JURNAL TEKNOINFO
Volume 17, Nomor 2, Juli 2023, Page 548-556
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

ANALISIS PERFORMANSI *ENTITY MATCHING* DENGAN *FUZZY WUZZY* PADA ARTIKEL *FAIRNESS* AI

Indira Salsabila Ardan[1)], Miftakhul Janah Sulastri[2)] , Nur Aini Rakhmawati[3)]

[1,2,3]Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember
[1,2,3]Jl. Teknik Kimia, Keputih, Sukolilo, Surabaya
Email: [1]indi.salsa@gmail.com, [2]janah1448@gmail.com, [3]nur.aini@is.its.ac.id

Abstract

In today's digital era, finding information on a particular topic is increasingly easy to do through search engines such as Google Scholar or scientific article databases. Google Scholar allows users to find scholarly articles or journals from various fields of science and serves as a personal library that allows users to save selected journals. However, due to the large number of scientific articles, it is often difficult to determine which articles are most relevant to a particular topic and have a high level of accuracy. One of the techniques used to select scientific articles that are relevant to a particular topic is by using entity matching. This research aims to analyze the performance of the entity matching technique using Fuzzy Wuzzy with the addition of blocking stop word removal and size blocking on articles with the theme Fairness in AI. The entity matching technique is performed by comparing titles with titles, authors with authors, and keywords with keywords. Weighting is applied to titles, authors, and keywords and there are four weighting variations used. Blocking is also applied to increase the speed and efficiency of the entity matching technique. The analysis shows that the weights used in the entity matching technique play an important role in achieving optimal performance. The weight of 0.5 for title, 0.1 for authors, and 0.4 for keywords produces the best performance with accuracy of 71.26%, recall of 48.34%, precision of 92.74%, and f-1 score of 63.56%. In addition, the application of size blocking can significantly speed up the data comparison process, with a running time of 2.56 seconds without sacrificing performance.

Keyword: *entity matching, fuzzy wuzzy, blocking, stop word removal, size blocking.*

Abstrak

Dalam era digital saat ini, mencari informasi mengenai topik tertentu semakin mudah dilakukan melalui mesin pencari seperti Google Scholar atau *database* artikel ilmiah. Google Scholar memungkinkan pengguna untuk menemukan artikel atau jurnal ilmiah dari berbagai bidang ilmu serta berfungsi sebagai perpustakaan pribadi yang memungkinkan pengguna untuk menyimpan jurnal-jurnal terpilih. Namun, karena jumlah artikel ilmiah yang begitu banyak, sering kali sulit untuk menentukan artikel mana yang paling relevan dengan topik tertentu dan memiliki tingkat keakuratan yang tinggi. Salah satu teknik yang digunakan untuk memilih artikel ilmiah yang relevan dengan topik tertentu adalah dengan menggunakan *entity matching*. Penelitian ini bertujuan untuk melakukan analisis performansi dari teknik entity matching menggunakan Fuzzy Wuzzy dengan penambahan *blocking stop word removal* dan *size blocking* pada artikel bertema Fairness in AI. Teknik *entity matching* dilakukan dengan membandingkan *title* dengan *title*, *authors* dengan *authors*, dan *keywords* dengan *keywords*. Pembobotan diterapkan pada *title*, *authors*, dan *keywords* serta terdapat empat variasi pembobotan yang digunakan. *Blocking* juga diterapkan untuk meningkatkan kecepatan dan efisiensi teknik *entity matching*. Hasil analisis menunjukkan bahwa bobot yang digunakan dalam teknik *entity matching* memiliki peran penting dalam mencapai performa yang optimal. Bobot 0.5 untuk *title*, 0.1 untuk *authors*, dan 0.4 untuk *keywords* menghasilkan performa terbaik dengan *accuracy* sebesar 71.26%, *recall* sebesar 48.34%, *precision* sebesar 92.74%, dan *f-1 score* sebesar 63.56%. Selain itu, penerapan *size blocking* dapat mempercepat proses perbandingan data secara signifikan, dengan *running time* sebesar 2.56 detik tanpa mengorbankan performa.

Kata Kunci: *entity matching, fuzzy wuzzy, blocking, stop word removal, size blocking.*

1. PENDAHULUAN

Saat ini, di era digital, mencari informasi mengenai topik tertentu semakin mudah dilakukan melalui mesin pencari seperti Google Scholar atau *database* artikel ilmiah. Google Scholar sendiri merupakan mesin pencari web yang menyimpan teks atau metadata dari literatur ilmiah dalam berbagai disiplin ilmu [1]. Mesin pencari ini memungkinkan

JURNAL TEKNOINFO
Volume 17, Nomor 2, Juli 2023, Page 548-556
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

pengguna untuk menemukan artikel atau jurnal ilmiah dari berbagai bidang ilmu [2], serta berfungsi sebagai perpustakaan pribadi yang memungkinkan pengguna untuk menyimpan jurnal-jurnal terpilih dan mengaksesnya melalui menu My Library yang tersedia di sidebar Google Scholar [1]. Dalam berbagai format publikasi ilmiah yang disajikan oleh Google Scholar, para peneliti dan mahasiswa dapat mengejar karir akademik dan mengembangkan pengetahuan. Namun, karena jumlah artikel ilmiah yang begitu banyak, seringkali sulit untuk menentukan artikel mana yang paling relevan dengan topik tertentu dan memiliki tingkat keakuratan yang tinggi [3].

