

LAMPIRAN

Lampiran. 1 Riwayat Hidup



Komang Tya Vinandita Kusuma Dewi lahir di Banyuning Singaraja pada tanggal 28 September 2004. Penulis lahir dari pasangan suami istri Bapak Gede Sudirga dan Ibu Ni Luh Ratmini sebagai anak kedua dari tiga bersaudara. Kini penulis beralamat di Jalan Pulau Kangean, Banyuning Selatan, Kecamatan Buleleng, Kabupaten Buleleng,

Provinsi Bali. Penulis menyelesaikan pendidikan dasar di SD N 2 Banyuasri dan lulus pada tahun 2016. Kemudian penulis melanjutkan pendidikan di SMP Negeri 2 Singaraja dan lulus pada tahun 2019. Pada tahun 2022, penulis lulus dari SMAS Laboratorium Undiksha Singaraja jurusan MIPA (Matematika dan Ilmu Pengetahuan Alam) dan melanjutkan studi (S1) di Universitas Pendidikan Ganesha dengan Program Studi Sistem Informasi, Jurusan Teknik Informatika. Perjalanan pendidikan yang telah dilalui penulis tidak terlepas dari dukungan keluarga, dosen, sahabat, dan berbagai pihak yang telah memberikan motivasi serta pengalaman berharga. Penulis berharap ilmu yang diperoleh selama masa studi dapat menjadi landasan untuk terus berkarya, mengembangkan potensi diri, dan memberikan manfaat bagi lingkungan sekitar maupun masyarakat luas.

Lampiran. 2 Surat Permohonan Pelabelan

1. Surat permohonan kepada SMPN 3 Kubutambahan



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: flk@undiksha.ac.id Laman: <http://flk.undiksha.ac.id>

Nomor : 1353/UN48.11.1/DI.03.00/2026
Perihal : Surat Permohonan Pengambilan Data

Singaraja, 13 Mei 2026

Yth. Kepala SMP Negeri 3 Kubutambahan
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 : Komang Tya Vinandita Kusuma Dewi
NIM : 2215091044
Program Studi : Sistem Informasi
Jurusan : Teknik Informatika
Data yang dibutuhkan : Data Pelabelan Skripsi.
Judul Penelitian : Analisis Sentimen Opini Publik Terhadap Inovasi Teknologi
Pada Admintrasi Publik Indonesia Menggunakan Metode
Indobert

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 tertanda diandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BsrE
- Surat ini dapat dibuktikan keasliannya dengan menggunakan *qr code* yang telah tersedia

2. Surat permohonan kepada SMPN 1 Abiansemal



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 : 1352/UN48.11.1/DI.03.00/2026

Singaraja, 13 Mei 2026

Perihal : Surat Permohonan Pengambilan Data

Yth. Kepala SMP Negeri 1 Abiansemal
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	: Komang Tya Vinandita Kusuma Dewi
NIM	: 2215091044
Program Studi	: Sistem Informasi
Jurusan	: Teknik Informatika
Data yang dibutuhkan	: Data Pelabelan Skripsi.
Judul Penelitian	: Analisis Sentimen Opini Publik Terhadap Inovasi Teknologi Pada Admintrasi Publik Indonesia Menggunakan Metode Indobert

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 tertanda ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BsRE
- Surat ini dapat dibuktikan keasliannya dengan menggunakan *qr code* yang telah tersedia

3. Surat permohonan kepada SMKS Karyawisata Singaraja



**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: ftk@undiksha.ac.id Laman: <http://ftk.undiksha.ac.id>

Nomor : 1354/UN48.11.1/DI.03.00/2026

Singaraja, 13 Mei 2026

Perihal : Surat Permohonan Pengambilan Data

Yth. Kepala SMKS Karyawisata Singaraja
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	: Komang Tya Vinandita Kusuma Dewi
NIM	: 2215091044
Program Studi	: Sistem Informasi
Jurusan	: Teknik Informatika
Data yang dibutuhkan	: Data Pelabelan Skripsi.
Judul Penelitian	: Analisis Sentimen Opini Publik Terhadap Inovasi Teknologi Pada Admintrasi Publik Indonesia Menggunakan Metode Indobert

