Naskah Publikasi-Nadya Bethry Balqies Tjikdaphia-19.01.55.0018-04072023

by Teteh Hayati

Submission date: 04-Jul-2023 09:00AM (UTC+0700)
Submission ID: 2126231767
File name: skah_Publikasi_Nadya_Bethry_Balqies_Tjikdaphia_19.01.55.0018.pdf (272.32K)
Word count: 2089
Character count: 12215

PERBANDINGAN ANALISIS SENTIMEN APLIKASI MOBILE JKN DENGAN NAIVE BAYES, SUPPORT VECTOR MACHINE, DAN K-NEAREST NEIGHBOR PADA ULASAN GOOGLE PLAY STORE

Nadya Bethry Balqies Tjikdaphia[1*], Sulastri[2*]
[1,2]Fakultas Teknologi Informasi dan Industri, Universitas Stikubank, Jl. Tri Lomba Juang, Mugassari, Kec. Semarang Sel., Kota Semarang, Jawa Tengah 50141
email: [1]nadyabethrybalqiestjikdaphia@mhs.unisbank.ac.id, [2]sulastri@edu.unisbank.ac.id.

Abstract: *The JKN Mobile application is a mobile application created to facilitate healthcare administration in Indonesia since 2017. The application has been downloaded by over 10 million users and has received 484,000 diverse reviews, including positive, negative, and neutral feedback. The average rating given by users is 4.5 out of 5 stars. This research aims to perform sentiment analysis on user reviews found in the Google Play Store review column. The methods used for sentiment analysis are Naive Bayes, K-Nearest Neighbor (K-NN), and Support Vector Machine (SVM). The test results show that with a 10% test data and 90% training data proportion, the SVM method achieves the highest accuracy of 95%. Naive Bayes follows with an accuracy of 87%, and K-NN with an accuracy of 75%.*

Keywords: *JKN Mobile application, sentiment analysis, Naive Bayes, K-Nearest Neighbor* (K-NN), *Support Vector Machine* (SVM).

Abstrak: Aplikasi Mobile JKN adalah sebuah aplikasi yang dibuat untuk mempermudah administrasi kesehatan di Indonesia sejak tahun 2017. Aplikasi ini telah diunduh lebih dari 10 juta pengguna dengan 484 ribu ulasan beragam positif, negatif, dan netral. Rata-rata rating yang diberikan pengguna adalah 4,5 bintang dari 5 bintang. Penelitian ini bertujuan untuk melakukan analisis sentimen terhadap ulasan pengguna yang terdapat di kolom *review Google Play Store*. Metode yang digunakan untuk analisis sentimen adalah Naive Bayes, K-Nearest Neighbor (K-NN), dan Support Vector Machine (SVM). Hasil pengujian menunjukkan bahwa dengan menggunakan proporsi data uji sebesar 10% dan data *training* sebesar 90%, metode SVM mencapai akurasi tertinggi sebesar 95%. Diikuti oleh Naive Bayes dengan akurasi 87%, dan K-NN dengan akurasi 75%.

Kata kunci: JKN Mobile, Analisis Sentimen, *Naïve Bayes, K-Nearest Neighbor (K-NN), Support Vector Machine (SVM)*.

PENDAHULUAN

Pada era globalisasi ini, BPJS Kesehatan telah mengembangkan aplikasi mobile bernama JKN *Mobile* untuk mempermudah pengguna dalam pendaftaran dan pengurusan kesehatan. Aplikasi ini telah diperkenalkan sejak tahun 2017[1] dan memiliki lebih dari 10 juta pengguna dengan rating rata-rata 4,5 bintang dari 5 bintang dan jumlah ulasan sebanyak 484 ribu[2]. Tujuan penelitian ini adalah untuk menganalisis sentimen ulasan aplikasi JKN *Mobile* di *Google Play Store* menggunakan metode *Naive Bayes, K-Nearest Neighbor* (K-NN), dan *Support Vector Machine* (SVM).

Data yang digunakan adalah dataset ulasan pengguna JKN *Mobile* yang diambil dari *Google Play Store*. Tahap

analisis melibatkan praproses data, seperti pembersihan teks dan ekstraksi fitur relevan. Selanjutnya, metode Naive Bayes, K-NN, dan SVM diterapkan pada *dataset* tersebut. Evaluasi dilakukan dengan menggunakan metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score*.