Salah satu teknik yang digunakan untuk memilih artikel ilmiah yang relevan dengan topik tertentu adalah dengan menggunakan *entity matching*. *Entity matching* adalah teknik yang digunakan untuk memilih artikel ilmiah yang relevan dengan topik tertentu. Teknik ini melibatkan perhitungan kesamaan atau kemiripan antara entitas yang terdapat pada artikel ilmiah dengan topik yang sedang dibahas. Entitas yang dimaksud adalah kata-kata atau frasa yang merepresentasikan konsep atau topik tertentu [4]. Dalam pemilihan artikel ilmiah, penulis dapat mengkaji secara langsung dari sumber-sumber dan topik yang relevan dengan permasalahannya, baik yang sejalan ataupun tidak. Namun, penulis harus cermat dan teliti dalam memilih rujukan yang sesuai dengan isu yang diangkat [5].

Proses perhitungan nilai *similarity* dilakukan dengan menggunalan *library fuzzy wuzzy. FuzzyWuzzy* merupakan salah satu *library Python* yang digunakan untuk melakukan pencocokan string dengan pendekatan *fuzzy*. Konsep pencocokan *string fuzzy* adalah proses mencari *string* yang memiliki kesamaan dengan pola yang ditentukan, meskipun tidak sepenuhnya identik. *Library* ini menggunakan Jarak *Levenshtein* untuk menghitung perbedaan antara urutan karakter pada string. *Library FuzzyWuzzy* menyediakan beberapa metode untuk membandingkan *string*, antara lain *FuzzyWuzzy Ratio*, *FuzzyWuzzy PartialRatio*, *FuzzyWuzzy TokenSortRatio*, *FuzzyWuzzy TokenSetRatio*, dan *FuzzyWuzzy WRatio*. Masing-masing metode memiliki kelebihan dan kekurangan dalam hal penggunaannya, tergantung pada kebutuhan pengguna. Dalam penggunaannya, *library FuzzyWuzzy* akan mengembalikan skor yang berkisar dari 0 hingga 100. Skor tersebut menunjukkan seberapa mirip kedua *string* yang dibandingkan, dimana semakin tinggi skor menunjukkan semakin mirip kedua string tersebut [6].

Untuk meningkatkan efektivitas dan efisiensi dari teknik entity matching, dapat digunakan teknik blocking untuk mengurangi jumlah pasangan artikel yang harus dibandingkan. Teknik blocking yang akan digunakan adalah stopwords removal dan size blocking. Stopword removal adalah kata-kata yang sangat umum digunakan sehingga tidak memberikan informasi yang berguna dalam analisis data atau pemrosesan bahasa alami seperti "dan", "atau", "yang", dan sebagainya. Stopwords removal dapat dihapus dari teks untuk mengurangi ukuran dokumen dan meningkatkan efisiensi pemrosesan teks sehingga dokumen menjadi lebih fokus pada kata-kata yang lebih penting dan informatif [7]. Size blocking adalah teknik blocking yang dilakukan dengan membatasi jumlah kata atau karakter dalam sebuah dokumen. Teknik ini dapat digunakan untuk membatasi jumlah kata dalam sebuah kalimat atau paragraf, sehingga dapat membantu dalam proses peringkasan teks [8].

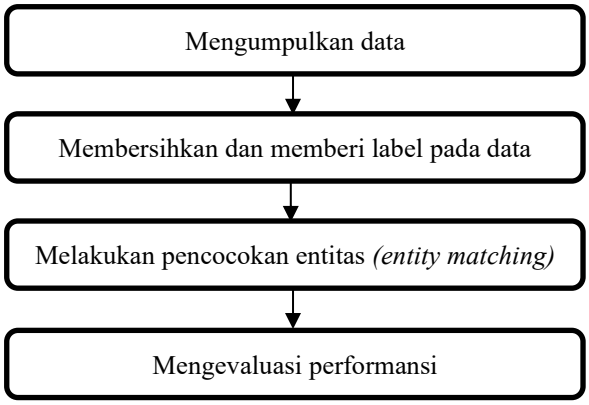
Beberapa penelitian telah dilakukan dalam pencarian artikel yang relevan. Penelitian Simple Query Suggestion untuk Pencarian Artikel Menggunakan Jaccard Similarity[9] berdasar pada variasi kata kunci. Analisis Similarity/Kemiripan Artikel Jurnal Online Terbitan Tahun 2019-2020 di ISI Yogyakarta[10] berdasar pada jurnal *online* terbitan tahun 2019-2020. Klasifikasi Multilabel Menggunakan Metode Fuzzy Similarity K-Nearest Neighbor Untuk Rekomendasi Pencarian Artikel Online [11] berdasar pada jumlah artikel *online*. Implementasi Metode Semantic Similarity untuk Pengukuran Kemiripan Makna antar Kalimat [12].

