Implementasi Content Based Filtering dalam Sistem Rekomendasi Jurnal Scopus Berbasis Web Untuk Menunjang Pelaksanaan Penelitian dan Tugas Mahasiswa Program Studi Bisnis Digital

Dion Sandy Ara Tambunan[1] , Dodik Arwin Dermawan[2]

Manajemen Informatika, Fakultas Vokasi, Universitas Negeri Surabaya
Jl. Ketintang, Ketintang, Kec. Gayungan, Surabaya, Jawa Timur 60231
[1]dion.20056@mhs.unesa.ac.id
[2]dodikdermawan@unesa.ac.id

Abstrak*— *Jurnal merupakan sumber informasi penting bagi mahasiswa, namun proses pencariannya seringkali sulit dan memakan waktu lama. Penelitian ini bertujuan untuk mengoptimalkan penelitian dan tugas akhir mahasiswa Program Studi Bisnis Digital Universitas Negeri Surabaya melalui sistem rekomendasi jurnal terindex Scopus berbasis web yang mengimplementasikan metode content-based filtering. Metode content-based filtering membandingkan konten jurnal yang telah dianalisis dengan jurnal yang dicari oleh pengguna, dengan mempertimbangkan topik bisnis digital dan kata kunci yang dimasukkan. Dataset jurnal dari berbagai topik Bisnis Digital digunakan untuk melatih model, dengan data jurnal diberi pembobotan TF-IDF dan dicari nilai cosine similaritynya. Hasil penelitian menunjukkan bahwa sistem rekomendasi jurnal dengan metode content-based filtering dapat memberikan rekomendasi jurnal yang relevan kepada pengguna. Selain itu, tingkat akurasi precision dari sistem ini sebesar 85%. Secara keseluruhan, sistem rekomendasi jurnal ini mendukung pelaksanaan tugas dan penelitian mahasiswa Program Studi Bisnis Digital Universitas Negeri Surabaya.

***Kata kunci*— content based filtering, sistem rekomendasi, sistem rekomendasi jurnal, jurnal bisnis digital, cosine similarity**

Abstract*— *Journals are a crucial information source for students; however, the search process is often tedious and time-consuming. This study aims to optimize the research and final projects of Digital Business Study Program students at Universitas Negeri Surabaya by implementing a content-based filtering recommendation system for Scopus-indexed journals within a web-based platform. The content-based filtering method compares analyzed journal content with user-searched journals, considering digital business topics and entered keywords. A dataset of journals from various digital business topics is utilized to train the model, with journal data weighted using TF-IDF and cosine similarity calculated. The research findings demonstrate that the content-based filtering recommendation system effectively provides relevant journal recommendations to users. Additionally, the system exhibits a precision accuracy rate of 85%. Overall, this journal recommendation system supports the execution of assignments and research projects for Digital Business Study Program students at Universitas Negeri Surabaya.

***Keywords*— content-based filtering, recommendation system, journal recommendation system, digital bussines journal, cosine similarity**

I. PENDAHULUAN

Jurnal merupakan salah satu sumber informasi yang penting bagi mahasiswa, baik untuk keperluan penelitian, tugas akhir, maupun referensi. Jurnal berisi berbagai informasi ilmiah yang dapat digunakan untuk mendukung penelitian, tugas akhir, maupun penulisan karya ilmiah lainnya. Sering sekali mahasiswa diwajibkan untuk menggunakan jurnal *international* yang terindex scopus. Saat ini, banyak media yang menyajikan jurnal-jurnal variatif yang bisa digunakan oleh mahasiswa sebagai referensi atau penunjang penelitan maupun tugas mereka. Namun, mencari jurnal yang terindex scopus yang sesuai dengan kebutuhan mahasiswa seringkali menjadi hal cukup sulit dan memakan waktu. Hal ini dikarenakan banyaknya jurnal yang tersedia di internet, sehingga mahasiswa harus mencarinya satu persatu secara manual, terlebih mencari jurnal yang terindex Scopus. Ketika terdapat banyak sekali informasi relevan dan berpotensi berharga, hal ini mungkin akan menjadi penghalang, bukan bantuan [1]. Proeses pencarian jurnal yang terindex Scopus secara manual dapat memakan waktu dan tidak efektif. Selain itu, mahasiswa juga harus memiliki pengetahuan yang cukup tentang jurnal untuk dapat memilih jurnal yang sesuai dengan kebutuhannya.

Dengan banyaknya media yang memberikan informasi jurnal, sistem rekomendasi menjadi sangat cocok dimanfaatkan untuk membantu mahasiswa menemukan jurnal yang relevan sesuai dengan kebutuhan mereka. Untuk mengatasi permasalahan tersebut, sistem rekomendasi (jurnal) dapat menjadi solusi yang efektif. Saat ini, banyak sistem rekomendasi yang digunakan untuk memecahkan masalah kelebihan informasi di berbagai bidang seperti e-commerce, hiburan, dan media sosial [2]. Sistem rekomendasi terbagi menjadi 6 kategori,

yaitu *Collaborative Filtering, Content Based Filtering, Demographic Based Recommendation System (RS), Utility Based RS, Knowledge Based RS, Hybrid Based RS* [3]. Collaborative Filtering berfungsi untuk menemukan kemiripan di antara pengguna dan menyarankan item yang relevan. Sistem rekomendasi ini memberikan rekomendasi item yang disukai oleh pengguna lain yang memiliki kesamaan. Collaborative filtering dapat dikategorikan menjadi dua bentuk utama. Bentuk pertama adalah Collaborative Filtering *memory based*, yang beroperasi dengan mengevaluasi kesamaan antara pengguna atau antar item. Selain itu ada juga, *model based* yang memprediksi peringkat pengguna untuk objek yang belum dinilai. Content Based Filtering adalah sebuah teknik rekomendasi yang memberikan rekomendasi berdasarkan preferensi pengguna dan hubungan antara deskripsi item. Hal ini dilakukan dengan cara memilih item yang memiliki kesamaan paling tinggi dengan item yang diinginkan oleh pengguna. Sedangkan Hybrid Based adalah gabungan dari berbagai jenis kategori atau model sistem rekomendasi.

