

BAB IV HASIL DAN PEMBAHASAN

4.1 Pengumpulan dan Deskripsi Data

Pengumpulan data pada tahap awal ini memakai dataset publik dari Kaggle, gabungan dua sumber utama mengenai opini masyarakat terhadap Program Makan Bergizi Gratis (MBG): kontribusi Almuhammad (2025) sejumlah 10.533 data, dan kontribusi Ravhi Haris Wibowo (2025) sebanyak 11.185 data. Penggabungan keduanya menghasilkan satu kesatuan data penelitian sejumlah 21.718 dokumen teks.

4.2 Hasil Pra-pemrosesan Data (*Preprocessing*)

Karena data mentah dari media sosial umumnya tidak terstruktur dan sarat *noise*, tahap *preprocessing* perlu dilalui agar data siap diolah mencakup *cleaning* untuk membuang elemen tidak relevan, *case folding* untuk menyeragamkan huruf, normalisasi kata tidak baku, tokenisasi, *stopword removal*, hingga *stemming* dengan algoritma yang disesuaikan kaidah morfologi Bahasa Indonesia agar pengembalian ke kata dasar berlangsung tepat.

Hasil transformasi data pada tiap tahap yang dilalui disajikan sebagai berikut:

4.2.1 *Cleaning*

Dengan memanfaatkan pustaka Regular Expression, tahap ini menyaring data dari elemen tidak relevan seperti URL, mention (@), hashtag (#), angka, tanda baca, dan emoji.

```

import re

def clean_tweet(text):
    if not isinstance(text, str):
        return text

    # 1. Hapus URL, Username (@), dan Hashtag (#)
    text = re.sub(r'http\S+|www\S+|@|\#|%', '', text)

    # 2. Hapus Emoji (Gabungan rentang Unicode utama)
    text = re.sub(r'[\U0001F600-\U0001FAFF\U0001F300-\U0001F5FF\U0001F004-\U0001F0CF\U0001F1E0-\U0001F1FF]+', '', text)

    # 3. Hapus Simbol & Angka (Hanya menyisakan huruf dan spasi)
    text = re.sub(r'^a-zA-Z\s', '', text)

    return text.strip()

# Terapkan sekaligus ke DataFrame
df['cleaning'] = df['full_text'].apply(clean_tweet)
df.head(10)

```

Gambar 4.1 Memanggil Fungsi cleaning

Setelah fungsi `clean_text` dijalankan pada data teks mentah, seluruh elemen berpotensi noise tautan URL, simbol tagar, angka, hingga emoji berhasil disingkirkan, menyisakan teks yang hanya berupa karakter alfabet murni. Perbandingan teks sebelum dan sesudah cleaning ini dapat disimak pada Tabel 4.1 berikut.

Tabel 4.1 Hasil Cleaning

Sebelum	Sesudah
Program Makan Bergizi Gratis (MBG) yang digagas oleh Prabowo Subianto-Gibran Rakabuming Raka akan diluncurkan pada 6 Januari 2025. [MBG] itu menjangkau seluruh titik sasaran. Dan yang akan kami laksanakan secara serentak tanggal 6 Januari 2025 adalah untuk anak usia sekolah https://t.co/cg0c86T78L	Program Makan Bergizi Gratis MBG yang digagas oleh Prabowo SubiantoGibran Rakabuming Raka akan diluncurkan pada Januari MBG itu menjangkau seluruh titik sasaran Dan yang akan kami laksanakan secara serentak tanggal Januari adalah untuk anak usia sekolah

4.2.2 Case Folding

Memakai fungsi bawaan string Python, tahap ini menyeragamkan format kata dengan mengubah seluruh huruf kapital menjadi huruf kecil (lowercase)

```
def case_folding(text):
    if isinstance(text, str):
        lowercase_text = text.lower()
        return lowercase_text
    else:
        return text

df['case_folding'] = df['cleaning'].apply(case_folding)
df.head(10)
```

Gambar 4.2 Memanggil Fungsi Case Folding

Teks bersih dari tahap sebelumnya lalu diproses melalui fungsi `case_folding` langkah yang penting agar algoritma machine learning tidak keliru menganggap kata yang sama sebagai dua kata berbeda hanya karena perbedaan kapitalisasi, seperti pada "MBG" dan "mbg". Hasil penyeragaman huruf ini ditampilkan pada Tabel 4.2

Tabel 4.2 Hasil Case Folding

Sebelum	Sesudah
Program Makan Bergizi Gratis (MBG) yang digagas oleh Prabowo Subianto-Gibran Rakabuming Raka akan diluncurkan pada 6 Januari 2025. [MBG] itu menjangkau seluruh titik sasaran. Dan yang akan kami laksanakan secara serentak tanggal 6 Januari 2025 adalah untuk anak usia sekolah https://t.co/cg0c86T78L	program makan bergizi gratis mbg yang digagas oleh prabowo subiantogibran rakabuming raka akan diluncurkan pada januari mbg itu menjangkau seluruh titik sasaran dan yang akan kami laksanakan secara serentak tanggal januari adalah untuk anak usia sekolah