Dengan melakukan penelitian ini, diharapkan dapat ditemukan cara terbaik untuk memilih artikel ilmiah yang relevan dengan topik tertentu dan meningkatkan efektivitas serta efisiensi teknik entity matching pada artikel bertema *Fairness in AI*. Artikel ini juga dapat memberikan manfaat bagi para peneliti dan praktisi dalam pengembangan teknologi AI yang adil dan tidak diskriminatif.

2. METODE PENELITIAN

Penelitian ini terdiri dari empat tahapan yang dilakukan secara berurutan, yaitu mengumpulkan data, membersihkan dan memberi label pada data, melakukan pencocokan entitas (*entity matching*), dan evaluasi performansi. Tahapan-tahapan penelitian ini terlihat pada **Error! Reference source not found.** dan akan dijelaskan lebih detail pada bab ini.

JURNAL TEKNOINFO
Volume 17, Nomor 2, Juli 2023, Page 548-556
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index



Gambar 1. Metode Penelitian

2.1 Pengumpulan Data

Penelitian ini menggunakan aplikasi *Publish or Perish 8* dalam mengumpulkan data. Aplikasi ini berfungsi untuk mengumpulkan metadata bibliografi karya ilmiah secara gratis [13]. Data yang dikumpulkan terdiri dari lima puluh artikel mengenai *fairness* di bidang *Artificial Intelligence* (AI), yang diambil dari *Google Scholar* dalam rentang waktu 2018 hingga 2022 dengan kata kunci "*fairness in AI*". Hasil pengumpulan data disimpan dalam format *file Comma Separated Value* (CSV), dan setiap artikel mencakup informasi seperti *title*, *authors*, *year*, *keywords*, dan lain-lain.

2.2 Pembersihan dan Pemberian Label pada Data

Dataset dalam format CSV dibersihkan dengan menghapus kolom-kolom yang tidak diperlukan sehingga hanya tersisa kolom *title*, *authors*, dan *keywords*. Kemudian, dilakukan *pre-processing* pada keseluruhan *dataset* dengan mengubah semua menjadi huruf kecil agar data seragam. Selanjutnya, menggunakan *fuzzy wuzzy*, dilakukan perhitungan kemiripan antara *tile* dengan *title*, *authors* dengan *authors*, dan *keywords* dengan *keywords*. *Weight* (bobot) *title*, *authors*, dan *keywords* masing-masing diberikan nilai 0, 0.5, dan 0.5. *Weight title* diberikan nilai 0 karena penilaian kemiripan judul dilakukan secara manual. Setiap baris pada *dataset* kemudian diberi label yang menunjukkan tingkat kemiripan antara judul artikel yang bersangkutan. Artikel yang tidak mirip diberi label 0, sedangkan artikel yang mirip diberi label 1. Label-label tersebut kemudian digunakan sebagai acuan untuk menghitung *accuracy, precision, recall,* dan *f1-score* pada model *entity matching*. *Dataset* ini terdiri dari 1225 baris dan 8 kolom secara keseluruhan. Contoh *dataset* dapat dilihat pada Tabel 1.

Tabel 1. Contoh *Dataset*

Title1	Title2	Authors1	Authors2	Keywords1	Keywords2	Similarity	Label
bias and fairness in machine learning and artificial intelligence	exploring bias and fairness in artificial intelligence and machine learning algorithms	cirillo d,rementeria mj	khakurel ub,abdelmoumin g,bajracharya a,rawat db	artificial intelligence,bias,fairness,model development	artificial intelligence,detection and tracking algorithms,machine learning,performance modeling,systems modeling	0.463685065	1
how self-perceived reputation affects fairness towards humans and artificial	artificial intelligence and the public sectorâ€"applications and challenges	russo pa,duradoni m,guazzini a	wirtz bw,weyerer jc,geyer c	artificial intelligence,fairness,reputation, ultimatum game	ai applications,ai challenges,artificial intelligence,public sector	0.30615543	1