Berdasarkan penjelasan sebelumnya, penulis merasa terpanggil untuk menciptakan sistem rekomendasi yang dapat menyediakan referensi jurnal bagi pengguna dalam menambah pilihan referensi yang tepat. Data untuk membangun sistem rekomendasi ini diperoleh dari situs web Scopus. Melimpahnya jurnal dan informasi pada situs Scopus, membuat mahasiswa atau pengguna membutuhkan waktu lebih lama untuk mendapatkan jurnal yang mereka inginkan. Setiap jurnal yang ada di web tersebut mempunyai hubungan antara satu jurnal dengan yang lainnya, baik dari topik, abstrak, penulis, kata kunci, dan judul dari jurnal tersebut. Informasi ini dapat digunakan untuk mengembangkan sistem rekomendasi yang memudahkan pengguna dalam menemukan jurnal yang dibutuhkan. Sistem rekomendasi merupakan solusi efektif untuk mengatasi kesulitan dalam mencari artikel yang sesuai dengan preferensi mahasiswa. Selain itu, sistem ini juga memperluas referensi pengetahuan terhadap artikel-artikel serupa. Cosine similarity (dalam content based filtering) digunakan karena akurasi sudut kosinus. [4]. Implementasi metode content-based filtering menggunakan informasi yang terdapat dalam setiap jurnal untuk mengukur sejauh mana kesamaannya dengan data jurnal lainnya. Dengan mengukur kemiripan antara jurnal-jurnal tersebut, sistem memberikan skor sebagai acuan atau parameter dalam memberikan rekomendasi berdasarkan informasi yang terdapat dalam jurnal tersebut. Diharapkan sistem rekomendasi ini dapat membantu mahasiswa Program Studi Bisnis Digital menambah referensi dan mendukung tugas serta penelitian mereka.

II. DASAR TEORI

A. Sistem ekomendasi

Sistem rekomendasi digunakan untuk memberikan saran item kepada pengguna berdasarkan keputusan yang mereka inginkan. Sistem rekomendasi adalah alat yang efisien untuk menyaring informasi online, yang tersebar luas karena perubahan kebiasaan pengguna komputer, tren personalisasi, dan akses yang muncul ke internet [5]. Secara keseluruhan, sistem rekomendasi merupakan alat yang efektif untuk mengkurasi informasi.

B. Content Based Filtering

Content Based Filtering (CBF) menyajikan rekomendasi item kepada pengguna berdasarkan korelasi antara deskripsi item. Proses ini melibatkan pembentukan hubungan antara item dan atributnya dalam bentuk matriks. Selanjutnya, penting untuk mengidentifikasi item yang menunjukkan tingkat kemiripan tertinggi dengan item target. Proses ini melibatkan penghitungan metrik kesamaan berdasarkan fitur yang terkait dengan item yang dibandingkan, menggunakan berbagai fungsi matematika. Metode yang umum digunakan untuk menilai kesamaan meliputi Adjusted Cosine, Cosine Similarity, atau Pearson Coefficient. Pendekatan content based filtering, sebaliknya, hanya mempertimbangkan preferensi masa lalu dari pengguna individu dan mencoba mempelajari model preferensi berdasarkan representasi berbasis fitur dari konten item yang direkomendasikan [6].

C. Web Scrapping

Web Scraping merupakan metode penting untuk menarik data yang tidak terorganisir dari web dan mengkonversinya menjadi data yang terorganisir. Teknik ini juga dikenal sebagai ekstraksi data web, scraping data web, pemanenan web, atau pengambilan layar. Web scraping adalah suatu bentuk dari penambangan data. Tujuan utama web scraping ialah untuk mengekstrak informasi dari berbagai situs web yang tidak terstruktur dan mengubahnya menjadi format yang mudah dipahami misalnya spreadsheet, database, atau file CSV. Web scraping adalah metode yang digunakan untuk menarik data dari web secara terstruktur dan mengubahnya menjadi sekumpulan data yang terorganisir. Web scraping memfasilitasi pengumpulan data dalam volume yang besar dalam waktu yang relative singkat dan secara otomatis, yang mengurangi kemungkinan kesalahan [7].

D. Term Frequency (TF)

Metode TF-IDF digunakan untuk menghitung nilai bobot kata (term) dalam sebuah dokumen [8]. TF IDF digunakan untuk menemukan seberapa relevan kata tersebut dalam dokumen. Relevansi kata adalah jumlah informasi yang memberi tentang konteks. Term frequency mengukur seberapa sering istilah muncul dalam dokumen, dan istilah tersebut memiliki relevansi lebih dari istilah lain untuk dokumen tersebut [9]. TF-IDF menggabungkan dua konsep, yaitu TF

(Term Frequency) dan IDF (Inverse Document Frequency). TF berfungsi dalam mengukur seberapa sering suatu istilah muncul dalam sebuah dokumen. Sebagai contoh, dalam sebuah dokumen "A" yang berisi 1000 kata, kata "informasi" muncul sebanyak 10 kali. dalam dokumen tersebut. Berikut adalah persamaan untuk menghitung TF:

$$tf(t, d) = f(t, d) / N(d) \quad (2.1)$$

Keterangan:

$tf(t, d)$= Nilai TF kata t dalam dokumen d.
$f(t, d)$ = Jumlah kemunculan kata t dalam dokumen d.
$N(d)$ = Jumlah total kata dalam dokumen d.

