

LAMPIRAN

Lampiran 1 Kartu Bimbingan Dosen

	DOKUMEN KARTU BIMBINGAN SKRIPSI	Nomor Dokumen	PP.05/02/55701/AMIKOMPWT
		Revisi	1
		Halaman	1 dari 1
		Tanggal	18 Mei 2015
		Token :	

KARTU BIMBINGAN SKRIPSI

Nama Mahasiswa : DAMAS BHANUARTA
NPM : 22SA21A176
Program Studi : SISTEM INFORMASI S1
Judul Skripsi : ANALISIS SENTIMEN KEPUASAN PELANGGAN BERBASIS ULASAN GOOGLE MAPS MENGGUNAKAN METODE NAIVE BAYES DAN SUPPORT VECTOR MACHINE (Studi Kasus: Kedai Kopi Bathok Purbalingga)
Dosen Pembimbing : Ali Nur Ikhlan

Histori Bimbingan Skripsi

Bimbingan Ke - : 1
Hari / Tanggal : Selasa / 14 Oktober 2025
Keterangan : Bimbingan judul dan tema skripsi

Bimbingan Ke - : 2
Hari / Tanggal : Selasa / 28 Oktober 2025
Keterangan : Latar belakang masalah diperkuat, sebutkan masalah yang real, belum ada penelitian terdahulu yang mendukung/ sebagai penguat penggunaan metode/ algoritme yang dipakai

Bimbingan Ke - : 3
Hari / Tanggal : Selasa / 04 November 2025
Keterangan : Perbaiki latar belakang masalah, belum ada masalah yg real yang diangkat pada LBM

Bimbingan Ke - : 4
Hari / Tanggal : Kamis / 06 November 2025
Keterangan : revisi Bab 1

Bimbingan Ke - : 5
Hari / Tanggal : Selasa / 18 November 2025
Keterangan : Pertajam lagi LBM, dan cari permasalahan yg real

Bimbingan Ke - : 6
Hari / Tanggal : Selasa / 02 Desember 2025
Keterangan : paragraf dipadatkan agar lebih akademis, batasan masalah tidak konsisten, tidak ada bagian urgensi penelitian secara eksplisit

Bimbingan Ke - : 7
Hari / Tanggal : Selasa / 02 Desember 2025
Keterangan : duplikasi penomoran, state of the art masih hanya deskriptif, belum membandingkan secara kritis

Bimbingan Ke - : 8
Hari / Tanggal : Kamis / 11 Desember 2025
Keterangan : pada hasil belum dijelaskan total data ulasanya

Bimbingan Ke - : 9
Hari / Tanggal : Selasa / 23 Desember 2025
Keterangan : kesimpulan terlalu menjelaskan ulang penelitian

Bimbingan Ke - : 10
Hari / Tanggal : Rabu / 21 Januari 2026
Keterangan : ACC

Lampiran 2 Kartu Bimbingan Dosen 2

	DOKUMEN KARTU BIMBINGAN SKRIPSI	Nomor Dokumen	PP.05.02/55701/AMIKOMPWT
		Revisi	1
		Halaman	1 dari 1
		Tanggal	18 Mei 2015
		Token :	

KARTU BIMBINGAN SKRIPSI

Nama Mahasiswa : DAMAS BHANUARTA
NPM : 22SA21A176
Program Studi : SISTEM INFORMASI S1
Judul Skripsi : ANALISIS SENTIMEN KEPUASAN PELANGGAN BERBASIS ULASAN GOOGLE MAPS MENGGUNAKAN METODE NAÏVE BAYES DAN SUPPORT VECTOR MACHINE (Studi Kasus: Kedai Kopi Bathok Purbalingga)
Dosen Pembimbing : Agus Pramono

Histori Bimbingan Skripsi

Bimbingan Ke- : 1. 
Hari / Tanggal : Sabtu / 04 Oktober 2025
Keterangan : - Konsultasi Judul

