) lebih sedikit dibanding total data berlabel positif (131.379).

Pengambilan data berlabel positif dilakukan secara acak, sedangkan data berlabel negatif diambil semuanya. Namun, data yang digunakan adalah data yang memiliki kata-kata penanda negasi serta kata setelah kata penanda negasi yang memiliki antonim. Setelah dilakukan pengecekan, dari 20.120 baris data yang ada, didapat dataset yang digunakan memiliki 393 baris data berlabel positif dan 393 data berlabel negatif dengan 603 baris memiliki kata penanda negasi dan kata antonim dari total 786 baris data yang digunakan.

Selain data utama untuk proses pelatihan dan pengujian model, penelitian ini juga menggunakan data tambahan berupa kamus kata antonim yang berisi pasangan kata dan antonimnya. Kamus kata antonim ini didapat dari sebuah repositori *GitHub* yang berisi 1.692 baris data. Setiap barisnya terdiri dari 2 kolom, yaitu kata dan antonimnya.

### 2.2. *Data Preprocessing*

Tahap *preprocessing* ini merupakan proses untuk memproses data mentah menjadi data dengan format yang telah ditentukan. Hal tersebut meliputi pembersihan data, pengubahan kapitalisasi, pengoreksian kesalahan pengetikan, dan 2 langkah tambahan penanganan negasi yang merupakan inti utama penelitian ini, yaitu langkah Next Word Negation dan langkah penggantian dengan antonim.

#### *a. Cleaning*

Tahap *cleaning* merupakan tahap untuk membersihkan data teks. Pembersihan ini meliputi membersihkan dari tanda baca, *whitespaces*, angka, emotikon, dan penyamaan kapitalisasi.

#### *b. Typo Correction*

Tahap *typo correction* atau pengoreksian kesalahan pengetikan merupakan tahap untuk memperbaiki kesalahan pengetikan yang ada pada data teks. Ada 2 macam teknik *typo correction* yang dilakukan, yaitu membersihkan kata dengan huruf berulang dan memperbaiki kesalahan pengetikan.

Pembersihan kata dengan huruf berulang dilakukan menggunakan *Regular Expression* (regex) dengan mencari pola huruf yang sama yang terjadi secara berulang. Apabila perulangan terjadi lebih dari 2 kali, maka huruf yang sama yang berulang sisanya akan dihapus dan akan menyisakan sebuah kata dengan huruf berulang sepanjang 2 kali. Setelah itu akan dicek apakah kata yang dihasilkan terdapat dalam kamus kata baku bahasa Indonesia. Apabila tidak ada, maka akan dicoba dengan menghapus pengoreksian kesalahan pengetikan. Hal ini dapat dilihat pada Gambar 1.

Memperbaiki kesalahan pengetikan dilakukan dengan menggunakan algoritma Norvig's Spelling Correction Algorithm. Cara kerjanya adalah dengan menggunakan pendekatan probabilistik untuk mencari kata pengganti terbaik di antara beberapa kata kandidat. Algoritma ini akan dimulai dengan mencari kata-kata kandidat. Kata-kata kandidat ini dicari dengan melakukan satu atau lebih penyuntingan (*edit*) yang dapat berupa penghapusan (*deletion*), penyisipan (*insertion*), perubahan (*replacement*), atau transposisi (*transpose*) huruf. Nantinya, dari semua kata kandidat, akan dicari kata yang memiliki probabilitas paling tinggi.