Panjang total suatu dokumen bisa berkisar dari sangat pendek hingga panjang, sehingga kemungkinan istilah apa pun muncul lebih sering dalam dokumen yang lebih besar daripada dokumen yang lebih kecil. Untuk mengatasi variabilitas ini, penting untuk menghitung frekuensi istilah (TF) dengan mempertimbangkan berapa kali suatu istilah muncul dalam dokumen dibandingkan dengan jumlah total istilah dalam dokumen tersebut. Jadi, frekuensi istilah untuk kata "informasi" dalam dokumen "A" dapat dihitung sebagai TF = 10/1000 = 0,01. Frekuensi kemunculan sebuah kata dalam dokumen tertentu menandakan pentingnya kata tersebut dalam konteks dokumen tersebut [8].

E. Inverse Document Frequency (IDF)

Proses Inverse Document Frequency (IDF) digunakan untuk mengukur seberapa pentingnya kata dalam sebuah dokumen saat menghitung Term Frequency (TF) dari dokumen tersebut. Algoritma memperlakukan semua kata kunci secara setara, termasuk kata-kata stopwords seperti 'dari'. Namun, setiap kata kunci memiliki tingkat penting yang berbeda. Sebagai contoh, jika kata stopwords 'dari' muncul 2000 kali dalam sebuah dokumen tanpa memberikan makna yang signifikan, IDF memberikan bobot yang lebih tinggi pada kata-kata yang jarang muncul dan bobot yang lebih rendah pada kata-kata yang sering muncul. Inverse Document Frequency (IDF) menggambarkan distribusi suatu istilah di seluruh koleksi dokumen. Nilai IDF akan semakin tinggi ketika nilai TF semakin rendah [10]. Sebagai ilustrasi, dalam sebuah koleksi 10 dokumen di mana istilah 'teknologi' muncul dalam 5 di antaranya, nilai IDF dapat dihitung sebagai log(10/5) = 0.3010.

Proses IDF bertolak belakang dengan TF, yaitu semakin sering kata atau term muncul, semakin rendah bobotnya [11]. Berikut persamaan dari IDF:

$$IDF(t) = \log\left(\frac{N}{N(t)} + 1\right) \quad (2.2)$$

Keterangan :

IDF(t)= IDF dari kata t.
N = Total jumlah teks atau dokumen dalam kumpulan data Anda.
N(t) = Jumlah teks atau dokumen yang mengandung kata t.
log = Logaritma natural (basis e). Jika basis tidak ditentukan, biasanya diasumsikan sebagai logaritma natural.
+1 = Penambahan 1 dilakukan untuk mencegah nilai IDF menjadi tak terdefinisi ketika N(t)=0

Bobot TF-IDF akan memiliki nilai yang tinggi jika nilai TF besar dan kata yang sedang diperhatikan tidak muncul dalam sejumlah besar dokumen [12]. Oleh karena itu, untuk menghitung nilai TF-IDF, menggunakan persamaan berikut :

$$TF\text{-}IDF(t, d) = TF(t,d) \times IDFt \quad (2.3)$$

$$TF\text{-}IDF(t, d) = TF(t,d) \times \ln\left(\frac{N}{N(t)}\right) \quad (2.4)$$

Keterangan :

TF-IDF(t, d) = Bobot TF-IDF kata t dalam dokumen d.
TF(t, d) = Frekuensi kemunculan kata t dalam dokumen d.
IDF(t) = Inverse Document Frequency dari kata t.
N = Total jumlah dokumen dalam kumpulan data.
N(t) = Jumlah dokumen yang mengandung kata t.
log = Logaritma natural (basis e).
+1 = Penambahan 1 untuk mencegah nilai IDF tak terdefinisi.

Berdasarkan persamaan 2.3 maupun 2.4, jika nilai $D = dfj$, maka berapapun nilai dari $TFij$ akan menjadi 0. Untuk menghindari nilai Wij menjadi 0, maka ditambahkan nilai 1 [11]. Persamaan dari TF-IDF akan menjadi seperti berikut:

$$TF\text{-}IDF(t, d) = TF(t,d) \times \left(\ln\left(\frac{N}{N(t)}\right) + 1\right) \quad (2.5)$$

F. Cosine Similarity

Cosine similarity adalah algoritma yang digunakan untuk mengukur sejauh mana dua dokumen memiliki kesamaan satu sama lain. Secara inti, algoritma ini menghitung kesamaan dengan mempertimbangkan dokumen sebagai vektor dalam ruang vektor dan mengukur kesamaannya. [13]. Dalam konteks lain, algoritma cosine similarity antara dua vektor (atau dua dokumen dalam ruang vektor) diukur dengan menghitung sudut kosinus antara keduanya. Untuk menghitung nilai cosine similarity, dapat menggunakan persamaan (2.6).

$$Sim(q, d_{j)} = \frac{q \times d_j}{|q| \times |d_j|} = \frac{\sum_{i=1}^{n} W_{i,q} \times W_{i,j}}{\sqrt{\sum_{i=1}^{t}(W_{i,q})^2}\sqrt{\sum_{i-1}^{t}(W_{i,j})^2}} \quad (2.6)$$

Dalam persamaan tersebut, $Sim(q, d_j)$ adalah ukuran kemiripan antara query dan dokumen. $|q|$ adalah panjang query. $|d_j|$ adalah panjang dokumen. $W_{i,j}$ adalah bobot dari dokumen ke-i. $W_{i,q}$ adalah bobot dari query dokumen ke-i. Kemiripan antara query dan dokumen berbanding lurus dengan hasil perkalian jumlah bobot query (q) dengan bobot dokumen (d_j) dan berbanding terbalik dengan hasil perkalian dari akar jumlah kuadrat q ($|q|$) dengan akar jumlah kuadrat dokumen $|d_j|$. Hasil cosine similarity akan menghasilkan nilai bobot dokumen yang mendekati nilai 1.

