

Analisis Sentimen Terhadap Aplikasi Lowongan Kerja Menggunakan *Lexicon Based Features* Dan *Support Vector Machine*

Muhammad Nurkarifin^{1*}, Teguh Iman Hermanto², Mochzen G Resmi³
 Teknik Informatika, STT Wastukencana Purwakarta, Purwakarta, Indonesia
 *e-mail *Corresponding Author*: m.nurkarifin@gmail.com

Abstract

The use of job vacancy platforms is often used by job seekers. After users utilize the platform or job vacancy application, quite a few users leave reviews on the application so it is necessary to conduct a study regarding sentiment analysis of job vacancy applications using Lexicon based features and Support vector machines (SVM). The aim of this research is to find out the results of applying the Lexicon based method features and support vector machine for sentiment analysis of job vacancy applications. The research method used is Lexicon based features and Support vector machine with 3 research objects, namely Glints, Pintarnya and Kupu. Data collection was carried out through a scraping process on job vacancy applications in the Google Play Store. The collected data will be processed, carried out by TF-IDF, studied using Lexicon based features and support vector machines and then evaluated so that research results can be obtained that can be accounted for. The research results show that the approach through a combination of TF-IDF, Lexicon based features, and SVM with hyperparameter tuning produces good performance in sentiment analysis on text data with accuracies of 81% (Glints), 76% (Butterfly), and 82% for the Smart application

Keywords: *Sentiment Analysis; Job Vacancies Application; Lexicon Based Features; Support Vector Machine*

Abstrak

Setelah pengguna memanfaatkan *platform* atau aplikasi lowongan kerja dalam mencari kerja, tidak sedikit para pengguna meninggalkan ulasan pada aplikasi tersebut sehingga perlu dilakukan kajian mengenai analisis sentimen terhadap aplikasi lowongan kerja tersebut. Tujuan penelitian ini adalah untuk mengetahui hasil dari penerapan metode *Lexicon based features* dan *Support Vector Machine* (SVM) terhadap analisis sentimen aplikasi lowongan kerja. Metode penelitian yang digunakan ialah *Lexicon based features* dan *Support vector machine* dengan 3 objek penelitian yaitu *Glints*, *Pintarnya* dan *Kupu*. Pengumpulan data dilakukan melalui proses *scraping* pada aplikasi lowongan kerja yang ada di *Google play store*. Data yang terkumpul akan di proses, dilakukan TF-IDF, dikaji melalui *Lexicon based features* dan *Support vector machine* kemudian dievaluasi sehingga diperoleh hasil penelitian yang dapat dipertanggung jawabkan. Hasil penelitian menunjukkan bahwa pendekatan melalui kombinasi TF-IDF, fitur berbasis leksikon, dan SVM dengan penyetulan hyperparameter menghasilkan performa yang baik dalam analisis sentimen pada data teks dengan akurasi 81% (*Glints*), 76% (*Kupu*), dan 82% untuk aplikasi *Pintarnya*.

Kata kunci: *Analisis Sentimen; Aplikasi; Lexicon Based Features; Lowongan Kerja; Support Vector Machine*

1. Pendahuluan

Informasi mengenai lowongan pekerjaan merupakan hal penting bagi masyarakat yang belum memiliki pekerjaan. Karena melalui bekerja maka seseorang dapat meningkatkan taraf hidupnya dan melanjutkan kehidupannya dengan baik [1]. Pekerjaan memiliki kaitan erat dengan ekonomi [2][3] sehingga tidak sedikit masyarakat yang sering mencari lowongan pekerjaan melalui media *smarthphone* atau android yang dimilikinya. Informasi mengenai data platform lowongan kerja yang paling banyak digunakan oleh masyarakat pada tahun 2022 yaitu *jobstreet* dengan persentase 51,4%. Kemudian pada posisi berikutnya ialah *Linkedin* dengan

persentase sebanyak 38%, *Karier.com* dengan persentase senilai 22.9%, *Jobs.id* dengan presentase 19%, *Glints* senilai 10.7%, dan *Kalibr* senilai 9.2%, dimana data ini diperoleh dari survei yang dilakukan melalui aplikasi Jakpat pada 3.473 respondeng pada 1 Agustus 2022. Beragam aplikasi yang digunakan, tentu saja memberikan impresi atau kesan tersendiri bagi para penggunanya dimana kesan ini biasanya disampaikan melalui ulasan yang diberikan pada aplikasi. Opini yang diberikan publik melalui ulasan yang ditinggalkan dapat dimanfaatkan untuk mencari informasi mengenai evaluasi dan penilaian layanan aplikasi yang digunakan termasuk pada aplikasi *jobstreet* yang telah menerapkan analisa sentimen [4].