Hasil pengujian menunjukkan bahwa SVM memiliki akurasi tertinggi sebesar 85.71%, diikuti oleh Naive Bayes dengan akurasi 76.70%, dan K-NN dengan akurasi 52.74%. Penelitian sebelumnya juga telah menguji tiga algoritma tersebut dengan hasil yang serupa.[3]

METODE

Model penelitian

1. *Knowledge Discovery on Database* (KDD) merupakan metode yang dinilai dapat digunakan dalam penelitian ini, merujuk pada proses ekstraksi pengetahuan yang bermanfaat dari data dalam *database*. Metode ini terdiri dari 4 tahap utama yaitu pengumpulan data, *preprocessing data*, penambangan data, dan evaluasi [4].

1.1. Pengumpulan Data

Pengumpulan data dilakukan dengan menyaring ulasan pengguna aplikasi JKN *Mobile* dari *Google Play Store* menggunakan *Google Colab*. Ulasan tersebut telah diberi label sentimen positif, negatif, dan netral berdasarkan isi ulasan.

Jumlah ulasan negatif dengan rating 1-2 adalah 858, ulasan netral dengan rating 3 berjumlah 80, dan ulasan positif dengan rating 4-5 berjumlah 262. Total keseluruhan data yang dikumpulkan adalah 1200 data.

```
[ ] df_busu['score'].value_counts()

    1    749
    5    211
    2    109
    3     80
    4     51
    Name: score, dtype: int64
```

Gambar 1. Hasil Pengumpulan Data

1.2. *Preprocessing Data*

Preprocessing data adalah langkah penting dalam persiapan data sebelum melakukan analisis atau pemrosesan lebih lanjut. Tujuannya adalah untuk membersihkan, mengatur, dan mengubah data mentah menjadi bentuk yang lebih terstruktur, mudah dipahami, dan sesuai dengan kebutuhan analisis atau pemodelan yang akan dilakukan.

Beberapa metode yang umum digunakan dalam preprocessing data yaitu, case folding, tokenizing, filtering, stopword, stemming dan pembobotan term menggunakan TF-IDF [5].

1.2.1. *Case Folding*

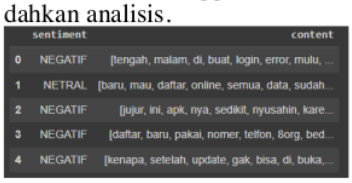
Case Folding adalah Mengubah semua karakter teks menjadi huruf kecil atau huruf besar, untuk menghindari perbedaan penulisan yang tidak relevan.

	sentiment	content
0	NEGATIF	tengah malam di buat login error mulu padahal ...
1	NETRAL	baru mau daftar online semua data sudah saya i...
2	NEGATIF	jujur ini apk nya sedikit nyusahin karena lagi...
3	NEGATIF	daftar baru pakai nomer telfon 8org beda kok g...
4	NEGATIF	kenapa setelah update gak bisa di buka klo se...

Gambar 2. Hasil *Case Floding*

1.2.2. *Tokenizing*

Tokenizing Memisahkan teks menjadi unit-unit yang lebih kecil, seperti kata-kata atau token, sehingga memudahkan analisis.



	sentiment	content
0	NEGATIF	[tengah, malam, di, buat, login, error, mulu, ...
1	NETRAL	[baru, mau, daftar, online, semua, data, sudah...
2	NEGATIF	[jujur, ini, apk, nya, sedikit, nyusahin, kare...
3	NEGATIF	[daftar, baru, pakai, nomer, telfon, 8org, bed...
4	NEGATIF	[kenapa, setelah, update, gak, bisa, di, buka,...

Gambar 3. Hasil *Tokenizing*

1.2.3. *Filtering*

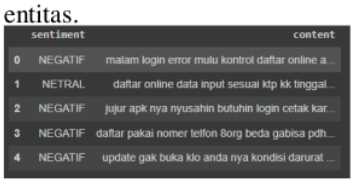
Filtering Menghilangkan kata-kata yang tidak relevan atau tidak berguna dalam analisis, seperti kata-kata umum (stopwords).

	sentiment	content
0	NEGATIF	[malam, login, error, mulu, kontrol, daftar, o...
1	NETRAL	[daftar, online, data, input, sesuai, ktp, kk,...
2	NEGATIF	[jujur, apk, nya, nyusahin, butuhin, login, ce...
3	NEGATIF	[daftar, pakai, nomer, telfon, 8org, beda, gab...
4	NEGATIF	[update, gak, buka, , klo, seandai, nya, kondi...

