DAFTAR LAMPIRAN

Lampiran 1. Surat Bebas Plagiat

UNIVERSITAS DARMA PERSADA
UPT PERPUSTAKAAN
Gedung Rektorat Lantai 3,
Jl.Taman Malaka Selatan, Pondok Kelapa – Jakarta Timur 13450

SURAT KETERANGAN
HASIL PENGECEKAN TURNITIN

UPT Perpustakaan Universitas Darma Persada menerangkan telah selesai melakukan pemeriksaan duplikasi/*similarity* menggunakan perangkat lunak Turnitin terhadap hasil karya sebagai berikut:

Judul : ANALISIS SENTIMEN PUBLIK DI INSTAGRAM DAN X TERHADAP PEMBATALAN PEMBELIAN BBM DARI PERTAMINA OLEH SPBU SWASTA MENGGUNAKAN METODE BERT

Penulis : Alfian Rusydi Hakim
NIM : **2019230021**
Tgl pemeriksaan : 13 Februari 2026

Dengan hasil Tingkat Kesamaan (*similarity index*) **24%**

Demikian Surat Keterangan kami buat, untuk dipergunakan sebagaimana mestinya.

Jakarta, 13 Februari 2026

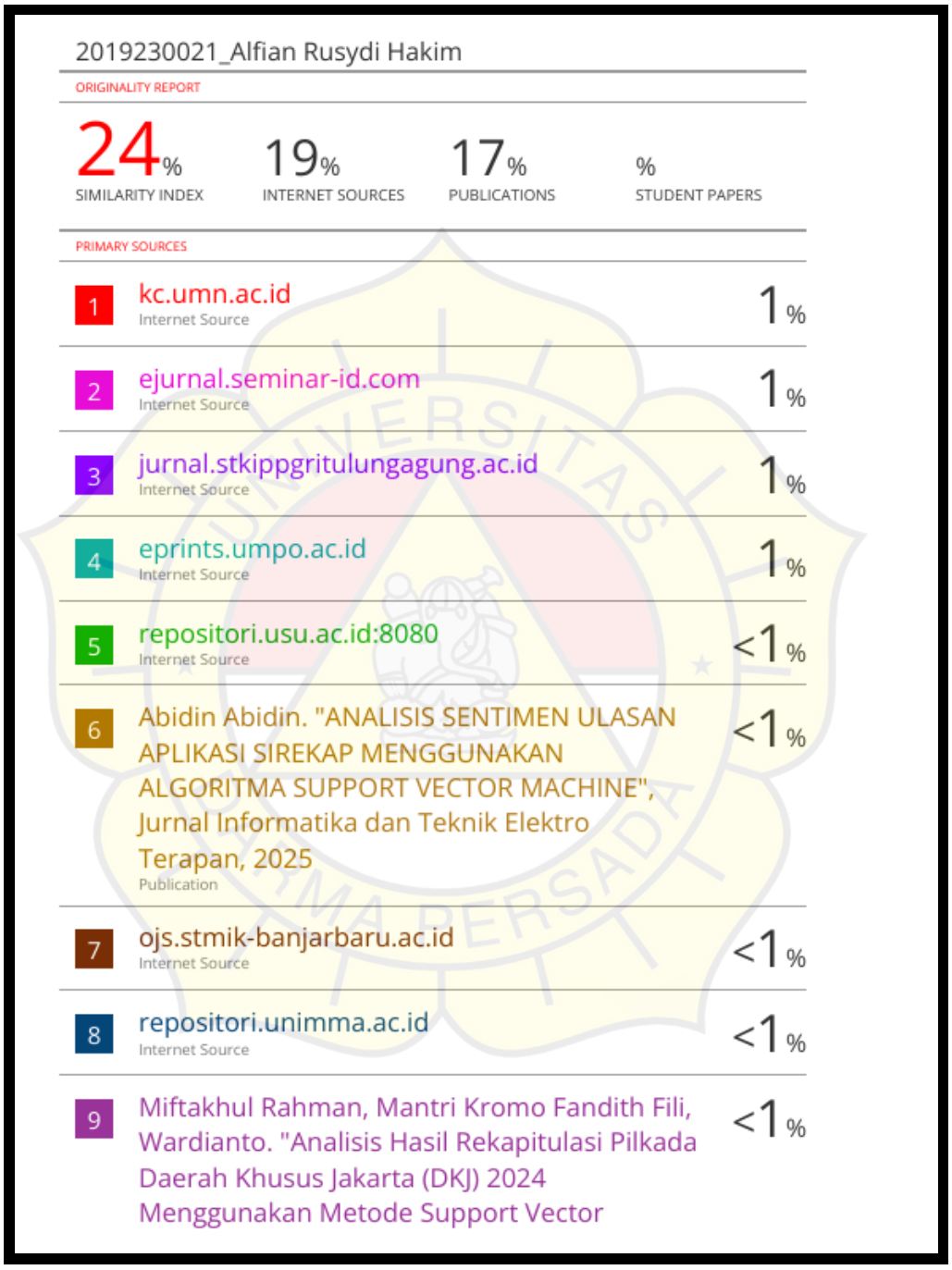
Ka.UPT Perpustakaan Unsada

Yus Rusmiyati, SS., MM

Batas maksimal similarity 30% untuk Fakultas Sastra dan Ekonomi

Batas maksimal similarity 25% untuk Fakultas Teknik, Kelautan dan Pasca Sarjana

Lampiran 2. Bebas Turnitin



2019230021_Alfian Rusydi Hakim

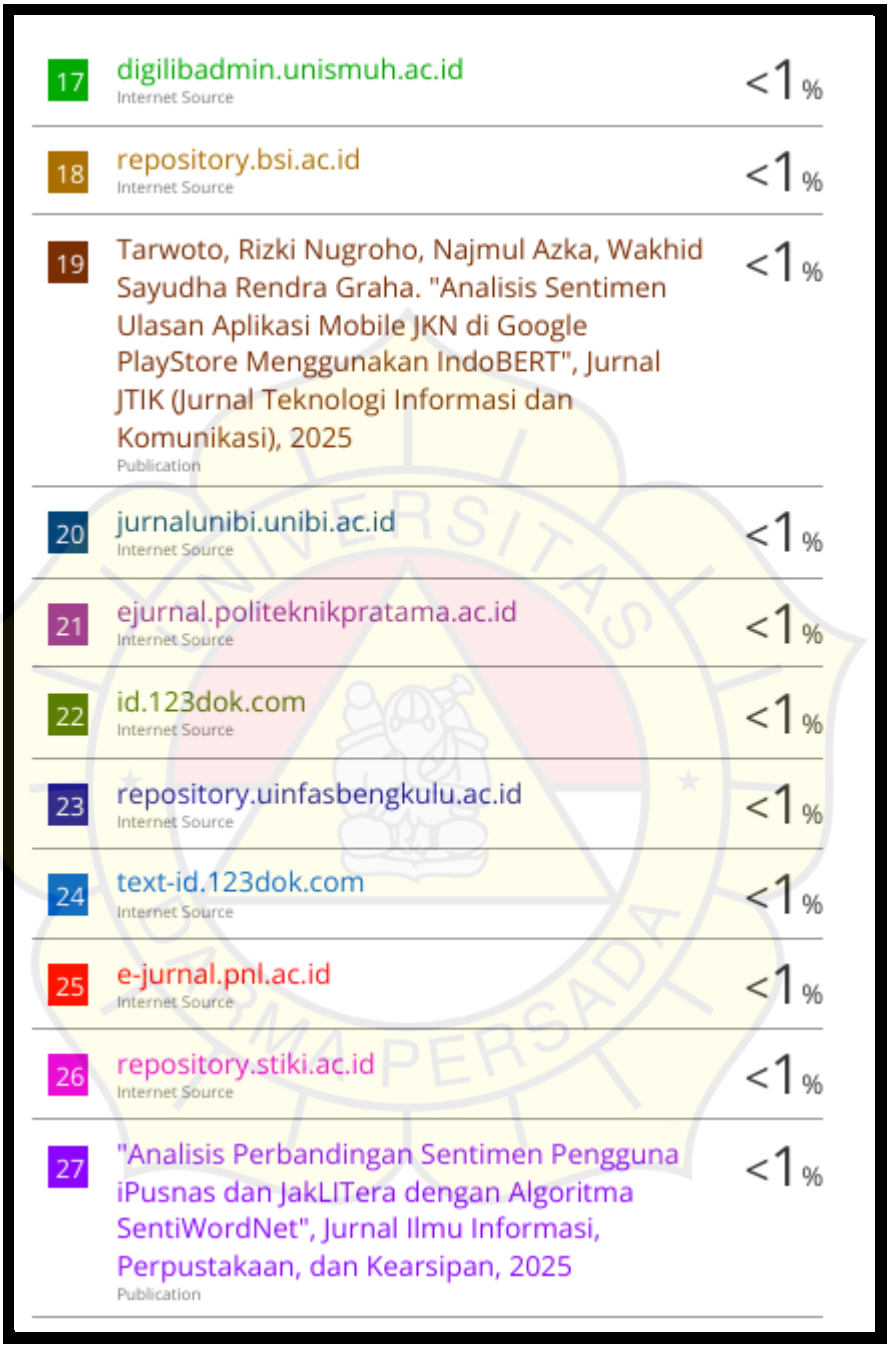
ORIGINALITY REPORT

24%	19%	17%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

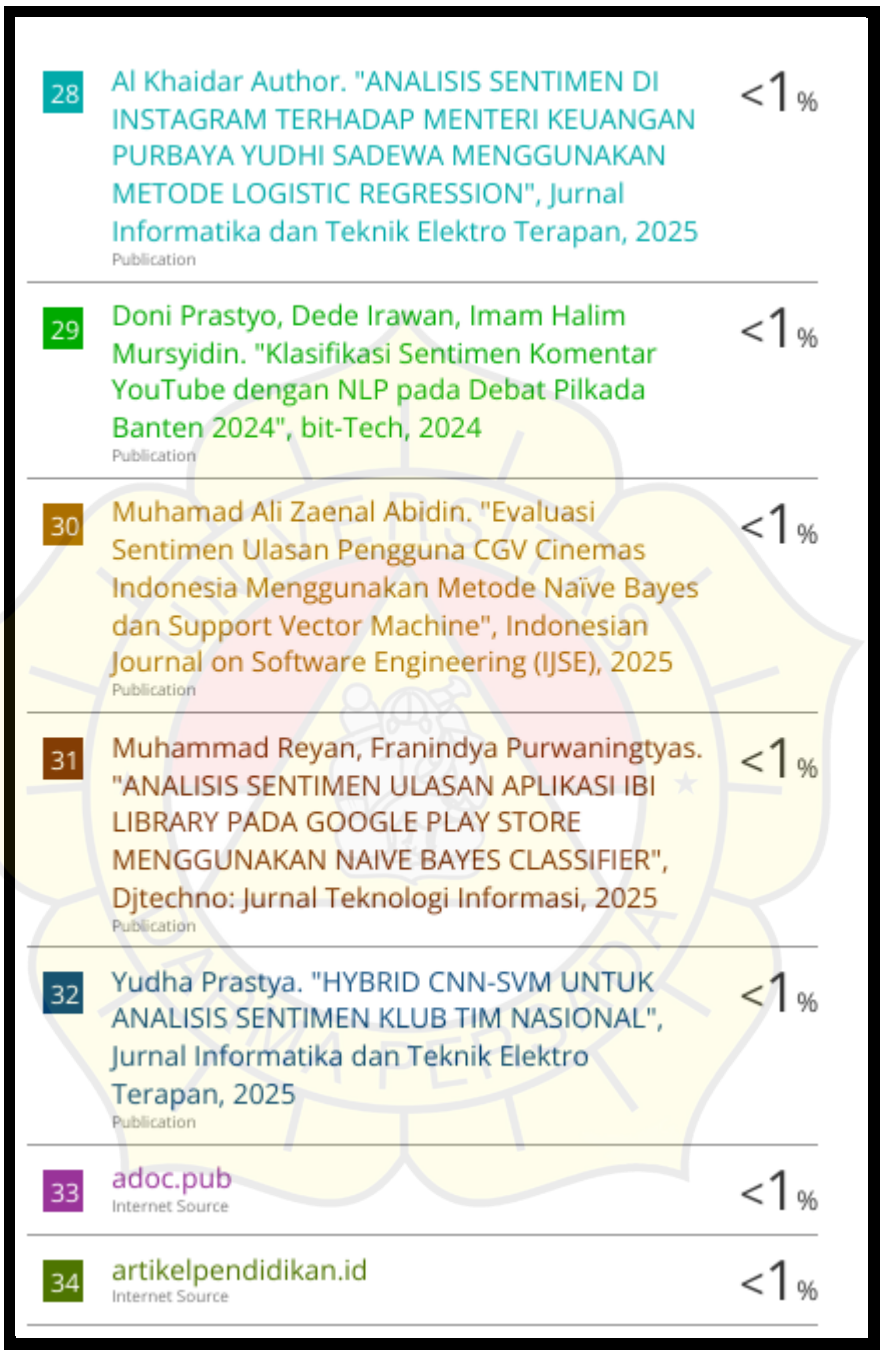
PRIMARY SOURCES

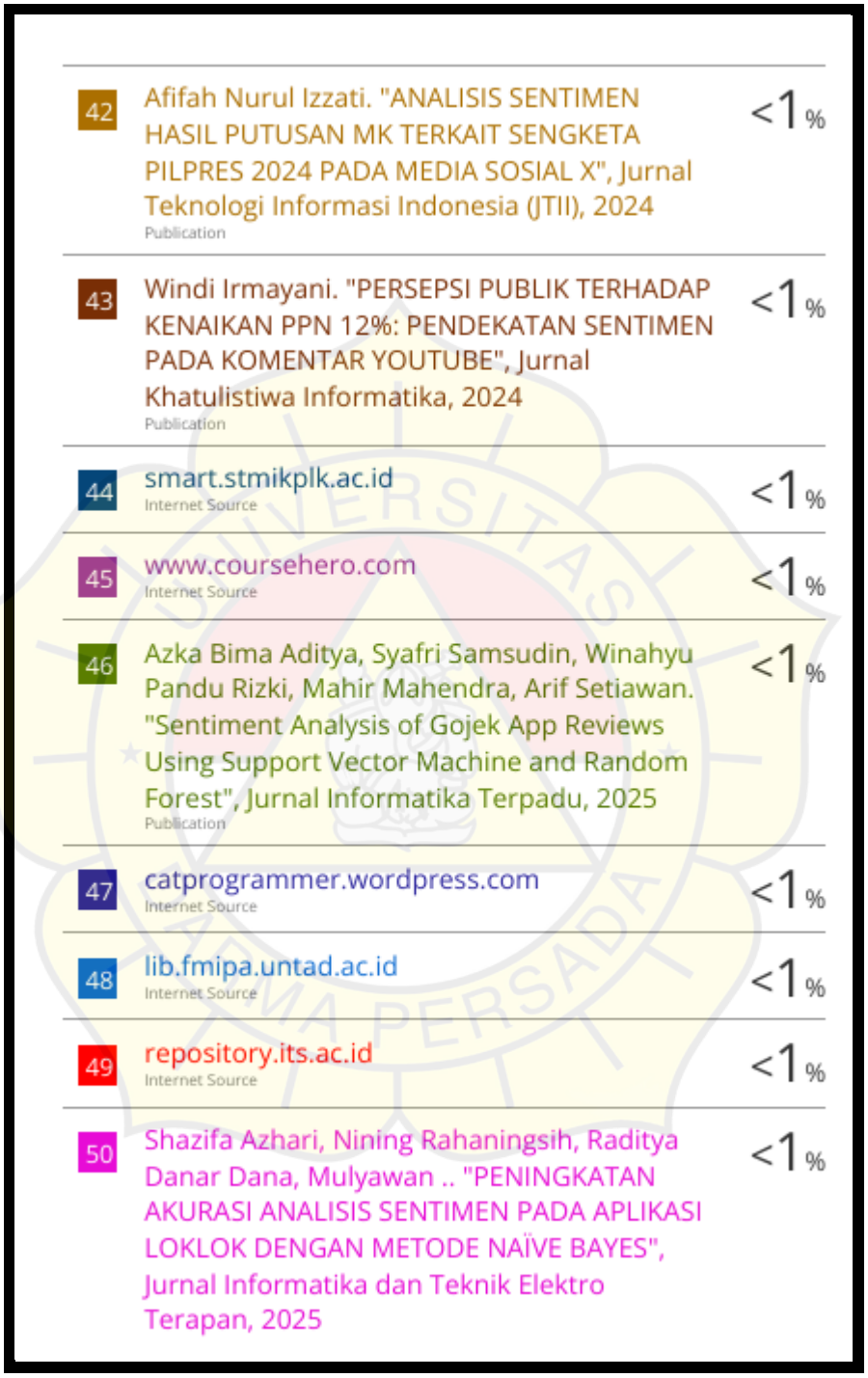
1	kc.umn.ac.id Internet Source	1%
2	ejurnal.seminar-id.com Internet Source	1%
3	jurnal.stkippgritulungagung.ac.id Internet Source	1%
4	eprints.umpo.ac.id Internet Source	1%
5	repositori.usu.ac.id:8080 Internet Source	<1%
6	Abidin Abidin. "ANALISIS SENTIMEN ULASAN APLIKASI SIREKAP MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE", Jurnal Informatika dan Teknik Elektro Terapan, 2025 Publication	<1%
7	ojs.stmik-banjarbaru.ac.id Internet Source	<1%
8	repositori.unimma.ac.id Internet Source	<1%
9	Miftakhul Rahman, Mantri Kromo Fandith Fili, Wardianto. "Analisis Hasil Rekapitulasi Pilkada Daerah Khusus Jakarta (DKJ) 2024 Menggunakan Metode Support Vector	<1%