Pengenalan dan ekstraksi opini dalam teks adalah fokus analisis sentiment dengan sebuah sub bidang *Natural Language Processing* (NLP). Penggunaan analisis teks yang diperoleh dari platform media sosial memungkinkan peneliti mengumpulkan umpan balik pengguna dan memperoleh sentimen yang mendasari suatu bagian tertentu. Penambahan opini, juga dikenal sebagai analisis sentimen adalah gabungan pengumpulan data dan teks yang digunakan untuk menentukan sejauh mana suatu topik, produk, layanan, organisasi, orang, atau serangkaian aktivitas tertentu menarik bagi pembicara atau penulis tertentu. Analisis sentimen mengungkapkan keseluruhan nada dari sebuah tulisan. Karena sentimen bersifat spesifik terhadap suatu subjek. Pernyataan serupa yang dibuat sehubungan dengan dua subjek berbeda mungkin mempunyai arti yang sangat berbeda. Dalam beberapa proyek penelitian, langkah pertama untuk menganalisis sentiment adalah mengidentifikasi fitur produk yang sedang dibahas [5]. Salah satu metode yang digunakan untuk mencapai tujuan pencapaian unsur-unsur tersebut adalah metode yang didasarkan pada *Lexicon based*.

Metode berbasis leksikon adalah metode yang didasarkan pada pendapat orang-orang yang mengungkapkan pendapatnya melalui penggunaan leksikon pendapat yang disebut juga dengan leksikon opini. Metode ini didasarkan pada cara penulisan kalimat, dan cara kalimat tersebut mengungkapkan sentimen positif atau negative [6][7]. Pada penelitian yang dilakukan dengan metodologi berbasis leksikon, hasil akurasi analisis sentimen efektif sebesar 66%. Selain itu ada metode lain yang banyak digunakan yaitu teknik *machine learning*. Salah satu algoritma klasifikasi yang sering digunakan untuk analisis sentimen dikenal dengan *Support Vector Machine* (SVM). Untuk keperluan mentransfer data ke dalam beberapa kelas sentimen yang berbeda, SVM melakukan klasifikasi dengan mencari *hyperplane* (garis pemisah) yang paling sesuai. Salah satu kelebihan SVM adalah kemampuannya dalam memberikan hasil yang akurat terutama dalam hal klasifikasi untuk dua kelas [8]-[10].

Maka dari itu, tidak sedikit masyarakat yang menginstal aplikasi-aplikasi ini sehingga tidak sedikit pula para pengguna yang meninggalkan ulasan mengenai aplikasi yang digunakan. Dengan ini maka pemeriksaan komentar opini publik dapat digunakan untuk mencari informasi mengenai evaluasi dan penilaian layanan aplikasi pencari kerja itu sendiri. Dari setiap aplikasi pencari kerja yang sering digunakan tidak menutup kemungkinan akan terjadi penilaian (*review* dan *rating*) yang diberikan sistem kepada pengguna setelah menjalankan sebuah aplikasi. Disetiap penilaian pasti ada pengguna berkomentar yang bersifat negatif dan positif pada sebuah aplikasi pencari kerja. Maka dengan banyaknya komentar dari pengguna sulit untuk mengetahui seberapa banyak komentar positif atau negatif dan apakah aplikasi tersebut lebih cenderung ke komentar positif atau negatif, sehingga perlu dilakukan kajian yang lebih mendalam untuk perbaikan performa juga layanan di kemudian hari. Kajian mengenai penggunaan metode Leksikon dan SVM telah dilakukan oleh beberapa peneliti diantaranya, penelitian yang dilakukan oleh Hermawan dkk. pada tahun 2018 tentang metode berbasis leksikon dan *support vector machine* untuk menganalisis sentimen di media social sebagai rekomendasi oleh-oleh favorit dan penelitian yang dilakukan oleh Ruslim dkk. pada tahun 2019 tentang analisis sentimen pada review aplikasi mobile banking menggunakan metode *Support vector machine* dan fitur berbasis leksikon [11], Kedua penelitian ini merupakan contoh penelitian yang menyelidiki sentimen dalam kaitannya dengan suatu aplikasi. Namun dari beragam penelitian yang dilakukan, belum ada kajian yang melakukan analisis terhadap aplikasi lowongan kerja melalui *Lexicon based feature* dan *Support vector machine*, sehingga mendorong peneliti untuk melakukan penelitian dengan judul "Analisis Sentimen Terhadap Aplikasi Lowongan Kerja Menggunakan *Lexicon based Features dan Support Vector Machine*".

Tujuan dari penelitian ini ialah untuk mengetahui hasil dari penerapan metode *Lexicon based features* dan *Support vector machine* terhadap analisis sentimen aplikasi lowongan kerja, dimana hasil penelitian dapat digunakan sebagai dasar perbaikan performa dan layanan aplikasi lowongan kerja.

2. Tinjauan Pustaka

Terdapat beberapa penelitian terdahulu yang relevan dengan penelitian ini diantaranya: *Pertama*, penelitian yang dilakukan oleh Mohammad Aldinugroho Abdullah dan Deni Mahdiana pada tahun 2022 dimana penelitian ini bertujuan untuk melakukan kajian mendalam mengenai peluang kerja yang ada di Indonesia saat masa *Covid-19* dengan metode *Naïve Bayes*. Hasil penelitian menunjukkan bahwa nilai AUC yang paling tinggi terdapat pada *Random forest* dengan nilai 79,40%, sedangkan nilai presisi yang paling akurat ialah dengan menggunakan model *Naïve Bayes* senilai 87,88% [12]. Persamaan penelitian Mohammad Aldinugroho Abdullah dan Deni Mahdiana dengan penelitian yang peneliti lakukan ialah sama-sama melakukan analisis sentimen pada aplikasi lowongan kerja, sedangkan perbedaannya ialah penggunaan metode atau *Modle* yang digunakan untuk menganalisis, dimana pada penelitian Mohammad Aldinugroho Abdullah dan Deni Mahdiana menggunakan model *Naïve Bayes* sedangkan pada penelitian yang peneliti lakukan menggunakan *Lexicon based features* dan *Support Vector Machine*.

