

DAFTAR PUSTAKA

Ali, M. Z., Javed, K., Haq, E. ul, & Tariq, A. (2021). *Sentiment and Emotion Classification of Epidemic Related Bilingual data from Social Media* (arXiv:2105.01468). arXiv. <http://arxiv.org/abs/2105.01468>

Al-Saqqa, S., & Awajan, A. (2019, December 12). *The Use of Word2vec Model in Sentiment Analysis: A Survey*. <https://doi.org/10.1145/3388218.3388229>

Astriratma, H. N. I. R. (2020). *Klasifikasi Jenis Pantun Dengan Metode Support Vector Machines (SVM) | Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*.
<https://www.jurnal.iaii.or.id/index.php/RESTI/article/view/2313>

Atmaja, R. M. R. W. P. K., & Yustanti, W. (2021). Analisis Sentimen Customer Review Aplikasi Ruang Guru Dengan Metode BERT (Bidirectional Encoder Representations from Transformers). *Journal of Emerging Information System and Business Intelligence (JEISBI)*, 2(3), Article 3.
<https://ejournal.unesa.ac.id>

Bhavsar, H., & Panchal, M. H. (2012). *A Review on Support Vector Machine for Data Classification*. 1(10).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (arXiv:1810.04805). arXiv. <http://arxiv.org/abs/1810.04805>



Dinar, A. N., Irawan, A. S. Y., & Umaidah, Y. (2023). ANALISIS SENTIMEN PADA PENGGUNA TWITTER TERHADAP PROGRAM KAMPUS MERDEKA MENGGUNAKAN NAÏVE BAYES. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1), Article 1. <https://doi.org/10.36040/jati.v7i1.6552>

Drajana, I. C. R. (2017). METODE SUPPORT VECTOR MACHINE DAN FORWARD SELECTION PREDIKSI PEMBAYARAN PEMBELIAN BAHAN BAKU KOPRA. *ILKOM Jurnal Ilmiah*, 9(2), Article 2. <https://doi.org/10.33096/ilkom.v9i2.134.116-123>

Fauzan, M. F., Purnamasari, A. I., & Dwilestari, G. (2023). PENERAPAN DATA MINING UNTUK MENGANALISIS PENJUALAN AIR MINUM DALAM KEMASAN SELAMA MASA PANDEMI COVID-19. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1), Article 1. <https://doi.org/10.36040/jati.v7i1.6290>

Halim, A., & Safuwan, A. (2023). ANALISIS SENTIMEN OPINI WARGANET TWITTER TERHADAP TES SCREENING GENOSE PENDETEKSI VIRUS COVID-19 MENGGUNAKAN METODE NAÏVE BAYES BERBASIS PARTICLE SWARM OPTIMIZATION. *Jurnal Informatika Teknologi dan Sains (Jinteks)*, 5(1), Article 1. <https://doi.org/10.51401/jinteks.v5i1.2229>



Ikhromr, F. N., Sugiyarto, I., Faddillah, U., & Sudarsono, B. (2023). Implementasi Data Mining Untuk Memprediksi Penyakit Diabetes Menggunakan Algoritma Naives Bayes dan K-Nearest Neighbor. *INTECOMS: Journal of Information Technology and Computer Science*, 6(1), 416–428. <https://doi.org/10.31539/intecom.v6i1.5916>

Insan, M. K. K., Hayati, U., & Nurdian, O. (2023). ANALISIS SENTIMEN APLIKASI BRIMO PADA ULASAN PENGGUNA DI GOOGLE PLAY MENGGUNAKAN ALGORITMA NAIVE BAYES. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1), Article 1. <https://doi.org/10.36040/jati.v7i1.6373>

Jaafar, G. B., Farida, N., & Putri, K. N. I. (2022). ANALISIS STRUKTUR DAN KARAKTER JARINGAN KOMUNIKASI OLAHRAGA TAGAR #TIMNASDAY DI TWITTER. *Mediakom : Jurnal Ilmu Komunikasi*, 6(2), 220–231. <https://doi.org/10.35760/mkm.2022.v6i2.7412>

Khairudin, M., Sukendar, A., & Somantri, A. (2023). ANALISIS SENTIMEN FILM DI TWITTER MENGGUNAKAN METODE SUPPORT VECTOR MACHINE. *Jurnal Sains dan Sistem Teknologi Informasi*, 5(1), Article 1. <https://doi.org/10.59811/sandi.v5i1.47>