III. METODE PENELTIAN

A. *Pendekatan Penelitian*

1. Analisa Masalah

   Penelitian ini mengkaji tentang sistem rekomendasi jurnal yang menggunakan metode content-based filtering untuk menentukan kemiripan antar item jurnal melalui penerapan algoritma TF-IDF dan Cosine Similarity. Dalam proses ini, diharapkan dapat memberikan referensi jurnal yang relevan dengan kebutuhan mahasiswa. Di samping itu, ada tantangan lain yaitu bagaimana sistem mampu menampilkan rekomendasi jurnal dengan lebih baik dan mengurangi kesalahan dalam hasil rekomendasi..

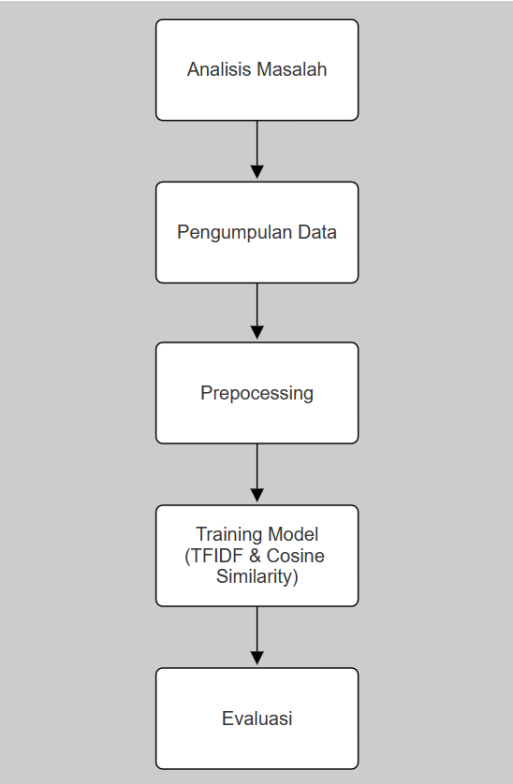
2. Studi Pustaka dan Literatur

   Untuk mencapai tujuan penelitian ini, penulis memerlukan sejumlah referensi dan informasi yang berkaitan dengan sistem rekomendasi dan algoritma yang relevan. Penulis melakukan pengumpulan data melalui kombinasi tinjauan literatur dan tinjauan pustaka. Proses tinjauan pustaka melibatkan sumber referensi dari jurnal dan situs web yang berkaitan dengan ruang lingkup penelitian ini. Penulis melakukan pencarian referensi secara online melalui internet. Setelah beberapa referensi terkumpul, penulis kemudian mengumpulkan informasi yang diperlukan untuk menyelesaikan penelitian.

   Metode berikutnya yang digunakan adalah studi literatur. Di tahap ini, penulis mengumpulkan dan menganalisa literatur yang terkait dengan penelitian yang sedang dilakukan. Literatur yang ditemukan bisa berupa jurnal, skripsi, dan karya tulis lainnya yang relevan. Dari beberapa literatur yang telah dikumpulkan, penulis melakukan pendalaman dan membuat perbandingan antara penelitian yang sedang dilakukan dengan penelitian-penelitian terdahulu. Perbandingan ini bertujuan untuk membuat penelitian penulis dapat melengkapi atau memperbaiki penelitian yang telah ada sebelumnya. Penulis mengumpulkan referensi berdasarkan ruang lingkup sebagai berikut:

   a) Penelitian yang berkaitan dengan sistem rekomendasi
   b) Penelitian yang berkaitan dengan pendekatan Content Based Filtering
   c) Penelitian yang berkaitan dengan TF-IDF dan Cosine Similarity
   d) Penelitian yang menggunakan bahasa pemrograman Flask sebagai basis pengembangan web

   Untuk mengembangkan sistem rekomendasi jurnal terindex scopus dalam penelitian ini, terdapat beberapa langkah yang harus dijalankan, termasuk analisa masalah, pengumpulan data, *preprocessing*, *cosine similarity*, dan evaluasi sistem.



Gambar 1 Alur Penelitian

B. *Pengumpulan Data*

Data yang digunakan pada penelitian untuk sistem rekomendasi jurnal terindeks scopus ini menggunakan data jurnal yang ada pada situs Scoopus dengan rentang waktu 2019-2024 (5 tahun terakhir). Di dalam situs

Scopus terdapat informasi detail dari jurnal, informasi ini sangat penting dalam Pembangunan model TF-IDF dan Cosine Similarity. Data jurnal/artikel ilmiah yang ada di Scopus diambil menggunakan scrapping, dan akan disimpan dalam file berekstensi csv.. File tersebut berisi detail dari jurnal yang disimpan.