10	Saikin Saikin, Mohammad Taufan Asri Zaen, Sofiansyah Fadli, Hairul Fahmi. "Penerapan Algoritma BERT dalam Analisis Sentimen Opini Publik terhadap Destinasi Wisata dengan Metode CRISP-DM", RIGGS: Journal of Artificial Intelligence and Digital Business, 2025 Publication	<1 %
11	repository.umsu.ac.id Internet Source	<1 %
12	Ichsani Mursidah, Remi Sanjaya, Bambang Yulianto, Dhian Sweetania, Puji Sularsih. "Klasifikasi Sentimen Google Play Store Aplikasi ChatGPT Berbahasa Indonesia Berbasis IndoBERT", Jurnal Minfo Polgan, 2025 Publication	<1 %
13	journal.admi.or.id Internet Source	<1 %
14	Diah Fatma Sjoraida, Bucky Wibawa Karya Guna, Dudi Yudhakusuma. "Analisis Sentimen Film Dirty Vote Menggunakan BERT (Bidirectional Encoder Representations from Transformers)", Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 2024 Publication	<1 %
15	docplayer.info Internet Source	<1 %
16	etheses.uin-malang.ac.id Internet Source	<1 %

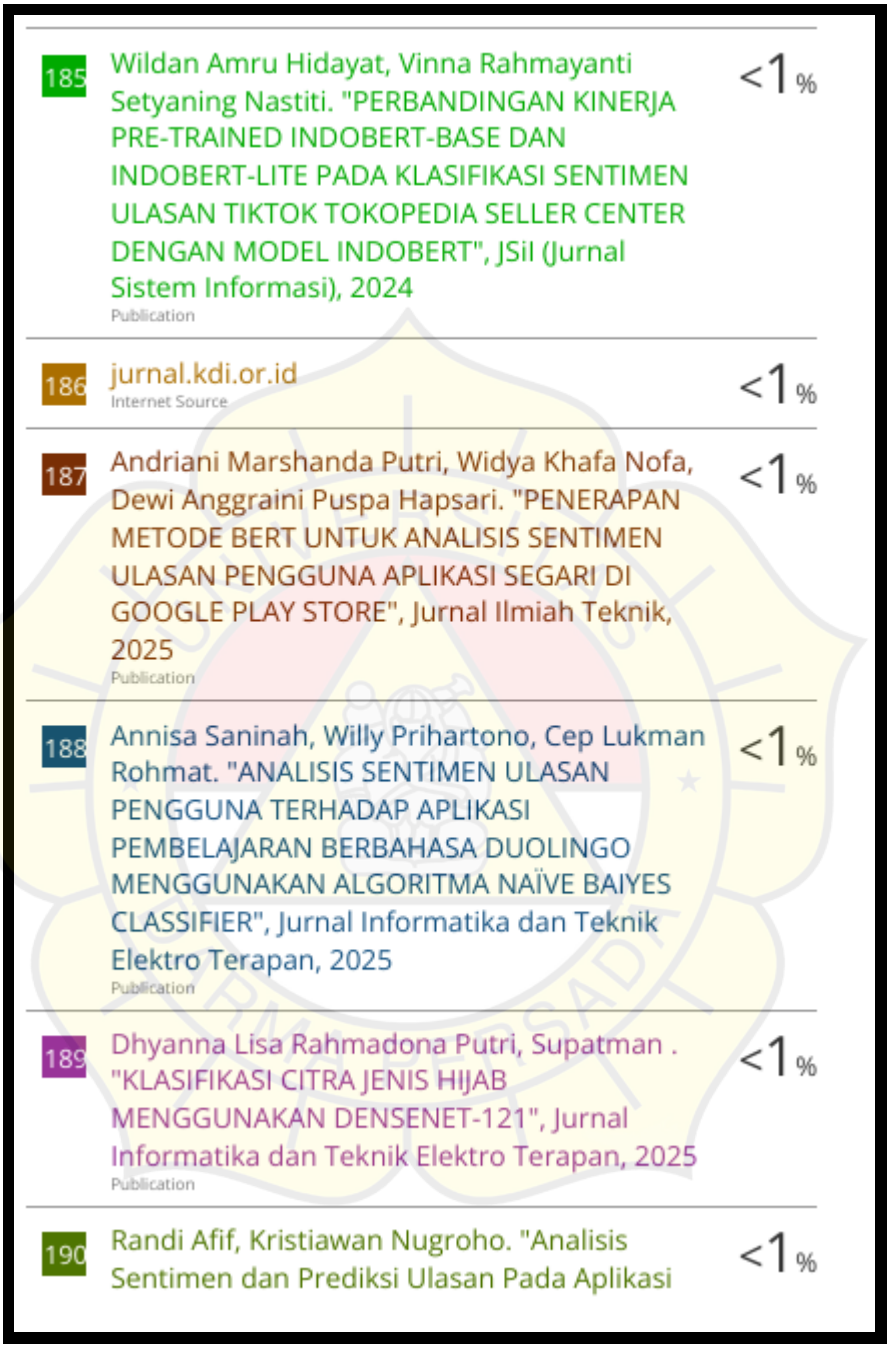


17	digilibadmin.unismuh.ac.id Internet Source	<1%
18	repository.bsi.ac.id Internet Source	<1%
19	Tarwoto, Rizki Nugroho, Najmul Azka, Wakhid Sayudha Rendra Graha. "Analisis Sentimen Ulasan Aplikasi Mobile JKN di Google PlayStore Menggunakan IndoBERT", Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 2025 Publication	<1%
20	jurnalunibi.unibi.ac.id Internet Source	<1%
21	ejurnal.politeknikpratama.ac.id Internet Source	<1%
22	id.123dok.com Internet Source	<1%
23	repository.uinfasbengkulu.ac.id Internet Source	<1%
24	text-id.123dok.com Internet Source	<1%
25	e-jurnal.pnl.ac.id Internet Source	<1%
26	repository.stiki.ac.id Internet Source	<1%
27	"Analisis Perbandingan Sentimen Pengguna iPusnas dan JakLITera dengan Algoritma SentiWordNet", Jurnal Ilmu Informasi, Perpustakaan, dan Kearsipan, 2025 Publication	<1%





42	Afifah Nurul Izzati. "ANALISIS SENTIMEN HASIL PUTUSAN MK TERKAIT SENGKETA PILPRES 2024 PADA MEDIA SOSIAL X", Jurnal Teknologi Informasi Indonesia (JTII), 2024 Publication	<1 %