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

a.n Dekan
Wakil Dekan Bidang Akademik,



Made Windu Antara Kesiman
NIP 19821112008121001



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 Bsrf
- Surat ini dapat dibuktikan keasliannya dengan menggunakan *qr code* yang telah tersedia

Lampiran. 3 Surat Validasi Pelabelan Data

1. Surat Validasi Pelabelan Data oleh Guru Bahasa Indonesia SMPN 3 Kubutambahan

	KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI UNIVERSITAS PENDIDIKAN GANESHA FAKULTAS TEKNIK DAN KEJURUAN Jalan Udayana Singaraja-Bali Kode Pos 81116 Tlp. (0362) 22570 fax. (0362) 25735 Laman: www.undiksha.ac.id
SURAT KETERANGAN	
LABELING DATA	
Yang bertanda tangan dibawah ini:	
Nama	: Ketut Suryaningsih, S.Pd., Gr.
Jabatan	: Guru Bahasa Indonesia SMPN 3 Kubutambahan
Menerangkan bahwa Mahasiswa Universitas Pendidikan Ganesha dibawah ini :	
Nama	: Komang Tya Vinandita Kusuma Dewi
NIM	: 2215091044
Prodi	: S1 Sistem Informasi
Dengan ini menerangkan bahwa proses pelabelan dataset telah dilakukan dan diselesaikan pada tanggal 14 Mei 2026.	
Demikian surat keterangan ini dibuat dengan sebenarnya untuk dapat digunakan sebagaimana mestinya.	
Singaraja, 14 Mei 2026	
Ahli/Pakar Bahasa,	
	
Ketut Suryaningsih, S.Pd., Gr	

CS Dipindai dengan CamScanner

2. Surat Validasi Pelabelan Data oleh Guru Bahasa Indonesia SMPN 1 Abiansemal

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

SURAT KETERANGAN
LABELING DATA

Yang bertanda tangan dibawah ini:

Nama : I Dewa Ayu Gede Wuwar Ratih, S.pd., Gr.
Jabatan : Guru Bahasa Indonesia di SMPN 1 Abiansemal

Menerangkan bahwa Mahasiswa Universitas Pendidikan Ganesha dibawah ini :

Nama : Komang Tya Vinandita Kusuma Dewi
NIM : 2215091044
Prodi : SI Sistem Informasi

Dengan ini menerangkan bahwa proses pelabelan dataset telah dilaksanakan dan diselesaikan pada tanggal 14 Mei 2026.

Demikian surat keterangan ini dibuat dengan sebenarnya untuk dapat digunakan sebagaimana mestinya.

Singaraja, 14 Mei 2026

Ahli/Pakar Bahasa,



I Dewa Ayu Gede Wuwar Ratih, S.pd., Gr.

CS Dipindai dengan CamScanner

3. Surat Validasi Pelabelan Data oleh Guru Bahasa Indonesia SMKS
Karyawisata Singaraja



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

SURAT KETERANGAN
LABELING DATA

Yang bertanda tangan dibawah ini:

Nama : Kadek Indah Kusuma Dewi, S.Pd, Gr.
Jabatan : Guru Bahasa Indonesia SMAS Karyawisata Singaraja

Menerangkan bahwa Mahasiswa Universitas Pendidikan Ganesha dibawah ini :

Nama : Komang Tya Vinandita Kusuma Dewi
NIM : 2215091044
Prodi : S1 Sistem Informasi

Dengan ini menerangkan bahwa proses pelabelan dataset telah dilaksanakan dan diselesaikan pada tanggal 14 Mei 2026.

Demikian surat keterangan ini dibuat dengan sebenarnya untuk dapat digunakan sebagaimana mestinya.