1. Teknik Pengumpulan Data

   Pengumpulan data pada step awal menggunakan teknik scrapping pada web Scopus. Penulis akan mengambil data-data jurnal beserta informasi dari jurnal tersebut dalam bentuk file csv. Dalam file tersebut berisi informasi detail dari jurnal dalam bentuk kolom. Informasi ini penting untuk membangun model TF-IDF dan Cosine Similarity. Penulis menggunakan keyword spesifik ”Business Digital” dalam mencari artikel ilmiah untuk dijadikan dataset. Untuk mendapatkan hasil data jurnal yang lebih akurat, penulis membatasi filter *Subject Area*. Pembatasan subject area itu dilandasi oleh hasil riset dan anlisa penulis dari kurikulum program studi S1 Bisnis Digital Unesa, maupun dari literatur-literaur tentang business digital, sehingga penulis menarik kesimpulan dan menentukan batas subject hanya pada area berikut:

   a) Business, Management, and Accounting
   b) Social Sciences
   c) Engineering
   d) Computer Science
   e) Economics, Econometrics, and Finance
   f) Humanities
   g) Mathematics
   h) Decision Science

   Namun, penulis juga terbuka akan masukan atau saran dari user untuk menambahkan topik lain yang relevan dengan Bisnis Digital. Untuk itu, penulis akan menyediakan halaman “Saran” pada sistem ini, sehingga dataset tentang topik Bisnis Digital akan selalu relevan dengan kebutuhan user yang merupakan mahasiswa prodi Bisnis Digital Universitas Negeri Surabaya. Selanjutnya, hasil dari pencarian ini kemudian disimpan dalam format file csv menggunakan fitur 'export as csv' yang tersedia di situs Scopus.

   Penulis mengumpulkan total 20.000 data artikel ilmiah tentang topik bisnis digital, tentunya dari 20.000 judul artikel tersebut akan mempunyai variasi judul yang tidak terpatok pada kata “*Business Digital”*, misalnya tidak menutup kemungkinan judul tersebut mengenai *E-commerce* tanpa ada kata *Business Digital’* sedikitpun. Salah satu penyebabnya ialah informasi dari judul tersebut memiliki kata “business digital” misalnya pada kolom abstract atau keyword. Variabel-variabel penting yang diperlukan untuk pembuatan sistem rekomendasi dalam penelitian ini meliputi judul artikel, sumber sumber terbitan, penulis, tahun publikasi, abstrak, kata kunci, doi, dan url artikel

C. Preprocessing

Preprocessing adalah proses mengonversi *raw data* menjadi data informasi yang terstruktur, dan siap untuk diproses lebih lanjut.. Dalam penelitian ini, *preprocessing* dilakukan dengan menggunakan library spaCy, yakni salah satu library Natural Language Processing (NLP). NLP adalah cabang kecerdasan buatan yang bertujuan untuk memungkinkan komputer memahami teks dan bahasa dengan cara yang mirip dengan pemahaman manusia [14]. Spacy akan digunakan untuk memuat model bahasa *en_core_web_sm*.

Pada tahap preprocessing ini, penulis memanfaatkan korpus, yaitu kumpulan teks tertulis atau lisan yang disimpan untuk keperluan investigasi dan penelitian, termasuk dalam bidang Natural Language Processing (NLP). Dalam proses *preprocessing*, akan dilakukan 2 tahap *preprocessing* yang berbeda. *Preprocessing* text dataset akan dilakukan terpisah (tidak di sistem) dan hasil *preprocessing*nya akan disimpan kemudian akan “dipanggil” melalui kode program. Sedangkan *preprocessing* text keyword akan dilakukan langsung oleh sistem.

1. Case Folding

   Pada tahap *Case folding*, kalimat akan diubah melalui beberapa proses. Langkah pertama adalah mengonversi seluruh huruf dalam kalimat menjadi huruf kecil (lowercase) [15]. Proses *case folding* bertujuan untuk mengonversi kata atau kalimat dalam dokumen, termasuk judul, jenis, dan penulis, menjadi huruf kecil. Tujuan dari proses ini untuk menciptakan keseragaman antara setiap kata dalam data jurnal

Table 1 Case Folding

Sebelum	**Sesudah**
Contoh dari Case Folding	Contoh dari case folding
Financial, Business, and Accounting	financial, business, and accounting

2. Tokenizing

   Tokenisasi adalah langkah untuk membagi string input menjadi kata-kata. Selama proses ini, data dari setiap jurnal juga mengalami tahap pembersihan untuk menghilangkan simbol, tanda baca, angka, backslash, karakter non-ASCII, dan URL, spasi di awal dan akhir kalimat, mengubah spasi berlebih menjadi spasi tunggal, dan karakter tunggal. Dengan melakukan tokenizing, ukuran dokumen menjadi relatif kecil, sehingga proses training model nantinya dapat lebih cepat.

Table 2 Tokenizing

Sebelum	Sesudah
Contoh dari Tokenizing	Contoh, dari, Tokenizing

3. Filtering

Filtering, juga dikenal sebagai stopword removal, melibatkan ekstraksi kata-kata penting dari dokumen data jurnal. Selama proses ini, istilah-istilah yang kurang deskriptif, seperti konjungsi, preposisi, dan kata ganti—contohnya mencakup "dan", "dari", "untuk", "tetapi", dan "kepada"—juga diidentifikasi. Tujuan pemfilteran adalah untuk mengurangi jumlah kata dalam dokumen, sehingga meningkatkan keakuratan kumpulan data melalui pemanfaatan kata-kata penting.