43	Windi Irmayani. "PERSEPSI PUBLIK TERHADAP KENAIKAN PPN 12%: PENDEKATAN SENTIMEN PADA KOMENTAR YOUTUBE", Jurnal Khatulistiwa Informatika, 2024 Publication	<1 %
44	smart.stmikplk.ac.id Internet Source	<1 %
45	www.coursehero.com Internet Source	<1 %
46	Azka Bima Aditya, Syafri Samsudin, Winahyu Pandu Rizki, Mahir Mahendra, Arif Setiawan. "Sentiment Analysis of Gojek App Reviews Using Support Vector Machine and Random Forest", Jurnal Informatika Terpadu, 2025 Publication	<1 %
47	catprogrammer.wordpress.com Internet Source	<1 %
48	lib.fmipa.untad.ac.id Internet Source	<1 %
49	repository.its.ac.id Internet Source	<1 %
50	Shazifa Azhari, Nining Rahaningsih, Raditya Danar Dana, Mulyawan .. "PENINGKATAN AKURASI ANALISIS SENTIMEN PADA APLIKASI LOKLOK DENGAN METODE NAÏVE BAYES", Jurnal Informatika dan Teknik Elektro Terapan, 2025	<1 %



185	Wildan Amru Hidayat, Vinna Rahmayanti Setyaning Nastiti. "PERBANDINGAN KINERJA PRE-TRAINED INDOBERT-BASE DAN INDOBERT-LITE PADA KLASIFIKASI SENTIMEN ULASAN TIKTOK TOKOPEDIA SELLER CENTER DENGAN MODEL INDOBERT", JSil (Jurnal Sistem Informasi), 2024 Publication	<1 %
186	jurnal.kdi.or.id Internet Source	<1 %
187	Andriani Marshanda Putri, Widya Khafa Nofa, Dewi Anggraini Puspa Hapsari. "PENERAPAN METODE BERT UNTUK ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI SEGARI DI GOOGLE PLAY STORE", Jurnal Ilmiah Teknik, 2025 Publication	<1 %
188	Annisa Saninah, Willy Prihartono, Cep Lukman Rohmat. "ANALISIS SENTIMEN ULASAN PENGGUNA TERHADAP APLIKASI PEMBELAJARAN BERBAHASA DUOLINGO MENGGUNAKAN ALGORITMA NAÏVE BAIYES CLASSIFIER", Jurnal Informatika dan Teknik Elektro Terapan, 2025 Publication	<1 %
189	Dhyanna Lisa Rahmadona Putri, Supatman . "KLASIFIKASI CITRA JENIS HIJAB MENGGUNAKAN DENSENET-121", Jurnal Informatika dan Teknik Elektro Terapan, 2025 Publication	<1 %
190	Randi Afif, Kristiawan Nugroho. "Analisis Sentimen dan Prediksi Ulasan Pada Aplikasi	<1 %