Gambar 4. Hasil *Filtering*

1.2.4. *Stemming*

Stemming mengubah kata-kata menjadi bentuk dasarnya dengan menghapus imbuhan atau akhiran, sehingga kata-kata dengan akar yang sama dapat dianggap sebagai satu entitas.



	sentiment	content
0	NEGATIF	malam login error mulu kontrol daftar online a...
1	NETRAL	daftar online data input sesuai ktp kk tinggal ...
2	NEGATIF	jujur apk nya nyusahin butuhin login cetak kar...
3	NEGATIF	daftar pakai nomer telfon 8org beda gabisa pdh...
4	NEGATIF	update gak buka klo anda nya kondisi darurat ...

Gambar Hasil 5. *Stemming*

1.2.5. *Term Frequency-Inverse Document Frequency* (TF-IDF)

Term frequency (TF) adalah jumlah kata/term dalam suatu dokumen, sedangkan *Inverse Document Frequency* (IDF) adalah frekuensi kemunculan kata/term diseluruh dokumen[4]. TF–IDF bermanfaat dalam meningkatkan performa model.

$$TF - IDF = \frac{n_{yx}}{\sum t_y} . log \frac{\sum d}{n_{dx}}$$

keterangan:

x = kata ke-x y = dokumen ke-y n = jumlah

t = *term*/kata d = dokumen

Gambar 6. Rumus TF-IDF

1.2.6. SMOTE (*Synthetic Minority Over-sampling Technique*) *Handling Imbalance*

SMOTE adalah teknik oversampling yang digunakan dalam machine learning untuk mengatasi masalah kelas yang tidak seimbang pada data.

Teknik SMOTE bekerja dengan menciptakan data sintetis baru dari kelas minoritas melalui interpolasi antara titik-titik data yang ada. Data sintetis yang dihasilkan oleh SMOTE berada dalam ruang fitur, bukan dalam ruang data.

Metode ini menambahkan contoh dari kelas minoritas dengan cara mengekstrak sampel data minoritas yang ada dan menggunakan sampel acak dari k tetangga terdekat.[6]

1.3. Naïve Bayes

Naive Bayes salah satu algoritma klasifikasi yang berdasarkan teorema Bayes dengan asumsi independensi antara fitur-fitur yang ada. Algoritma ini sangat populer karena sederhana, efisien dalam penggunaan sumber daya, dan mampu memberikan hasil yang baik dalam banyak aplikasi.[7]

1.4. Support Vactor Machine

Support Vector Machine (SVM) adalah algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi dan regresi. SVM bekerja dengan membangun sebuah model yang dapat memisahkan dua kelas dengan mencari hyperplane terbaik yang memiliki margin maksimal antara kelas-kelas tersebut [5].

Support Vector Machine juga dapat mengatasi masalah data yang tidak linier dengan menggunakan kernel untuk mentransformasi data ke dalam ruang fitur yang lebih tinggi dimensi.[8]

1.5. K-Nearst Neighbor

K-Nearest Neighbor (K-NN) algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi dan regresi. KNN bekerja dengan menghitung jarak antara data yang akan diklasifikasikan dengan data latihan yang ada.

Kemudian, KNN akan menentukan klasifikasi data baru berdasarkan mayoritas klasifikasi tetangga terdekat (dengan nilai k tertentu) dari data tersebut. KNN sangat sederhana dan mudah dipahami, tetapi juga bisa menghadapi tantangan dalam menangani data dengan dimensi tinggi. [9]

1.6. Evaluasi Permodelan

Pada penelitian ini menggunakan perhitungan *performance matrix* yaitu *Confusion Matrix, confusing matrix* sendiri merupakan tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi atau algoritma prediksi. Tabel ini membandingkan nilai prediksi dari model dengan nilai sebenarnya dari data yang diamati.