4.2.3 Normalization

Dengan memanfaatkan kamus (dictionary) normalisasi khusus, tahap ini mengonversi kata tidak baku, singkatan, atau bahasa gaul menjadi kata baku sesuai KBBI. Mengingat tingginya pemakaian bahasa tidak baku di media sosial, normalisasi kata ini sangat memengaruhi akurasi akhir.

```
import pandas as pd

# Fungsi penggantian kata tidak baku
def replace_taboo_words(text, kamus_tidak_baku):
    # Jika bukan string, langsung kembalikan nilai kosong
    if not isinstance(text, str):
        return '', [], [], []

    replaced_words, kalimat_baku, kata_diganti, kata_tidak_baku_hash = [], [], [], []

    for word in text.split():
        baku_word = kamus_tidak_baku.get(word) # Ambil kata baku jika ada di kamus

        if isinstance(baku_word, str) and baku_word.isalpha():
            replaced_words.append(baku_word)
            kalimat_baku.append(baku_word)
            kata_diganti.append(word)
            kata_tidak_baku_hash.append(hash(word))
        else:
            replaced_words.append(word)

    return ' '.join(replaced_words), kalimat_baku, kata_diganti, kata_tidak_baku_hash

# Memilih kolom dari dataset (gunakan .copy() untuk menghindari warning SettingWithCopy)
kolom_pilihan = ['created_at', 'username', 'full_text', 'cleaning', 'case_folding']
data = df[kolom_pilihan].copy()

data.head()
```

Gambar 4.3 Memuat Kamus Slang dan Fungsi Normalization

Setiap kata pada tahap normalisasi dicocokkan dengan kamus `slang_dict` yang telah disusun sebelumnya; bila ditemukan kata tidak baku atau singkatan, sistem otomatis menggantinya dengan padanan kata baku. Hasil perbaikan ejaan dan singkatan ini diuraikan pada Tabel 4.3

Tabel 4.3 Hasil *Normalization*

Sebelum	Sesudah
Program Makan Bergizi Gratis (MBG) yang digagas oleh Prabowo Subianto-Gibran Rakabuming Raka akan diluncurkan pada 6 Januari 2025.	program makan bergizi gratis mbg yang digagas oleh prabowo subiantogibran rakabuming raka akan diluncurkan pada januari mbg itu

[MBG] itu menjangkau seluruh titik sasaran. Dan yang akan kami laksanakan secara serentak tanggal 6 Januari 2025 adalah untuk anak usia sekolah https://t.co/cg0c86T78L	menjangkau seluruh titik sasaran dan yang akan kami laksanakan secara serentak tanggal januari adalah untuk anak usia sekolah
---	---

4.2.4 Tokenizing

Agar tiap kata dapat dianalisis secara terpisah, proses ini memecah kalimat menjadi satuan kata tunggal (token).

```
def tokenize(text):
    tokens = text.split()
    return tokens

df['tokenize'] = df['normalisasi'].apply(tokenize)

df.head(10)
```

Gambar 4.4 Memanggil Fungsi Tokenizing

Lewat fungsi `tokenize_text`, kalimat baku hasil tahap normalisasi dipecah menggunakan pemisah spasi mengubah teks yang semula berupa satu untaian kalimat panjang (string) menjadi struktur data senarai (list) berisi kumpulan kata tunggal. Hasil pemecahan kata ini tersaji pada Tabel 4.4

Tabel 4.4 Hasil *Tokenizing*

Sebelum	Sesudah
Program Makan Bergizi Gratis (MBG) yang digagas oleh Prabowo Subianto-Gibran Rakabuming Raka akan diluncurkan pada 6 Januari 2025. [MBG] itu menjangkau seluruh titik sasaran. Dan yang akan kami laksanakan secara serentak tanggal 6	['program', 'makan', 'bergizi', 'gratis', 'mbg', 'yang', 'digagas', 'oleh', 'prabowo', 'subiantogibran', 'rakabuming', 'raka', 'akan', 'diluncurkan', 'pada', 'januari', 'mbg', 'itu', 'menjangkau', 'seluruh', 'titik', 'sasaran', 'dan', 'yang', 'akan', 'kami', 'laksanakan', 'secara', 'serentak',

Januari 2025 adalah untuk anak usia sekolah https://t.co/cg0c86T78L	'tanggal', 'januari', 'adalah', 'untuk', 'anak', 'usia', 'sekolah']
---	---

4.2.5 Stopword Removal

Memakai daftar stopwords standar, tahap ini membuang kata hubung umum yang tidak membawa makna sentimen berarti, seperti "wah", "ini", atau "buat"

```
from nltk.corpus import stopwords
nltk.download('stopwords')
stop_words = stopwords.words('indonesian')

def remove_stopwords(text):
    return [word for word in text if word not in stop_words]

df['stopword removal'] = df['tokenize'].apply(lambda x: remove_stopwords(x))
```

Gambar 4.5 Memanggil Fungsi Stopword Removal

Daftar token yang telah terbentuk disaring kembali untuk memangkas redundansi data: fungsi `remove_stopwords` memeriksa setiap token, dan bila termasuk daftar kata umum tanpa bobot sentimen, kata itu akan dibuang seperti tampak pada Tabel 4.5

Tabel 4.5 Hasil *Stopword Removal*

Sebelum	Sesudah
Program Makan Bergizi Gratis (MBG) yang digagas oleh Prabowo Subianto-Gibran Rakabuming Raka akan diluncurkan pada 6 Januari 2025. [MBG] itu menjangkau seluruh titik sasaran. Dan yang akan kami laksanakan secara serentak tanggal 6 Januari 2025 adalah untuk anak usia sekolah https://t.co/cg0c86T78L	['program', 'makan', 'bergizi', 'gratis', 'mbg', 'digagas', 'prabowo', 'subiantogibran', 'rakabuming', 'raka', 'diluncurkan', 'januari', 'mbg', 'menjangkau', 'titik', 'sasaran', 'laksanakan', 'serentak', 'tanggal', 'januari', 'anak', 'usia', 'sekolah']

