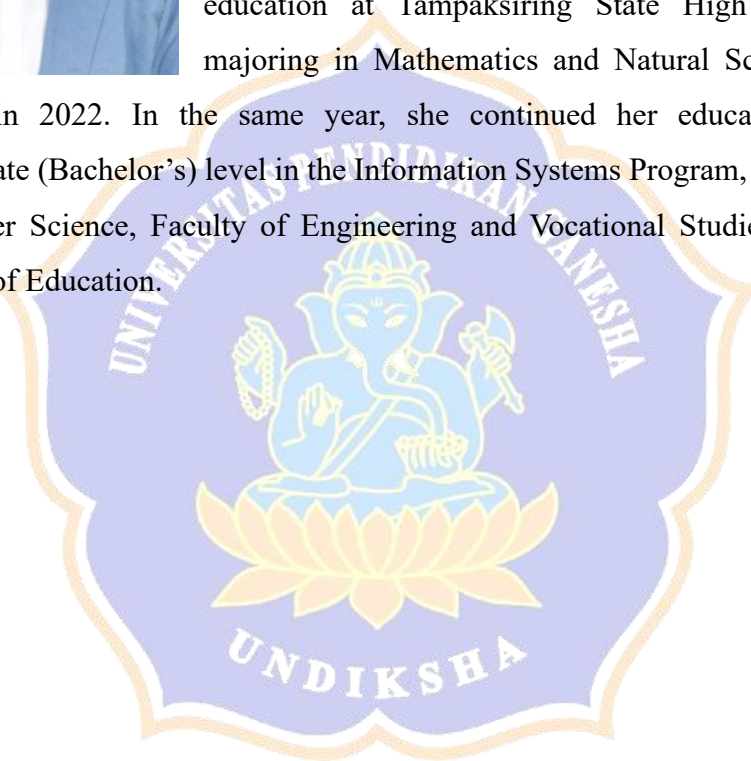


ATTACHMENTS

Attachment 1. Profile



Ni Ketut Artini Artalia was born in Gianyar on August 27, 2004. The author is the daughter of I Made Karna and Ni Ketut Muspa. She completed her elementary education at SD Negeri 1 Manukaya in 2016. Subsequently, she continued her junior high school education at SMP Negeri 1 Tampaksiring and graduated in 2019. She pursued her high school education at Tampaksiring State High School 1, majoring in Mathematics and Natural Sciences, and graduated in 2022. In the same year, she continued her education at the undergraduate (Bachelor's) level in the Information Systems Program, Department of Computer Science, Faculty of Engineering and Vocational Studies, Ganesha University of Education.



Attachment 2. Cleansing Data Code

```

import pandas as pd
import re

df = pd.read_excel('dataset/dataset_sentiment.xlsx')

def preprocess_text(text):
    if not isinstance(text, str):
        return ""

    # 1. Hapus mention (@user)
    text = re.sub(r'@\w+', ' ', text)

    # 2. Hapus hashtag (#tag)
    text = re.sub(r'#\w+', ' ', text)

    # 3. Hapus URL/link
    text = re.sub(r'http\S+', ' ', text)

    # 4. Hapus angka
    text = re.sub(r'[0-9]+', ' ', text)

    # 5. Hapus HTML tag
    text = re.sub(r'<[^>]+>', ' ', text)

    # 6. Hapus emoji dan karakter non-ASCII
    text = text.encode('ascii', 'ignore').decode('ascii')

    # 7. Hapus huruf berulang 3x atau lebih (contoh: "baaaagus"
    → "bagus")
    text = re.sub(r'(\.)\1{2,}', r'\1', text)

    # 8. Lowercase
    text = text.lower()

    # 9. Hapus spasi berlebih
    text = re.sub(r'\s+', ' ', text).strip()

    # 10. Hapus karakter non-printable
    text = re.sub(r'^\x20-\x7E\n\t', ' ', text)

    return text

df['Clean_data'] = df['Text'].apply(preprocess_text)
print(df[['Text', 'Clean_data']].head())
df.to_excel('dataset/sentiment_dataset_clean.xlsx', index=False)