Table 3 Filtering

Sebelum	Sesudah
Contoh dari Filtering	Contoh, Filtering
Japanese journal of public health	Japanese, journal, public, health

4. Lemmatization

Lemmatisasi atau Lemmatization merupakan proses mengubah kata ke bentuk dasarnya atau lemma. Proses ini bertujuan untuk mengkonsolidasikan berbagai bentuk kata yang memiliki makna dasar yang sama ke dalam satu bentuk dasar, memfasilitasi analisis semantik teks yang lebih efisien. Lemmatisasi mempertimbangkan konteks dan bagian dari ucapan kata dalam proses pengembalian ke bentuk dasar atau lemma. Lemmatisasi dan stemming sering kali disamakan, namun keduanya memiliki perbedaan yang signifikan. Stemming adalah proses yang lebih primitif, memotong bagian akhir kata untuk menghasilkan bentuk dasar atau stem, yang bisa menghasilkan kata yang tidak ada dalam kamus. Sebagai contoh, “eating” menjadi “eat” (benar), tetapi “caring” menjadi “car” (bukan kata yang valid).

Sebaliknya, Lemmatisasi menggunakan pemahaman morfologis dan memperhatikan konteks kata dalam prosesnya. Lemmatisasi menghasilkan lemma atau bentuk dasar yang pasti merupakan kata yang valid. Contoh: “was” menjadi “be”, “mice” menjadi “mouse”, “running” menjadi “run”, “better” menjadi “good”. Dalam konteks ini, Lemmatisasi cenderung mempunyai tingkat presisi yang lebih tinggi dibandingkan stemming karena mempertimbangkan konteks kata dalam kalimat untuk mengembalikannya ke bentuk dasar atau lemma yang benar. Ini sangat penting dalam pembangunan sistem rekomendasi yang efektif dan akurat.

D. Training Model

Proses penghitungan skor kemiripan antara setiap jurnal dilakukan melalui dua langkah. Langkah awal, menggunakan teknik Term Frekuensi-Invers Dokumen Frekuensi (TF-IDF) untuk memberikan bobot pada setiap istilah dalam informasi jurnal. Selanjutnya dihitung kemiripan antar jurnal dengan menggunakan metode cosine kesamaan.

1. Pembobotan TF IDF Vectorizer

Proses pemberian bobot pada kata dilakukan dengan menggunakan algoritma TF-IDF. Bobot sebuah kata mengukur signifikansinya dalam mewakili data jurnal. Dalam pemberian bobot TF-IDF, bobot suatu kata akan bertambah jika frekuensi kemunculannya dalam dokumen tertentu semakin tinggi. Sebaliknya, bobotnya akan berkurang jika kata tersebut lebih sering muncul di dokumen lain. Misalnya saja tiga kalimat yang telah melalui tahap preprocessing sebagai berikut:

a) Kalimat a: “new local startup technology”
b) Kalimat b: “international scale startup business”
c) Kalimat c: “international startup in the information technology era”

Dari 3 kalimat contoh, perhitungan menggunakan persamaan (2.5) mendapatkan hasil sebagai berikut:

No	Kata	Doc 1	Doc 2	Doc 3	df	Idf	Tf.idf		
							Doc 1	Doc 2	Doc 3
1	technology	1	0	1	2	Log(3/2)=0.1760	0.1760	0	0.1760
2	startup	1	1	1	3	Log(3/3)=0	0	0	0
3	new	1	0	0	1	Log(3/1)=0.4771	0.4771	0	0
4	local	1	0	0	1	Log(3/1)=0.4771	0.4771	0	0
5	business	0	1	0	1	Log(3/1)=0.4771	0	0.4771	0
6	scale	0	1	0	1	Log(3/1)=0.4771	0	0.4771	0
7	international	0	1	1	2	Log(3/2)=0.1760	0	0.1760	0.1760
8	era	0	0	1	1	Log(3/1)=0.4771	0	0	0.4771
9	information	0	0	1	1	Log(3/1)=0.4771	0	0	0.4771

Gambar 2 Hasil Pembobotan TF-IDF

2. Implementasi Algoritma Cosine Similarity

Cosine similarity digunakan untuk menilai sejauh mana dua vektor atau dua dokumen dalam ruang vektor memiliki kesamaan. Dengan metrik ini, sistem dapat menentukan sejauh mana satu data jurnal serupa dengan data jurnal lainnya. Perbandingan antara dokumen yang telah melalui

tahap pra-pemrosesan tidak hanya mempertimbangkan frekuensi kata (TF-IDF), tetapi juga sudut antara dokumen-dokumen. Perhitungan cosine similarity dapat dilakukan menggunakan Persamaan (2.6).

Misalnya, terdapat 2 buah dokumen, yaitu Doc 1 dan Doc 2, yang akan diuji pada sistem. Dokumen-dokumen ini telah melalui preprocessing. Nilai term Doc 1 dan term Doc 2 diperoleh dari frekuensi kata unik, misalnya kata 'startup' pada term Doc 1 memiliki nilai 2 karena muncul dua kali, begitu juga dengan term Doc 2. Selanjutnya, semua nilai dari kata-kata unik dimasukkan ke dalam rumus cosine similarity.