Confusion matrix terdiri dari empat komponen utama: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) [4]. Setelah dilakukannya perhitungan dari *Confusion Matrix*, maka untuk mengukur *Performance Metrics* pada permodelan algoritma dapat menggunakan rumus *accuracy, precission, recall* dan *F1-score* [10].

$$Accuracy = \frac{TP+TN}{(TP+FP+FN+TN)} \; x \; 100\%$$

$$Precision = \frac{TP}{TP+FP} \; x \; 100\%$$

$$Recall = \frac{TP}{TP+FN} \; x \; 100\%$$

$$F - score = \frac{2 \times precision \; x \; recall}{precision + recall}$$

Gambar.7 Rumus Perhitungan Evaluasi Kerja.

HASIL DAN PEMBAHASAN

2. Pembagian data (*Splitting data*)

Data hasil *preprocessing* kemudian displit data yang terdiri dari data uji dan data latih. Untuk mengetahui hasil yang optimal maka dilakukan pembagian ada 5 yaitu percobaan 1 data uji 10% data latih 90%, percobaan 2 data uji 20% data latih 80 %, percobaan 3 data uji 30% data latih 70%, percobaan 4 data uji 40% data latih 60%, dan yang terakhir percobaan 5 data latih 50% data

uji 50%.

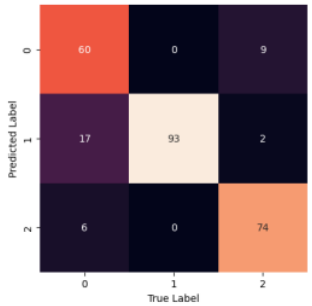
2.1. Klasifikasi Naïve Bayes

Pada setiap percoban pembagian menghasilkan *accuracy* yang berbeda-beda seperti pada tabel.

Tabel 1. Hasil Percobaan Klasifikasi *Naïve Bayes*

Percobaan	precision	recall	f1-score	accuracy
1	88%	87%	86%	87%
2	86%	85%	85%	86%
3	86%	85%	85%	86%
4	86%	85%	85%	85%
5	83%	82%	81%	81%

Dari data tabel yang bisa menjadi acuan hasil paling optimal yaitu pada percobaan pertama dengan hasil *accuracy* 87%, *Precision* 88%, *Recall* 87%, dan *F1-Score* 86%. Penerapan klasifikasi paling akurat menghasilkan *Confusion Matrix* sebagai berikut.



Gambar 8. *Confusion Matrix Naïve Bayes*

```
MultinomialNB Accuracy: 0.8697318007662835
MultinomialNB Precision: 0.8749741200828157
MultinomialNB Recall: 0.8644932671863926
MultinomialNB f1_score: 0.864586818116985
confusion matrix:
 [[60 17  6]
 [ 0 93  0]
 [ 9  2 74]]
================================================

              precision    recall  f1-score   support

     NEGATIF       0.87      0.72      0.79        83
      NETRAL       0.83      1.00      0.91        93
     POSITIF       0.93      0.87      0.90        85

    accuracy                           0.87       261
   macro avg       0.87      0.86      0.86       261
weighted avg       0.87      0.87      0.87       261
```

Gambar 9. Hasil klasifikasi Naïve Bayes

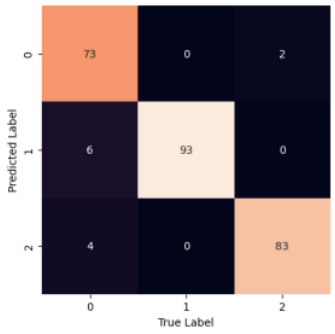
2.2. Klasifikasi *Support Vector Machine* (SVM)

Pada setiap percoban pembagian menghasilkan *accuracy* yang berbeda-beda seperti pada tabel.

Tabel 2. Hasil Percobaan Klasifikasi *Support Vector Machine*

Percobaan	precision	recall	f1-score	accuracy
1	96%	95%	95%	95%
2	94%	94%	94%	94%
3	93%	93%	93%	93%
4	93%	93%	93%	93%
5	92%	92%	92%	92%

Dari data tersebut yang bisa menjadi acuan hasil paling optimal yaitu pada percobaan pertama dengan hasil *accuracy* 95%, *Precision* 96%, *Recall* 95%, dan *F1-Score* 95%. Penerapan klasifikasi paling akurat menghasilkan *Confusion Matrix* sebagai berikut.



