

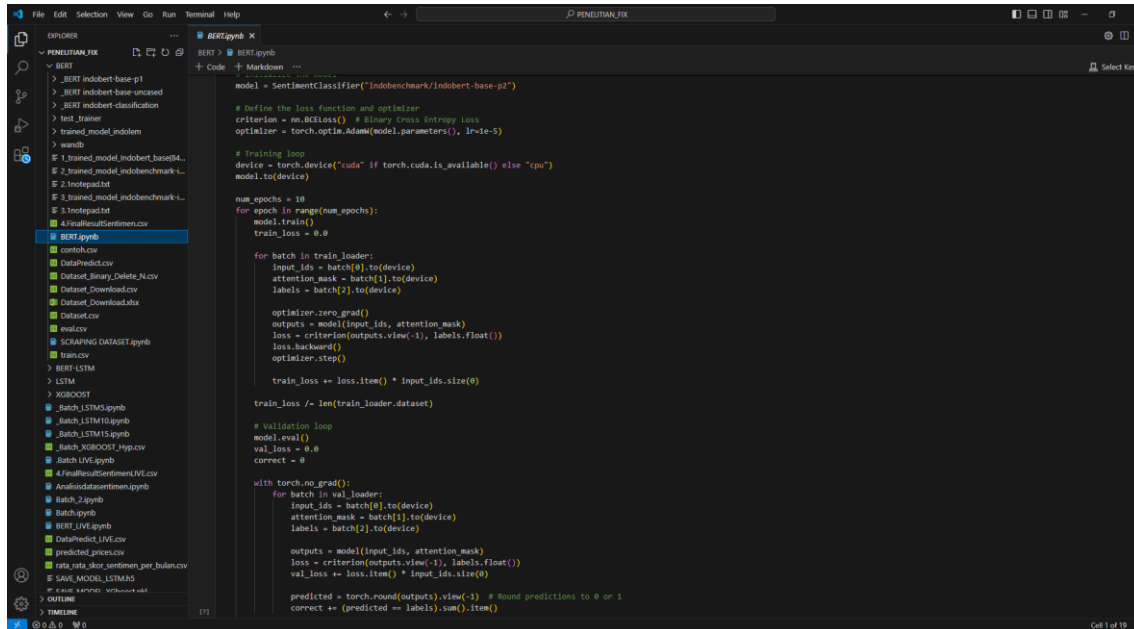
DAFTAR PUSTAKA

- [1] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock Closing Price Prediction using Machine Learning Techniques," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 599–606, 2020, doi: 10.1016/j.procs.2020.03.326.
- [2] M. R. Ramdhan Reska Anung Rustam Hidayat Sri Sulasmianti, "Pengaruh analisis fundamental pendekatan price earning ratio per dalam pengambilan keputusan investasi saham," *J. Adm. Bisnis (JAB)|Vol*, vol. 40, no. 2, 2016.
- [3] X. Weng, X. Lin, and S. Zhao, "Stock Price Prediction Based On Lstm And Bert," *Proc. - Int. Conf. Mach. Learn. Cybern.*, vol. 2022-Sept, pp. 12–17, 2022, doi: 10.1109/ICMLC56445.2022.9941293.
- [4] S. M. Isa, G. Nico, and M. Permana, "Indobert for Indonesian Fake News Detection," *ICIC Express Lett.*, vol. 16, no. 3, pp. 289–297, 2022, doi: 10.24507/icicel.16.03.289.
- [5] L. Y. Kurniawati, H. Tjandrasa, and I. Ariesanti, "Prediksi pergerakan harga saham menggunakan support vector regression," *J. SimanteC*, vol. VIII, no. 1, pp. 11–21, 2013.
- [6] N. E. Monika and M. W. Yusniar, "Analisis Teknikal Menggunakan Indikator MACD dan RSI pada Saham JII," *J. Ris. Inspirasi Manaj. dan Kewirausahaan*, vol. 4, no. 1, pp. 1–8, 2020, doi: 10.35130/jrimk.v4i1.76.
- [7] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, pp. 1–25, 2018, doi: 10.1002/widm.1253.
- [8] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 4171–4186, 2019.
- [9] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, 2014, doi: 10.1016/j.asej.2014.04.011.
- [10] dkk Mas Raden, "Analisis Sentiment Customer Review Aplikasi Ruang Guru dengan Metode BERT (Bidirectional Encoder Representations from