Markoulidakis, I., Rallis, I., Georgoulas, I., Kopsiaftis, G., Doulamis, A., & Doulamis, N. (2021). Multiclass *Confusion Matrix* Reduction Method and



Its Application on Net Promoter Score Classification Problem.
Technologies, 9(4), Article 4. <https://doi.org/10.3390/technologies9040081>

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space* (arXiv:1301.3781). arXiv.
<http://arxiv.org/abs/1301.3781>

Nardilasari, A. P., Hananto, A. L., Hilabi, S. S., & Priyatna, B. (2023). *Analisis Sentimen Calon Presiden 2024 Menggunakan Algoritma SVM Pada Media Sosial Twitter. 1.*

Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). *–Teori dan Aplikasinya dalam Bioinformatika1–.*

Nurdin, A., Aji, B. A. S., Bustamin, A., & Abidin, Z. (2020). PERBANDINGAN KINERJA WORD EMBEDDING WORD2VEC, GLOVE, DAN FASTTEXT PADA KLASIFIKASI TEKS. *Jurnal Tekno Kompak*, 14(2), Article 2. <https://doi.org/10.33365/jtk.v14i2.732>

Pakaya, R., Katili, Y., & Latuda, F. (2022). SISTEM PEMILU PROPORSIONAL TERTUTUP DALAM ANALISIS PEMILU 2024. *Jurnal Analisis Sosial Politik*, 1(2), Article 2.



Putra, D., Nama, G. F., & Sulistiono, W. E. (2022). Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode

Naive Bayes Classifier. *Jurnal Informatika Dan Teknik Elektro Terapan*, 10(1), Article 1. <https://doi.org/10.23960/jitet.v10i1.2262>

Suparno, D. S. (2023). Pengenalan Pola Untuk Mengetahui Jumlah Target Pengunjung Mall Metode EDA, K-Means, Hierarchial Clustering, dan Confusion Matrix. *Sains, Aplikasi, Komputasi dan Teknologi Informasi*, 3(2), Article 2. <https://doi.org/10.30872/jsakti.v3i2.4445>

Susanto, H., & Sudiyatno, S. (2014). Data mining untuk memprediksi prestasi siswa berdasarkan sosial ekonomi, motivasi, kedisiplinan dan prestasi masa lalu. *Jurnal Pendidikan Vokasi*, 4(2). <https://doi.org/10.21831/jpv.v4i2.2547>

Tirtayasa, A., & Wibowo, A. L. (2023). Sentiment Analysis Tweet KTT G-20 di Media Sosial Twitter Menggunakan Metode Naïve Bayes. *Jurnal Pengembangan Sistem Informasi Dan Informatika*, 4(2), Article 2. <https://doi.org/10.47747/jpsii.v4i2.1097>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (arXiv:1706.03762). arXiv. <http://arxiv.org/abs/1706.03762>

Wirdayanti, W., Ardiansyah, R., Yudhaswana, Y., Anshori, Y., Pusadan, Y., Angreni, D. S., & Putri, T. I. (2023). IMPLEMENTASI NATURAL LANGUAGE PROCESSING (NLP) UNTUK DETEKSI KESALAHAN JAAN YANG DISEMPURNAKAN (EYD) PADA NASKAH SKRIPSI. *ScientiCO : Computer Science and Informatics Journal*, 6(1), Article 1.



Zulfa, I., & Winarko, E. (2017). Sentimen Analisis Tweet Berbahasa Indonesia Dengan Deep Belief Network. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 11(2), Article 2.
<https://doi.org/10.22146/ijccs.24716>



LAMPIRAN

Lampiran 1. Hasil Prediksi Sentimen

review_text	category	pred
bukti rangkul masyarakat desa bangun desa wawas lingkung cerdas sehat warga lumajang dukung presiden ri	positive	positive
enggak bahasa inggris jos enggak kayak jokowi byuh malu msluin presiden cuma tukang baca teksapes kareba cebong tolol	negative	negative
wes wayahe anies baswedan presiden ri nggih	negative	neutral
dpw p lampung deklarasi ganjar pranowo presiden sumatera utara pranowo	neutral	neutral
one marm anakkolong id setia jakarta warsono jakarta apa capres oligarki jk dampak buruk langsung demokrasi tanah air presiden boneka as cenderung citra bagi dana hibah ormas dukung organisasi larang kompensasi utang budi politik	negative	negative
satu dukung ganjar presiden dki jakarta dukung gp pranowo	neutral	neutral
enggak insya allah anies presiden	negative	neutral
id idha anis presiden mantabeg	neutral	positive
alumni hmi keluar manifesto dukung anies baswedan presiden ri	neutral	neutral
pesan rakyat mu calon presiden kasih sempat putra baik bangsa orang tua bangsa ayom mudamuda	positive	positive