Lampiran 3. Penyaringan Kata Slang dan Tidak Di Pakai

```
import re

import string

import pandas as pd

fromSastrawi.StopWordRemover.StopWordRemoverFactory            import
StopWordRemoverFactory


# --- 1. Load Data/Resources ---


# KAMUS SLANG: Tambahkan lebih banyak slang sesuai data Anda

SLANG_DICT = {

"yg": "yang", "tdk": "tidak", "bgt": "banget", "gpp": "tidak apa-apa",

"ga": "tidak", "klo": "kalau", "krn": "karena", "k": "ke", "dgn":
"dengan",

"bkn": "bukan", "dr": "dari", "udah": "sudah", "blm": "belum", "jgn":
"jangan",

"btw": "omong-omong", "wkwk": "tertawa", "si": "saja", "ok": "oke"

}


# STOPWORD: Menggunakan Sastrawi dan stopword kustom

factory = StopWordRemoverFactory()

stopword_sastrawi = factory.get_stop_words()


# Daftar kata tambahan yang sering muncul tapi tidak penting (kata
gajelas)

additional_stopwords = [

'sih', 'nya', 'kali', 'dong', 'deh', 'gue', 'lu', 'ini', 'itu', 'udah',
'gak',

'yang', 'di', 'dan', 'ke', 'dari', 'untuk', 'pada', 'adalah', 'ini',
'itu', 'atau',

'juga', 'dengan', 'saya', 'kita', 'kami', 'anda', 'dia', 'mereka',
'akan', 'bisa',
```

```
'ada', 'tidak', 'tak', 'tiada', 'bukan', 'belum', 'jangan', 'kenapa',
'mengapa',

'kapan', 'dimana', 'bagaimana', 'apa', 'siapa', 'aja', 'saja', 'kok',
'kan',

'mah', 'tuh', 'dong', 'yah', 'wow', 'lho', 'loh', 'yuk', 'mari',
'halo', 'hai',

'oke', 'ok', 'sip', 'ya', 'tidak', 'iya', 'enggak', 'nggak', 'gak',
'gk',

'yg', 'utk', 'dgn', 'kalo', 'klo', 'krn', 'karena', 'tapi', 'tp', 'jd',
'jdi',

'jadi', 'bgt', 'banget', 'sama', 'sm', 'sdh', 'sudah', 'blm', 'belum',
'dr',

'terima', 'kasih', 'mohon', 'tolong', 'silakan', 'selamat', 'pagi',
'siang',

'sore', 'malam', 'hari', 'ini', 'besok', 'kemarin', 'lalu', 'nanti',
'tadi',

'barusan', 'sekarang', 'saat', 'waktu', 'jam', 'menit', 'detik',
'tahun',

'bulan', 'minggu', 'tanggal', 'senin', 'selasa', 'rabu', 'kamis',
'jumat',

'sabtu', 'minggu', 'januari', 'februari', 'maret', 'april', 'mei',
'juni',

'juli', 'agustus', 'september', 'oktober', 'november', 'desember',

'satu', 'dua', 'tiga', 'empat', 'lima', 'enam', 'tujuh', 'delapan',
'sembilan', 'sepuluh',

'banyak', 'sedikit', 'kurang', 'lebih', 'cukup', 'hampir', 'sekitar',
'kira',

'sangat', 'paling', 'terlalu', 'makin', 'semakin', 'justru', 'malah',
'malahan',

'tentu', 'pasti', 'yakin', 'percaya', 'harap', 'semoga', 'mudah',
'sulit',

'bagus', 'baik', 'buruk', 'jelek', 'keren', 'mantap', 'oke', 'sip',

'dan', 'atau', 'tetapi', 'melainkan', 'padahal', 'sedangkan', 'serta',
'lagipula',

'karena', 'sebab', 'jika', 'kalau', 'jikalau', 'apabila', 'bila',
'bilamana',

'supaya', 'agar', 'biar', 'sehingga', 'maka', 'akibatnya',
'karenanya',
```

```
'seperti', 'bagai', 'bagaikan', 'laksana', 'ibarat', 'umpama',
'contoh',

'misal', 'misalnya', 'contohnya', 'yaitu', 'yakni', 'ialah', 'adalah',

'dalam', 'luar', 'atas', 'bawah', 'kiri', 'kanan', 'depan',
'belakang',

'samping', 'tengah', 'pusat', 'pinggir', 'ujung', 'pangkal', 'awal',
'akhir',

'bahlul', 'bapak', 'pak', 'mau', 'buat', 'pake', 'nan', 'lah', 'nih',
'tuh',

'emang', 'gw', 'kalian', 'cuma', 'gini', 'bikin', 'jadi', 'lebih',
'biar',

'masalah', 'karena', 'kalau', 'jika', 'seperti', 'yaitu', 'yakni',
'contoh', 'bahlil', 'kontol', 'bgst', 'anjing',

'bangsat', 'ngentot', 'goblok', 'tolol', 'bego', 'kampret', 'jancuk',
'tai', 'tahi', 'bajingan', 'brengsek', 'sialan',

'setan', 'fuck', 'fuckup', 'lanjing', 'ngopo', 'oplosan'

]


custom_stopwords = set(stopword_sastrawi + additional_stopwords)


# --- 2. Fungsi Utama Preprocessing ---


def normalize_slang(text):

"""Mengganti kata-kata slang dengan kata baku."""

words = text.split()

normalized_words = [SLANG_DICT.get(word, word) for word in words]

return " ".join(normalized_words)


def remove_stopwords(text):

"""Menghapus kata-kata tidak penting (stopwords)."""

words = text.split()

filtered_words = [word for word in words if word not in
custom_stopwords]

return " ".join(filtered_words)
```

```
def preprocess_text(text):
"""Melakukan semua langkah preprocessing NLP kompleks."""
if pd.isna(text) or not text:
return ""

text = str(text).lower()

# 1. Cleaning Teks Medsos: URL, Mentions, Hashtags, Newlines
text = re.sub(r'http\S+|www\S+|https\S+|ftp\S+', '', text, flags=re.MULTILINE) # URL
text = re.sub(r'@\w+', '', text) # Mentions
text = re.sub(r'#\w+', '', text) # Hashtags
text = re.sub(r'\n', ' ', text) # Newline

# 2. Menghapus Emoji
emoji_pattern = re.compile(
'['
'\U0001F600-\U0001F64F'
'\U0001F300-\U0001F5FF'
'\U0001F680-\U0001F6FF'
'\U0001F1E0-\U0001F7FF'
'\U00002702-\U000027B0'
']+',
flags=re.UNICODE
)
text = emoji_pattern.sub(r'', text)

# 3. Menghapus Tanda Baca dan Angka
text = text.translate(str.maketrans('', '', string.punctuation))
text = re.sub(r'\d+', '', text)
```

```
# 4. Normalisasi Slang (NLP Kunci)
text = normalize_slang(text)

# 5. Penghapusan Stopword (NLP Kunci)
text = remove_stopwords(text)

# 6. Menghapus spasi berlebihan
text = re.sub(r'\s+', ' ', text).strip()

return text

def preprocess_dataframe(df, text_column):
"""Menerapkan preprocessing ke seluruh kolom DataFrame."""
if text_column not in df.columns:
raise KeyError(f"Kolom sumber '{text_column}' tidak ditemukan. Kolom
tersedia: {list(df.columns)}")

df = df.dropna(subset=[text_column])
df['text_clean'] = df[text_column].apply(preprocess_text)
return df
```