- Transformers),” *Jeisbi*, vol. 02, no. 03, pp. 55–62, 2021.
- [11] P. A. Riyantoko and dkk, “Analisis Prediksi Harga Saham Sektor Perbankan Menggunakan Algoritma Long-Short Terms Memory (Lstm),” *Semin. Nas. ...*, vol. 2020, no. Semnasif, pp. 427–435, 2020, [Online]. Available: <http://www.jurnal.upnyk.ac.id/index.php/semnasif/article/view/4135>
 - [12] M. R. Givari, M. R. Sulaeman, and Y. Umaidah, “Perbandingan Algoritma SVM , Random Forest Dan XGBoost Untuk Penentuan Persetujuan Pengajuan Kredit,” vol. 16, pp. 141–149, 2022.
 - [13] B. Jange, “Prediksi Harga Saham Bank BCA Menggunakan Prophet,” *J. Trends Econ. Account. ...*, vol. 2, no. 1, pp. 1–5, 2021, doi: 10.47065/arbitrase.v3i2.495.
 - [14] T. W. A. Khairi, R. M. Zaki, and W. A. Mahmood, “Stock Price Prediction using Technical, Fundamental and News based Approach,” *SCCS 2019 - 2019 2nd Sci. Conf. Comput. Sci.*, pp. 177–181, 2019, doi: 10.1109/SCCS.2019.8852599.
 - [15] Y. J. Chen and Y. M. Chen, “A fundamental analysis-based method for stock market forecasting,” *Proc. 2013 Int. Conf. Intell. Control Inf. Process. ICICIP 2013*, pp. 354–359, 2013, doi: 10.1109/ICICIP.2013.6568097.
 - [16] A. Namdari and Z. S. Li, “Integrating Fundamental and Technical Analysis of Stock Market through Multi-layer Perceptron,” *2018 IEEE Technol. Eng. Manag. Conf. TEMSCON 2018*, pp. 2–7, 2018, doi: 10.1109/TEMSCON.2018.8488440.
 - [17] A. Sarkar, A. K. Sahoo, S. Sah, and C. Pradhan, “LSTMSA: A Novel Approach for Stock Market Prediction Using LSTM and Sentiment Analysis,” *2020 Int. Conf. Comput. Sci. Eng. Appl. ICCSEA 2020*, no. i, pp. 4–9, 2020, doi: 10.1109/ICCSEA49143.2020.9132928.
 - [18] Z. Zhang, Y. Shen, G. Zhang, Y. Song, and Y. Zhu, “Short-term prediction for opening price of stock market based on self-adapting variant PSO-Elman neural network,” *Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS*, vol. 2017-Novem, pp. 225–228, 2018, doi: 10.1109/ICSESS.2017.8342901.
 - [19] danu salam, M. Musaruddin, and H. T. Mokui, “Kombinasi Analisis Teknis Dan Algoritma Gradient Boosting Untuk Memprediksi Indeks Harga Saham Gabungan,” pp. 1–6, 2020.
 - [20] D. Shah, H. Isah, and F. Zulkernine, “Predicting the Effects of News Sentiments

- on the Stock Market,” *Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018*, pp. 4705–4708, 2019, doi: 10.1109/BigData.2018.8621884.
- [21] C. A. Putri, “Analisis Sentiment Review Film Berbahasa Inggris Dengan Pendekatan Bidirectional Encoder Representations from Transformers,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 6, no. 2, pp. 181–193, 2020, doi: 10.35957/jatisi.v6i2.206.
- [22] R. Man and K. Lin, “Sentiment analysis algorithm based on bert and convolutional neural network,” *Proc. IEEE Asia-Pacific Conf. Image Process. Electron. Comput. IPEC 2021*, pp. 769–772, 2021, doi: 10.1109/IPEC51340.2021.9421110.
- [23] S. N. Himawan, R. Sohiburoyyan, and I. Iryanto, “Hyperparameter Tuning on Graph Neural Network for the Classification of SARS-CoV-2 Inhibitors,” *J. Appl. Informatics Comput.*, vol. 7, no. 2, pp. 186–191, 2023, doi: 10.30871/jaic.v7i2.6735.
- [24] N. L. Chusna, D. Mei, R. Tari, and A. Khumaidi, “Konfigurasi Hyperparameter Long Short Term Memory untuk Optimalisasi Prediksi Penjualan,” *Fakt. Exacta*, vol. 15, no. 4, pp. 1979–276, 2022, doi: 10.30998/faktorexacta.v15i4.15286.
- [25] Sabar Sautomo and Hilman Ferdinandus Pardede, “Prediksi Belanja Pemerintah Indonesia Menggunakan Long Short-Term Memory (LSTM),” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 1, pp. 99–106, 2021, doi: 10.29207/resti.v5i1.2815.