```

Attachment 2. Translation Pipeline Code

```
!pip install -U transformers
```

```

# Use a pipeline as a high-level helper
from transformers import pipeline

pipe = pipeline("translation", model="Helsinki-NLP/opus-mt-id-en")

# Load model directly
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM

tokenizer = AutoTokenizer.from_pretrained("Helsinki-NLP/opus-mt-en-id")
model = AutoModelForSeq2SeqLM.from_pretrained("Helsinki-NLP/opus-mt-en-id")

def translate_text(text):
    if not isinstance(text, str) or text.strip() == "":
        return ""
    inputs = tokenizer(text, return_tensors="pt",
truncation=True, padding=True)
    translated = model.generate(**inputs)
    return tokenizer.decode(translated[0],
skip_special_tokens=True)

df['Translation'] = df['Clean_data'].apply(translate_text)
df.to_excel('dataset/sentiment_dataset_translation.xlsx',
index=False)

```

Attachment 3. Post Translation Cleansing Code

```

df = pd.read_excel('dataset/sentiment_dataset_translation.xlsx')

def final_preprocess_text(text):
    # if not isinstance(text, str):
    #     return ""

    # 1. Hapus formatting tags seperti {\cHffffff}
    text = re.sub(r"\{\cH[0-9A-Fa-f]+\}", "", text)

    # 2. Hapus teks diapit #
    text = re.sub(r"#.*?#", "", text)

    # 3. Hapus tanda pagar di awal/akhir
    text = re.sub(r"^#|#$", "", text)

    # 4. Ganti tanda baca & karakter non-alphabetic jadi spasi
    text = re.sub(r'[^A-Za-z\s]', ' ', text)

    # 5. Normalisasi spasi
    text = re.sub(r"\s+", " ", text).strip()

    # 6. Lowercase
    text = text.lower()

    return text

```

```
df['Post-Cleansing']
=df['Translation'].apply(final_preprocess_text)
df.to_excel('dataset/sentiment_dataset_post-cleansing.xlsx',
index=False)
```

Attachment 4. Normalization Code

```
df = pd.read_excel('dataset/sentiment_dataset_post-
cleansing.xlsx')

# Membaca file kamus slang dari lokal
kbba_dict = pd.read_csv ("dictionary/kbba_backup.txt",
delimiter='\t', names=['slang', 'formal'], header=None,
encoding='utf-8')

# Membuat dictionary slang -> formal
slang_dict = dict(zip(kbba_dict['slang'], kbba_dict['formal']))

# Fungsi normalisasi slang word

def convert_slangword(text):
    if not isinstance(text, str):
        return ""
    words = text.split()
    normalized = [slang_dict[word] if word in slang_dict else word
for word in words]
    return " ".join(normalized)

df['Normalized'] = df['Post-Cleansing'].apply(convert_slangword)
df.to_excel('dataset/sentiment_dataset_normalized.xlsx',
index=False)
```

Attachment 5. Stemming Code

```
df = pd.read_excel('dataset/sentiment_dataset_normalized.xlsx')

!pip install Sastrawi

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

# 1. Buat stemmer dari Sastrawi
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# 2. Fungsi Stemming
def apply_stemming(text):
    if isinstance(text, str):
        return stemmer.stem(text)
    return text

# 3. Menerapkan pada kolom normalized
df['Stemming'] = df['Normalized'].apply(apply_stemming)
df.to_excel('dataset/sentiment_dataset_stemming.xlsx',
index=False)
```

Attachment 6. InSet Lexicon Labelling Code

```

import pandas as pd
import numpy as np
import re

df = pd.read_excel("dataset/final_preprocessing.xlsx")

print(df['Stemming'].head(10))
print(type(df['Stemming'].iloc[0]))

import pandas as pd
import re

# Load lexicon positif & negatif
positive_url =
"https://raw.githubusercontent.com/fajri91/InSet/master/positive
.tsv"
negative_url =
"https://raw.githubusercontent.com/fajri91/InSet/master/negative
.tsv"

positive_lexicon = set(pd.read_csv(positive_url, sep='\t',
header=None)[0])
negative_lexicon = set(pd.read_csv(negative_url, sep='\t',
header=None)[0])

def sentiment_analysis_lexicon_indonesia(text):
    if pd.isna(text):
        return 0, '0', {}

    words = re.findall(r'[a-zA-Z]+', str(text).lower())

    score = 0
    word_scores = {}

    for word in words:
        if word in positive_lexicon:
            word_scores[word] = 1 # default weight
            score += 1
        elif word in negative_lexicon:
            word_scores[word] = -1 # default weight
            score -= 1
        else:
            word_scores[word] = 0

    if score > 0:
        sentiment = '1'
    elif score < 0:
        sentiment = '-1'
    else:
        sentiment = '0'

    return score, sentiment, word_scores