Singaraja, 14 Mei 2026

Ahli/Pakar Bahasa,



Kadek Indah Kusuma Dewi, S.Pd., Gr

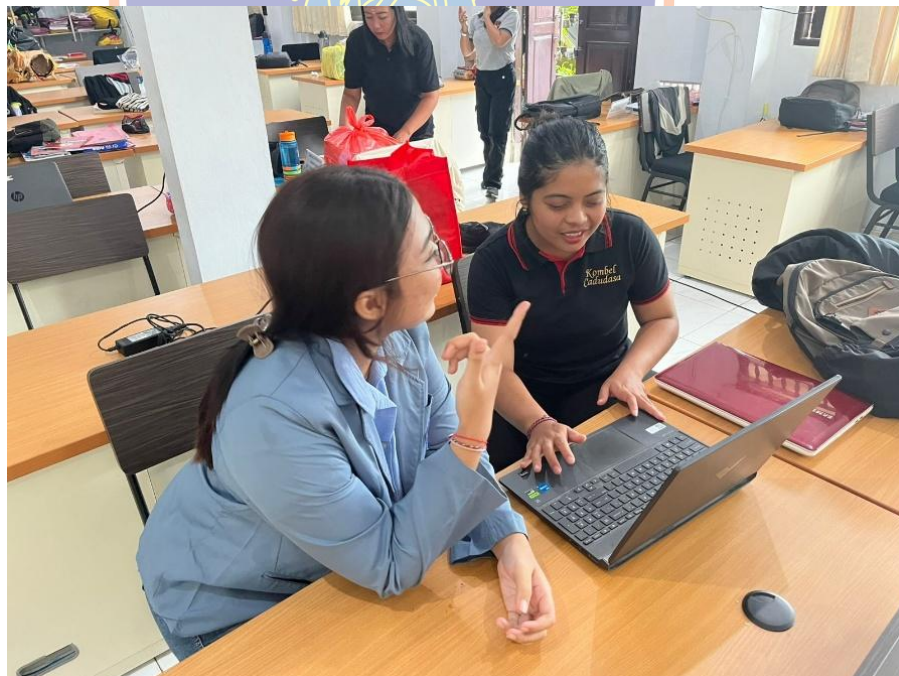
CS Scanned with CamScanner

Lampiran. 4 Dokumentasi Pelabelan Data

1. Dokumentasi Pelabelan Data di SMPN 3 Kubutambahan



2. Dokumentasi Pelabelan Data di SMPN 1 Abiansemal



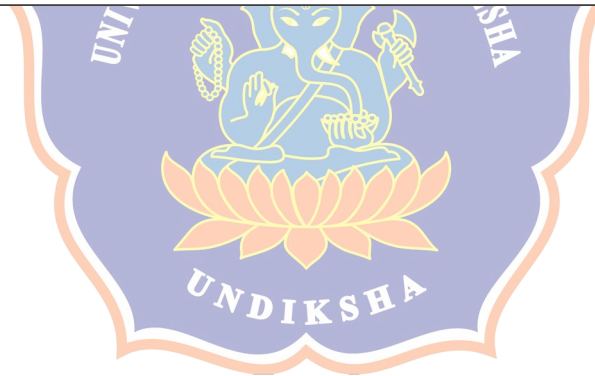
3. Dokumentasi Pelabelan Data di SMKS Karyawisata Singaraja



Lampiran. 5 Code Crawling dan Scraping Data

1. *Crawling Data*

```
#@title Twitter Auth Token  
twitter_auth_token = #AUTH_TOKEN  
  
# Import required Python package  
!pip install pandas  
  
# Crawl Data  
  
filename = 'inovasiteknologi.csv'  
  
search_keyword = 'inovasi teknologi since:2024-01-01  
until:2024-12-31 lang:id'  
  
limit = 10000  
  
!npx -y tweet-harvest@2.6.1 -o "{filename}" -s  
"{search_keyword}" --tab "LATEST" -l {limit} --token  
{twitter_auth_token}
```