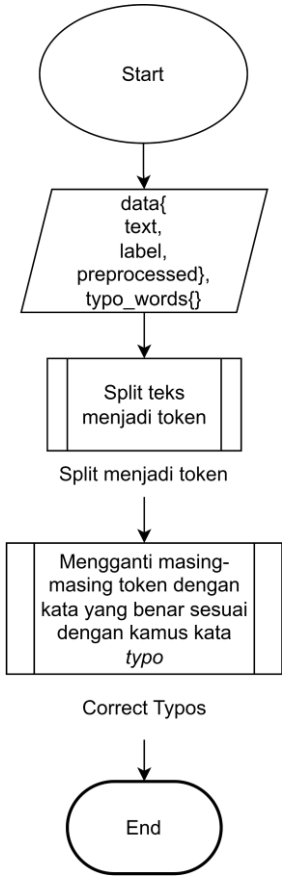
Setelah dilakukan kedua teknik tersebut, hal yang dilakukan selanjutnya adalah melakukan pembersihan kesalahan pengetikan (*typo*) dengan membandingkan masing-masing kata yang ada di dalam data teks dengan kata-kata yang ada pada sebuah kamus *typo words* seperti yang dapat dilihat pada Gambar 2. Kamus *typo words* ini merupakan sebuah *file* berformat .csv yang terdiri dari 2 kolom, yaitu 'kata salah' dan 'kata benar'. Nantinya, apabila kata yang ada pada data teks ditemukan di dalam kolom 'kata salah', maka kata tersebut akan digantikan dengan 'kata benar'. Contohnya apabila terdapat kalimat "barangnya gk bgs", dengan kata "gk" termasuk ke dalam kolom 'kata salah' dan kata "tidak" merupakan kolom 'kata benar'-nya, maka kata "gk" tersebut akan diganti menjadi kata "tidak".

### *c. Stemming*

Tahap *stemming* atau pengakaran kata merupakan tahap untuk menghapus imbuhan yang ada di dalam sebuah kata dan menyisakan akar kata atau kata dasarnya saja. Algoritma *stemming* yang digunakan adalah algoritma sederhana yang akan menghapus kombinasi awalan (*prefix*) dan akhiran (*suffix*) apabila awalan dan akhiran tersebut ada di dalam suatu kata.

### *d. Next Word Negation*

Tahap *Next Word Negation* (NWN) merupakan tahap tambahan pada tahap *preprocessing* yang merupakan teknik penanganan negasi dengan menggabungkan kata penanda negasi, seperti "tidak",