Table 4 Cosine Similarity

No	Term	T(1)	T(2)
1	startup	2	2
2	Business	1	1
3	digital	1	0
4	era	1	0
5	technology	1	1
6	information	1	0

Vector 1 dan 2 dalam tabel masing-masing merepresentasikan term "Doc 1" dan "Doc 2". Tujuannya adalah untuk menghitung jumlah nilai yang diperoleh untuk setiap kata unik dalam dokumen. Selanjutnya, perhitungan ini akan menggunakan metode cosine similarity.

a. Vektor 1 = (2,1,1,1,1,1)
b. Vektor 2 = (2,1,0,0,1,0)

Misalnya, untuk mendapatkan hasil perhitungan antara term 1 dan term 2 pada Tabel 4, dapat melakukan perhitungan cosine similarity. Dengan menggunakan persamaan pada (2.6), maka dapat menghasilkan nilai sebagai berikut:

$$\text{similarity}\ (1,2) = \frac{(2x2)+(1x1)+(1x0)+(1x1)+(1x0)}{\sqrt{2^2+1^2+1^2+1^2+1^2+1^2} \times \sqrt{2^2+1^2+0^2+0^2+1^2+0^2}}$$

$$\text{similarity}\ (1,2) = \frac{4+1+0+0+1+0}{\sqrt{4+1+1+1+1+1} \times \sqrt{4+1+0+0+1+0}}$$

$$\text{similarity}\ (1,2) = \frac{6}{\sqrt{9} \times \sqrt{6}}$$

$$\text{similarity}\ (1,2) = \frac{6}{7.35}$$

$$\text{similarity}\ (1,2) = \frac{6}{7.35} = 0.82$$

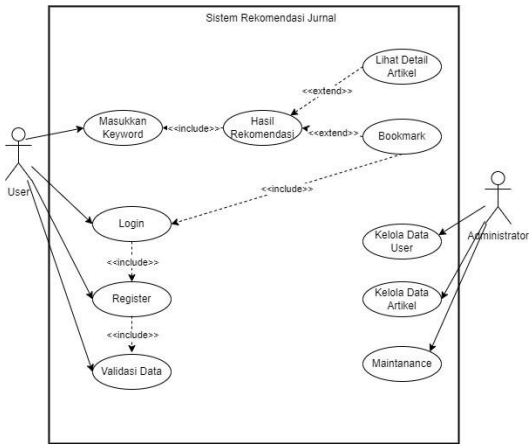
Setelah cosine similarity nya dihitung, maka didapatkan hasil ukuran kesamaan antara term Doc 1 dan term Doc 2 dalam ruang vektor tersebut, yaitu 0.82. Hasil ini menggambarkan bahwa term Doc 1 dan term Doc 2 memiliki kesamaan yang signifikan karena nilainya mendekati 1.

E. Perancngan Sistem

Berikut adalah perancangan dari sitem rekomendasi jurnal yang dibuat oleh penulis, terdiri dari dua macam, yakni Use Case Diagram, Flowchart Diagram Sistem, dan User Interface (UI):

1. Use Case Diagram

Use Case Diagram adalah representasi grafis dalam Unified Modeling Language (UML) yang menggambarkan interaksi antara aktor (pengguna) dan sistem. Diagram ini membantu memvisualisasikan fungsionalitas sistem dan bagaimana sistem tersebut digunakan oleh aktor. Berikut adalah perancangan use case diagram ditunjukkan pada Gambar 3.



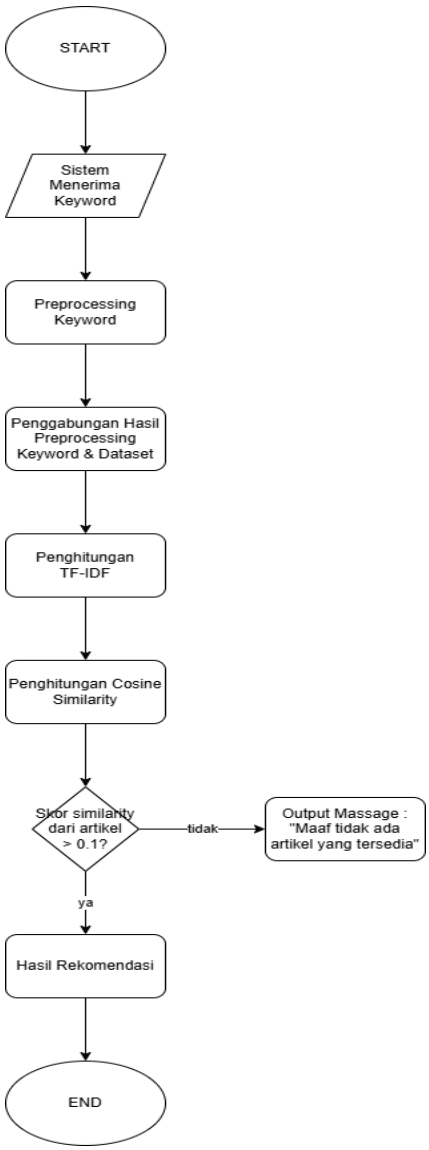
Gambar 3 Use Case Diagram

Pada diagram use case pada gambar 3, terlihat gambaran sederhana bagaimana interaksi antara sistem dan pengguna. Pengguna dapat menggunakan sistem tanpa login/register untuk mencari rekemendasi artikel dengan memasukkan keyword. Kemudian user dapat melihat detail dari artikel yang direkomendasikan. Namun yang membedakan antara user yang login dengan yang tidak login ialah, user yang login dapat menggunakan fitur bookmark untuk menyimpan judul artikel.