4.2.6 Stemming

Dengan memanfaatkan pustaka Sastrawi, tahap terakhir ini mengembalikan kata berimbuhan ke bentuk dasarnya. Optimalisasi teknik stemming pada teks Bahasa Indonesia terbukti mampu mendongkrak performa model klasifikasi

```
!pip install 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 stem_text(text):
    return [stemmer.stem(word) for word in text]

df['stemming_data'] = df['stopword removal'].apply(lambda x: ' '.join(stem_text(x)))
```

Gambar 4.6 Memanggil Fungsi Stemming

Sebagai penutup pra-pemrosesan data, fungsi `stemming_text` memproses token-token yang tersisa agar kembali ke bentuk dasarnya (akar kata). Merujuk Tabel 4.6, pustaka Sastrawi terbukti berhasil memangkas imbuhan pada kata "bergizi" hingga kembali ke bentuk dasarnya, yaitu "gizi".

Tabel 4.6 Hasil *Stemming*

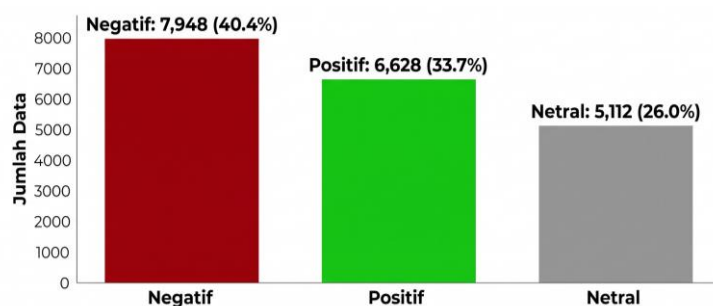
Sebelum	Sesudah
Program Makan Bergizi Gratis (MBG) yang digagas oleh Prabowo Subianto-Gibran Rakabuming Raka akan diluncurkan pada 6 Januari 2025. [MBG] itu menjangkau seluruh titik sasaran. Dan yang akan kami laksanakan secara serentak tanggal 6 Januari 2025 adalah untuk anak usia sekolah https://t.co/cg0c86T78L	program makan gizi gratis mbg gagasan prabowo subiantogibran rakabuming raka luncur januari mbg jangkauan titik sasaran laksanakan serentak tanggal januari anak usia sekolah

4.3 Hasil Pelabelan Data (*Data Labeling*)

Setelah data mentah dibersihkan lewat text preprocessing termasuk pembuangan baris data yang kosong akibat penghapusan noise sebanyak 19.688 dokumen teks memasuki tahap pelabelan kelas sentimen (data labeling). Sesuai rancangan sistem penelitian, pelabelan ini dilakukan secara semi-otomatis melalui pendekatan leksikon (lexicon-based), yang kemudian diverifikasi manual oleh peneliti demi menjaga validitas hasilnya.

Pendekatan leksikon bekerja dengan mencocokkan setiap kata dalam dokumen terhadap kamus leksikon yang telah memiliki nilai polaritas sentimen positif atau negatif; setelah skor terakumulasi, data dikelompokkan secara final ke tiga kelas sentimen: Positif, Negatif, dan Netral.

Rincian distribusi hasil pelabelan dari 19.688 data tersebut dijabarkan pada Gambar 4.7 berikut:



Gambar 4.7 Jumlah Analisis Sentimen

4.3.1 Sentimen Positif

Komentar yang mengungkapkan dukungan, kepuasan, apresiasi, atau harapan baik masyarakat atas keberlanjutan Program Makan Bergizi Gratis (MBG) direpresentasikan oleh kelas sentimen positif. Pada kelompok data ini, sistem banyak mendeteksi kata berpolaritas tinggi, seperti "dukung", "bantu", "sehat", "gizi", "maju", "tepat", dan "terima kasih".

1. Contoh Data Asli: "Mari kita dukung implementasi program makan gizi gratis demi kemajuan generasi penerus bangsa!"
2. Contoh Hasil Preprocessing: ['mari', 'dukung', 'implementasi', 'program', 'makan', 'gizi', 'gratis', 'maju', 'generasi', 'bangsa']

4.3.2 Sentimen Negatif

Sebaliknya, kelas sentimen negatif mencakup opini yang menyuarakan kritik, kekhawatiran, ketidakpuasan, keraguan, atau penolakan terhadap pelaksanaan maupun sumber anggaran kebijakan MBG dengan kata-kata berbobot negatif yang kerap terdeteksi antara lain "korupsi", "boros", "utang", "pajak", "potong", "beban", dan "kurang".

1. Contoh Data Asli: "Ruang fiskal makin tipis, target pajak sulit tercapai, ekonomi lesu, belanja tidak efisien, MBG boros, dan beban jatuh tempo utang kian besar."
2. Contoh Hasil Preprocessing: ['ruang', 'fiskal', 'tipis', 'target', 'pajak', 'sulit', 'capai', 'ekonomi', 'belanja', 'efisiensi', 'boros', 'mbg', 'beban', 'jatuh', 'tempo', 'utang']

4.3.3 Sentimen Netral

Adapun kelas sentimen netral terdiri atas cuitan berupa pernyataan fakta empiris, pembagian tautan berita tanpa disertai opini pribadi penulisnya, atau kalimat informatif yang sama sekali tidak menyiratkan muatan emosi yang jelas.