Lampiran 2. Support Vector Machine

```
# Buat objek SVM
SVM_model = SVC()

# Definisikan grid parameter yang akan dioptimisasi
param_grid = {
    'C': [0.1, 1, 10],
    'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
    'gamma': ['scale', 'auto']
}

# Penalaan hyperparameter menggunakan GridSearchCV
grid_search = GridSearchCV(estimator=SVM_model, param_grid=param_grid, cv=5,
n_jobs=-1)
grid_search.fit(X_train, y_train)

# Hasil dari penalaan hyperparameter terbaik
best_params = grid_search.best_params_
print("Best hyperparameters:", best_params)

# Mencetak nilai akurasi dari setiap kombinasi parameter yang diuji
means = grid_search.cv_results_['mean_test_score']
stds = grid_search.cv_results_['std_test_score']
for mean, std, params in zip(means, stds, grid_search.cv_results_['params']):
    print("Parameters:", params)
    print("Accuracy:", mean)
    print("Standard Deviation:", std)
    print()

# Membuat model SVM dengan hyperparameter terbaik
best_SVM_model = SVC(C=best_params['C'], kernel=best_params['kernel'],
best_params['gamma'])
best_SVM_model.fit(X_train, y_train)
```




```
# Prediksi menggunakan model terbaik
y_pred = best_SVM_model.predict(X_test)

# Evaluasi model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("Classification Report:")
print(classification_report(y_test, y_pred))
```



Lampiran 3. BERT

```
# Function to set seed
def set_seed(seed):
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed(seed)

# Function to count parameters
def count_param(module, trainable=False):
    if trainable:
        return sum(p.numel() for p in module.parameters() if p.requires_grad)
    else:
        return sum(p.numel() for p in module.parameters())

# Function to get learning rate
def get_lr(optimizer):
    for param_group in optimizer.param_groups:
        return param_group['lr']

# Function to convert metrics to string
def metrics_to_string(metric_dict):
    string_list = []
    for key, value in metric_dict.items():
        string_list.append('{:}.{:2f}'.format(key, value))
    return ' '.join(string_list)

# Load Tokenizer and Config
tokenizer = BERTTokenizer.from_pretrained('indobenchmark/indoBERT-base-p1')
config = BERTConfig.from_pretrained('indobenchmark/indoBERT-base-p1')
config.num_labels = DocumentSentimentDataset.NUM_LABELS
```

iate model



```

model = BERTForSequenceClassification.from_pretrained('indobenchmark/indoBERT-
base-p1', config=config)

# Training loop
n_epochs = 5
history = defaultdict(list)

for epoch in range(n_epochs):
    model.train()
    torch.set_grad_enabled(True)

    total_train_loss = 0
    list_hyp_train, list_label = [], []

    train_pbar = tqdm(train_loader, leave=True, total=len(train_loader))
    for i, batch_data in enumerate(train_pbar):
        # Forward model
        loss, batch_hyp, batch_label = forward_sequence_classification(model,
batch_data[:-1], i2w=i2w, device='cuda')

        # Update model
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        tr_loss = loss.item()
        total_train_loss = total_train_loss + tr_loss

    # Calculate train metrics score
    list_hyp_train += batch_hyp
    list_label += batch_label

    train_pbar.set_description("(Epoch {}) TRAIN LOSS:{:.4f}
)".format((epoch+1),
total_train_loss/(i+1), get_lr(optimizer)))

```



```

metrics = document_sentiment_metrics_fn(list_hyp_train, list_label)
print("(Epoch {}) TRAIN LOSS:{:.4f} {} LR:{:.8f}".format((epoch+1),
    total_train_loss/(i+1), metrics_to_string(metrics), get_lr(optimizer)))

# Save train acc for learning curve
history['train_acc'].append(metrics['ACC'])

# Evaluation on the validation set
model.eval()
torch.set_grad_enabled(False)

total_loss, total_correct, total_labels = 0, 0, 0
list_hyp, list_label = [], []

pbar = tqdm(valid_loader, leave=True, total=len(valid_loader))
for i, batch_data in enumerate(pbar):
    batch_seq = batch_data[-1]
    loss, batch_hyp, batch_label = forward_sequence_classification(model,
batch_data[:-1], i2w=i2w, device='cuda')

    # Calculate total loss
    valid_loss = loss.item()
    total_loss = total_loss + valid_loss

    # Calculate evaluation metrics score
    list_hyp += batch_hyp
    list_label += batch_label
    metrics = document_sentiment_metrics_fn(list_hyp, list_label)

    pbar.set_description("VALID LOSS:{:.4f} {}".format(total_loss/(i+1),
metrics_to_string(metrics)))