2. *Scraping data*

```
from apify_client import ApifyClient

# Initialize the ApifyClient with your API token
client = ApifyClient("<YOUR_API_TOKEN>")

# Prepare the Actor input
run_input = {
    "postURLs": [#URL_TIKTOK]
    "commentsPerPost": 100,
    "topLevelCommentsPerPost": None,
    "maxRepliesPerComment": 0,
    "profiles": None,
    "resultsPerPage": 100,
    "profileScrapeSections": ["videos"],
    "profileSorting": "latest",
    "oldestPostDateUnified": None,
    "newestPostDate": None,
    "excludePinnedPosts": False,
}

# Run the Actor and wait for it to finish
run =
client.actor("BDec00yAmCm1QbMEI").call(run_input=run_input)

# Fetch and print Actor results from the run's dataset (if
there are any)
for item in
client.dataset(run["defaultDatasetId"]).iterate_items():
    print(item)
```


Lampiran. 6 Code Preprocessing Data

1. *Cleaning*

```
def remove_URL(tweet):
    if tweet is not None and isinstance(tweet, str):
        url = re.compile(r'https?://\S+|www\.\S+')
        return url.sub(r'', tweet)
    else:
        return tweet

def remove_html(tweet):
    if tweet is not None and isinstance(tweet, str):
        html = re.compile(r'<.*?>')
        return html.sub(r'', tweet)
    else:
        return tweet

def remove_emoji_with_regex(tweet):
    if tweet is not None and isinstance(tweet, str):
        emoji_pattern = re.compile(
            "["
            u"\U0001F600-\U0001F64F" # Emoticons
            u"\U0001F300-\U0001F5FF" # Symbols &
            pictographs
            u"\U0001F680-\U0001F6FF" # Transport & map
            symbols
            u"\U0001F700-\U0001F77F" # Alchemical symbols
            u"\U0001F780-\U0001F7FF" # Geometric shapes
            u"\U0001F800-\U0001F8FF" # Supplemental
            symbols
            u"\U0001F900-\U0001F9FF" # Symbols and
            pictographs
            u"\U0001FA00-\U0001FA6F" # Chess symbols
            u"\U0001FA70-\U0001FAFF" # Symbols &
            pictographs extended-A
            u"\U00002702-\U000027B0" # Dingbats
            u"\U000024C2-\U0001F251" # Enclosed characters
            "]" +, flags=re.UNICODE
```

```

        return emoji_pattern.sub(r'', tweet)
    return tweet

def remove_symbols(tweet):
    if tweet is not None and isinstance(tweet, str):
        tweet = re.sub(r'^a-zA-Z0-9\s', '', tweet)
    return tweet

def remove_numbers(tweet):
    if tweet is not None and isinstance(tweet, str):
        tweet = re.sub(r'\d', '', tweet)
    return tweet

data['Content'] = data['Content'].astype(str)
data['Cleaning'] = data['Content'].apply(lambda x:
remove_URL(x))

data['Cleaning'] = data['Cleaning'].apply(lambda x:
remove_html(x))

data['Cleaning'] = data['Cleaning'].apply(lambda x:
remove_emoji_with_regex(x))

data['Cleaning'] = data['Cleaning'].apply(lambda x:
remove_symbols(x))

data['Cleaning'] = data['Cleaning'].apply(lambda x:
remove_numbers(x))

data['Cleaning'] = data['Cleaning'].str.strip()
data = data[data['Cleaning'].str.split().str.len() > 1]
data = data.reset_index(drop=True)

data.head(10)

```

2. Case Folding

```
def case_Folding(text):
    if isinstance(text, str):
        Lowercase_text = text.lower()
        return Lowercase_text
    else:
        return text

data['case_Folding'] = data['Cleaning'].apply(case_Folding)
data.head(10)
```