LAMPIRAN



```
model = SentimentClassifier("indobenchmark/indobert-base-p2")

# Define the loss function and optimizer
criterion = nn.BCELoss() # Binary Cross Entropy Loss
optimizer = torch.optim.Adam(model.parameters(), lr=1e-5)

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

num_epochs = 10
for epoch in range(num_epochs):
    model.train()
    train_loss = 0.0

    for batch in train_loader:
        input_ids = batch[0].to(device)
        attention_mask = batch[1].to(device)
        labels = batch[2].to(device)

        optimizer.zero_grad()
        outputs = model(input_ids, attention_mask)
        loss = criterion(outputs.view(-1), labels.float())
        loss.backward()
        optimizer.step()

        train_loss += loss.item() * input_ids.size(0)

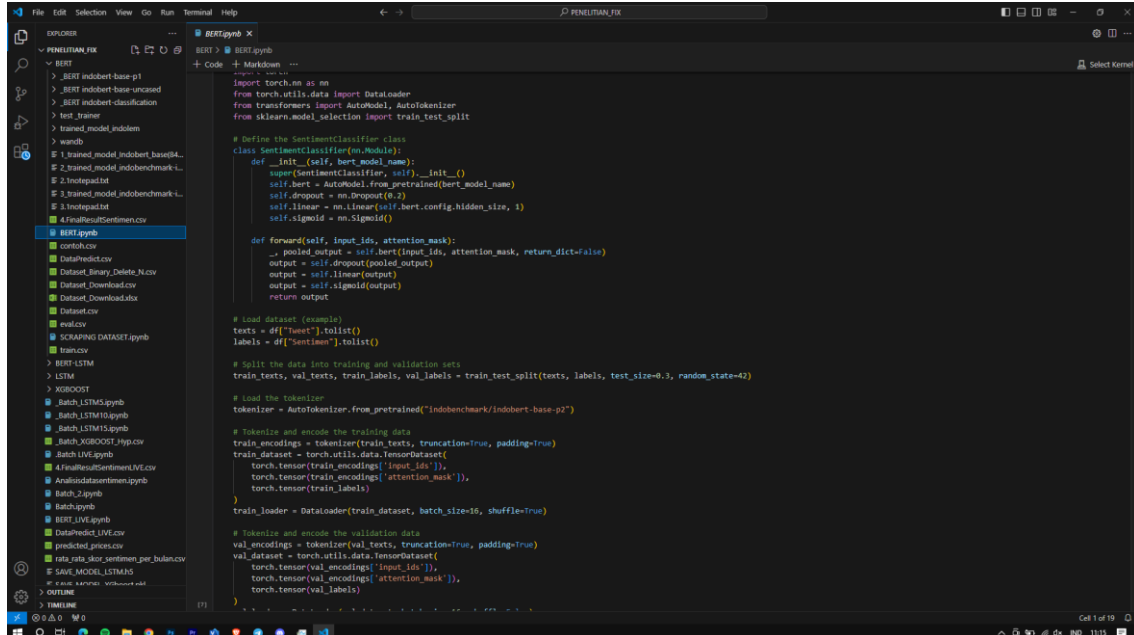
    train_loss /= len(train_loader.dataset)

    # Validation loop
    model.eval()
    val_loss = 0.0
    correct = 0

    with torch.no_grad():
        for batch in val_loader:
            input_ids = batch[0].to(device)
            attention_mask = batch[1].to(device)
            labels = batch[2].to(device)

            outputs = model(input_ids, attention_mask)
            loss = criterion(outputs.view(-1), labels.float())
            val_loss += loss.item() * input_ids.size(0)

            predicted = torch.round(outputs.view(-1)) # Round predictions to 0 or 1
            correct += (predicted == labels).sum().item()
```



```
import torch as nn
from torch.utils.data import DataLoader
from transformers import AutoModel, AutoTokenizer
from sklearn.model_selection import train_test_split

# Define the SentimentClassifier class
class SentimentClassifier(nn.Module):
    def __init__(self, bert_model_name):
        super(SentimentClassifier, self).__init__()
        self.bert = AutoModel.from_pretrained(bert_model_name)
        self.dropout = nn.Dropout(0.2)
        self.linear = nn.Linear(self.bert.config.hidden_size, 1)
        self.sigmoid = nn.Sigmoid()

    def forward(self, input_ids, attention_mask):
        _, pooled_output = self.bert(input_ids, attention_mask, return_dict=False)
        output = self.dropout(pooled_output)
        output = self.linear(output)
        output = self.sigmoid(output)
        return output

# Load dataset (example)
texts = df["Tweet"].tolist()
labels = df["Sentimen"].tolist()

# Split the data into training and validation sets
train_texts, val_texts, train_labels, val_labels = train_test_split(texts, labels, test_size=0.3, random_state=42)

# Load the tokenizer
tokenizer = AutoTokenizer.from_pretrained("indobenchmark/indobert-base-p2")

# Tokenize and encode the training data
train_encodings = tokenizer(train_texts, truncation=True, padding=True)
train_dataset = torch.utils.data.TensorDataset(
    torch.tensor(train_encodings["input_ids"]),
    torch.tensor(train_encodings["attention_mask"]),
    torch.tensor(train_labels)
)
train_loader = DataLoader(train_dataset, batch_size=16, shuffle=True)

# Tokenize and encode the validation data
val_encodings = tokenizer(val_texts, truncation=True, padding=True)
val_dataset = torch.utils.data.TensorDataset(