```



```

s = document_sentiment_metrics_fn(list_hyp, list_label)
(Epoch {}) VALID LOSS:{:.4f} {}".format((epoch+1),

```

```
total_loss/(i+1), metrics_to_string(metrics)))  
  
# Save validation acc for learning curve  
history['val_acc'].append(metrics['ACC'])
```



Lampiran 4. Cara Kerja SVM Secara Sederhana

Misalkan terdapat data:

dim1	dim2	Label
0.34	-0.45	1
1.45	0.34	1
0.32	0.67	0
-0.12	-0.56	0
0.66	0.75	2
0.45	0.32	2

1. Hitung Vektor Bobot w dan Konstanta Bias b

Contoh ini akan menggunakan kernel linear untuk menghitung nilai hyperplane sesuai dengan persamaan (3)

Data Latih:

a. Data latih dari kelas 1:

$$X_1 = \begin{bmatrix} 0.34 & -0.45 \\ 1.45 & 0.34 \end{bmatrix}, \quad y_1 = [1, 1]$$

b. Data latih dari kelas 2:

$$X_2 = \begin{bmatrix} 0.66 & 0.75 \\ 0.45 & 0.32 \end{bmatrix}, \quad y_2 = [2, 2]$$

c. Data latih dari kelas 0:

$$X_0 = \begin{bmatrix} 0.32 & 0.67 \\ -0.12 & -0.56 \end{bmatrix}, \quad y_0 = [0, 0]$$

2. Hitung nilai w dan b

a. Hitung nilai w untuk kelas 1

$$\begin{aligned} w_1 &= \frac{1}{2}((0.34 \cdot 1) + (1.45 \cdot 1)) - \frac{1}{2}((-0.45 \cdot 1) + (0.34 \cdot 1)) \\ w_1 &= \frac{1}{2}(0.34 + 1.45 - 0.45 + 0.34) \\ w_1 &= \frac{1}{2}(1.68) \\ w_1 &= 0.84 \end{aligned}$$



b. Hitung nilai w untuk kelas 0

$$w_0 = 0$$

$$w_0 = \frac{5}{7}((0.35 \cdot 0) + (-0.15 \cdot 0)) - \frac{5}{7}((0.05 \cdot 0) + (-0.20 \cdot 0))$$

c. Hitung nilai w untuk kelas 2

$$w_2 = \frac{1}{2}((0.66 \cdot 2) + (0.45 \cdot 2)) - \frac{1}{2}((0.75 \cdot 2) + (0.32 \cdot 2))$$

$$w_2 = \frac{1}{2}(1.32 + 0.90 - 1.50 - 0.64)$$

$$w_2 = \frac{1}{2}(0.08)$$

$$w_2 = 0.04$$

d. Hitung b

Untuk menghitung nilai b kita dapat menggunakan salah satu contoh data. Misalnya, kita ambil data $(0.34, -0.45)$ dari kelas 1

$$b = -1 + 0.84 \cdot 0.34$$

$$b = -1 + 0.2856$$

$$b = -0.7144$$

3. Prediksi Label

Untuk memprediksi label dari data uji atau data lainnya. Misalnya, kita ingin memprediksi label untuk data $(0.66, 0.75)$:

$$y = \text{sign}(0.04 \cdot 0.66 + 0.84 \cdot 0.75 - 0.7144)$$

$$y = \text{sign}(0.0264 + 0.63 - 0.7144)$$

$$y = \text{sign}(-0.058)$$

Berdasarkan hasil ini, kita dapat mengklasifikasikan data tersebut ke kelas 0 karena hasilnya negatif.

Cara sederhana klasifikasi menggunakan OvA:

1. Pisahkan Data Pisahkan data menjadi data latih dan data uji. Untuk contoh ini, mari kita gunakan dua data sebagai data latih dan empat data lainnya sebagai data uji.