Kedua, penelitian yang dilakukan oleh Hermawan, et al pada tahun 2018 dengan judul *Metode Lexicon Based Dan Support Vector Machine Untuk Menganalisis Sentimen Pada Media Sosial Sebagai Rekomendasi oleh-oleh Favorit*. Berdasarkan temuan penelitian yang dilakukan Hermawan dan kawan-kawan, terdapat data yang memiliki hasil presisi tertinggi yakni 94,23% dan akurasi maksimal yakni 87,78% bila berbasis *Lexicon*. Selama ini, pendekatan *Support Vector Machine* memberikan hasil recall tertinggi, khususnya 100% [10]. Adanya kesamaan antara penelitian yang dilakukan oleh Hermawan dkk. dan penelitian yang dilakukan oleh peneliti lainnya adalah kedua jenis penelitian ini menggunakan metodologi *Lexicon Based* dan *Support Vector Machine*. Penelitian yang dilakukan Hermawan dan rekan untuk menilai sentimen di media sosial sebagai saran oleh-oleh favorit berbeda dengan penelitian yang dilakukan peneliti untuk menganalisis sentimen pada lamaran lowongan kerja. Shemawan dan rekannya melakukan penelitian sebelumnya.

Ketiga, penelitian yang dilakukan oleh Nova Andre Saputra, Jessica Alexandra dan Indra Budi Trisno pada tahun 2023, dimana pada penelitiannya bertujuan untuk melakukan kajian secara mendalam mengenai pemanfaatan obrolan grup telegram mengenai beragam informasi lowongan kerja dengan metode *Naïve bayes classifier*. Hasil penelitian menunjukkan bahwa dengan memanfaatkan *text mining*, grup telegram yang digunakan untuk berbagi lowongan kerja menjadi lebih tertata dan dapat diakses secara publik tanpa adanya *spam* yang mengganggu anggota lain, sehingga tujuan pembuatan grup menjadi lebih optimal [13]. Persamaan penelitian Nova Andre Sputra, Jessica Alexandra dan Indra Budi Trisno ialah sama-sama melakukan analisis sentiment pada lowongan kerja. Sedangkan perbedaan dengan penelitian yang peneliti lakukan ialah penelitian Nova Andre Saputra, Jessica Alexandra, dan Indra Budi Trisno melakukan penelitian mengenai analisis sentimen lowongan kerja melalui grup telegram, sedangkan penelitian yang peneliti lakukan ialah menganalisis sentiment pada aplikasi lowongan kerja dengan metode *Lexicon based features* dan *Support vector machine*.

Penelitian ini akan menguji penggunaan metode *Lexicon based features*, dan *Support vector machine* untuk menganalisis *sentiment*, dimana ekstraksi yang dilakukan digunakan untuk memperoleh informasi tertentu secara optimal.

3. Metodologi

3.1 Prosedur Penelitian

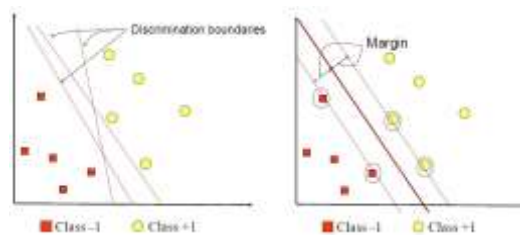
Metode penelitian yang digunakan pada penelitian ini ialah *Lexicon based features* dan *Support Vector Machine* dengan objek penelitian ialah aplikasi lowongan kerja yaitu *Slints*, *Pintarnya* dan *KUPU*. Teknik pengumpulan data dilakukan melalui teknik *scarping* data menggunakan *Anaconda* dengan panel *Jupyter* diperoleh data total melalui *Glints* 640 dengan 320 sentimen positif dan 320 sentimen negative. *Pintarnya* total 1000 data dengan 500 sentimen positif dan 500 sentimen negatif, dan *KUPU* total 1000 data dengan 500 sentimen positif dan 500 sentimen negatif. Total dataset keseluruhan 3 aplikasi yaitu 2640 data. Data yang telah terkumpul kemudian diolah untuk mengetahui bobot (w) dengan memanfaatkan *Term Frequency Invers Dokumen Frequency* (TF-IDF) untuk memberi bobot pada kata. Mulailah dengan mencari *Term Frekuensi* (TF), *Dokumen Frekuensi* (DF), *Inverse Dokumen Frekuensi* (IDF), dan nilai TF-IDF. Tujuan dari pembobotan kata ini adalah untuk memastikan bahwa setiap kata mengurangi kesulitan dalam proses kategorisasi. Setelah itu akan dilakukan

analisis sentimen untuk mengekstrak data opini sedemikian rupa sehingga dapat dipahami dan diolah secara tekstual guna menentukan sentimen yang terkandung di dalamnya [14].