```

```
df['Word_Scores'] = df['Stemming'].apply(lambda x:
sentiment_analysis_lexicon_indonesia(x)[2])
df['Polarity_Score'] = df['Stemming'].apply(lambda x:
sentiment_analysis_lexicon_indonesia(x)[0])
df['Sentiment_Class_Inset'] = df['Stemming'].apply(lambda x:
sentiment_analysis_lexicon_indonesia(x)[1])

df.to_excel('dataset/InSet_Lexicon_Labelling.xlsx', index=False)
```

Attachment 7. IndoBERT Modelling Code

```
pip install transformers datasets scikit-learn torch pandas
numpy

from transformers import AutoTokenizer,
AutoModelForSequenceClassification, TrainingArguments, Trainer
from sklearn.model_selection import KFold
from sklearn.metrics import accuracy_score, f1_score,
classification_report, confusion_matrix
from datasets import Dataset
import pandas as pd
import numpy as np
import torch
import os
import shutil
from google.colab import drive
```

Load Data

```
df = pd.read_excel("dataset/manual_dataset.xlsx")

# Convert original labels (-1,0,1) → (0,1,2)
label_mapping = {-1: 0, 0: 1, 1: 2}
df["label"] = df["Sentiment Class"].map(label_mapping)

# Extract columns
texts = df["Text"].tolist()
labels = df["label"].tolist()

# Number of classes
num_labels = 3
```

IndoBERT Tokenizer

```
MODEL_NAME = "indobenchmark/indobert-base-p1"
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
```

Storage Path Setup

```
from google.colab import drive
```

```

import os
import numpy as np
import pandas as pd

# 1. Mount Google Drive
drive.mount('/content/drive')

# 2. Create a folder to store the results
output_dir =
"/content/drive/MyDrive/Penelitian_IndoBERT/Results_Manual"
if not os.path.exists(output_dir):
    os.makedirs(output_dir)

print(f"The results of each fold will be automatically saved in:
{output_dir}")

```

K-Fold Config

```

k = 10 # number of folds
kf = KFold(n_splits=k, shuffle=True, random_state=42)

accuracy_scores = []
macro_f1_scores = []
micro_f1_scores = []
confusion_matrices = []

fold = 1

```

Start K-Fold Loop

```

for train_index, test_index in kf.split(texts):

    print(f"\n=====")
    print(f"                FOLD {fold} / {k}")
    print(f"===== \n")

    # -----
    # 5.1 Prepare train & test dataframe
    # -----
    train_df = df.iloc[train_index].reset_index(drop=True)
    test_df = df.iloc[test_index].reset_index(drop=True)

    train_dataset = Dataset.from_pandas(train_df)
    test_dataset = Dataset.from_pandas(test_df)

    # -----
    # 5.2 Tokenization process
    # -----
    def tokenize(batch):
        return tokenizer(
            batch["Text"],
            max_length=128,
            truncation=True,
            padding="max_length"

```

```

)

train_dataset = train_dataset.map(tokenize, batched=True)
test_dataset = test_dataset.map(tokenize, batched=True)

# Hapus kolom teks mentah & format ke torch
train_dataset = train_dataset.remove_columns(["Text"])
test_dataset = test_dataset.remove_columns(["Text"])

train_dataset.set_format("torch")
test_dataset.set_format("torch")

# -----
# 5.3 Initialize new model for each fold
# -----
model = AutoModelForSequenceClassification.from_pretrained(
    MODEL_NAME,
    num_labels=num_labels
)

# -----
# 5.4 Training configuration
# -----
# Temporary local output directory (per fold)
temp_fold_dir = f"./temp_fold_{fold}"

training_args = TrainingArguments(
    output_dir=f"./indobert_temp_fold_{fold}",
    num_train_epochs=3,
    learning_rate=2e-5,
    weight_decay=0.01,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    logging_steps=100,
    save_strategy="no",
    report_to="none"
)

# -----
# 5.5 Metric calculation (Trainer evaluation)
# -----
def compute_metrics(eval_pred):
    logits, y_true = eval_pred
    y_pred = np.argmax(logits, axis=1)
    return {
        "accuracy": accuracy_score(y_true, y_pred),
        "macro_f1": f1_score(y_true, y_pred,
average="macro"),
        "micro_f1": f1_score(y_true, y_pred,
average="micro"),
    }

# -----
# 5.6 Trainer setup
# -----
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,