1. Contoh Data Asli: "Pemerintah meluncurkan program makan gizi gratis (MBG) secara serentak pada bulan Januari. Kepala Badan Gizi Nasional menyebutkan program ini telah melalui uji coba implementasi di beberapa provinsi."
2. Contoh Hasil Preprocessing: ['perintah', 'luncur', 'program', 'makan', 'gizi', 'gratis', 'mbg', 'serentak', 'januari', 'kepala', 'badan', 'gizi', 'nasional', 'uji', 'coba', 'implementasi', 'provinsi']

4.3.4 Analisis Distribusi dan Verifikasi Manual

Meski pendekatan leksikon sanggup memberi label awal secara cepat pada puluhan ribu data, verifikasi manual tetap dilakukan secara acak (random sampling) pada sebagian data sebuah intervensi yang krusial untuk mengoreksi kesalahan klasifikasi mesin, terutama pada kalimat bermajas ironi, sarkasme, atau bahasa kiasan yang sering luput diidentifikasi secara akurat oleh perhitungan matematis.

Sebagaimana terlihat pada Gambar 4.7, respons masyarakat di media sosial terhadap Program Makan Bergizi Gratis memperlihatkan sebaran yang cukup bervariasi namun tetap didominasi opini bersentimen Negatif, yaitu 7.948 data atau 40,4% dari keseluruhan. Sentimen Positif menempati posisi kedua dengan 6.628 data (33,7%), disusul sentimen Netral sebanyak 5.112 data (26,0%).

Kendati kelas Negatif tetap menjadi mayoritas, ketimpangan kelas (imbalanced data) yang terjadi tidak seekstrem sebelumnya karena jumlah kelas mayoritas tidak lagi melampaui gabungan dua kelas lainnya pola distribusi yang masih tergolong wajar pada ekstraksi data organik opini publik di media sosial. Keseluruhan 19.688 data berlabel ini selanjutnya langsung dipakai pada tahap split data, untuk melatih dan membentuk model klasifikasi algoritma Multinomial Naïve Bayes dan Support Vector Machine (SVM).

4.3.5 Visualisasi Word Cloud

Untuk melengkapi analisis pelabelan data, penelitian ini menghadirkan visualisasi Word Cloud guna menangkap gambaran besar mengenai topik atau kata yang paling sering diperbincangkan masyarakat terkait Program Makan Bergizi Gratis (MBG) pada dataset publik Kaggle. Ukuran tiap kata pada visualisasi ini mencerminkan frekuensi kemunculannya di seluruh dataset (19.688 dokumen teks) semakin besar ukuran hurufnya, semakin sering kata tersebut dipakai publik.

Kata-kata inti yang merepresentasikan entitas kebijakan seperti "program", "n", "gizi", dan "gratis" tampak mendominasi wacana publik pada Gambar 1. Hal ini wajar mengingat kata-kata tersebut memang menjadi topik utama penelitian.

Di sisi lain, kata-kata yang mencerminkan dukungan dan harapan tetap tampil dalam Word Cloud, menegaskan keberadaan sentimen positif sebagai kelompok terbesar kedua dengan porsi 33,7%. Kata seperti "dukung", "sehat", "bantu", dan "anak" tampak cukup menonjol, berdampingan dengan narasi sentimen netral (26,0%) yang umumnya bersifat informatif menunjukkan bahwa di tengah derasny kritk seputar anggaran, tetap ada porsi masyarakat yang cukup solid mendukung esensi program demi pemenuhan gizi dan kesehatan anak.

Secara keseluruhan, visualisasi ini membuktikan bahwa metode pra-pemrosesan teks (preprocessing) berhasil mengekstraksi kata-kata bermakna.

Dataset yang dipakai pun terbukti sangat dinamis: mampu menangkap realitas pro dan kontra di masyarakat, sekaligus cukup relevan dan kaya muatan opini untuk diproses lebih lanjut menggunakan algoritma klasifikasi.

4.4 Pembagian Data (*Split Data*) dan Ekstraksi Fitur

```
# 1. Split Data 80:20
X_tr, X_te, y_tr, y_te = train_test_split(X, y_enc, test_size=0.2, random_state=42, stratify=y_enc)
counts, total = [len(X_tr), len(X_te)], len(X_tr) + len(X_te)

# 2. Visualisasi Barplot
plt.figure(figsize=(5, 3.5))
ax = sns.barplot(x=['Train (80%)', 'Test (20%)'], y=counts, hue=['Train', 'Test'], palette='Blues', legend=False)

# 3. Label Jumlah & Persentase
for i, val in enumerate(counts):
    ax.text(i, val + (total * 0.01), f'{val} ({val/total*100:.1f}%)', ha='center', va='bottom', fontweight='bold')

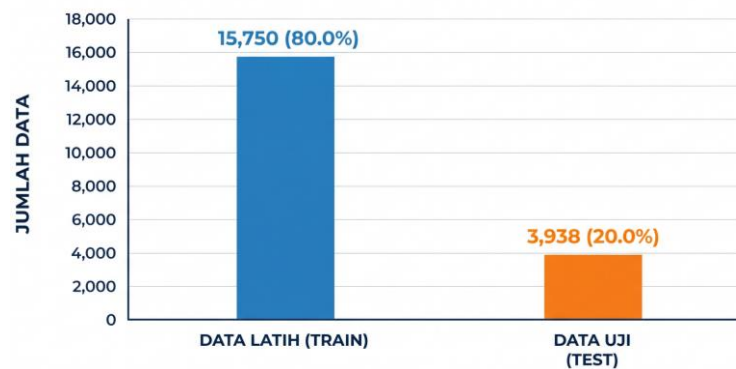
plt.title('Pembagian Data 80:20')
plt.ylabel('Jumlah Sampel')
plt.tight_layout()
plt.show()
```