3.2 Support Vector Machine

Support Vector Machine (SVM) adalah teknik yang relatif baru untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi [14]. SVM adalah pengklasifikasi linier non-probabilistik, biner. Pemahaman klasifikasi menggunakan SVM seringkali diawali dengan pencarian garis ideal (*hyperplane*). Nilai positif (+1) dan negatif (-1) begitu dibedakan satu sama lain. Gambar 2 menggambarkan data positif berwarna kuning (+1) dan data negatif berwarna merah (-1).

Gambar 1 menggambarkan prosedur menyeluruh analisis SVM. Gambar 3 di sebelah kiri menggambarkan grafik yang menunjukkan berbagai garis partisi (batas diskriminasi) di SVM untuk mengevaluasi kumpulan data. Sementara itu, grafik sebelah kanan menunjukkan margin diskriminasi terbesar. Batas, atau margin, adalah jarak *hyperplane* antara dua kelas data terdekat. Hasil generalisasi dan klasifikasi terbaik akan diperoleh dari *hyperplane* dengan margin terbesar.



Gambar 1. Proses SVM dalam Menemukan *Hyperline*

4. Hasil dan Pembahasan

4.1 Hasil Analisis

Setelah dilakukan beragam tahapan pengolahan data maka diketahui hasil analisis data yang dilakukan dengan dua pendekatan utama. Pendekatan pertama adalah dengan menerapkan metode TF-IDF untuk menghasilkan representasi numerik dari teks berdasarkan bobot *term frekuensi-inverse* dokumen. Sedangkan, pendekatan kedua melibatkan ekstraksi fitur berbasis leksikon, di mana kamus leksikon kata dan kata keterangan digunakan untuk menentukan bobot sentimen pada teks. Hasil analisis pada data penelitian ini melalui *Lexicon Based Features* yaitu:

```
lexicon_words_df.head()
```

	word	weight	number_of_words
0	hai	3	1
1	merekam	2	1
2	ekstensif	3	1
3	paripurna	1	1
4	detail	2	1

Gambar 2. *Lexicon Based Features*

Hasil analisis pembobotan TF-IDF disajikan pada Gambar 3.

```

X_train:
TF-IDF_0 TF-IDF_1 TF-IDF_2 TF-IDF_3 TF-IDF_4 TF-IDF_5 TF-IDF_6 \
0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
1 0.0 0.0 0.0 0.0 0.0 0.0 0.0
2 0.0 0.0 0.0 0.0 0.0 0.0 0.0
3 0.0 0.0 0.0 0.0 0.0 0.0 0.0
4 0.0 0.0 0.0 0.0 0.0 0.0 0.0
5 0.0 0.0 0.0 0.0 0.0 0.0 0.0
6 0.0 0.0 0.0 0.0 0.0 0.0 0.0
7 0.0 0.0 0.0 0.0 0.0 0.0 0.0
8 0.0 0.0 0.0 0.0 0.0 0.0 0.0
9 0.0 0.0 0.0 0.0 0.0 0.0 0.0

TF-IDF_7 TF-IDF_8 TF-IDF_9 ... TF-IDF_622 TF-IDF_623 TF-IDF_624 \
0 0.0 0.0 0.0 ... 0.0 0.0 0.0
1 0.0 0.0 0.0 ... 0.0 0.0 0.0
2 0.0 0.0 0.0 ... 0.0 0.0 0.0
3 0.0 0.0 0.0 ... 0.0 0.0 0.0
4 0.0 0.0 0.0 ... 0.0 0.0 0.0
5 0.0 0.0 0.0 ... 0.0 0.0 0.0
6 0.0 0.0 0.0 ... 0.0 0.0 0.0
7 0.0 0.0 0.0 ... 0.0 0.0 0.0
8 0.0 0.0 0.0 ... 0.0 0.0 0.0
9 0.0 0.0 0.0 ... 0.0 0.0 0.0

...
8 -2.00
9 1.25

```

Gambar 3. Hasil Pembobotan TF-IDF

Setelah dilakukan ekstraksi fitur, algoritme *Support Vector Machine* (SVM) digunakan sebagai metode klasifikasi. Dalam penggunaan SVM, dilakukan penyetelan *hyperparameter* menggunakan metode *Grid Search* untuk mencari kombinasi parameter terbaik yang menghasilkan performa model yang optimal.

```
Kombinasi parameter terbaik: {'C': 1, 'gamma': 0.1, 'kernel': 'linear'}
```