```

```

        eval_dataset=test_dataset,
        compute_metrics=compute_metrics,
    )

    # -----
    # 5.7 Model training (Fine-tuning)
    # -----
    trainer.train()

    # -----
    # 5.8 Evaluation on this fold
    # -----
    predictions = trainer.predict(test_dataset)
    y_pred = np.argmax(predictions.predictions, axis=1)
    y_true = predictions.label_ids

    # -----
    # 5.9 Classification Report
    # -----
    print(f"\nClassification Report Fold-{{fold}}")
    print(classification_report(
        y_true,
        y_pred,
        target_names=["Negatif", "Netral", "Positif"],
        digits=4
    ))

    # -----
    # 5.10 Store evaluation metrics
    # -----
    acc = accuracy_score(y_true, y_pred)
    accuracy_scores.append(acc)
    macro_f1_scores.append(f1_score(y_true, y_pred,
    average="macro"))
    micro_f1_scores.append(f1_score(y_true, y_pred,
    average="micro"))
    confusion_matrices.append(confusion_matrix(y_true, y_pred,
    labels=[0, 1, 2]))

    print(f"Fold-{{fold}} Accuracy : {{acc:.4f}}")

    # =====
    # AUTOMATIC SAVING TO GOOGLE DRIVE FEATURE
    # =====

    # 1. Save Metrics to CSV
    current_results_df = pd.DataFrame({
        "fold": list(range(1, fold + 1)),
        "accuracy": accuracy_scores,
        "macro_f1": macro_f1_scores,
        "micro_f1": micro_f1_scores
    })

    # 1. Save/Overwrite rekap nilai
    current_results_df.to_csv(os.path.join(output_dir,
    "ongoing_results.csv"), index=False)
    np.save(os.path.join(output_dir,
    "ongoing_confusion_matrices.npy"), np.array(confusion_matrices))

```

```

# 2. SAVE MODEL FISIK PER FOLD (Ini request barumu)
# Kita buat sub-folder: Results_Inset/Saved_Model_Fold_1
fold_model_path = os.path.join(output_dir,
f"Saved_Model_Fold_{fold}")

# Buat folder jika belum ada
if not os.path.exists(fold_model_path):
    os.makedirs(fold_model_path)

print(f"Sedang menyimpan Model Fold {fold} ke Drive...")

# Save Model & Tokenizer
trainer.save_model(fold_model_path)
tokenizer.save_pretrained(fold_model_path)

print(f"Model Fold {fold} AMAN tersimpan di:
{fold_model_path}")

# 3. CLEANUP MEMORY (Supaya Colab tidak Crash)
# Hapus folder temp lokal
if os.path.exists(temp_fold_dir):
    shutil.rmtree(temp_fold_dir)

# Hapus model dari RAM
del model
del trainer
torch.cuda.empty_cache()

# -----
fold += 1

print("\n\nSelesai! Semua Fold telah dijalankan dan disimpan.")