Gambar 4.9 Proses Split Data

Proses split data bertujuan membagi keseluruhan dataset berlabel menjadi dua bagian terpisah, untuk kebutuhan pelatihan dan pengujian model. Mengacu skenario pengujian yang ditetapkan pada Bab III, data dibagi dengan rasio 80:20 80% sebagai data latih (training data) dan 20% sebagai data uji (testing data) rasio yang lazim dipakai dalam riset machine learning untuk mencapai keseimbangan optimal antara pembelajaran pola dan validasi model.

Proses pembagian ini dijalankan memakai fungsi `train_test_split` dari pustaka `scikit-learn`, dengan parameter `stratify` diaktifkan agar proporsi sebaran kelas (Positif, Negatif, Netral) tetap seimbang antara data latih dan data uji. Guna memvalidasi dan menampilkan perbandingan distribusi data secara visual, pustaka `matplotlib` dan `seaborn` turut dipakai untuk menghasilkan grafik batang (bar chart).

Dari eksekusi kode tersebut pada total 19.688 dokumen teks berlabel, diperoleh grafik visual sekaligus rincian pembagian data: porsi data latih (Train) berjumlah 15.750 data (80%), dan data uji (Test) berjumlah 3.938 data (20%). Porsi data latih ini dipakai algoritma Naïve Bayes dan SVM untuk mengenali pola kata pada tiap kelas sentimen, sementara data uji berperan sebagai data baru untuk mengevaluasi performa model secara objektif



Gambar 4.10 Jumlah Sampel Data

4.4.1 Ekstraksi Fitur dan Pembobotan Kata (*Term Weighting*)

Ekstraksi fitur merupakan tahap mengonversi data teks hasil praproses menjadi representasi numerik yang dapat diolah algoritma machine learning. Metode TF-IDF (Term Frequency-Inverse Document Frequency) dipakai penelitian ini sebagai metode pembobotan kata, yang memadukan dua komponen: TF yang mengukur frekuensi kata dalam satu dokumen, dan IDF yang mengukur seberapa jarang kata tersebut muncul di seluruh korpus. Rumusnya adalah sebagai berikut:

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \log\left(\frac{N}{1 + \text{df}(t)}\right)$$

Ekstraksi fitur ini diterapkan pada dataset sebanyak 19.688 dokumen teks berbahasa Indonesia yang telah melalui tahap stemming. Dengan pustaka Scikit-learn dan parameter `max_features=5.000`, dihasilkan matriks TF-IDF berukuran 19.688 x 5.000 dengan tingkat sparsity mencapai 99,78% ciri umum pada representasi teks berdimensi tinggi.

Sepuluh kata dengan rata-rata TF-IDF tertinggi disajikan pada Tabel 4.7, di mana kata "makan", "gratis", dan "gizi" mendominasi dengan nilai TF-IDF tertinggi sekaligus IDF terendah menandakan kata-kata tersebut muncul di hampir seluruh dokumen dan menjadi topik utama dataset. Matriks TF-IDF ini selanjutnya dipakai sebagai masukan pada proses pelatihan model klasifikasi.

Tabel 4.7 Sepuluh Kata dengan Rata-rata TF-IDF Tertinggi

No	Kata	Rata-rata TF-IDF	Nilai IDF
1	makan	0.076675	1.3107
2	gratis	0.072539	1.3634
3	siang	0.063085	1.7803
4	gizi	0.053090	1.9977
5	program	0.046448	2.1183
6	mbg	0.045142	2.1325
7	anak	0.026794	2.9891
8	indonesia	0.018519	3.4339
9	prabowo	0.014987	3.6261
10	sekolah	0.014923	3.6842

4.5 Implementasi dan Evaluasi Model Klasifikasi

Sebagai inti dari penelitian ini, tahap implementasi memakai data latih untuk melatih algoritma Multinomial Naïve Bayes dan Support Vector Machine (SVM). Implementasinya dijalankan lewat metode Pipeline dari pustaka scikit-learn, untuk memadukan proses ekstraksi fitur TF-IDF dan proses klasifikasi dalam satu alur kerja yang konsisten.

4.5.1 Implementasi dan Evaluasi Multinomial Naïve Bayes (MNB)

```
# ===== BASELINE: MultinomialNB (Original) =====
baseline_nb = Pipeline([
    ('tfidf', TfidfVectorizer(max_features=50000, ngram_range=(1, 3))),
    ('clf', MultinomialNB())
])
baseline_nb.fit(X_train, y_train)
y_pred_nb_base = baseline_nb.predict(X_test)
results_nb_base = evaluate_and_report
('Baseline MultinomialNB (Original)', y_test, y_pred_nb_base)
```