Gambar 10. *Confusion Matrix* SVM

```
Support Vector Machine Accuracy: 0.9540229885057471
Support Vector Machine Precision: 0.9555834204110066
Support Vector Machine Recall: 0.9519962201748169
Support Vector Machine F1 score: 0.9526389706603866
Confusion matrix:
[[73  6  4]
 [ 0 93  0]
 [ 2  0 83]]
--------------------------------------------------

              precision    recall  f1-score   support

     NEGATIF       0.97      0.88      0.92        83
      NETRAL       0.94      1.00      0.97        93
     POSITIF       0.95      0.98      0.97        85

    accuracy                           0.95       261
   macro avg       0.96      0.95      0.95       261
weighted avg       0.95      0.95      0.95       261
```

Gambar 11. Hasil Klasifikasi SVM

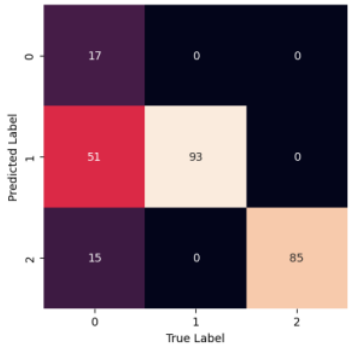
2.3. Klasifikasi K-Nearest Neighbor (K-NN)

Pada setiap percoban pembagian menghasilkan *accuracy* yang berbeda-beda seperti pada tabel.

Tabel 3. Hasil Percobaan Klasifikasi *K-Nearest Neighbor* (K-NN)

Percobaan	precision	recall	f1-score	accuracy
1	83%	73%	68%	75%
2	82%	75%	67%	73%
3	81%	72%	66%	72%
4	81%	71%	66%	72%
5	80%	70%	64%	70%

Dari data tersebut yang bisa menjadi acuan hasil paling optimal yaitu pada percobaan pertama dengan hasil *accuracy* 75%, *Precision* 83%, *Recall* 73%, dan *F1-Score* 68%. Penerapan klasifikasi paling akurat menghasilkan *Confusion Matrix* sebagai berikut.



Gambar 12. *Confusion Matrix* KNN

```
KNeighbors Accuracy: 0.7471264367816092
KNeighbors Precision: 0.8319444444444445
KNeighbors Recall: 0.7349397590361445
KNeighbors f1_score: 0.6812430151670658
confusion matrix:
 [[17 51 15]
 [ 0 93  0]
 [ 0  0 85]]
--------------------------------------------------

              precision    recall  f1-score   support

     NEGATIF       1.00      0.20      0.34        83
      NETRAL       0.65      1.00      0.78        93
     POSITIF       0.85      1.00      0.92        85

    accuracy                           0.75       261
   macro avg       0.83      0.73      0.68       261
weighted avg       0.82      0.75      0.69       261
```

Gambar 13. Hasil Klasifikasi KNN.

2.4. Hasil Klasifikasi keseluruhan

Dari kelima percobaan yang dilakukan dari 3 algoritma didapatkan hasil yang terbaik dipercobaan data uji 10% data latih 90%.

Klasifikasi keseleruhan nilai *precision, recall, f1 score* serta akurasi.

Tabel 4. Perbandingan Hasil Klasifikasi

Perbandingan antar hasil klasifikasi			
Hasil dengan akurasi tertinggi			
	NB	SVM	KNN
Precision	88%	96%	83%
Recall	87%	95%	73%
F1-score	86%	95%	68%
Accuracy	87%	95%	68%

Dari tabel diatas dapat disimpulkan bahwa nilai klasifikasi SVM lebih tinggi dari Naïve bayes dan KNN. Dengan hasil *Accuracy* 95%, *Precision* 96%, *Recall* 95%, dan *F1-Score* 95%. Sedangkan NB *accuracy* 87%, *Precision* 88%, *Recall* 87%, dan *F1-Score* 86% dan KNN *accuracy* 75%, *Precision*

83%, *Recall* 73%, dan *F1-Score* 68%.

SIMPULAN

Berdasarkan hasil pengujian Analisa sentiment pada aplikasi JKN Mobile dapat disimpulkan bahwa dari 1200 data set dan dengan berbagai proporsi data latih dan data uji dapat disimpukan bahwa dengan percobaan 1 data uji 10% dan data latih 90% memiliki hasil lebih optimal dengan akurasi *Naïve Bayes* 87%, *Support Vector Machine* 95% dan *K-Nearest Neighbor* 75%.