JURNAL TEKNOINFO
Volume 17, Nomor 2, Juli 2023, Page 548-556
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

intelligence							
beyond fairness metrics: roadblocks and challenges for ethical ai in practice	artificial intelligence, intersectionality, and the future of public health	chen j,storchan v,kurshane	bauer gr,lizotte dj	algorithmic fairness,discrimination,explainability,machine learning,supervised learning	ai development,ai ethics,accountability,artificial intelligence,research framework,responsibility,transparency	0.24234127	0
artificial intelligence in human resources management: challenges and a path forward	flexible and adaptive fairness-aware learning in non-stationary data streams	tambep,cappelli p,yakubovich v	zhang w,zhang m,zhang j,liu z	data analysis,decision-making tools,hiring and recruitment,human capital,human resource ethics,information systems	ai fairness,flexible fairness,online classification	0.252163743	0
rawls's original position and algorithmic fairness	artificial intelligence in human resources management: challenges and a path forward	franke u	tambep,cappelli p,yakubovich v	algorithmic fairness,original position,veil of ignorance,veil of uncertainty	data analysis,decision-making tools,hiring and recruitment,human capital,human resource ethics,information systems	0.220470914	0

2.3 *Entity Matching*

Penelitian ini menggunakan metode *Fuzzy Wuzzy* sebagai teknik pengukuran kesamaan antara dua *string* dalam proses *entity matching* [14]. *Fuzzy Wuzzy* adalah metode yang menggunakan teknik *fuzzy string matching* untuk mencocokkan entitas pada dua sumber data yang berbeda. Proses *entity matching* dilakukan dengan membandingkan *title* dengan *title*, *authors* dengan *authors*, dan *keywords* dengan *keywords*. Metode *Fuzzy Wuzzy Token Set Ratio* dipilih karena dapat mengatasi perbedaan urutan *token* pada suatu *string*.

Pembobotan juga diterapkan dalam proses *entity matching* untuk memberikan prioritas pada aspek yang lebih penting. Pembobotan digunakan pada *title*, *authors*, dan *keywords* dan terdapat empat variasi pembobotan yang digunakan secara berurutan, yaitu: (1) 0.33, 0.33, 0.33, (2) 0.5, 0.1, 0.4, (3) 0.4, 0.5, 0.1, dan (4) 0.1, 0.4, 0.5. Selain itu, *blocking* juga diterapkan untuk meningkatkan kecepatan dan efisiensi proses *entity matching*. Hasil *entity matching* dihitung dengan menggunakan nilai *similarity* antara entitas yang dibandingkan. Semakin tinggi nilai *similarity*, semakin mirip kedua *string* yang dibandingkan. Nilai *similarity* akhir dihitung dengan menggunakan Persamaan 1.

$$Similarity = (Title\ Similarity \times Title\ Weight) + (Authors\ Similarity \times Authors\ Weight) + (Keywords\ Similarity \times Keywords\ Weight) \quad (1)$$

2.4 Evaluasi Performansi

Penelitian ini melakukan evaluasi performansi dengan menggunakan beberapa metrik, yaitu *accuracy, precision, recall,* dan *f1-score*. *Accuracy* mengukur seberapa akurat sistem dalam mengenali entitas yang sama antara kedua sumber data, sedangkan *precision* mengukur seberapa banyak entitas yang diidentifikasi oleh sistem yang benar-benar cocok dengan

JURNAL TEKNOINFO
Volume 17, Nomor 2, Juli 2023, Page 548-556
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

entitas di sumber data lain. *Recall*, di sisi lain, mengukur seberapa banyak entitas yang diidentifikasi oleh sistem yang benar-benar ada di sumber data lain. *F1-score* merupakan gabungan antara *precision* dan *recall* yang memberikan gambaran holistik tentang performansi sistem. Rumus untuk menghitung masing-masing metrik dapat ditemukan pada Persamaan 2, Persamaan 3, Persamaan 4, dan Persamaan 5.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$F1\ Score = 2 \times \frac{Recall \times Precision}{Recall+Precision} \quad (5)$$

Evaluasi performansi yang dilakukan dalam penelitian ini digunakan untuk membandingkan performa dari *entity matching* dengan mempertimbangkan berbagai bobot dan perbandingan antara penggunaan *blocking* dan tanpa *blocking*. Evaluasi ini bertujuan untuk menemukan bobot *entity matching* terbaik yang dapat memberikan hasil yang paling optimal dalam menentukan tingkat kemiripan antar artikel yang berkaitan.