Gambar 4. Hasil Parameter SVM

Gambar 5 berikut ini adalah implementasi kode dari tahapan analisis data:

```

kode:
def analyze_sentiment(df):
    # Baca kamus leksikon-words dan kata-keterangan
    lexicon_words_df = pd.read_csv('data/lexicon-word.csv')
    kata_keterangan_df = pd.read_csv('data/kata-keterangan.csv')

    # Membangun kamus Lexicon-words
    lexicon_words = {}
    for _, row in lexicon_words_df.iterrows():
        lexicon_words[row['word']] = {
            'weight': row['weight'],
            'number_of_words': row['number_of_words']
        }

    # Membangun kamus Kata-keterangan
    kata_keterangan = {}
    for _, row in kata_keterangan_df.iterrows():
        kata_keterangan[row['words']] = row['weight']

    # Bagi dataset menjadi data latih dan data uji
    X_train, X_test, y_train, y_test = train_test_split(df['final'], df['sentimen'], test_size=0.1,
        random_state=42)

    # Ekstraksi fitur menggunakan TF-IDF
    vectorizer = TfidfVectorizer()
    X_train_tfidf = vectorizer.fit_transform(X_train)
    X_test_tfidf = vectorizer.transform(X_test)

    # Ekstraksi fitur berbasis leksikon
    sia = SentimentIntensityAnalyzer()

    def get_lexicon_score(text):
        tokens = nltk.word_tokenize(text)
        compound_score = 0.0
        for token in tokens:
            if token in lexicon_words:
                lexicon_weight = lexicon_words[token]['weight']
                number_of_words = lexicon_words[token]['number_of_words']
                compound_score += (lexicon_weight * number_of_words)
            elif token in kata_keterangan:
                compound_score += kata_keterangan[token]
        return compound_score

    X_train_lexicon = [get_lexicon_score(text) for text in X_train]
    X_train_lexicon = np.array(X_train_lexicon).reshape(-1, 1)
    X_test_lexicon = [get_lexicon_score(text) for text in X_test]
    X_test_lexicon = np.array(X_test_lexicon).reshape(-1, 1)

    # Menggabungkan fitur TF-IDF dan fitur leksikon
    X_train_combined = hstack((X_train_tfidf, X_train_lexicon))
    X_test_combined = hstack((X_test_tfidf, X_test_lexicon))

```

Gambar 5. Kode pada tahapan analisis data

```

# Mengambil nama kolom dari fitur TF-IDF
tfidf_columns = ['TF-IDF_' + str(i) for i in range(X_train_tfidf.shape[1])]

# Mengambil nama kolom dari fitur leksikon
lexicon_columns = ['Lexicon_' + str(i) for i in range(X_train_lexicon.shape[1])]

# Membuat DataFrame untuk nilai komponen TF-IDF dan Lexicon
X_train_df = pd.DataFrame(X_train_combined.toarray(), columns=tfidf_columns +
lexicon_columns)
X_test_df = pd.DataFrame(X_test_combined.toarray(), columns=tfidf_columns +
lexicon_columns)

# Definisikan hyperparameter yang akan disetel
parameters = {
    'C': [0.1, 1, 10],
    'kernel': ['linear', 'rbf'],
    'gamma': [0.1, 0.01, 0.001]
}

# Buat objek GridSearchCV
grid_search = GridSearchCV(SVC(), parameters, cv=5)

# Lakukan penyetelan parameter menggunakan data pelatihan
grid_search.fit(X_train_combined, y_train)

# Dapatkan kombinasi parameter terbaik
best_params = grid_search.best_params_
print("Kombinasi parameter terbaik:", best_params)

# Buat model SVM dengan kombinasi parameter terbaik
svm = SVC(**best_params)

# Latih model dengan data pelatihan yang sudah digabungkan
svm.fit(X_train_combined, y_train)

# Evaluasi model menggunakan data uji yang sudah digabungkan
accuracy = svm.score(X_test_combined, y_test)
print("Akurasi:", accuracy)

# Membuat prediksi pada data uji
y_pred = svm.predict(X_test_combined)

# Menghitung metrik evaluasi
f1 = f1_score(y_test, y_pred, average='weighted')
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')

# Classification report
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d", xticklabels=svm.classes_,
yticklabels=svm.classes_)
plt.xlabel("Prediksi")

plt.ylabel("Aktual")
plt.title("Confusion Matrix")
plt.show()

return accuracy, f1, precision, recall

# Panggil fungsi analyze_sentiment untuk setiap dataset dan simpan hasil analisisnya
accuracy_glints, f1_glints, precision_glints, recall_glints = analyze_sentiment(df_glints)
accuracy_kupu, f1_kupu, precision_kupu, recall_kupu = analyze_sentiment(df_kupu)
accuracy_pintarnya, f1_pintarnya, precision_pintarnya, recall_pintarnya =
analyze_sentiment(df_pintarnya)