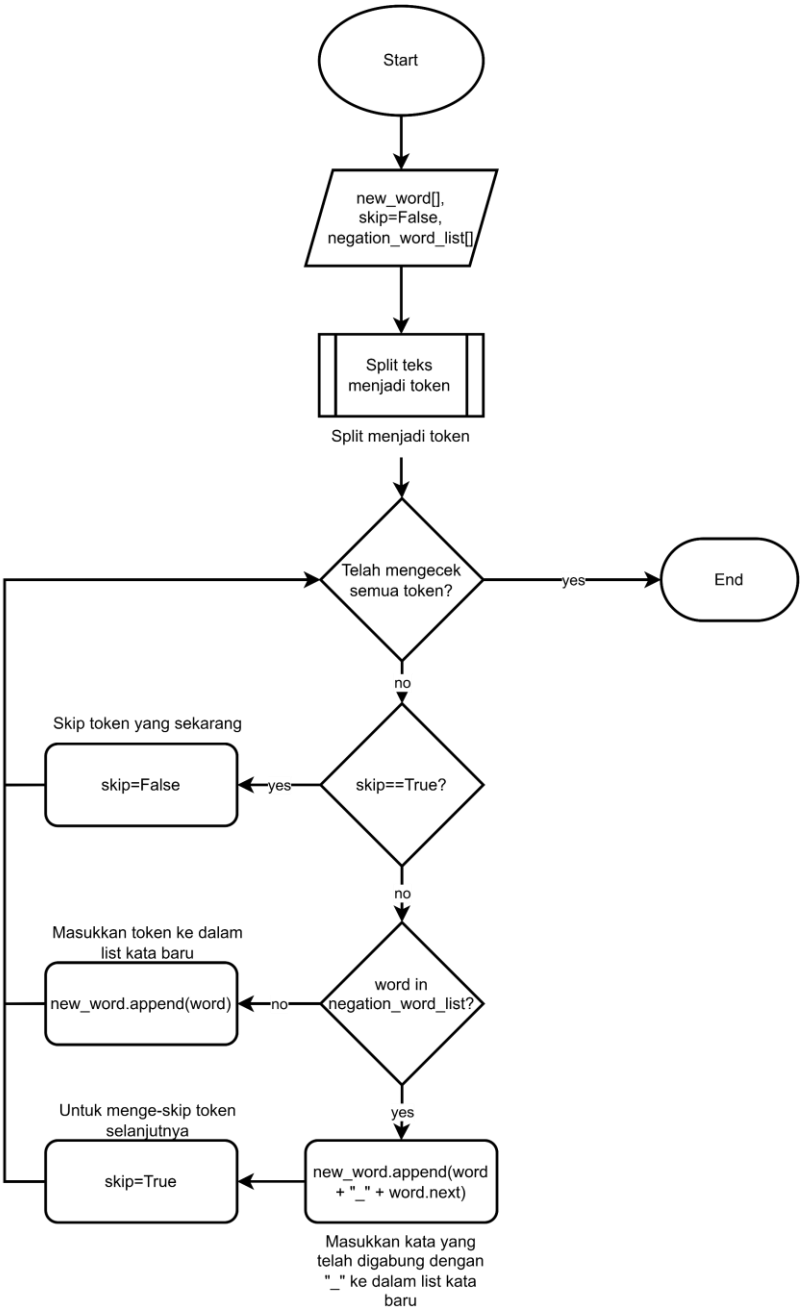

Gambar 2. *Typo Correction* dengan *Typo Words Dictionary*

"bukan", "kurang", dll, dengan kata setelahnya menggunakan suatu simbol tertentu, seperti *underscore* (_). Contohnya apabila terdapat kalimat: "barangnya gk bener ukurannya" akan diubah menjadi "barangnya gk_bener ukurannya". Untuk lebih jelasnya dapat dilihat pada Gambar 3.

Dengan proses ini, kata-kata yang telah digabungkan dengan kata negasi akan dianggap sebagai satu *token* yang memungkinkan *token* ini untuk diabaikan pada tahap penghapusan *stopword*. Hal ini akan membuat teks masih memiliki konteks negasinya setelah dilakukan penghapusan *stopword* karena kata-kata penanda negasinya telah digabung menjadi satu *token* unik yang tidak termasuk sebagai *stopword*.

### e. Penggantian dengan Antonim

Tahap penggantian dengan antonim merupakan tahap tambahan yang merupakan salah satu teknik penanganan negasi. Mekanismenya adalah dengan mengecek apakah kata setelah kata penanda negasi memiliki antonim, apabila iya, maka kata penanda negasi dan kata setelahnya akan diganti menjadi antonim tersebut. Contohnya apabila terdapat kalimat: "barangnya gak bener ukurannya" akan akan diubah menjadi "barangnya salah ukurannya". Untuk lebih jelasnya dapat melihat Gambar 4.