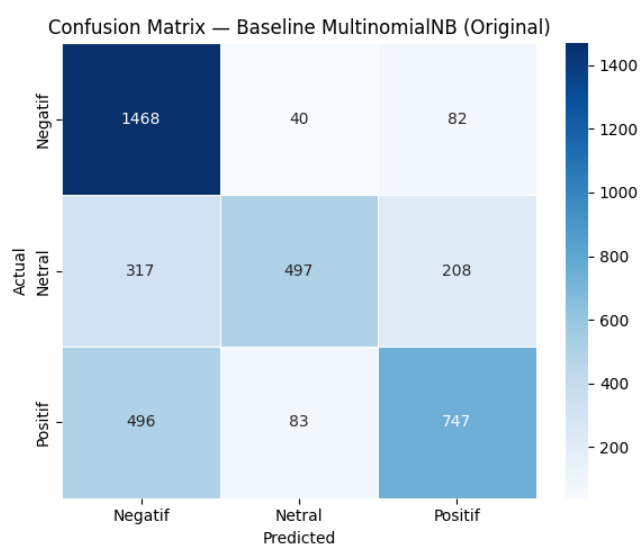
```
=====
Baseline MultinomialNB (Original)
=====
Accuracy : 0.6887 (68.87%)
Precision : 0.7104
Recall : 0.6887
F1-Score : 0.6762
```

	precision	recall	f1-score	support
Negatif	0.64	0.92	0.76	1590
Netral	0.80	0.49	0.61	1022
Positif	0.72	0.56	0.63	1326
accuracy			0.69	3938
macro avg	0.72	0.66	0.67	3938
weighted avg	0.71	0.69	0.68	3938

Gambar 4.11 Hasil Evaluasi Klasifikasi Multinomial Naïve Bayes

Model pertama dibangun menggunakan algoritma Multinomial Naïve Bayes, dengan konfigurasi ekstraksi fitur `max_features=50000` dan `ngram_range=(1,3)` untuk menangkap konteks mulai dari kata tunggal hingga rangkaian tiga kata (trigram).

Sebagaimana terlihat pada Gambar 4.11, model MNB mencatatkan akurasi keseluruhan sebesar 68,87%, meski metrik per kelas menunjukkan ketimpangan performa yang cukup mencolok. Recall model ini sangat tinggi pada kelas Negatif (0,92 atau 92%), tetapi menurun cukup signifikan pada kelas Netral (0,49) dan Positif (0,56) mengindikasikan kecenderungan model bias ke arah sentimen negatif, yang pada data uji ini kebetulan merupakan kelas dengan jumlah sampel terbanyak (support = 1.590 data



Gambar 4.12 *Confusion Matrix* Multinomial Naïve Bayes

Ketimpangan prediksi tersebut tampak semakin nyata pada *Confusion Matrix* di Gambar 4.12, yang menunjukkan model MNB sangat agresif menebak data uji sebagai kelas "Negatif" terlihat dari tingginya angka True Positive pada kelas Negatif, yang mencapai 1.468 data terprediksi benar.

Namun demikian, tingginya jumlah tebakan benar pada kelas Negatif itu dibarengi angka kesalahan prediksi (False Positive) yang cukup ekstrem pada kolom yang sama: sebanyak 317 opini yang sekadar informatif (Netral) dan 496 opini berisi dukungan (Positif) justru keliru diklasifikasikan model MNB sebagai sentimen Negatif. Fenomena ini menegaskan karakteristik MNB yang sangat bergantung pada probabilitas awal (prior probability), di mana fitur kata dari kelas mayoritas cenderung mendominasi fitur kelas lain yang lebih sedikit, sehingga model menjadi sensitif terhadap imbalanced data.

4.5.2 Implementasi dan Evaluasi Support Vector Machine (SVM)

Model kedua dibangun menggunakan algoritma SVM dengan pendekatan LinearSVC, dengan konfigurasi ekstraksi fitur `max_features=20000` dan `ngram_range=(1,2)` untuk menangkap konteks kata tunggal dan pasangan kata (bigram); parameter `max_iter=10000` ditambahkan agar model tetap konvergen pada ruang fitur yang luas.

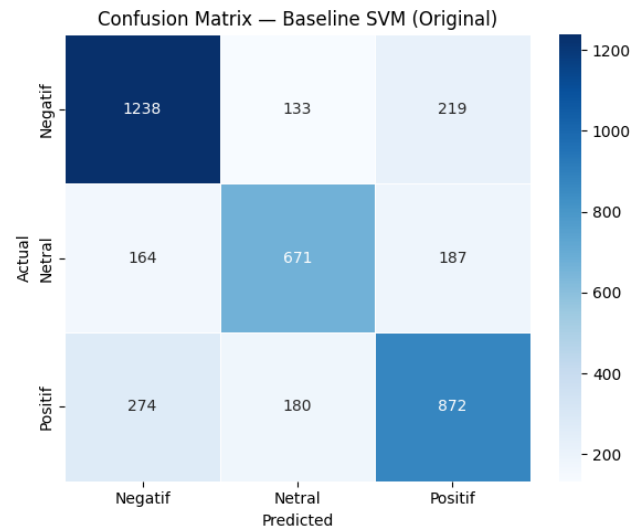
```
# ===== BASELINE: SVM (Original) =====
baseline_svm = Pipeline([
    ('tfidf', TfidfVectorizer(max_features=20000, ngram_range=(1, 2))),
    ('clf', LinearSVC(random_state=42, max_iter=10000))
])
baseline_svm.fit(X_train, y_train)
y_pred_baseline = baseline_svm.predict(X_test)
results_baseline = evaluate_and_report
('Baseline SVM (Original)', y_test, y_pred_baseline)
```

```
=====
Baseline SVM (Original)
=====
Accuracy : 0.7062 (70.62%)
Precision : 0.7050
Recall : 0.7062
F1-Score : 0.7052
```

	precision	recall	f1-score	support
Negatif	0.74	0.78	0.76	1590
Netral	0.68	0.66	0.67	1022
Positif	0.68	0.66	0.67	1326
accuracy			0.71	3938
macro avg	0.70	0.70	0.70	3938
weighted avg	0.70	0.71	0.71	3938