```

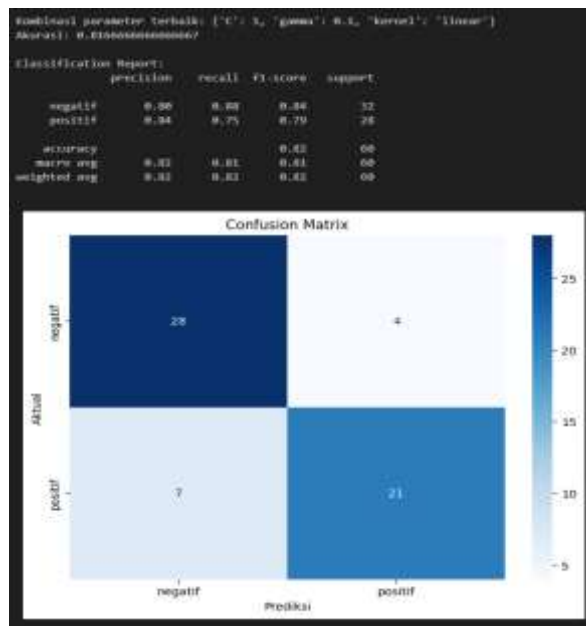
Gambar 5. Kode pada tahapan analisis data (lanjutan)

Pada kode Gambar 5, dilakukan analisis sentimen menggunakan pendekatan gabungan antara metode ekstraksi fitur TF-IDF dan fitur berbasis leksikon. Pertama, dilakukan pembacaan kamus leksikon kata dan kata keterangan dari file CSV yang telah disiapkan sebelumnya. Selanjutnya, dataset dibagi menjadi data latih dan data uji menggunakan fungsi `train_test_split` dari *library Scikit-learn*. Fitur TF-IDF diekstraksi menggunakan `TfidfVectorizer`, di mana representasi numerik dari teks dilakukan berdasarkan bobot term frekuensi-inverse dokumen (TF-IDF). Selain itu, fitur berbasis leksikon juga diekstraksi menggunakan kamus leksikon kata dan kata keterangan yang telah dibangun sebelumnya. Setelah itu, fitur-fitur TF-IDF dan leksikon digabungkan menggunakan metode *hstack*, sehingga menghasilkan representasi fitur yang lebih lengkap. Dilakukan penyetelan *hyperparameter* menggunakan metode *Grid Search* dan model *Support Vector Machine* (SVM) dengan parameter terbaik dibuat.

Model SVM dilatih menggunakan data latih yang telah digabungkan, kemudian dievaluasi menggunakan data uji yang juga telah digabungkan. Metrik evaluasi yang dihitung meliputi akurasi, F1-score, presisi, dan *recall*. Selain itu, dilakukan pembuatan *classification report* yang menampilkan hasil metrik evaluasi secara detail. Terakhir, *confusion matrix* digunakan untuk visualisasi hasil prediksi pada data uji. *Confusion matrix* ditampilkan dalam bentuk *heatmap* menggunakan *library seaborn*. Hasil analisis sentimen dari masing-masing dataset (*glints*, *kupu*, *pintarnya*) disimpan dalam variabel yang sesuai. Data tersebut kemudian dilakukan evaluasi dengan hasil:

1) *Glints*

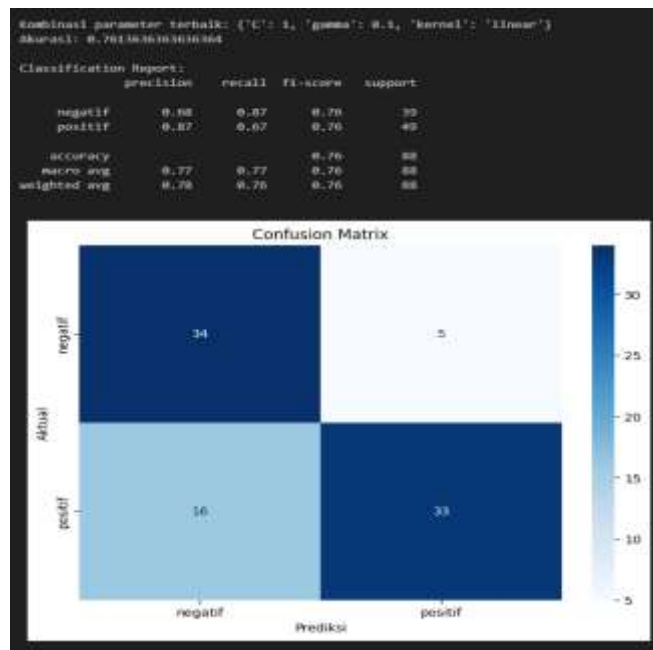
Hasil evaluasi sentimen pada aplikasi *glints* disajikan pada Gambar 5.