Lampiran 4. Pelatihan Data Sentimen

```
import numpy as np

import torch

import joblib

import os

import sys


from sklearn.model_selection import train_test_split

from sklearn.metrics import classification_report, f1_score,
recall_score, confusion_matrix

from transformers import BertTokenizer,
BertForSequenceClassification, Trainer, TrainingArguments

from datasets import Dataset


# --- Import NLP Utilitas Kompleks ---

try:

    from nlp_utils import preprocess_dataframe

except ImportError:

    print(" FATAL ERROR: Modul 'nlp_utils.py' tidak ditemukan.")

    sys.exit(1)


# --- 1. Konfigurasi Global ---

MODEL_NAME = "indobenchmark/indobert-base-p1"

MAX_LENGTH = 128

NUM_LABELS = 3

OUTPUT_DIR = "./sentiment_model"

DATA_PATH = "data_training.xlsx"

LABEL_MAPPER = {'negatif': 0, 'positif': 1, 'netral': 2}

TEXT_COL_ORIGINAL = 'text'
```

```
os.makedirs(OUTPUT_DIR, exist_ok=True)


# --- 2. Load Dataset dan Preprocessing (NLP) ---
print(f"Mengambil data dari: {DATA_PATH}")


try:
    if DATA_PATH.endswith('.csv'):
        df = pd.read_csv(DATA_PATH, encoding='latin-1')
    elif DATA_PATH.endswith('.xlsx') or DATA_PATH.endswith('.xls'):
        df = pd.read_excel(DATA_PATH)
    else:
        raise ValueError("Format file data tidak didukung. Gunakan
.csv, .xlsx, atau .xls.")


    if 'text' not in df.columns or 'label' not in df.columns:
         raise KeyError("Kolom 'text' atau 'label' tidak ditemukan
di file data.")


except Exception as e:
    print(f" ERROR Pemuatan Data: {e}")
    data = {'text': ["SPBU swasta batal, ini keputusan yang sangat
bagus!", "Kebijakan ini merugikan kita semua, saya sangat kecewa.",
"Tidak ada komentar, saya netral dan menunggu keputusan
Pertamina."],
            'label': ['positif', 'negatif', 'netral']}
    df = pd.DataFrame(data * 50)
    df.to_excel(DATA_PATH, index=False)
    print(f"Data simulasi dibuat di {DATA_PATH}. Jalankan ulang
skrip.")
    sys.exit(1)


# ----------------------------------------------------
# PENANGANAN DATA CLEANING (NaN, Mapping, Minority Class)
```

```
# ----------------------------------------------------
print(f"Jumlah baris awal: {len(df)}")

# 1. Cleaning NaN dan Label Mapping
df.dropna(subset=['label'], inplace=True)
df['label'] = df['label'].astype(str).str.lower().str.strip()
df['label'] = df['label'].map(LABEL_MAPPER)
df.dropna(subset=['label'], inplace=True)
df['label'] = df['label'].astype(int)

print(f"Jumlah baris setelah cleaning dan mapping: {len(df)}")

# Lakukan Preprocessing
print("Mulai Preprocessing Teks (NLP Kompleks)...")
df_processed = preprocess_dataframe(df, TEXT_COL_ORIGINAL)

# 2. Penanganan Kelas Minoritas (Untuk Stratifikasi)
class_counts = df_processed['label'].value_counts()
print("\nDistribusi Kelas Sebelum Split:")
print(class_counts)

classes_to_remove = class_counts[class_counts < 2].index

if not classes_to_remove.empty:
    print(f"\n Peringatan: Kelas-kelas dengan label
{classes_to_remove.tolist()} dihapus karena hanya memiliki kurang
dari 2 anggota.")
    df_processed =
df_processed[~df_processed['label'].isin(classes_to_remove)]
    print(f"Jumlah baris setelah menghapus kelas minoritas:
{len(df_processed)}")
```

```
# Cek final sebelum split
if len(df_processed) < 2:
    print(" ERROR: Data terlalu sedikit setelah pembersihan. Tidak bisa melakukan splitting.")
    sys.exit(1)

# Pisahkan data (Aman dari error minority class)
train_df, test_df = train_test_split(df_processed, test_size=0.2, stratify=df_processed['label'], random_state=42)