Gambar 3. Teknik Penanganan Negasi dengan *Next Word Negation*

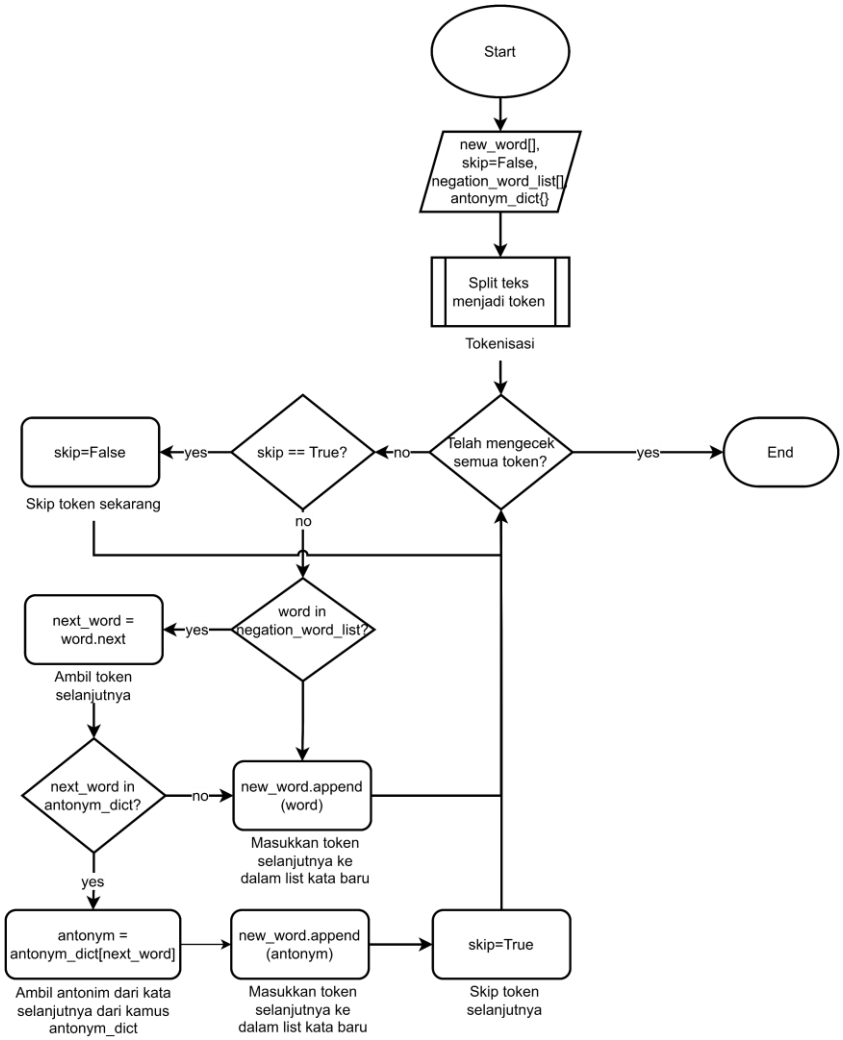
### f. *Stopword Removal*

Tahap *stopword removal* merupakan tahap penghapusan *stopword* atau kata-kata yang sering muncul yang tidak memberikan sentimen pada suatu data teks. Mekanismenya adalah dengan mengecek masing-masing kata apabila kata tersebut tidak ada di dalam suatu kamus *stopword*, maka tambahkan kata akan membuat kalimat baru tidak berisi *stopword*.

Nantinya, proses penghapusan *stopword* ini akan dilakukan 3 kali, yaitu pada data yang hanya dilakukan *preprocessing* sampai *typo correction*, pada data yang dilakukan *Next Word Negation*, dan pada data yang dilakukan penggantian dengan antonim. Hal ini akan membuat adanya 3 jenis dataset berbeda yang akan digunakan untuk melatih 3 jenis model yang berbeda. Ketiga jenis model tersebutlah yang akan dibandingkan performanya.

## 2.3. *Naïve Bayes Classifier*

Pada model *Naïve Bayes Classifier*, hal pertama yang dilakukan adalah menghitung jumlah dataset pada masing-masing kelas. Dilanjutkan dengan menghitung probabilitas kemunculan dataset pada masing-masing kelas dengan membagi banyak data pada suatu kelas dengan jumlah keseluruhan data, biasa disebut sebagai *prior probability* $P(c)$. Setelah itu adalah menghitung kemunculan tiap fitur pada masing-masing kelas, biasa disebut sebagai *likeli-*




Gambar 4. Teknik Penanganan Negasi dengan Penggantian Antonim

*hood probability* $P(x|c)$, dengan asumsi bahwa tiap fitur independen terhadap fitur lain. Hal ini yang membuat metode ini dikatakan *naive*. Selanjutnya adalah melakukan iterasi untuk masing-masing label, hitung probabilitas kemunculan masing-masing fitur dan kalikan hasilnya untuk masing-masing fitur yang ada, atau biasa disebut sebagai *posterior probability* $P(c|x)$ seperti yang dapat dilihat pada Persamaan (1). Hal ini dapat dirumuskan sebagai berikut:

$$P(c|x) = P(x|c) \times P(c) \quad (1)$$

Nantinya akan didapat probabilitas untuk masing-masing kelas, yaitu $P(pos|x)$ dan $P(neg|x)$. Di akhiri dengan membandingkan $P(pos|x)$ dan $P(neg|x)$ dan memilih kelas mana yang memiliki nilai yang lebih besar yang menandakan hasil prediksi label oleh Naïve Bayes Classifier untuk suatu data.