Gambar 5. Hasil Evaluasi Sentimen pada aplikasi *Glints*

2) *Kupu*

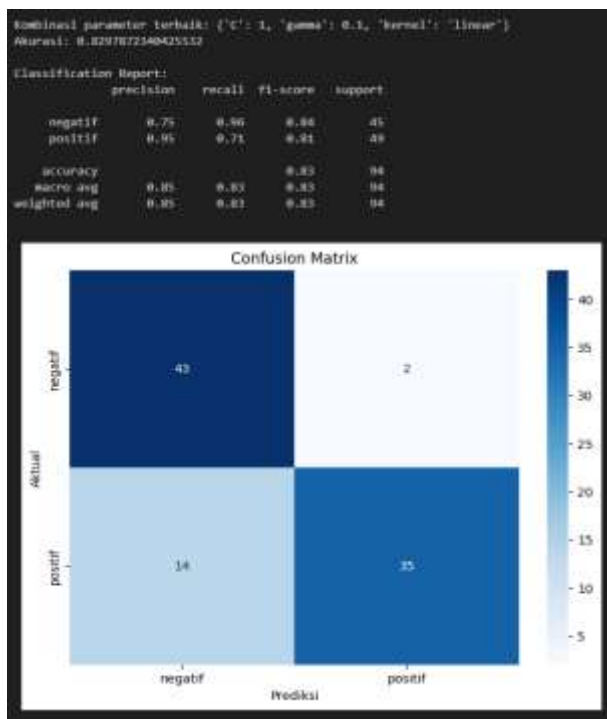
Hasil evaluasi sentimen pada aplikasi *Kupu* disajikan pada Gambar 6.



Gambar 6. Hasil Evaluasi Sentimen pada aplikasi *Kupu*

3) *Pintarnya*

Hasil evaluasi sentimen pada aplikasi *Pintarnya* disajikan pada Gambar 7.



Gambar 7. Hasil Evaluasi Sentimen pada aplikasi *Pintarnya*

Dari hasil evaluasi yang sudah dihitung, didapatkan akurasi pada masing-masing aplikasi yaitu *Glints* memiliki akurasi sebesar 81%, *Kupu* sebesar 76%, dan *Pintarnya* sebesar 82%.

4.2 Pembahasan

Hasil analisis data di atas diperoleh bahwa nilai akurasi pada masing-masing aplikasi yaitu *Glints* memiliki akurasi sebesar 81%, *Kupu* sebesar 76%, dan *Pintarnya* sebesar 82%. Dengan ini maka diketahui bahwa pendekatan menggunakan kombinasi TF-IDF, fitur berbasis leksikon, dan SVM dengan penyetelan *hyperparameter* menghasilkan performa yang baik dalam analisis sentimen pada data teks dengan akurasi pada masing-masing aplikasi yaitu *Glints* memiliki akurasi sebesar 81%, *Kupu* sebesar 76%, dan *Pintarnya* sebesar 82%. Maka dari itu perlu adanya ruang yang digunakan untuk meningkatkan performa analisis sentimen pada tiga aplikasi tersebut diantaranya: *Pertama*, pada aplikasi *glints* dan *Kupu* dapat dilakukan peninjauan ulang pada kamus leksikon kata dan kata keterangan untuk memberikan perbaikan pada pengenalan sentimen yang lebih akurat. *Kedua*, perlu adanya eksplorasi lebih mendalam terhadap teknik ekstraksi fitur tambahan dan penyetelan *hyperparameter* yang lebih cermat pada SV, sehingga dapat memberikan peningkatan lebih lanjut dalam akurasi analisis sentimen. *Ketiga*, pada aplikasi *Pintarnya* yang telah mencapai akurasi tinggi, disarankan untuk terus memantau dan mengelola sentimen pengguna secara efektif untuk menjaga kepuasan pengguna dan meningkatkan pengalaman pengguna yang lebih baik. Hasil penelitian yang menunjukkan bahwa penggunaan metode *lexicon based features* dan SVM akurat sebagaimana penelitian yang dilakukan oleh Mahendrajaya, et al pada tahun 2019 dimana penelitiannya dilakukan pada aplikasi *Gopay* dengan metode *Lexicon based features* dan SVM [15].

Dengan dilakukannya beberapa hal di atas maka dapat mendorong peningkatan layanan pada aplikasi lowongan kerja khususnya *Glints*, *Kupu* dan *Pintarnya* sehingga para pengguna dapat merasa puas dan terbantu dengan pelayanan yang diberikan. Selain itu, jika para pengguna merasa puas dan jumlah pengguna meningkat maka akan berdampak pula pada laju keuangan perusahaan aplikasi.

5. Simpulan

Setiap perusahaan tentu saja berusaha untuk memberikan pelayanan terbaik bagi usaha yang diupayakan sebagaimana yang dilakukan oleh para pelaku usaha bidang aplikasi khususnya *Glints*, *Kupu* dan *Pintarnya*. Namun, setiap perusahaan tentu selalu ingin meningkatkan kualitasnya sehingga menerima befragam ulasan yang disampaikan oleh para penggunanya. Beragam ulasan tersebut dikaji melalui *Lexicon based features* dan *Support vector machine* dengan hasil bahwa pendekatan melalui kombinasi TF-IDF, fitur berbasis leksikon, dan SVM dengan penyetelan *hyperparameter* menghasilkan performa yang baik dalam analisis sentimen pada data teks dengan akurasi 81% (*Glints*), 76% (*Kupu*), dan 82% untuk aplikasi *Pintarnya*. Dengan ini maka setiap aplikasi penitng untuk melakukan peningkatan fasilitas melalui peninjauan ulang pada kamus leksikon ataupun dengan melakukan beragam tindakan lain yang dapat meningkatkan performa layanan aplikasi yang dikembangkan.