Gambar 4.13 Hasil Evaluasi Klasifikasi Support Vector Machine (SVM)

Berbeda dari MNB yang cenderung bias ke kelas Negatif, algoritma SVM justru menunjukkan kemampuan klasifikasi yang jauh lebih seimbang dan proporsional di seluruh kelas, dengan akurasi global 70,62%. Nilai F1-score yang lebih stabil 0,76 pada kelas Negatif, 0,67 pada kelas Netral, dan 0,67 pada kelas Positif mencerminkan keseimbangan ini, dengan peningkatan paling krusial tampak pada kemampuan SVM mengenali opini Netral dan Positif secara lebih akurat, yang sebelumnya banyak disalahartikan oleh Naïve Bayes.



Gambar 4.14 *Confusion Matrix* Support Vector Machine (SVM)

Pada *Confusion Matrix* SVM di Gambar 4.14, sebaran angka terpusat dengan baik pada garis diagonal utama (dari kiri atas ke kanan bawah) indikator bahwa model berhasil membuat tebakan True Positive yang merata di berbagai kelas, yakni 1.238 data Negatif, 671 data Netral, dan 872 data Positif.

Meski jumlah tebakan benar pada kelas Negatif (1.238) tampak lebih rendah dibandingkan MNB (1.468), hal ini justru menjadi bentuk kompromi (trade-off) matematis yang positif untuk menekan bias. SVM berhasil memangkas angka kesalahan prediksi (False Positive) yang sebelumnya ekstrem pada MNB, dan mengalihkannya menjadi tebakan benar pada kelas Netral dan Positif. Kendati masih ada sedikit kekeliruan model yang beririsan antar kelas akibat kemiripan kosakata, secara keseluruhan *Confusion Matrix* SVM memperlihatkan pola klasifikasi yang jauh lebih adil, cerdas, dan tangguh menghadapi ketidakseimbangan data.

4.5.3 Implementasi dan Evaluasi Support Vector Machine (SVM) + Word & Char N-Gram

Eksperimen lanjutan dijalankan pada algoritma SVM dengan teknik Feature Fusion (penggabungan fitur) tingkat kata (Word N-Gram) dan karakter (Character N-Gram), dengan tujuan mengoptimalkan kemampuan model mengenali variasi

ejaan, salah ketik (typo), dan imbuhan kompleks pada opini media sosial yang rawan menjadi Out-of-Vocabulary bila hanya mengandalkan fitur kata.

Penggabungan ini diterapkan melalui fungsi FeatureUnion pada scikit-learn, yang memadukan dua skema pembobotan TF-IDF berikut:

1. Word N-Gram, yang menangkap makna semantik teks secara menyeluruh (ngram_range=(1, 3), max_features=25000).
2. Char N-Gram: Menangkap pola morfologis dan afiksasi di dalam batas kata (analyzer='char_wb', ngram_range=(3, 5), max_features=15000).

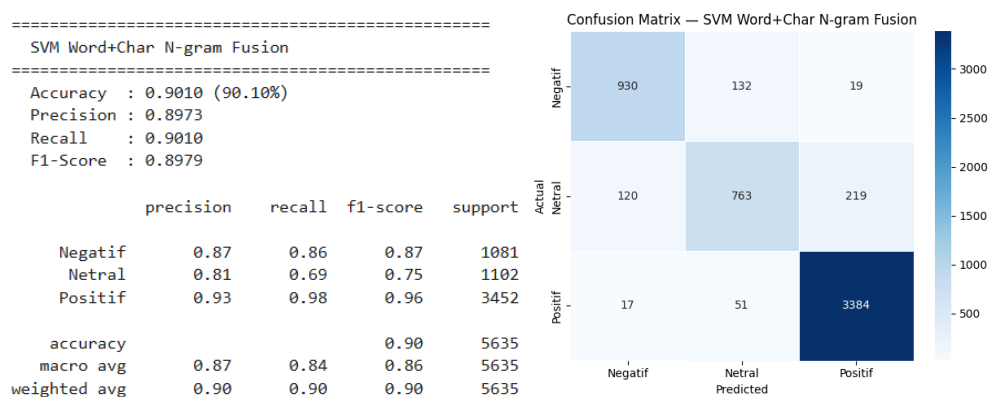
```
from sklearn.pipeline import Pipeline, FeatureUnion
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC

# 1. Membuat Pipeline Gabungan (Kata + Karakter) & Model SVM
svm_model = Pipeline([
    ('features', FeatureUnion([
        ('word', TfidfVectorizer(ngram_range=(1, 3), analyzer='word')),
        ('char', TfidfVectorizer(ngram_range=(3, 5), analyzer='char_wb'))
    ])),
    ('clf', LinearSVC(class_weight='balanced', random_state=42))
])

# 2. Melatih model dan melakukan prediksi
svm_model.fit(X_train, y_train)
y_pred = svm_model.predict(X_test)
```

Gambar 4.15 Kode Evaluasi Klasifikasi SVM Word+Char N-gram Fusion

Hasil gabungan dengan maksimal 40.000 dimensi fitur ini kemudian dilatih menggunakan algoritma LinearSVC.