Bimbingan Ke- : 2. 
Hari / Tanggal : Jumat / 28 November 2025
Keterangan : - Ganti Judul dengan menggunakan 2 metode yang berbeda
- Lanjutkan Penulisan Bab 1

Bimbingan Ke- : 3. 
Hari / Tanggal : Sabtu / 06 Desember 2025
Keterangan : - ACC bab 1 lanjut bab 2

Bimbingan Ke- : 4. 
Hari / Tanggal : Senin / 08 Desember 2025
Keterangan : - ACC bab 2 lanjut bab 3

Bimbingan Ke- : 5. 
Hari / Tanggal : Senin / 15 Desember 2025
Keterangan : - ACC bab 3 lanjut bab 4

Bimbingan Ke- : 6. 
Hari / Tanggal : Sabtu / 27 Desember 2025
Keterangan : - Revisi Bab 4

Bimbingan Ke- : 7. 
Hari / Tanggal : Senin / 05 Januari 2026
Keterangan : - Revisi bab 4

Bimbingan Ke- : 8. 
Hari / Tanggal : Sabtu / 24 Januari 2026
Keterangan : - ACC bab 4 lanjut bab 5

Lampiran 3 Kode

```
!pip install Sastrawi
```

```
Collecting Sastrawi
  Downloading Sastrawi-1.0.1-py2.py3-none-any.whl.metadata (909 bytes)
Downloading Sastrawi-1.0.1-py2.py3-none-any.whl (209 kB)
----- 209.7/209.7 kB 5.5 MB/s eta 0:00:00
Installing collected packages: Sastrawi
Successfully installed Sastrawi-1.0.1
```

```
!pip install nltk
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.3.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5.3)
Requirement already satisfied: regex<=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nltk) (2025.11.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.1)
```

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

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

preprocessing

```
[ ]
# Import library
import pandas as pd
import re
from collections import Counter
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import nltk

# Unduh resource NLTK
nltk.download('punkt')
nltk.download('punkt_tab')
nltk.download('stopwords')

# Load file Excel
df = pd.read_excel('data_ulasan.xlsx')

print("Kolom dalam dataset:", df.columns)

# Kolom teks ulasan (pakai 'snippet')
text_data = df['snippet'].astype(str)

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt_tab.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
Kolom dalam dataset: Index(['name', 'link', 'thumbnail', 'date', 'snippet'], dtype='object')
```

case folding

```
import re
import string

def preprocess_text(text):
    # Memeriksa jika text adalah string
    if isinstance(text, str):
        # Mengubah text menjadi huruf kecil
        lower_case = text.lower()

        # Menghapus angka dari kalimat
        text = re.sub(r"\d+", "", lower_case)

        # Menghapus spasi pada awal dan akhir kalimat
        text = text.strip()

        # Menghapus Username
        text = re.sub(r'@[A-Za-z0-9_]+', '', text)

        # Menghapus https dan http
        text = re.sub(r"(?:\@|http?:\/\/|https?:\/\/|www)\S+", "", text)

        # Menghapus ..<br>
        text = re.sub(r'\.\.\<br>', ' ', text)

        # Mengganti <br> menjadi spasi
        text = re.sub(r'\<br\s*/?>', ' ', text)

        # Mengganti & dengan spasi
        text = re.sub(r'&', ' ', text)

        # Menghilangkan Tanda Baca
        text = text.translate(str.maketrans('', '', string.punctuation))
```

```

# Mengganti karakter HTML dengan tanda petik
text = re.sub('<.*?>', ' ', text)

# Mempertimbangkan huruf dan angka
text = re.sub('[^a-zA-Z0-9]', ' ', text)

# Mengganti line baru dengan spasi
text = re.sub("\n", " ", text)

# Menghilangkan emoji
emoji_pattern = re.compile("[
    u"\U0001F600-\U0001F64F"
    u"\U0001F300-\U0001F5FF"
    u"\U0001F680-\U0001F6FF"
    u"\U0001F1E0-\U0001F1FF"
    u"\U00002500-\U00002BEF"
    u"\U00002702-\U000027B0"
    u"\U000024C2-\U0001F251"
    u"\U0001f926-\U0001f937"
    u"\U00010000-\U0010ffff"
    u"\u200d"
    u"\u2640-\u2642"
    u"\u2600-\u2B55"
    u"\u23cf"
    u"\u23e9"
    u"\u231a"
    u"\u3030"
    u"\ufe0f"
    ]+", flags=re.UNICODE)
text = emoji_pattern.sub(r'', text)