    torch.tensor(val_encodings["input_ids"]),
    torch.tensor(val_encodings["attention_mask"]),
    torch.tensor(val_labels)
)
val_loader = DataLoader(val_dataset, batch_size=16, shuffle=False)
```



```

df_sentimen.tail()

...

      Date                Tweet  Sentimen
2764  2022-12-30 02:37:55+00:00  bbi akan membagi dividen interim tahun buku 2...  0.998930
2765  2022-12-30 03:44:49+00:00  teams ada yang udah akum bbi awal tahun mau...  0.996459
2766  2022-12-30 04:15:31+00:00  investor bbi bergembira dapat dividen interim...  0.860599
2767  2022-12-30 07:47:27+00:00  terima kasih 2022 untuk tmg aai meds abmm ps...  0.999397
2768  2022-12-30 13:04:36+00:00  akhirnya bbi kjual juga depan bisa beli lagi...  0.995887

df_sentimen['Date'] = pd.to_datetime(df_sentimen['Date'])
df_sentimen['Date'] = df_sentimen['Date'].dt.date
df_sentimen.head()

...

      Date                Tweet  Sentimen
0  2020-07-22  barusan cek bursa karena sehanan full istirahat...  0.310727
1  2020-07-23  ah bbi kapan kau lyk kimo fomo  0.064664
2  2020-07-23  anyinglah bbi sing nyangkul utang sedikit bl...  0.028234
3  2020-07-24  asing gempur telkom idat tampung the next bbi  0.514676
4  2020-07-26  kapan bumi konstruksi mulai bikin anak cucu sa...  0.059208

# Mengonversi kolom "tanggal" menjadi tipe datetime
df_sentimen['Date'] = pd.to_datetime(df_sentimen['Date'])
# Menghitung skor data berdasarkan tanggal dan menjumlahkannya
jumlah_data_per_tanggal = df_sentimen.groupby('Date')['Sentimen'].sum()
# Menghitung jumlah data pada setiap tanggal
jumlah_data = df_sentimen['Date'].value_counts()
# Mengisi hasil jumlah data per tanggal dengan jumlah data pada setiap tanggal
rata_rata_data_per_tanggal = jumlah_data_per_tanggal / jumlah_data
# Menampilkan hasil
df_SentimenSkor = pd.DataFrame(rata_rata_data_per_tanggal)
df_SentimenSkor = df_SentimenSkor.rename(columns={'Sentimen': 'SkorSentimen'})
df_SentimenSkor.head()

...

      SkorSentimen
0  0.310727
1  0.064664
2  0.028234
3  0.514676
4  0.059208

```

```

# https://www.kaggle.com/code/yassinesfahri/lstm-s-p908-stocks-time-series-forecasting/notebook#evaluate-the-model
import pandas as pd
import numpy as np
import os
from keras.models import Sequential
from keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler
import yfinance as yf
import ta

...

folder_path_bert = 'E:\\Dokumen\\S2-Perkuliahan\\Proposal Fik\\Program\\PENELITIAN_FIK\\BERT' # ganti dengan path folder yang sesuai
file_path_bert = os.path.join(folder_path_bert, '4.FinalResultSentimen.csv')

...

# mengambil data sentimen dari folder part_bert
df_sentimen = pd.read_csv(file_path_bert)
df_sentimen.info()

...

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2768 entries, 0 to 2768
Data columns (total 3 columns):
 # Column  Non-Null Count  Dtype
---  --
 0 Date      2769 non-null    object
 1 Tweet     2769 non-null    object
 2 Sentimen  2769 non-null    float64
dtypes: float64(1), object(2)
memory usage: 65.0+ KB

df_sentimen.head()

...

      Date                Tweet  Sentimen
0  2020-07-22 10:23:25+00:00  barusan cek bursa karena sehanan full istirahat...  0.310727

```