```

K-Fold Visualization

```

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# =====
# VISUALISASI PERFORMA PER FOLD
# =====

# 1. Menyiapkan Data Sumbu X (Fold 1 sampai Fold Terakhir)
folds = np.arange(1, len(accuracy_scores) + 1)

# 2. Pengaturan Ukuran Gambar
plt.figure(figsize=(12, 6))

# 3. Plotting Garis (Warna Hijau seperti referensi)
plt.plot(folds, accuracy_scores, marker='o', color='green',
linewidth=2, markersize=8, label='Accuracy Validation')

# 4. Menambahkan Label Nilai di Atas Setiap Titik
for x, y in zip(folds, accuracy_scores):

```

```

    # Menempatkan text sedikit di atas titik (y + offset)
    plt.text(x, y + 0.001, f'{y:.4f}', ha='center', va='bottom',
             fontsize=10, fontweight='bold')

# 5. Dekorasi Grafik
plt.title('Performance K-Fold (Validation) - IndoBERT',
          fontsize=14, pad=20)
plt.xlabel('Fold -', fontsize=12)
plt.ylabel('Accuracy', fontsize=12)
plt.grid(True, linestyle='-', alpha=0.7)
plt.xticks(folds) # Memastikan sumbu X hanya menampilkan angka
bulat (1, 2, 3...)
plt.legend()

# 6. Menampilkan Grafik
plt.tight_layout()
plt.show()

# =====
# MENAMPILKAN RATA-RATA AKHIR
# =====
mean_acc = np.mean(accuracy_scores)
std_acc = np.std(accuracy_scores)

print("-" * 30)
print("FINAL RESULTS OF CROSS-VALIDATION")
print("-" * 30)
print(f"Average Accuracy : {mean_acc:.4f}")
print(f"Standard Deviation : {std_acc:.4f}") # Semakin kecil
semakin stabil modelnya
print("-" * 30)

```

Final Result Summary

```

import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
import os
from google.colab import drive

# 1. Mount Drive
drive.mount('/content/drive', force_remount=True)

print("\n=====")
print("      FINAL SUMMARY & VISUALIZATION")
print("=====\n")

# =====
output_dir =
"/content/drive/MyDrive/Penelitian_IndoBERT/Results_Manual"

results_path = os.path.join(output_dir, "ongoing_results.csv")
cm_path = os.path.join(output_dir,
"ongoing_confusion_matrices.npy")

```

```

# Check if the file is found
if os.path.exists(results_path) and os.path.exists(cm_path):
    print(f"File found in: {output_dir}\n")

    # --- LOAD DATA ---
    results_df = pd.read_csv(results_path)
    loaded_cms = np.load(cm_path)

    # --- TEXT SUMMARY ---
    print(f"Total Folds Completed : {len(results_df)}")
    print(f"Mean Accuracy          :
{results_df['accuracy'].mean():.4f}")
    print(f"Mean Macro-F1             :
{results_df['macro_f1'].mean():.4f}")
    print(f"Mean Micro-F1            :
{results_df['micro_f1'].mean():.4f}")

    # --- VISUALIZATION ---
    avg_cm = np.mean(loaded_cms, axis=0)

    plt.figure(figsize=(8, 6))
    label_names = ["Negative (-1)", "Neutral (0)", "Positive
(1)"]

    sns.heatmap(
        avg_cm,
        annot=True,
        fmt=".2f",
        cmap="Blues",
        xticklabels=label_names,
        yticklabels=label_names
    )

    plt.xlabel("Predicted Label")
    plt.ylabel("True Label")
    plt.title(f"Average Confusion Matrix Manual Labelling +
ROS")
    plt.tight_layout()
    plt.show()

else:
    print(f"Still not found in: {output_dir}")
    # print("Coba cek apakah nama folder 'Penelitian_IndoBERT'
ada spasi atau huruf besar/kecilnya beda?")
    # Debugging Help: Show the contents of the parent folder
    parent_dir = "/content/drive/MyDrive/Penelitian_IndoBERT"
    if os.path.exists(parent_dir):
        print(f"\nIsi folder '{parent_dir}':")
        print(os.listdir(parent_dir))