```

```

]
    # Memisahkan dan menggabungkan kata
    text = ' '.join(text.split())

    return text
else:
    # Mengembalikan string kosong untuk nilai non-string
    return ""

]
▶ df['clean'] = df['snippet'].apply(preprocess_text)
# Menampilkan hanya kolom 'komentar' dan 'clean'
df[['snippet', 'clean']].head(736)

```

tokenizing

```
1 ]
▶ def tokenize_text(kalimat):
    tokens = nltk.tokenize.word_tokenize(kalimat)
    return tokens

df['token'] = df['clean'].apply(tokenize_text)
# Menampilkan hanya kolom 'clean' dan 'token'
df[['clean', 'token']].head(732)
```

stopword removal

```
1 ]
▶ from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
factory = StopWordRemoverFactory()
stopwords = factory.get_stop_words()

print(stopwords)

def stopword_text(tokens):
    cleaned_tokens = []
    for token in tokens:
        if token not in stopwords:
            cleaned_tokens.append(token)
    return cleaned_tokens

df['stop'] = df['token'].apply(stopword_text)
# Menampilkan hanya kolom 'token' dan 'stop'
df[['token', 'stop']].head(732)
```

stemming

```
1 ]

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
stem_factory = StemmerFactory()
stemmer = stem_factory.create_stemmer()

def stemming_text(tokens):
    hasil = [stemmer.stem(token) for token in tokens]
    return hasil

df['stemmed'] = df['stop'].apply(stemming_text)
# Menampilkan hanya kolom 'stop' dan 'stemmed'
df[['stop', 'stemmed']].head(732)
```

frekuensi kata

```
1 ]
▶ # Hitung frekuensi kata
# Gabungkan semua token jadi satu list
all_words = []

for tokens in df['stemmed']:
    all_words.extend(tokens)

word_freq = Counter(all_words)

print("\n20 kata yang paling sering muncul:")
for word, count in word_freq.most_common(20):
    print(f"{word}: {count}")

wc = WordCloud(
    width=800,
    height=400,
    background_color='white'
).generate_from_frequencies(word_freq)

plt.figure(figsize=(10,5))
plt.imshow(wc, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud dari Kolom Snippet', fontsize=16)
plt.show()
```

labeling

```
▶ # Load dataset
df = pd.read_excel("data_preprocessing.xlsx")

# Labeling Otomatis Berdasarkan Kata Kunci
positif_words = ['anak', 'nyaman', 'ramah', 'mantap', 'cocok', 'bagus', 'bersih', 'murah', 'recommended', 'recomend', 'puas', 'rekomended', 'top', 'juara', 'luas', 'terjangkau', 'w',
'vibesnya', 'jawa', 'santay', 'mantul', 'istimewa', 'lengkap', 'adem', 'bgus', 'banyak varian', 'gacor', 'bintang', 'fav', 'panteng', 'dikunjungi', 'berkualitas', 'asyik',
'seger', 'masyaallah', 'mampir', 'best', 'great', 'signature', 'sedia', 'gede', 'maxxook', 'manyap', 'enjoy', 'geerrrrr', 'cool', 'mantap', 'enaakk', 'chapter', 'sering',
negatif_words = ['mahal', 'tidak', 'kurang', 'jelek', 'kotor', 'lama', 'buruk', 'mengecewakan', 'parah', 'cuman', 'mati listrik', 'menyebalkan', 'belum standart', 'cancel', 'maaf']

def label_sentiment(text):
    text = text.lower()
    if any(word in text for word in positif_words):
        return 'positif'
    elif any(word in text for word in negatif_words):
        return 'negatif'

df['sentiment'] = df['clean'].astype(str).apply(label_sentiment)

# Lihat hasil
df[['clean', 'sentiment']].head(732)

#Catatan buat penambahan kata untuk negatif atau positif words

...
clean sentiment
```