3. Normalization



```
# Path file kamus slang
file_path = '/Content/kbba_backup.txt'

# Membaca kamus slang dari file
kbba_Dictionary = pd.read_csv(file_path, delimiter='\t',
                              names=['slang', 'formal'], header=None, encoding='utf8')

# Membuat kamus slang_dict
slang_dict = dict(zip(kbba_Dictionary['slang'],
                      kbba_Dictionary['formal']))

# Fungsi untuk mengganti slang word
def convert_slangword(text):
    # Mengganti kata slang dengan kata formal
    normalized_words = [slang_dict[word] if word in
                        slang_dict else word for word in text.split()]

    return ' '.join(normalized_words) # Mengembalikan
    dalam bentuk string

data['Normalization'] = data['case_Folding'].apply(lambda
x: ' '.join(x) if isinstance(x, list) else
x).apply(convert_slangword)

data[['Normalization']]

data.head(5)
```

Lampiran. 7 Code Transformasi Data

1. *Splitting data*

```
from sklearn.model_selection import train_test_split
X = data['Normalization']
y = data['Sentiment']
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42,
    stratify=y
)
train_df = pd.DataFrame({
    'Normalization': X_train,
    'Sentiment': y_train
})
test_df = pd.DataFrame({
    'Normalization': X_test,
    'Sentiment': y_test
})
```



2. Back Translation

```
def batch_translate(texts, tokenizer, model,
                    batch_size=16):
    results = []
    model.eval()
    for i in range(0, len(texts), batch_size):
        batch = texts[i:i+batch_size]
        inputs = tokenizer(
            batch,
            return_tensors="pt",
            padding=True,
            Truncation=True,
            max_length=256
        ).to(device)
        with torch.no_grad():
            outputs = model.generate(**inputs)
        decoded = tokenizer.batch_decode(
            outputs,
            skip_special_tokens=True
        )
        results.extend(decoded)
    return results

train_pos_sample = train_pos.sample(n=500, random_state=42)
texts = train_pos_sample['Normalization'].tolist()
en1 = batch_translate(texts, tok_id_en, mod_id_en,
                      batch_size=16)
de = batch_translate(en1, tok_en_de, mod_en_de,
                     batch_size=16)
en2 = batch_translate(de, tok_de_en, mod_de_en,
                     batch_size=16)
idn = batch_translate(en2, tok_en_id, mod_en_id,
                      batch_size=16)
train_pos_sample['bt_text'] = idn
```

3. Tokenisasi

```

train_aug.head()

texts_train =
train_aug['Normalization'].astype(str).tolist()

labels_train = train_aug['label'].astype(int).tolist()

from Transformers import AutoTokenizer

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

train_encodings = tokenizer(
    texts_train,
    padding=True,
    Truncation=True,
    max_length=128,
    return_tensors="pt"
\

```

4. *Count Vectorizer*

```

#positif
vectorizer_pos = CountVectorizer(
    max_df=0.9,
    min_df=5,
    stop_words=list(stop_words))

dtm_pos =
vectorizer_pos.fit_transform(data_pos['Normalization'])

vocab_pos = vectorizer_pos.get_feature_names_out()

#negatif
vectorizer_neg = CountVectorizer(
    max_df=0.95,
    min_df=5,
    stop_words=list(stop_words))

dtm_neg =
vectorizer_neg.fit_transform(data_neg['Normalization'])

vocab_neg = vectorizer_neg.get_feature_names_out()

```


Lampiran. 8 Code Modeling

1. *Modeling* IndoBERT

a) *Training*

```

model = BertForSequenceClassification.from_pretrained(
    "indobenchmark/indobert-base-p1",
    num_labels=2)

device = torch.device("cuda" if torch.cuda.is_available()
else "cpu")
model.to(device)

Optimizer = torch.optim.AdamW(
    model.parameters(),
    lr=2e-5)

model.train()

Epochs = 3
for Epoch in range(EPOCHS):
    total_loss = 0
    for batch in train_loader:
        Optimizer.zero_grad()
        Input_ids = batch['Input_ids'].to(device)
        Attention_mask = batch['Attention_mask'].to(device)
        labels = batch['labels'].to(device)
        outputs = model(
            Input_ids=Input_ids,
            Attention_mask=Attention_mask,
            labels=labels)
        loss = outputs.loss
        loss.backward()
        Optimizer.step()
        total_loss += loss.item()
    avg_loss = total_loss / len(train_loader)
    print(f"Epoch {Epoch+1} | Loss: {avg_loss:.4f}")