### 2.4. *Support Vector Machine Classifier*

Pada model SVM, hal yang dilakukan pada masing-masing iterasi adalah menghitung skor yang merupakan hasil perkalian dot antara bobot $(w)$ dan fitur $(x)$ ditambah dengan *bias* $(b)$. Selanjutnya adalah melakukan *update* nilai bobot dan *bias*. Setelah didapat bobot dan bias yang baru, maka satu iterasi telah selesai dan akan dilanjutkan ke iterasi selanjutnya dengan menggunakan bobot dan *bias* yang baru. Hal ini dapat dirumuskan sebagai sebuah fungsi sebagai berikut:

$$f(x) = w \cdot x + b \quad (2)$$

Setelah bobot (*weights*) dan bias diperbarui pada setiap iterasi, proses iterasi ini terus berlanjut hingga model mencapai kondisi yang optimal, yaitu kondisi ketika model menemukan *hyperplane* yang memisahkan kelas-kelas data dengan *margin* maksimum. *Margin* adalah jarak antara *hyperplane* dan data terdekat dari kedua kelas, yang dikenal sebagai *support vectors*. Tujuan SVM adalah memaksimalkan *margin* ini untuk meningkatkan kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya.

### 2.5. Evaluasi Model

Dalam proses pelatihan model, digunakan *K-Fold Cross Validation* sebagai metode evaluasi untuk

mengukur seberapa bagus model yang telah dilatih. Dataset dibagi menjadi $k$ bagian sama besar, dan model akan dilatih sebanyak $k$ kali, dengan data latih menggunakan $k-1$ bagian dan 1 bagian sebagai data uji (Gorriz *et al*, 2024). Dengan demikian, pada masing-masing iterasi pelatihan, model dilatih menggunakan data yang berbeda dan divalidasi pada bagian yang berbeda pula. Hal ini memungkinkan model untuk dievaluasi secara komprehensif (Attila & Kovács, 2024), karena setiap titik data menjadi bagian dari data uji sebanyak satu kali dan bagian dari data latih sebanyak $k-1$ kali.

Proses ini bertujuan untuk mengurangi bias dalam melakukan evaluasi model dan memberikan gambaran yang lebih akurat tentang performa model pada data yang belum pernah dilihat sebelumnya. Setelah semua iterasi selesai, hasil evaluasi dari setiap iterasi dirata-rata untuk mendapatkan estimasi performa model yang dapat diandalkan.

Metrik evaluasi yang digunakan untuk mengukur performa model adalah dengan menggunakan metrik akurasi. Akurasi mengukur proporsi prediksi yang benar dari total prediksi yang dilakukan, dan merupakan metrik yang umum digunakan untuk masalah klasifikasi. Masing-masing model selama proses pelatihan dilatih menggunakan data latih, lalu diuji menggunakan data uji yang sama sekali belum pernah dilihat oleh model dan dihitung akurasinya dengan menghitung jumlah prediksi sentimen yang benar dibandingkan dengan sentimen asli.

## 3. HASIL DAN PEMBAHASAN

### 3.1. Hasil Pelatihan Model

Sesuai dengan yang disampaikan pada bagian Evaluasi Model, pelatihan model dilakukan pada 2 jenis model yang berbeda (*Naïve Bayes* dan SVM) dengan 3 jenis dataset hasil *preprocessing* yang berbeda juga, yaitu tanpa penanganan negasi, dengan penanganan negasi *Next Word Negation*, dan dengan penanganan negasi penggantian antonim. Keenam model ini akan dilatih dan divalidasi menggukan *K-Fold Cross Validation.*

Tabel 2. Hasil Pelatihan Model *Naïve Bayes*

| fold | NB Prep | NB NWN | NB Antonim |
|---|---|---|---|
| 1 | 74,65% | 80,28% | 76,06% |
| 2 | 70,42% | 84,51% | 73,24% |
| 3 | 78,87% | 77,46% | 81,69% |
| 4 | 83,10% | 74,65% | 74,65% |
| 5 | 76,06% | 76,06% | 71,83% |
| 6 | 69,01% | 90,14% | 76,06% |
| 7 | 77,14% | 77,14% | 84,29% |
| 8 | 78,57% | 81,43% | 84,29% |
| 9 | 75,71% | 84,29% | 81,43% |
| 10 | 74,29% | 81,43% | 82,86% |
| Rerata | 75,78% | 80,74% | 78,64% |