hitung jumlah masing - masing label

```
▶ label_counts = df['sentiment'].value_counts()

print("Jumlah ulasan per label:")
print(label_counts)
```

```
*** Jumlah ulasan per label:
sentiment
positif      710
negatif       22
Name: count, dtype: int64
```

visualisasi

```
import matplotlib.pyplot as plt

plt.figure(figsize=(6,4))
label_counts.plot(kind='bar', color=['green', 'red', 'gray'])
plt.title('Distribusi Label Sentimen')
plt.xlabel('Kategori Sentimen')
plt.ylabel('Jumlah Ulasan')
plt.show()
```

balancing data

```
▶ from sklearn.utils import resample

# Pisahkan tiap label
df_pos = df[df['sentiment'] == 'positif']
df_neg = df[df['sentiment'] == 'negatif']

# Oversampling: samakan jumlah dengan kelas positif
df_neg_upsampled = resample(df_neg, replace=True, n_samples=len(df_pos), random_state=42)
# df_net_upsampled = resample(df_net, replace=True, n_samples=len(df_pos), random_state=42)

# Gabungkan kembali
df_balanced = pd.concat([df_pos, df_neg_upsampled])

print(df_balanced['sentiment'].value_counts())
```

```

▶ from sklearn.model_selection import train_test_split
  from sklearn.feature_extraction.text import TfidfVectorizer

  X = df_balanced['snippet']
  y = df_balanced['sentiment']

  X_train, X_test, y_train, y_test = train_test_split(
      X, y, test_size=0.2, random_state=42, stratify=y
  )

  tfidf = TfidfVectorizer(
      max_features=5000,
      ngram_range=(1,2), # unigram + bigram bisa mengenali frasa seperti "tidak enak"
      # stop_words='english' # boleh dihapus kalau full Bahasa Indonesia
  )

  X_train_tfidf = tfidf.fit_transform(X_train)
  X_test_tfidf = tfidf.transform(X_test)

```

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer

```

```

1 ▶ import pandas as pd
  from sklearn.model_selection import train_test_split
  from sklearn.feature_extraction.text import TfidfVectorizer

  # 1. Split Data

  X = df_balanced['snippet']
  y = df_balanced['sentiment']

  X_train, X_test, y_train, y_test = train_test_split(
      X, y,
      test_size=0.2,
      random_state=42,
      stratify=y
  )

  # 2. TF-IDF

  tfidf = TfidfVectorizer(
      max_features=5000,
      ngram_range=(1, 2)
  )

  X_train_tfidf = tfidf.fit_transform(X_train)
  X_test_tfidf = tfidf.transform(X_test)

```

```
[ ] ▶ # 3. Ubah ke DataFrame

feature_names = tfidf.get_feature_names_out()

tfidf_df = pd.DataFrame(
    X_train_tfidf.toarray(),
    columns=feature_names
)

# 4. PILIH FITUR PALING PENTING
# (AGAR TIDAK 0 SEMUA)
# Hitung rata-rata TF-IDF tiap fitur
tfidf_mean = tfidf_df.mean()

# Ambil 15 fitur terpenting
top_features = tfidf_mean.sort_values(ascending=False).head(15).index

# 5. TAMPILKAN BEBERAPA DOKUMEN

# Ambil 5 dokumen pertama
tfidf_display = tfidf_df.loc[:4, top_features]

print("Contoh Matriks TF-IDF (5 Dokumen, 15 Fitur Terpenting):")
print(tfidf_display)
```