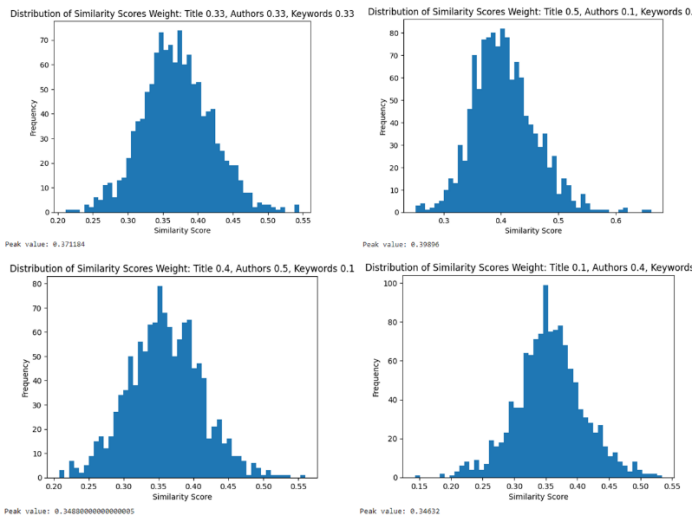
3. HASIL DAN PEMBAHASAN

3.1 Penerapan *Weight*

Penerapan bobot dilakukan dengan empat variasi guna menemukan bobot yang paling optimal, yaitu sebagai berikut:

a. *Title* 0.33, *authors* 0.33, dan *keywords* 0.33: Pemilihan *weight* ini didasarkan pada asumsi bahwa setiap elemen (*title, authors, keywords*) memiliki kontribusi yang sama penting dalam menentukan kesamaan antara dua dokumen.
b. *Title* 0.5, *authors* 0.1, dan *keywords* 0.4: Pemilihan *weight* ini didasarkan pada pertimbangan bahwa informasi mengenai *title* dan *keywords* lebih penting daripada informasi mengenai penulis dalam menentukan kesamaan antara dokumen.
c. *Title* 0.4, *authors* 0.5, dan *keywords* 0.1: Pemilihan *weight* ini didasarkan pada pertimbangan bahwa informasi mengenai *authors* lebih penting daripada informasi mengenai *keywords* dalam menentukan kesamaan antara dokumen.
d. *Title* 0.1, *authors* 0.4, dan *keywords* 0.5: Pemilihan *weight* ini didasarkan pada pertimbangan bahwa informasi mengenai *keywords* lebih penting daripada informasi mengenai *title* dalam menentukan kesamaan antara dokumen, dan informasi mengenai *authors* tetap cukup penting.

Dari keempat varian tersebut, akan dihasilkan nilai *similarity* yang kemudian diberi label untuk mengukur performanya. Label diberikan dengan menggunakan nilai *threshold* yang diperoleh dari perhitungan *peak value* yang ditunjukkan dalam Gambar 2.



Gambar 2. Hasil *Peak Value*

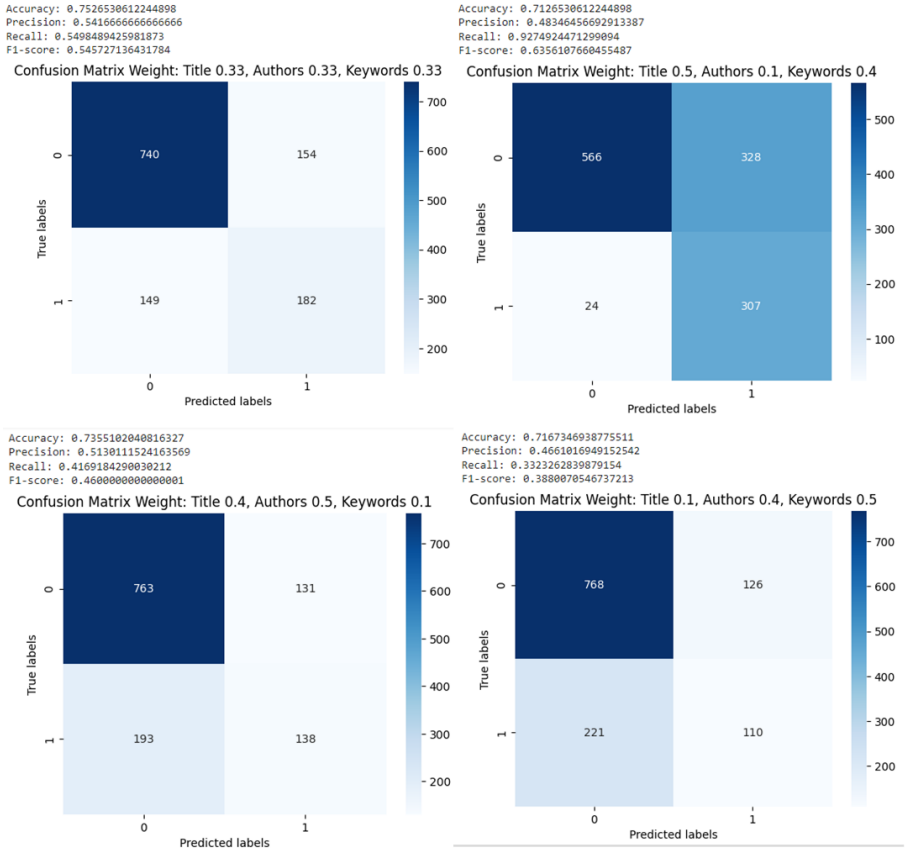
JURNAL TEKNOINFO
Volume 17, Nomor 2, Juli 2023, Page 548-556
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