Dalam konteks analisis sentimen aplikasi JKN Mobile, disarankan untuk menggunakan algoritma *Support Vector Machine* (SVM) karena memiliki akurasi yang lebih tinggi dibandingkan dengan *Naïve Bayes* dan *K-Nearest Neighbor*. Penggunaan *Support Vector Machine* dapat membantu dalam memprediksi sentimen dengan lebih akurat.

DAFTAR PUSTAKA

[1] Humas, "BPJS Kesehatan Ikuti Perkembangan Zaman, Mobile JKN Satu Genggaman Untuk Berbagai Kemudahan," 2020. https://www.bpjs-kesehatan.go.id/bpjs/post/read/2020/1671/Ikuti-Perkembangan-Zaman-Mobile-JKN-Satu-Genggaman-Untuk-Berbagai-Kemudahan (accessed Jul. 02, 2023).

[2] B. Kesehatan, "Mobile JKN," *Google Play Store*, 2023. https://play.google.com/store/apps/details?id=app.bpjs.mobile (accessed Mar. 24, 2023).

[3] S. Lestari, M. Mupaat, and A. Erfina, "Analisis Sentimen Masyarakat Indonesia terhadap Pemindahan Ibu Kota Negara Indonesia pada Twitter," *JUSIFO (Jurnal Sist. Informasi)*, vol. 8, no. 1, pp. 13–22, 2022, doi: 10.19109/jusifo.v8i1.12116.

[4] D. S. Putri, A. Sentimen, U. Aplikasi, and T. Ridwan, "Analisis Sentimen Ulasan Aplikasi Pospay dengan Algoritma Support Vector Machine," no. 2018, 2023.

[5] Z. Zaenal and I. R. I. Astutik, "Sentiment Analysis of OYO App Reviews Using the Support Vector Machine Algorithm," *Procedia Eng. Life Sci.*, vol. 3, no. December, 2023, doi: 10.21070/pels.v3i0.1338.

[6] A. Nugroho and E. Rilvani, "Penerapan Metode Oversampling SMOTE Pada Algoritma Random Forest Untuk Prediksi Kebangkrutan Perusahaan," *Techno.Com*, vol. 22, no. 1, pp. 207–214, 2023, doi: 10.33633/tc.v22i1.7527.

[7] J. P. Tanjung, F. C. Tampubolon, A. W. Panggabean, and M. A. A. Nandrawan, "Customer Classification Using Naive Bayes Classifier With Genetic Algorithm Feature Selection," *Sinkron*, vol. 8, no. 1, pp. 584–589, Feb. 2023, doi: 10.33395/sinkron.v8i1.12182.

[8] N. H. Ovirianti, M. Zarlis, and H. Mawengkang, "Support Vector Machine Using A Classification Algorithm," *SinkrOn*, vol. 7, no. 3, pp. 2103–2107, 2022, doi: 10.33395/sinkron.v7i3.11597.

[9] D. Cheng, S. Zhang, Z. Deng, Y. Zhu, and M. Zong, "κ NN algorithm with data-driven k value," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8933, pp. 499–512, 2014, doi: 10.1007/978-

3-319-14717-8_39.
[10] R. Puspitasari, Y. Findawati, M. A. Rosid, P. S. Informatika, and U. M. Sidoarjo, “SENTIMENT ANALYSIS OF POST-COVID-19 INFLATION BASED ON TWITTER USING THE K-NEAREST NEIGHBOR AND SUPPORT VECTOR MACHINE ANALISIS SENTIMEN TERHADAP INFLASI PASCA COVID-19 BERDASARKAN TWITTER DENGAN METODE KLASIFIKASI K-NEAREST NEIGHBOR DAN,” vol. 4, no. 4, pp. 1–11, 2023, doi: 10.20884/jutif.

Naskah Publikasi-Nadya Bethry Balqies Tjikdaphia-19.01.55.0018-04072023

ORIGINALITY REPORT

3%	3%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.jurnal.uts.ac.id Internet Source	2%
2	www.researchgate.net Internet Source	2%

Exclude quotes	On	Exclude matches	< 2%
Exclude bibliography	On		