```
# 6. SIMPAN KE EXCEL (LAPORAN)
```

```
tfidf_display.to_excel(
    "tfidf.xlsx",
    index=True
)
```

```
from sklearn.naive_bayes import MultinomialNB
```

```
nb = MultinomialNB()  
nb.fit(X_train_tfidf, y_train)  
y_pred_nb = nb.predict(X_test_tfidf)
```

```
from sklearn.svm import LinearSVC
```

```
svm = LinearSVC()  
svm.fit(X_train_tfidf, y_train)  
y_pred_svm = svm.predict(X_test_tfidf)
```

```
from sklearn.metrics import classification_report, confusion_matrix  
  
print("=== Naive Bayes ===")  
print(classification_report(y_test, y_pred_nb))  
print(confusion_matrix(y_test, y_pred_nb))  
  
print("\n=== SVM ===")  
print(classification_report(y_test, y_pred_svm))  
print(confusion_matrix(y_test, y_pred_svm))
```

```
from sklearn.metrics import confusion_matrix  
import matplotlib.pyplot as plt  
import numpy as np  
  
def plot_cm(y_true, y_pred, labels, title, cmap="Blues"):  
    cm = confusion_matrix(y_true, y_pred)  
  
    fig, ax = plt.subplots()  
    img = ax.imshow(cm, cmap=cmap)  
  
    ax.set_title(title)  
    ax.set_xticks(np.arange(len(labels)))  
    ax.set_yticks(np.arange(len(labels)))  
    ax.set_xticklabels(labels)  
    ax.set_yticklabels(labels)  
  
    # Tampilkan nilai setiap kotak  
    for i in range(cm.shape[0]):  
        for j in range(cm.shape[1]):  
            ax.text(j, i, cm[i, j], ha="center", va="center", color="black")  
  
    plt.colorbar(img)  
    plt.xlabel("Predicted")  
    plt.ylabel("Actual")  
    plt.show()
```

```
labels = ["Negatif", "Positif"]
```

```
# SVM  
plot_cm(y_test, y_pred_svm, labels, "Confusion Matrix - SVM", cmap="Blues")
```

```
# Naive Bayes  
plot_cm(y_test, y_pred_nb, labels, "Confusion Matrix - Naive Bayes", cmap="Greens")
```

```

from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import cross_val_score

pipeline_svm = Pipeline([
    ('tfidf', TfidfVectorizer(max_features=5000, ngram_range=(1,2))),
    ('svm', SVC())
])

pipeline_nb = Pipeline([
    ('tfidf', TfidfVectorizer(max_features=5000, ngram_range=(1,2))),
    ('nb', MultinomialNB())
])

print("\n=== Cross Validation (SVM) ===")
scores_svm = cross_val_score(pipeline_svm, X, y, cv=5)
print("Fold scores :", scores_svm)
print("Mean :", scores_svm.mean())

print("\n=== Cross Validation (Naive Bayes) ===")
scores_nb = cross_val_score(pipeline_nb, X, y, cv=5)
print("Fold scores :", scores_nb)
print("Mean :", scores_nb.mean())

=== Cross Validation (SVM) ===
Fold scores : [1. 1. 1. 1. 1.]
Mean : 1.0

=== Cross Validation (Naive Bayes) ===
Fold scores : [1.          0.99295775 0.99295775 0.98591549 0.9084507 ]
Mean : 0.976056338028169

▶ plt.figure(figsize=(8,5))
plt.plot(scores_svm, marker='o', label='SVM')
plt.plot(scores_nb, marker='o', label='Naive Bayes')
plt.title("Cross-Validation Fold Scores")
plt.xlabel("Fold")
plt.ylabel("Accuracy")
plt.legend()
plt.tight_layout()
plt.show()

```