Seperti yang dapat dilihat pada Tabel 2, model Naïve Bayes ini dilatih menggunakan 3 jenis dataset hasil *preprocessing* yang berbeda, yaitu tanpa penanganan negasi (NB Prep), dengan penanganan negasi *Next Word Negation* (NB NWN), dan dengan penanganan negasi penggantian antonim (NB Antonim). Dapat dilihat, model yang memiliki akurasi pelatihan tertinggi adalah model Naïve Bayes dengan penanganan negasi menggunakan *Next Word Negation* (NB NWN) dengan akurasi rata-rata pada masing-masing iterasi K-Fold sebesar 80,74%.

Tabel 3. Hasil Pelatihan Model SVM

| Fold | SVM Prep | SVM NWN | SVM Antonim |
|---|---|---|---|
| 1 | 77,46% | 77,46% | 73,24% |
| 2 | 71,83% | 81,69% | 84,51% |
| 3 | 76,06% | 73,24% | 83,10% |
| 4 | 81,69% | 84,51% | 85,92% |
| 5 | 76,06% | 74,65% | 78,87% |
| 6 | 71,83% | 83,10% | 80,28% |
| 7 | 78,57% | 84,29% | 81,43% |
| 8 | 75,71% | 82,86% | 75,71% |
| 9 | 77,14% | 85,71% | 80,00% |
| 10 | 72,86% | 74,29% | 78,57% |
| Rerata | 75,92% | 80,18% | 80,16% |

Seperti yang dapat dilihat pada Tabel 3, model SVM ini dilatih menggunakan 3 jenis dataset hasil *preprocessing* yang berbeda, yaitu tanpa penanganan negasi (SVM Prep), dengan penanganan negasi *Next Word Negation* (SVM NWN), dan dengan penanganan negasi penggantian antonim (SVM Antonim). Dapat dilihat, model yang memiliki akurasi pelatihan tertinggi adalah model SVM dengan penanganan negasi menggunakan *Next Word Negation* (SVM NWN) dengan akurasi rata-rata pada masing-masing iterasi *K-Fold* sebesar 80,18%.

### 3.2. Hasil Pengujian Model

Setelah model dilatih, selanjutnya model perlu diuji pada data uji yang merupakan data yang sama sekali belum pernah dilihat oleh model, dengan kata lain data yang tidak digunakan selama proses pelatihan.

Data uji yang digunakan terdiri dari 170 baris dengan 85 baris data berlabel positif dan 85 baris data berlabel negatif yang didapat dari mengambil sampel dari dataset mentah. Data uji ini merupakan data uji yang memiliki label agar dapat diketahui akurasi model dalam memprediksi sentimen.

Tabel 4. Hasil Pengujian dengan Data Uji

| Model | Akurasi | Peningkatan Penanganan Negasi |
|---|---|---|
| NB Prep | 82,94% | 0.00% |
| NB NWN | 85,88% | 3.54% |
| NB Antonym | 87,64% | 5.67% |
| SVM Prep | 84,70% | 0.00% |
| SVM NWN | 88,23% | 4.17% |
| SVM Antonym | 89,41% | 5.56% |

Berdasarkan hasil pengujian model pada data uji, didapatkan hasil bahwa model dengan teknik penanganan negasi menggunakan antonim pada model SVM memiliki akurasi tertinggi sebesar 89,41% yang disusul oleh 2 model SVM lain yaitu

dengan penanganan negasi *Next Word Negation* yang memiliki akurasi 88,23% dan tanpa penanganan negasi yang memiliki akurasi 84,70%.