```

Attachment 5. Data Request Letter SMP Negeri 1 Sukawati

	<p>KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI UNIVERSITAS PENDIDIKAN GANESHA FAKULTAS TEKNIK DAN KEJURUAN Jalan Udayana Nomor 11 Singaraja Bali Laman: http://ftk.undiksha.ac.id</p>	
<hr/>		
Nomor	: 1977/UN48.11.1/KM/2025	Singaraja, 28 Juli 2025
Perihal	: Surat Permohonan Pengambilan Data	
<p>Yth. Kepala SMP Negeri 1 Sukawati di tempat</p>		
<p>Dengan hormat, sehubungan dengan proses penyelesaian Tugas Akhir/Skripsi, maka melalui surat ini kami mohon Bapak/Ibu berkenan memberikan data yang terkait dengan data yang dibutuhkan. Adapun mahasiswa yang akan melakukan pengambilan data seperti tersebut di bawah ini:</p>		
Nama	: Ni Ketut Artini Artalia	
NIM	: 2215091072	
Program Studi	: Sistem Informasi	
Jurusan	: Teknik Informatika	
Data yang dibutuhkan	: Pelabelan data skripsi	
Judul Penelitian	: Sentiment Analysis of Bali's Rapid Tourism Growth: Comparing IndoBERT Performance With InSet Lexicon and Manual Label Integration	
<p>Demikian kami sampaikan, atas perhatian dan kerjasamanya, diucapkan terimakasih.</p>		
		
		
<p>a.n Dekan Wakil Dekan Bidang Akademik, Made Windu Antara Kesiman NIP. 198211112008121001</p>		

KEMENTERIAN PENDIDIKAN TINGGI, SAINS DAN TEKNOLOGI
UNIVERSITAS PENDIDIKAN GANESHA
FAKULTAS TEKNIK DAN KEJURUAN
JURUSAN TEKNIK INFORMATIKA



Jalan Udayana Singaraja-Bali Kode Pos 81116
Tlp. (0362) 22570 Fax. (0362) 25735
Laman: www.undiksha.ac.id

Nomor : 413/UN48.11.5/KM/2025
Perihal : Surat Permohonan Pengambilan Data
Lampiran : -

Singaraja, 28 Juli 2025

Yth. Dekan FTK
Universitas Pendidikan Ganesha
Di tempat

Dengan hormat,
Sehubungan dengan proses penyelesaian Tugas Akhir / Skripsi yang dilaksanakan oleh saudara mahasiswa:

Nama	: Ni Ketut Artini Artalia
Nim	: 2215091072
Prodi/Jurusan	: Sistem Informasi / Teknik Informatika
Instansi yg ditujui	: SMP N 1 Sukawati
Jabatan Yang Dituju	: Kepala Sekolah SMP N 1 Sukawati
Data Yang Dibutuhkan	: Pelabelan Data Skripsi.
Judul	: Sentiment Analysis of Bali's Rapid Tourism Growth: Comparing IndoBERT Performance with InSet Lexicon and Manual Label Integration

Bersama ini kami mohonkan kepada Bapak untuk berkenan memfasilitasi kebutuhan data untuk Tugas Akhir / Skripsi mahasiswa yang bersangkutan.

Demikian kami sampaikan. Atas perhatian dan kerjasama Bapak, kami ucapkan terima kasih.

Kepala Jurusan Teknik Informatika,



Putu Hendra Suputra
NIP. 198212222006041001



**Balai
Sertifikasi
Elektronik**

Catatan :

- UU ITE No. 11 Tahun 2008 Pasal 5 ayat 1 "Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah"
- Dokumen ini tertanda ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BsrE
- Surat ini dapat dibuktikan keasliannya dengan menggunakan *qr code* yang telah tersedia

Attachment 6. Data Request Letter SMK Negeri 1 Tampaksiring



KEMENTERIAN PENDIDIKAN TINGGI,
SAINS, DAN TEKNOLOGI
UNIVERSITAS PENDIDIKAN GANESHA
FAKULTAS TEKNIK DAN KEJURUAN
Jalan Udayana Nomor 11 Singaraja Bali
Laman: <http://ftk.undiksha.ac.id>

Nomor : 1976/UN48.11.1/KM/2025

Singaraja, 28 Juli 2025

Perihal : Surat Permohonan Pengambilan Data

Yth. Kepala SMK Negeri 1 Tampaksiring
di tempat

Dengan hormat, sehubungan dengan proses penyelesaian Tugas Akhir/Skripsi, maka melalui surat ini kami mohon Bapak/Ibu berkenan memberikan data yang terkait dengan data yang dibutuhkan. Adapun mahasiswa yang akan melakukan pengambilan data seperti tersebut di bawah ini:

Nama : Ni Ketut Artini Artalia
NIM : 2215091072
Program Studi : Sistem Informasi
Jurusan : Teknik Informatika
Data yang dibutuhkan : Pelabelan data skripsi
Judul Penelitian : Sentiment Analysis of Bali's Rapid Tourism Growth: Comparing IndoBERT Performance With InSet Lexicon and Manual Label Integration

Demikian kami sampaikan, atas perhatian dan kerjasamanya, diucapkan terimakasih.

a.n Dekan
Wakil Dekan Bidang Akademik,

Made Windu Antara Kesiman
NIP. 198211112008121001

KEMENTERIAN PENDIDIKAN TINGGI, SAINS DAN TEKNOLOGI
UNIVERSITAS PENDIDIKAN GANESHA
FAKULTAS TEKNIK DAN KEJURUAN
JURUSAN TEKNIK INFORMATIKA
 Jalan Udayana Singaraja-Bali Kode Pos 81116
 Tlp. (0362) 22570 Fax. (0362) 25735
 Laman: www.undiksha.ac.id



1976

Nomor : 412/UN48.11.5/KM/2025
 Perihal : Surat Permohonan Pengambilan Data
 Lampiran : -

Singaraja, 28 Juli 2025

Yth. Dekan FTK
 Universitas Pendidikan Ganesha
 Di tempat

Dengan hormat,
 Schubungan dengan proses penyelesaian Tugas Akhir / Skripsi yang dilaksanakan oleh saudara mahasiswa:

Nama : Ni Ketut Artini Artalia
 Nim : 2215091072
 Prodi/Jurusan : Sistem Informasi / Teknik Informatika
 Instansi yg ditujui : SMK N 1 Tampaksiring
 Jabatan Yang Dituju : Kepala Sekolah SMK N 1 Tampaksiring
 Data Yang Dibutuhkan : Pelabelan Data Skripsi.
 Judul : Sentiment Analysis of Bali's Rapid Tourism Growth:
 Comparing IndoBERT Performance with InSet Lexicon and
 Manual Label Integration

Bersama ini kami mohonkan kepada Bapak untuk berkenan memfasilitasi kebutuhan data untuk Tugas Akhir / Skripsi mahasiswa yang bersangkutan.

Demikian kami sampaikan. Atas perhatian dan kerjasama Bapak, kami ucapkan terima kasih.

Ketua Jurusan Teknik Informatika,



Putu Hendra Suputra
 NIP. 198212222006041001

Balai
Sertifikasi
Elektronik

Catatan :

- UU ITE No. 11 Tahun 2008 Pasal 5 ayat 1 "Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah"
- Dokumen ini tertanda ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BsrE
- Surat ini dapat dibuktikan keasliannya dengan menggunakan *qr code* yang telah tersedia

Attachment 7. Data Request Letter Café Gabriel



KEMENTERIAN PENDIDIKAN TINGGI, SAINS,
DAN TEKNOLOGI
UNIVERSITAS PENDIDIKAN GANESHA
FAKULTAS TEKNIK DAN KEJURUAN
Jalan Udayana Nomor 11 Singaraja - Bali Kode Pos 81116
Telepon (0362) 22570 Email: fk@undiksha.ac.id Laman: <http://fk.undiksha.ac.id>

Nomor : 2651/UN48.11.1/DI.03.00/2025 Singaraja, 25 September 2025
Perihal : Surat Permohonan Pengambilan Data

Yth. Manager Cafe Gabriel
di tempat

Dengan hormat, sehubungan dengan proses penyelesaian Tugas Akhir/Skripsi, maka melalui surat ini kami mohon Bapak/Ibu berkenan memberikan data yang terkait dengan data yang dibutuhkan. Adapun mahasiswa yang akan melakukan pengambilan data seperti tersebut di bawah ini:

Nama : Ni Ketut Artini Artalia
NIM : 2215091072
Program Studi : Sistem Informasi
Jurusan : Teknik Informatika
Data yang dibutuhkan : Pelabelan data skripsi
Judul Penelitian : Sentiment Analysis of Bali's Rapid Tourism Growth:
Comparing IndoBERT Performance with InSet Lexicon and
Manual Label Integration.