```
# TABEL PERBANDINGAN KINERJA METODE
comparison_df = pd.DataFrame({
    "Metode": ["Naive Bayes", "SVM"],
    "Akurasi Rata-rata (%)": [
        scores_nb.mean() * 100,
        scores_svm.mean() * 100
    ]
})

comparison_df

# GRAFIK PERBANDINGAN METODE
plt.figure(figsize=(7,5))
plt.bar(
    comparison_df["Metode"],
    comparison_df["Akurasi Rata-rata (%)"]
)

plt.title("Perbandingan Akurasi Metode Naive Bayes dan SVM")
plt.xlabel("Metode Klasifikasi")
plt.ylabel("Akurasi (%)")
plt.ylim(0, 100)
plt.tight_layout()
plt.show()
```

Lampiran 4 Data Ulasan

Data Ulasan Lengkapnya :

<https://drive.google.com/drive/folders/1nyjK13l4OIQ6oY44lIZw8Qbyf6TBCFIQ?usp=sharing>

No	name	snippet
1.	AL	"Pulang dr Dieng saya lewat jalur selatan krn, direkomendasikan teman jika lewat purbalingga mampir ke cafe ini. 1.Cafe luas pilihan meja biasa atau saung (semua outdoor) saya senang suasananya cuma terganggu dng asap rokok krn kami satu keluarga tidak merokok, jd klu yg ga merokok cari tempat duduk yg agak jauh dr yg lain krn memang konsepnya terbuka 2.Parkir: bisa didepan cafe atau lapangan sebelah, parkirannya luas 3. Menu makanan rumahan, comfort food (konsep prasmanan seperti kopi kltk) untuk sayur yg saya pilih kebetulan pedas semua(yg ga suka pedas pilih yang lain saja) sayur lodeh dan tumis pepaya, rasa khas masakan rumah jawa, gurih dan agak manis cuma buat saya kepedesan aja. Ayam empuk, semua bumbu dilauk yang saya coba cocok dilidah saya 3. side dish mendoan, pisang, singkong semua enak (akhirnya dibungkus dan dimakan diperjalanan) 4. recommended coffee: -pistachio latte (blend coffee) - coffee batok (ciri khas pakai cup batok) 5. Affordable price (tergantung persepsi masing2) 6.Karyawan ramah2 dan sigap membantu kalau lewat purbalingga, nanti mampir lagi. semoga sukses selalu"
2.	Z	"Beberapa kali makan di sini bareng temen dan keluarga. Suasananya lumayan enak dan makanan ada yang prasmanan juga jadi gak menunggu lama. Menu yang menarik bisa pesan bebek cuma perlu nunggu aja. Kalau prasmanan pilihannya banyak ada telur, ikan, ayam. Ada mushola dan toiletnya juga bersih. Parkirnya juga cukup luas dan muat untuk banyak mobil."
3.	Djoko Setijono	"Resto yg cukup artistik terutama mebelernya dari kayu2 yg ukurannya cukup besar, menu makan n minumannya enak, ownernya ramah sekali"
4.	Filbert Tanaka	"Makanannya enak, harga relatif terjangkau. Pelayanan juga ramah dan cepat."
5.	eprina exa	"Tempat enak Pelayanan baik Makanan enak Harga terjangkau Tengkyu"
6.	Dita Partowihardjo	"Makanannya cenderung manis dan tidak pedas. Banyak pilihan menu bervariasi. Pelayanan sangat ramah, cepat tanggap. Suasana nyaman. Parkiran juga mudah."
7.	muhammad mufti ramadhan	"Tempatnya enak buat makan apalagi dengan suasana joglo. Untuk rasa tumben hari ini rasanya enak biasanya rasanya dominan manis sepertinya yang masak lagi happy karena enak rasanya. Tolong untuk yang masak di pertahankan rasanya ya karena kadang lebih dominan manis. Untuk kopi nya enak kebetulan pesen ice coffee late. Terus juga untuk pisang gorengnya juga enak. Untuk harga standard ga begitu mahal karena nasi sayur 11.500 sampai 12.500."
8.	Shiva Nabila	"Warung kopi bathok selalu rame, makanannya enak, wajarlah kalau ada kekurangannya pas rame tapi selama saya jadi pelanggan belum pernah kecewa."