Daftar Referensi

- [1] S. Ari Agustina, Khairul Saleh, "Hubungan Kesejahteraan Karyawan Dengan Kinerja Karyawan Pada Hotel Lusy," vol. 02, no. 02, pp. 22–28, 2018.
- [2] D. Dwi Wahyuni Ambali, L. Banne Allo, and J. Mambaya, "Hubungan Tingkat Pendapatan Dan Pekerjaan Orang Tua Dengan Gaya Hidup Mahasiswa S1 Keperawatan Semester Iv Stikes Tana Toraja Kabupaten Toraja Utara Tahun 2018," *J. Ilm. Kesehat. Promot.*, vol. 5, no. 2, pp. 165–185, 2021, doi: 10.56437/jikp.v5i2.61.
- [3] S. G. Nyoman, "Pengaruh Tingkat Pendidikan dan Jenis Pekerjaan Terhadap Produktivitas dan Pendapatan Keluarga Miskin Kecamatan Negara Jembrana," *E-Jurnal EP Unud*, vol. 9, pp. 2529–2556, 2020.
- [4] B. K. Widodo, N. H. Matondang, and D. S. Prasvita, "Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Penggunaan Aplikasi Jobstreet," *Techno.Com*, vol. 21, no. 3, pp. 523–533, 2022, doi: 10.33633/tc.v21i3.6361.
- [5] I. Ritonga, S. S., Setiawan, E. B., & Kurniawan, "Analisis Trending Topik Pada Twitter menggunakan Metode Naive Bayes dengan," *e-proceeding Eng.*, vol. 7, no. 1, pp. 2806–2816, 2020.
- [6] A. Syakur, "Implementasi Metode Lexicon Base Untuk Analisis Sentimen Kebijakan Pemerintah Dalam Pencegahan Penyebaran Virus Corona Covid-19 Pada Twitter," *J. Ilm. Inform. Komput.*, vol. 26, no. 3, pp. 247–260, 2021, doi: 10.35760/ik.2021.v26i3.4720.
- [7] A. A. Nurfalah, A., Adiwijawa. & Suryani, "Analisis Sentimen Berbahasa Indonesia dengan Pendekatan Lexicon-Based pada Media Sosial Twitter," *J. Masy. Inform. Indones.*, vol. 2, no. 1, pp. 1–8, 2017.
- [8] P. M. N. D. and N. L. P. Merawati, "Penerapan Algoritma Support Vector Machine Dalam Sentimen Analisis Terkait Kenaikan Tarif BPJS Kesehatan," *J. Bumigora Inf. Technol.*, vol. 2, no. 2, pp. 105–112, 2020, doi: 10.30812/bite.v2i2.904.
- [9] R. Andika, L. A., Azizah, P. A. N., & Respatiulwan, "Analisis Sentimen Masyarakat terhadap Hasil Quick Count Pemilihan Presiden Indonesia 2019 pada Media Sosial Twitter Menggunakan Metode Naive Bayes Classifier.," *Indones. J. Appl. Stat.*, vol. 2, no. 1, pp. 34–41, 2019, doi: <https://doi.org/10.13057/ijas.v2i1.29998>.
- [10] L. Himawan, H., Kaswidjanti, W., Sentimen, A., Sosial, M., & Based, "Metode Lexicon Based dan Support Vector Machine untuk Menganalisis Sentimen pada Media Sosial sebagai Rekomendasi Oleh-Oleh Favorit," in *Seminar Nasional Informatika*, 2018, pp. 235–244.
- [11] & I. Ruslim, K. I., Adikara, P. P., "nalisis Sentimen Pada Ulasan Aplikasi Mobile Banking Menggunakan Metode Support Vector Machine dan Lexicon Based Features," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 3, no. 7, pp. 6694–6702, 2019.
- [12] M. A. Abdullah and D. Mahdiana, "Analisis Sentimen terhadap Peluang Kerja di Indonesia selama Masa Pandemi COVID-19 dengan Metode Klasifikasi Naive Bayes," *J. Media Inform. Budidarma*, vol. 6, no. 2, pp. 1234, 2022, doi: 10.30865/mib.v6i2.3972.
- [13] I. B. T. Nova Andre Saputra, Jessica Alexandra, "Analisis Sentimen Pemanfaatan Obrolan Grup Telegram Berbagi Informasi Lowongan Kerja Menggunakan Metode Naive Bayes Classifier," *J. Mhs. Tek. Inform.*, Vol. 7, No. 2, pp. 1321–1327, 2023.
- [14] M. Handayani, Y., Hakim, A. R., "Sentiment analysis of Bank BNI user comments using the support vector machine method. Proceedings - 2020 International Seminar on Application for Technology of Information and Communication: IT Challenges for Sustainability,

Scalability, and Security in t,” in *ISemantic*, 2020, pp. 202–207. doi: <https://doi.org/10.1109/iSemantic50169.2020.9234230>.)

- [15] M. B. Mahendrajaya, R., Buntoro, G. A., & Setyawan, “Analisis Sentimen Pengguna Gopay Menggunakan Metode Lexicon Based Dan Support Vector Machine.,” *Komputek*, vol. 3, no. 2, pp. 52-57, 2019, doi: <https://doi.org/10.24269/jkt.v3i2.270>.