Sehingga diperoleh *peak value* tertinggi yaitu 0.39896 yang akan digunakan sebagai nilai *threshold*. Selanjutnya nilai *similarity* yang kurang dari *peak value* akan dilabeli sebagai 0 karena semakin kecil nilai *similarity* maka semakin kecil pula kemungkinan artikel tersebut memiliki makna yang sama, contoh hasil pelabelan ditunjukkan pada Tabel 2.

Tabel 2. Contoh Hasil Pelabelan dengan Nilai *Threshold*

Index	Title1	Title2	Authors1	Authors2	Keywords1	Keywords2	Similarity	Label
0	transparency,and,explainability,of,ai,systems:,ethical,guidelines,in,practice	artificial,intelligence,intersectionality,,and,the,future,of,public,health	balasubramaniam n,kauppinen m,hiekkanen k,kujala s	bauer gr,lizotte dj	ai systems,ethical guidelines,explainability,quality requirements,transparency	ajph,apha,american,association,health,journal,public,ethics,infrastructure,policy,practice	0.404	1
1	transparency,and,explainability,of,ai,systems:,ethical,guidelines,in,practice	fairness,&,friends,in,the,data,science,era	balasubramaniam n,kauppinen m,hiekkanen k,kujala s	catania b,guerrini g,accinelli c	ai systems,ethical guidelines,explainability,quality requirements,transparency	data-informed automated decision system,diversity,fairness,nondiscrimination,processing pipeline	0.399	1

Sehingga dari keempat variasi bobot tersebut didapatkan *confussion matrix* seperti yang ditunjukkan pada Gambar 3.



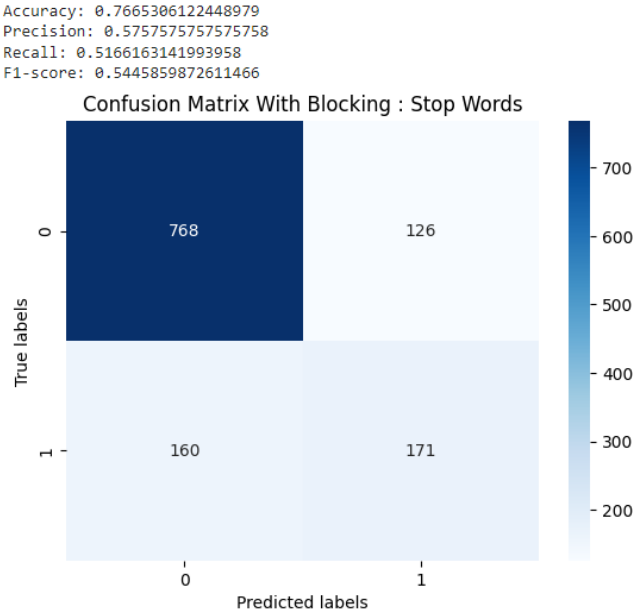
Gambar 3. *Confusion Matrix* dari Empat Variasi *Weight* yang Digunakan

JURNAL TEKNOINFO
Volume 17, Nomor 2, Juli 2023, Page 548-556
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

Pembobotan *title* 0.5, *authors* 0.1, dan *keywords* 0.4 memberikan hasil terbaik di antara variasi lain dengan *accuracy* sebesar 71.26%, *recall* sebesar 48.34%, *precision* sebesar 92.74%, dan *f-1 score* sebesar 63.56%. Hasil tersebut menunjukkan bahwa informasi mengenai *title* dan *keywords* pada artikel memiliki pengaruh yang lebih besar dalam menentukan kesamaan antar artikel, dibandingkan dengan informasi mengenai *authors*.