Sedangkan untuk model *Naïve Bayes* dengan akurasi tertinggi dimiliki oleh model *Naïve Bayes* dengan teknik penanganan negasi menggunakan antonim dengan akurasi 87,64% disusul oleh model *Next Word Negation* dengan akurasi 85,88% dan model *Naïve Bayes* tanpa penanganan negasi dengan akurasi 82,94%.

Berdasarkan Tabel 4, performa model dengan teknik penanganan negasi menggunakan antonim lebih baik dibandingkan dengan model tanpa penanganan negasi. Hal ini menandakan bahwa teknik penanganan negasi menggunakan antonim memiliki dampak yang positif terhadap performa model klasifikasi sentimen dibandingkan tanpa menggunakan teknik penanganan negasi sama sekali. Akan tetapi, perlu diketahui bahwa dataset yang digunakan terdiri dari 603 baris data memiliki kata penanda negasi dan antonim dari 786 baris.

## 4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan beserta hasil yang diperoleh, dapat disimpulkan bahwa integrasi teknik penanganan negasi, seperti *Next Word Negation* dan penggantian dengan antonim, memberikan dampak positif terhadap performa model klasifikasi pada *Naïve Bayes Classifier* (NWN) dan *Support Vector Machine*. Performa model klasifikasi pada dataset yang mengalami penanganan negasi menggunakan *Next Word Negation* dan penggantian antonim dibandingkan dengan dataset tanpa penanganan negasi menunjukkan hasil yang lebih baik pada masing-masing model *Naïve Bayes Classifier* dan *Support Vector Machine* (SVM). Model *Naïve Bayes* tanpa penanganan negasi memiliki akurasi sebesar 82,94%, sementara dengan penanganan negasi menggunakan NWN mencapai 85,88% dan penggantian antonim mencapai 87,64%. Model SVM tanpa penanganan negasi memiliki akurasi 84,70% sedangkan dengan NWN mencapai 88,23% dan penggantian antonim mencapai 89,41%.

Berdasarkan hasil tersebut, penerapan teknik penanganan negasi, seperti *Next Word Negation* dan penggantian antonim, terbukti efektif dalam menjaga makna negasi pada teks untuk kasus klasifikasi data tekstual. Kedua teknik ini memberikan pengaruh yang positif terhadap performa model *Naïve Bayes Classifier dan Support Vector Machine Classifier* dibandingkan dengan model yang tanpa menggunakan teknik penanganan negasi sama sekali.

## DAFTAR PUSTAKA

ATTILA, F., & KOVÁCS, G. 2024. Enumerating the k-fold configurations in multi-class classification problems. arXiv.org, abs/2401.13843.

DAS, B., & CHAKRABORTY, S. 2018. An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation. ArXiv, abs/1806.06407.

HARIYANTO, H.T., & TRISUNARNO, L. 2020. Analisis Pengaruh Online Customer Review, Online Customer Rating, dan Star Seller terhadap Kepercayaan Pelanggan Hingga Keputusan Pembelian pada Toko Online di Shopee. Jurnal Teknik ITS, 9(2).

GÓRRIZ, J.M., SEGOVIA, F., RAMIREZ, J., ORTÍZ, A., & SUCKLING, J. 2024. Is K-fold cross validation the best model selection method for Machine Learning? ArXiv, abs/2401.16407.

MARCO, A., PALOMINO, F., & AIT AIDER, A. 2022. Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis. Applied Sciences, 12(17), 8765-8765.

NAMA, G.F., PUTRI, D.D., & SULISTIONO, W.E. 2022. Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode Naive Bayes Classifier. Jurnal Informatika dan Teknik Elektro Terapan (JITET), 10(1), 34-40.

TARECHA, R.I., WAHYUDI, F., & JANNAH, U.M. 2022. Penanganan Negasi dalam Analisa Sentimen Bahasa Indonesia. Jurnal Sistem Informasi Dan Informatika (JUSIFOR), 1(1), 51–58.

WAWORUNDENG, J.M.S., SANDAG, G.A., SAHULATA, R.A., & RELLELY, G.D. 2022. Sentiment Analysis of Online Lectures Tweets using Naïve Bayes Classifier. CogITo Smart Journal, 8(2), 371–384.