```

b) Testing

```
import torch
from torch.utils.data import DataLoader
model.eval()
predictions = []
true_labels = []
with torch.no_grad():
    for batch in test_loader:
        Input_ids = batch["Input_ids"].to(device)
        Attention_mask = batch["Attention_mask"].to(device)
        labels = batch["labels"].to(device)
        outputs = model(
            Input_ids=Input_ids,
            Attention_mask=Attention_mask
        )
        logits = outputs.logits
        preds = torch.argmax(logits, dim=1)
        predictions.extend(preds.cpu().numpy())
        true_labels.extend(labels.cpu().numpy())
y_pred = predictions
y_test = true_labels
```

2. Topic Modeling

```
lda_pos = LatentDirichletAllocation(
    n_components=5,
    random_state=42)
lda_pos.fit(dtm_pos)
lda_neg = LatentDirichletAllocation(
    n_components=5,
    random_state=42)
lda_neg.fit(dtm_neg)
```

Lampiran. 9 Code Evaluasi Model

1. *K-Fold Cross Validation*

```

kf = KFold(n_splits=5, shuffle=True, random_state=42)
train_scores = []
val_scores = []
Fold_results = []
Fold = 1
class IndoDataset(torch.utils.data.Dataset):
    def __init__(self, encodings, labels):
        self.encodings = encodings
        self.labels = labels
    def __getitem__(self, idx):
        item = {key: torch.tensor(val[idx]) for key, val in
self.encodings.items()}
        item['labels'] = torch.tensor(self.labels[idx])
        return item
    def __len__(self):
        return len(self.labels)
for train_index, val_index in kf.split(texts):
    print(f"\n===== Fold {Fold} =====")
    train_encodings = {key: [val[i] for i in train_index]
for key, val in encodings.items()}
    val_encodings = {key: [val[i] for i in val_index] for
key, val in encodings.items()}
    train_labels = labels.iloc[train_index].tolist()
    val_labels = labels.iloc[val_index].tolist()
    train_dataset = IndoDataset(train_encodings,
train_labels)
    val_dataset = IndoDataset(val_encodings, val_labels)

```

```

model =
AutoModelForSequenceClassification.from_pretrained(
    "indobenchmark/indobert-base-p1",
    num_labels=2
)
Training_args = TrainingArguments(
    output_dir=f"./results_Fold_{Fold}",
    num_train_Epochs=2,
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    save_strategy="no",
    logging_dir="./logs",
    disable_tqdm=True)
trainer = Trainer(
    model=model,
    args=Training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset)
trainer.train()
train_pred = trainer.predict(train_dataset)
train_preds = np.argmax(train_pred.predictions, axis=1)
train_acc = Accuracy_score(train_labels, train_preds)
val_pred = trainer.predict(val_dataset)
val_preds = np.argmax(val_pred.predictions, axis=1)
val_acc = Accuracy_score(val_labels, val_preds)
train_scores.append(train_acc)
val_scores.append(val_acc)
print("Train Accuracy:", train_acc)
print("Validation Accuracy:", val_acc)
print(classification_report(val_labels, val_preds))
report = classification_report(val_labels, val_preds,
output_dict=True)

```

2. *Evaluation Matrix*

```
from sklearn.metrics import classification_report
report = classification_report(
    y_test,
    y_pred,
    target_names=["Negatif", "Positif"]
)
print(report)

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure()
sns.heatmap(cm, annot=True, fmt='d')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