Data Latih:

$(0.34, -0.45)$ $(0.34, -0.45)$ -> Label 1 (Kelas A)

$(0.32, 0.67)$ $(0.32, 0.67)$ -> Label 0 (Kelas B)



Data Uji:

(1.45 , 0.34) (1.45,0.34) -> Label 1 (Kelas A)

(- 0.12 , - 0.56) (-0.12,-0.56) -> Label 0 (Kelas B)

(0.66 , 0.75) (0.66,0.75) -> Label 2 (Kelas C)

(0.45 , 0.32) (0.45,0.32) -> Label 2 (Kelas C)

2. Latih Model SVM untuk Setiap Kelas

Model untuk Kelas A (Label 1) versus Kelas Lain (0 dan 2)

Ubah label: Label A menjadi 1, label lainnya menjadi -1.

a. Data latih: (0.34 , - 0.45) (0.34,-0.45) -> Label 1, (0.32 , 0.67) (0.32,0.67) -> Label -1

b. Model SVM menghasilkan vektor bobot w_A dan konstanta bias b_A .

Model untuk Kelas B (Label 0) versus Kelas Lain (1 dan 2)

Ubah label: Label B menjadi 1, label lainnya menjadi -1.

a. Data latih: (0.32 , 0.67) (0.32,0.67) -> Label 1, (0.34 , - 0.45) (0.34,-0.45) -> Label -1

b. Model SVM menghasilkan vektor bobot w_B dan konstanta bias b_B .

Model untuk Kelas C (Label 2) versus Kelas Lain (0 dan 1)

Ubah label: Label C menjadi 1, label lainnya menjadi -1.

a. Data latih: (0.66 , 0.75) (0.66,0.75) -> Label 1, (0.45 , 0.32) (0.45,0.32) -> Label -1

b. Model SVM menghasilkan vektor bobot w_C dan konstanta bias b_C .

3. Prediksi Label Data Uji

a. Data (1.45 , 0.34) (1.45,0.34) -> Label 1 (Kelas A):

- Gunakan model A: w_A dan b_A
- Hitung $f_A(x) = w_A \cdot x + b_A$
- Jika $f_A(x) > 0$, prediksi Kelas A; jika $f_A(x) < 0$, prediksi bukan Kelas A.



b. Data (- 0.12 , - 0.56) (-0.12,-0.56) -> Label 0 (Kelas B):

- Gunakan model B: w_B , b_B
- Hitung $f_B(x) = w_B \cdot x + b_B$
- Jika $f_B(x) > 0$, prediksi Kelas B; jika $f_B(x) < 0$, prediksi bukan Kelas B.

c. Data (0.66 , 0.75) (0.66,0.75) -> Label 2 (Kelas C):

- Gunakan model C: w_C , b_C
- Hitung $f_C(x) = w_C \cdot x + b_C$
- Jika $f_C(x) > 0$, prediksi Kelas C; jika $f_C(x) < 0$, prediksi bukan Kelas C.

d. Data (0.45 , 0.32) (0.45,0.32) -> Label 2 (Kelas C):

- Gunakan model C: w_C , b_C
- Hitung $f_C(x) = w_C \cdot x + b_C$
- Jika $f_C(x) > 0$, prediksi Kelas C; jika $f_C(x) < 0$, prediksi bukan Kelas C.

4. Voting atau Pemilihan

Dari hasil prediksi di atas, kita dapat menggunakan strategi voting atau pemilihan berdasarkan probabilitas untuk menentukan kelas akhir dari data uji.

Misalnya, jika $f_A(x) > 0$ dan $f_B(x) < 0$

untuk data uji pertama, maka dapat diputuskan bahwa data tersebut termasuk dalam Kelas A (Label 1).



Lampiran 5. Lembar Perbaikan Skripsi

LEMBAR PERBAIKAN SKRIPSI

“ANALISIS SENTIMEN KANDIDAT CALON PRESIDEN 2024 MENGUNAKAN *IndoBERT* DAN *SUPPORT VECTOR MACHINE*”

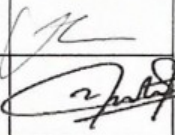
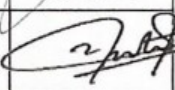
OLEH:

MUHAMMAD AKIB

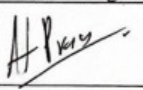
D121171304

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana pada tanggal 6 Mei 2024.
Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
I	Dr. Ir. Zahir Zainuddin, M.Sc	
II	Prof. Dr.Eng Intan Sari Areni., ST., MT.,	

Persetujuan perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	A. Ais Prayogi Alimuddin, S.T., M.Eng.	
II	Anugrayani Bustamin, S. T., M.T.	