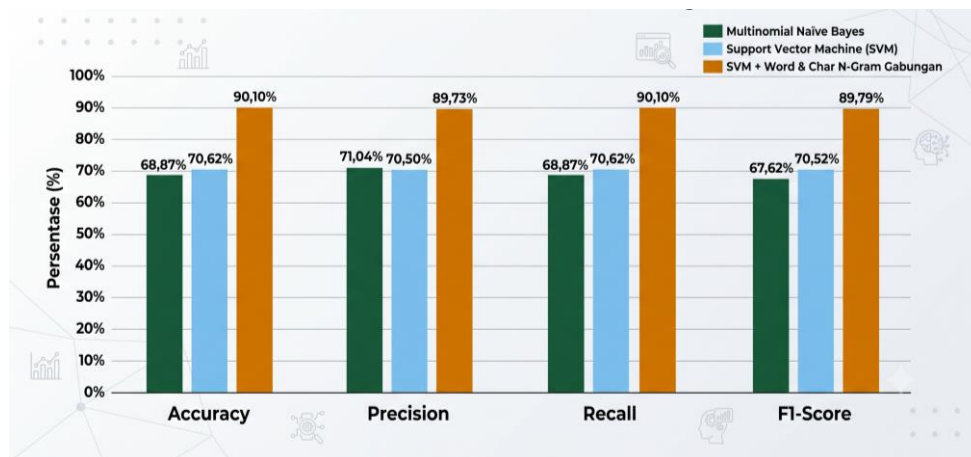
Gambar 4.16 Confusion Matrix SVM Word+Char N-gram Fusion

Analisis Confusion Matrix menunjukkan model ini sangat stabil mengatasi ketidakseimbangan data: SVM gabungan berhasil mengidentifikasi sentimen Positif dengan presisi tinggi (Recall 98%) dan mencatatkan lonjakan tajam dalam mengenali sentimen Negatif (Recall 86%). Representasi ganda dari kata dan karakter membuat batas keputusan hyperplane SVM menjadi lebih adaptif, sehingga menghasilkan prediksi klasifikasi paling tangguh dan seimbang dalam penelitian ini.

4.5.4 Rekapitulasi Perbandingan Performa

Sebagai rangkuman tahap implementasi dan evaluasi klasifikasi sentimen, perbandingan metrik performa global disusun untuk menilai tiga skenario pemodelan: Multinomial Naïve Bayes (MNB) standar, Support Vector Machine (SVM) standar, dan SVM yang dioptimasi dengan kombinasi fitur Word & Character N-Gram. Empat metrik evaluasi utama dipakai dalam perbandingan ini, yaitu Accuracy, Precision, Recall, dan F1-Score.

Kinerja ketiga model tersebut dapat dibandingkan lewat grafik batang (bar chart) berikut:



Gambar 4.17 Grafik Perbandingan Kinerja Klasifikasi Sentimen

Sebagai pelengkap visualisasi pada Gambar 4.17, rincian angka dari perbandingan metrik global ketiga skenario model disajikan pada Tabel 4.8 berikut.

Tabel 4.8 Perbandingan Metrik Global MNB dan SVM

Algoritma	Accuracy	Precision	Recall	F1-Score
Multinomial Naïve Bayes	68,87%	71,04%	68,87%	67,62%
Support Vector Machine	70,62%	70,50%	70,62%	70,52%
Support Vector Machine (SVM) + Word & Char N-Gram	90,10%	89,73%	90,10	89,79

Grafik dan tabel komparasi di atas memperlihatkan pergeseran performa yang cukup signifikan. Pada tahap pemodelan standar tanpa optimasi fitur gabungan SVM hanya unggul tipis atas MNB, dengan akurasi masing-masing 70,62% berbanding 68,87%; keduanya tampak memiliki batasan performa yang serupa pada ruang fitur ekstraksi kata standar.

Ketangguhan sesungguhnya baru tervalidasi saat SVM dioptimasi dengan ekstraksi fitur gabungan Word & Character N-Gram. Lonjakan performa yang sangat tajam pun dicatatkan model ketiga ini, dengan akurasi mencapai 90,10%.

Keunggulan model optimasi ini kian tak terbantahkan bila membandingkan metrik F1-Score, yang sebagai rata-rata harmonik dari Precision dan Recall jauh lebih representatif dan andal untuk mengukur keandalan model pada dataset opini publik yang distribusi kelasnya tidak seimbang (imbalanced data). Model MNB dan SVM standar masing-masing hanya mencapai F1-Score sebesar 67,62% dan 70,52%.

Sebaliknya, F1-Score sebesar 89,79% berhasil dicetak SVM dengan Word & Char N-Gram. Nilai yang tinggi dan merata ini membuktikan bahwa kombinasi N-Gram tingkat kata dan karakter memungkinkan model hyperplane linier pada SVM tidak hanya membaca kata secara utuh, tetapi juga mengenali pola sub-kata, imbuhan, dan konteks frasa sehingga model menjadi jauh lebih cerdas dan adil dalam membedakan batas ketiga kelas sentimen (Positif, Negatif, Netral) yang kerap ambigu.

Perbandingan ini menyajikan bukti empiris dan analitis yang kuat untuk menjawab rumusan masalah penelitian: bahwa Support Vector Machine (SVM) yang dipadukan dengan ekstraksi fitur Word & Char N-Gram terbukti menjadi metode paling optimal, stabil, dan andal untuk diterapkan dalam sistem klasifikasi analisis sentimen terkait respons publik terhadap Program Makan Bergizi Gratis (MBG).