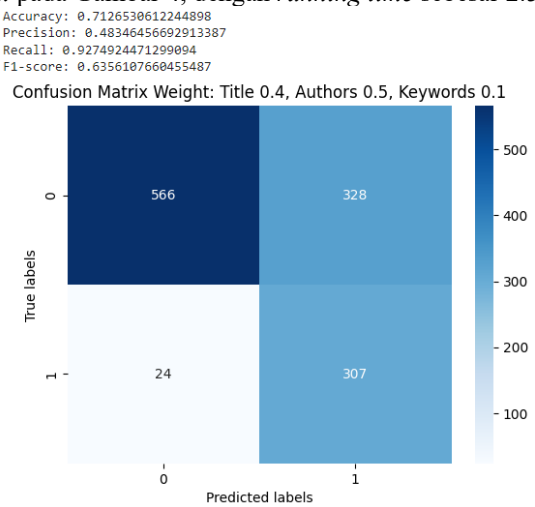
3.2 Penerapan *Blocking*

Setelah memperoleh nilai *weight* terbaik, langkah selanjutnya adalah menerapkan *blocking* untuk meningkatkan performa. Selain *accuracy*, *precision, recall, dan f1-score, running time* juga menjadi parameter yang penting. Terdapat dua jenis *blocking* yang digunakan yaitu *stop word removal* dan *size blocking*. *Stop word removal* diterapkan dengan menghapus kata-kata umum atau biasa yang sering muncul dalam suatu teks, seperti *"a", "an", "the", "and", "in", "of",* dan sebagainya. Hasil penerapan *blocking stop word removal* ditunjukkan dalam *confusion matrix* pada Gambar 4, dengan *running time* sebesar 2.79 detik.



Gambar 4. *Confusion Matrix* dari Penerapan *Blocking Stop Word Removal*

Dalam menerapkan *size blocking*, dilakukan pemilihan ukuran blok yang sesuai dengan mencoba beberapa nilai dan mengukur waktu eksekusi serta penggunaan memori untuk memastikan bahwa nilai yang dipilih dapat mempercepat waktu eksekusi tanpa menghabiskan terlalu banyak memori. Ukuran blok yang akhirnya dipilih adalah 100 dengan mempertimbangkan keseimbangan antara waktu komputasi dan performa yang dihasilkan. Hasil penerapan *size blocking* ditunjukkan dalam *confusion matrix* pada Gambar 4, dengan *running time* sebesar 2.56 detik.



Gambar 5. *Confusion Matrix* dari Penerapan *Size Blocking*

JURNAL TEKNOINFO
Volume 17, Nomor 2, Juli 2023, Page 548-556
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

3.3 Hasil Implementasi

Berdasarkan pengujian yang dilakukan terhadap empat variasi *weight* sebelumnya, didapatkan bahwa *weight* terbaik yang menghasilkan performa tertinggi adalah *title* sebesar 0.5, *authors* sebesar 0.1, dan *keywords* sebesar 0.4. Kemudian, performa dan *running time*-nya dibandingkan dalam tiga skenario, yaitu tanpa blocking, dengan *blocking stop word removal*, dan dengan *size blocking*. Perbandingan rinci performa dan *running time* dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan Performa dan *Running Time*

Skenario	Performa yang Dihasilkan (%)				*Running Time* (detik)
	Accuracy	*Precision*	*Recall*	*F1-Score*	
Weight terbaik tanpa *blocking*	71.26	48.34	92.74	63.56	2.78
Weight terbaik dengan *blocking stop word removal*	76.65	57.57	51.66	54.45	2.79
Weight terbaik dengan *size blocking*	71.26	48.34	92.74	63.56	2.56

Dalam penelitian ini, *f1-score* dianggap lebih penting daripada *accuracy*, *precision,* dan *recall*. Alasannya adalah karena *f1-score* memberikan pengukuran yang lebih lengkap dan akurat dibandingkan dengan *accuracy*, *precision,* dan *recall*. Hasil penelitian menunjukkan bahwa pada ketiga skenario yang diuji, nilai tertinggi *f1-score* dicapai ketika tidak melakukan blocking dan melakukan size blocking. Kedua skenario tersebut menunjukkan performa yang sama dengan nilai *accuracy* sebesar 71.26%, *precision* sebesar 48.34%, *recall* sebesar 92.74%, dan *f1-score* sebesar 63.56%.

Dilihat dari *running time,* skenario yang menggunakan *size blocking* memiliki *running time* yang lebih cepat dibandingkan dengan skenario lainnya. Sementara itu, skenario yang menggunakan *blocking stop word removal* memiliki *running time* yang hampir sama dengan skenario tanpa *blocking,* dengan selisih hanya 0.01. Hal tersebut dikarenakan pada penerapan *blocking stop word removal* diperlukan waktu untuk mencocokkan setiap kata dengan kamus *stop word* atau menerapkan aturan pemotongan akhiran dari kata tertentu, sedangkan pada penerapan *size blocking,* blok ukuran 100 digunakan untuk membatasi jumlah pasangan data yang dibandingkan pada setiap iterasi, sehingga mempercepat proses perbandingan data secara signifikan.