Demikian kami sampaikan, atas perhatian dan kerjasamanya, diucapkan terima kasih.

a.n Dekan
Wakil Dekan Bidang Akademik,



Made Windu Antara Kesiman
NIP 198211112008121001



Catatan :

- UU ITE No. 11 Tahun 2008 Pasal 5 ayat 1 "Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah"
- Dokumen ini setelah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BIRI.
- Surat ini dapat dibuktikan keasliannya dengan menggunakan qr code yang telah tersedia

Attachment 8. Data Annotation Statement Letter

SURAT PERNYATAAN ANNOTASI DATA

Saya yang bertanda tangan di bawah ini:

Nama : Komang Sri Ayu Sudarmini, A.Md.Par., S.Pd
NIP : 197802082023212007
Instansi : SMK Negeri 1 Tampaksiring
No.HP / Email : 082247979573

Dengan ini menyatakan bahwa saya benar-benar telah melakukan proses pelabelan (annotasi) data milik **Ni Ketut Artini Artalia**, yang digunakan untuk kepentingan penelitian dengan judul **"Sentiment Analysis of Bali's Rapid Tourism Growth: Comparing IndoBERT Performance with InSet Lexicon and Manual Label Integration"**.

Saya mengerjakan proses pelabelan tersebut secara manual dan telah berupaya memberikan label sesuai dengan petunjuk atau kriteria yang telah ditentukan oleh peneliti.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya untuk digunakan sebagaimana mestinya.

Gianyar, 14 Juli 2025

Hormat saya,



Komang Sri Ayu Sudarmini, A.Md.Par., S.Pd

NIP. 197802082023212007

SURAT PERNYATAAN ANNOTASI DATA

Saya yang bertanda tangan di bawah ini:

Nama : I Dewa Gede Pujana, S.Pd
NIP : -
Instansi : SMP Negeri 1 Sukawati
No.HP / Email : 085737882748 /dwpujana@gmail.com

Dengan ini menyatakan bahwa saya benar-benar telah melakukan proses pelabelan (annotasi) data milik **Ni Ketut Artini Artalia**, yang digunakan untuk kepentingan penelitian dengan judul "**Sentiment Analysis of Bali's Rapid Tourism Growth: Comparing IndoBERT Performance with InSet Lexicon and Manual Label Integration**".

Saya mengerjakan proses pelabelan tersebut secara manual dan telah berupaya memberikan label sesuai dengan petunjuk atau kriteria yang telah ditentukan oleh peneliti.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya untuk digunakan sebagaimana mestinya.

Gianyar, 12 Juli 2025

Hormat saya,



I Dewa Gede Pujana

NIP.

SURAT PERNYATAAN ANNOTASI DATA

Saya yang bertanda tangan di bawah ini:

Nama : I Made Adi Arta
NIP : -
Instansi : Café Gabriel - Ubud
No.HP / Email : 083115420874 /adiarta703@gmail.com

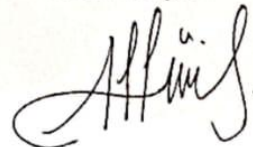
Dengan ini menyatakan bahwa saya benar-benar telah melakukan proses pelabelan (annotasi) data milik **Ni Ketut Artini Artalia**, yang digunakan untuk kepentingan penelitian dengan judul "**Sentiment Analysis of Bali's Rapid Tourism Growth: Comparing IndoBERT Performance with InSet Lexicon and Manual Label Integration**".

Saya mengerjakan proses pelabelan tersebut secara manual dan telah berupaya memberikan label sesuai dengan petunjuk atau kriteria yang telah ditentukan oleh peneliti.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya untuk digunakan sebagaimana mestinya.

Gianyar, 26 September 2025

Hormat saya,



I Made Adi Arta

NIP.

Attachment 9. Documentation with annotators

	<p>Annotator Name: Komang Ayu Sri Sudarmini, A.Md.Par., S.Pd</p> <p>Occupation: Indonesian Language and Hospitality Teacher</p> <p>Institution: SMK Negeri 1 Tampaksiring</p>
	<p>Annotator Name: I Dewa Gede Pujana, S.Pd</p> <p>Occupation: Indonesian Language Teacher</p> <p>Institution: SMP Negeri 1 Sukawati</p>