# Koreksi: MENGHINDARI KOLOM DUPLIKAT SEBELUM CONVERSI KE DATASET
cols_to_keep = ['text_clean', 'label']
train_df = train_df[cols_to_keep]
test_df = test_df[cols_to_keep]

# Konversi DataFrame ke format Hugging Face Dataset
train_dataset = Dataset.from_pandas(train_df.rename(columns={'text_clean': 'text'}))
test_dataset = Dataset.from_pandas(test_df.rename(columns={'text_clean': 'text'}))

# --- 3. Tokenizer dan Model ---
print("Memuat Tokenizer dan Model IndoBERT...")
tokenizer = BertTokenizer.from_pretrained(MODEL_NAME)
model = BertForSequenceClassification.from_pretrained(MODEL_NAME, num_labels=NUM_LABELS)

def tokenize_function(examples):
    """Fungsi Tokenisasi BERT/IndoBERT"""
    return tokenizer(examples['text'], truncation=True, padding='max_length', max_length=MAX_LENGTH)

tokenized_train = train_dataset.map(tokenize_function, batched=True)
```

```
tokenized_test = test_dataset.map(tokenize_function, batched=True)


# Format PyTorch dan hapus kolom yang tidak diperlukan

cols_to_remove = [col for col in ['__index_level_0__'] if col in
tokenized_train.column_names]


tokenized_train =
tokenized_train.remove_columns(cols_to_remove).rename_column("label"
, "labels")

tokenized_test =
tokenized_test.remove_columns(cols_to_remove).rename_column("label",
"labels")

tokenized_train.set_format("torch", columns=['input_ids',
'attention_mask', 'labels'])

tokenized_test.set_format("torch", columns=['input_ids',
'attention_mask', 'labels'])


# --- 4. Fungsi Metrik dan Training ---

def compute_metrics(p):

    """Fungsi untuk menghitung metrik utama: F1, Recall, Confusion
Matrix."""

    preds = np.argmax(p.predictions, axis=1)

    labels = p.label_ids


    # Hitung metrik

    f1_micro = f1_score(labels, preds, average='micro')

    recall_macro = recall_score(labels, preds, average='macro',
zero_division=0)


    #  KOREKSI FINAL: Tambahkan labels=[0, 1, 2] untuk mengatasi
missing classes di evaluation set

    report = classification_report(labels, preds, output_dict=True,
zero_division=0,

      labels=[0, 1, 2],

      target_names=['Negatif', 'Positif', 'Netral'])
```

```
    # Simpan metrik lengkap ke file joblib
    metrik_data = {
        'model_path': MODEL_NAME,
        'accuracy': report.get('accuracy', 0.0),
        'f1_score_micro': f1_micro,
        'recall_score_macro': recall_macro,
        'classification_report': report,
        'confusion_matrix': confusion_matrix(labels, preds).tolist()
    }
    joblib.dump(metrik_data, f"{OUTPUT_DIR}/training_metrics.pkl")
    print("\n Metrik training berhasil disimpan.")

    return {"f1_micro": f1_micro, "recall_macro": recall_macro,
"accuracy": report.get('accuracy', 0.0)}


training_args = TrainingArguments(
    output_dir=OUTPUT_DIR,
    num_train_epochs=3,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=100,
    eval_strategy="epoch", # Koreksi dari evaluation_strategy
    save_strategy="epoch",
    load_best_model_at_end=True,
    metric_for_best_model="f1_micro",
)
```

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_train,
    eval_dataset=tokenized_test,
    tokenizer=tokenizer,
    compute_metrics=compute_metrics,
)


print("\n--- Mulai Fine-Tuning IndoBERT ---")
trainer.train()


# Simpan model terbaik dan tokenizer
model.save_pretrained(OUTPUT_DIR)
tokenizer.save_pretrained(OUTPUT_DIR)
print(f"--- Model dan Tokenizer berhasil disimpan di {OUTPUT_DIR} ---")
```