4. KESIMPULAN

Berdasarkan hasil analisis, dapat disimpulkan bahwa bobot yang digunakan dalam proses *entity matching* memiliki peran penting dalam mencapai performa yang optimal. Hasil menunjukkan bahwa bobot 0.5 untuk *title*, 0.1 untuk *authors*, dan 0.4 untuk *keywords* menghasilkan performa terbaik dengan *accuracy* sebesar 71.26%, *recall* sebesar 48.34%, *precision* sebesar 92.74%, dan *f-1 score* sebesar 63.56%. Hal ini menunjukkan bahwa informasi mengenai *title* dan *keywords* lebih signifikan dalam menentukan kesamaan antar artikel, dibandingkan dengan informasi mengenai *authors*. Selain itu, penerapan *size blocking* dapat mempercepat proses perbandingan data secara signifikan, dengan *running time* sebesar 2.56 detik tanpa mengorbankan performa. Penelitian selanjutnya dapat mengembangkan teknik-teknik baru untuk meningkatkan performa *entity matching* lebih lanjut.

DAFTAR PUSTAKA

[1] A. Setiani Rafika, H. Yunan Putri, F. Diah Widiarti, D. STMIK Raharja Tangerang, M. STMIK Raharja Tangerang, and J. Jendral Sudirman No, "ANALISIS MESIN PENCARIAN GOOGLE SCHOLAR SEBAGAI SUMBER BARU UNTUK KUTIPAN," vol. 3, no. 2, 2017.

[2] "GOOGLE SEBAGAI SUMBER INFORMASI UNTUK MENULIS DI ERA DISRUPSI COVID-19 | Dinas Perpustakaan dan Kearsipan Provinsi Banten." https://dpk.bantenprov.go.id/Layanan/topic/265 (accessed Apr. 02, 2023).

[3] W. Khairiyah, "Nautical : Jurnal Ilmiah Multidisiplin Pemanfaatan google scholar dalam pemenuhan kebutuhan informasi

penelitian mahasiswa prodi perpustakaan dan ilmu informasi Universitas Negeri Padang," *J. Ilm. Multidisiplin Indones.*, vol. 1, no. 9, pp. 1058–1071, 2022.

[4] N. Barlaug and J. A. Gulla, "Neural Networks for Entity Matching: A Survey," *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 3, 2021, doi: 10.1145/3442200.

[5] "KARYA TULIS ILMIAH DALAM PENGEMBANGAN SUMBERDAYA MANUSIA DI ORGANISASI PEMERINTAH." https://ppsdmaparatur.esdm.go.id/artikel/karya-tulis-ilmiah-dalam-pengembangan-sumberdaya-manusia-di-organisasi-pemerintah (accessed Apr. 02, 2023).

[6] "FuzzyWuzzy Python library - GeeksforGeeks." https://www.geeksforgeeks.org/fuzzywuzzy-python-library/ (accessed Apr. 02, 2023).

[7] B. Alhadidi and M. Wedyan, "Hybrid Stop-Word Removal Technique for Arabic Language.," *Egypt. Comput. Sci. J.*, vol. 30, pp. 35–38, Jan. 2008.

[8] A. Jelita, "Effective Techniques for Indonesian Text Retrieval," *Ph.D Thesis*, pp. 1–286, 2007, [Online]. Available: https://researchbank.rmit.edu.au/view/rmit:6312

[9] K. Rinartha, "Simple Query Suggestion Untuk Pencarian Artikel Menggunakan Jaccard Similarity," *J. Ilm. Rekayasa dan Manaj. Sist. Inf.*, vol. 3, no. 1, pp. 30–34, 2017.

[10] A. Agustiawan, "Analisis Similarity/Kemiripan Artikel Jurnal Online Terbitan Tahun 2019-2020 Di ISI Yogyakarta," *ABDI PUSTAKA J. Perpust. dan Kearsipan*, vol. 2, no. 1, pp. 29–43, 2022, doi: 10.24821/jap.v2i1.6984.

[11] W. Lubis, Y. A. Sari, and M. A. Fauzi, "Klasifikasi Multilabel Menggunakan Metode Fuzzy Similarity K-Nearest Neighbor Untuk Rekomendasi Pencarian Artikel Online," vol. 3, no. 1, pp. 931–939, 2019, [Online]. Available: http://j-ptiik.ub.ac.id

[12] G. U. Abriani and M. A. Yaqin, "Implementasi Metode Semantic Similarity untuk Pengukuran Kemiripan Makna antar Kalimat," *Ilk. J. Comput. Sci. Appl. Informatics*, vol. 1, no. 2, pp. 47–57, 2019, doi: 10.28926/ilkomnika.v1i2.15.

[13] N. F. Azkia, "Meta-Analisis Pengaruh Media Pembelajaran Berbasis Digital terhadap Hasil Belajar Matematika," Feb. 2023, Accessed: Apr. 03, 2023. [Online]. Available: https://repository.uinjkt.ac.id/dspace/handle/123456789/67090

[14] A. D. I. RAHAYU, "Implementasi Fuzzy Mcdm Topsis Pada Functional Design Untuk Mendukung Proses Pengembangan Produk Ventela Public Low," Sep. 2021, Accessed: Apr. 03, 2023. [Online]. Available: https://dspace.uii.ac.id/handle/123456789/36264