2. Flowchart Diagram Sistem

Flowchart atau diagram alir adalah diagram yang menggunakan simbol-simbol standar untuk memvisualisasikan langkah-langkah dan keputusan dalam suatu proses. Flowchart adalah alat yang berharga untuk menggambarkan cara kerja sistem. Pada Flowchart yang penulis buat, akan terlihat bagaimana alur dari sistem rekomendasi ini bekerja. Dimulai dari user memasukkan keyword, lalu sistem menerima keyword, kemudian sistem melakukan preprocessing terhadap dataset yang sesuai dengan keyword dari user, selanjutnya sistem melakukan training model tf-idf, setelah dilakukan tf-idf, dilakukan penghitungan cosine

similarity. Apabila ada judul artikel yang nilai cosine similaritynya > 0.1, maka sistem akan memasukkan judul tersebut untuk hasil rekomendasi ke user. Apabila dalam tahap penghitungan cosine similarity tidak ada satupun yang nilainya >0.1, maka sistem tidak akan merekomendasikan apa-apa. Berikut ini adalah *flowchart* diagram dari Sistem Rekomendasi Jurnal yang penulis buat:



Gambar 4 Flowchart Sistem

F. Evaluasi

Evaluasi merupakan langkah untuk mengukur sejauh mana efektivitas sistem yang sudah dibangun. Data yang telah diproses dari preprocessing sampai perhitungan cosine similarity perlu ditinjau dan dievaluasi.

1. Pengujian User

   Penulis merasa perlunya dilakukan pengujian oleh user untuk memastikan apakah sistem rekomendasi jurnal berbasis web dengam metode content-based filtering ini dapat membantu mahasiswa Program Studi Bisnis Digital untuk mencari jurnal terindex scopus yang sesuai dengan kebutuhannya. Pengujian akan menggunakan metode survei menggunakan kuesioner kepada 30 responden yang akan mencoba sistem ini, sekaligus hasil kuesioner akan digunakan sebagai data untuk kesimpulan dari peneletian ini.

2. Pengujian Sistem

   Pengujian sistem dilakukan dengan 2 cara, cara pertama dengan menggunakan metode black box untuk menemukan kesalahan fungsional pada aplikasi, dan cara kedua dengan menggunakan precision, precision digunakan untuk menilai sejauh mana kecocokan antara informasi yang diinginkan pengguna dan respons yang diberikan oleh sistem. Melalui precision, maka dapat diukur dan dievaluasi akurasi sistem rekomendasi yang telah dikembangkan.

G. Implementasi GUI

Sistem rekomendasi jurnal yang berhasil dikembangkan selanjutnya akan diintegrasikan ke dalam website. Dalam ranah pengembangan web, penulis menggunakan bahasa pemrograman Python dengan menggunakan framework Flask, dengan tambahan dukungan JavaScript, Bootstrap, dan HTML.

H. Komponen Perancangan Sistem

Untuk mengembangkan sistem rekomendasi jurnal dan mengimplementasikannya sebagai situs web, komponen perancangan yang terstruktur dengan baik sangatlah penting. Komponen-komponen perancangan yang diperlukan untuk pengembangan sistem ini antara lain sebagai berikut:

:

a. Hardware
   1) Processor Intel Core i5 9th Gen (4 Core)
   2) RAM 8 GB
   3) SSD 512 GB

b. Software
   1) OS Windows 11
   2) Visual Studio Code
   3) Google Colab
   4) Microsoft Excel
   5) Google Chrome
   6) Microsoft Edge

IV. HASIL DAN PEMBAHASAN

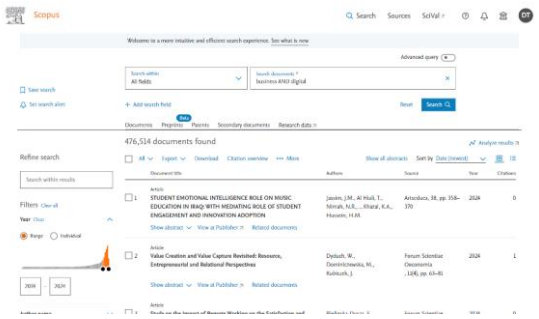
A. Pengumpulan Data

Data penelitian ini diperoleh dari website Scopus, dengan metode pengumpulan data melalui teknik web scraping. Proses ini melibatkan pencarian artikel terkait Bisnis Digital dalam periode 2019-2025, dengan keyword yang spesifik yakni Business Digital dan dengan pemfilteran subject area. Hasil dari pencarian ini

kemudian disimpan dalam format file csv menggunakan fitur 'export as csv' yang tersedia di situs Scopus. Penelitian ini mengumpulkan total 20.000 data artikel ilmiah. Variabel-variabel penting yang diperlukan dari data yang dikumpulkan untuk pembuatan sistem rekomendasi dalam penelitian ini meliputi judul artikel, penulis, tahun publikasi, abstrak, kata kunci, doi, dan url artikel. Variabel-variabel tersebut sangat penting untuk pembangunan model tf-idf dan cosine similarity.

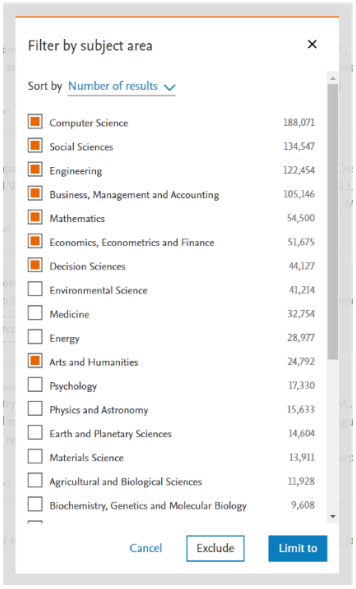
1. Web Scrapping

   Proses pengambilan data dengan metode scrapping dari situs Scopus. Proses ini akan mengumpulkan data jurnal/artikel ilmiah (selanjutnya akan disebut jurnal) beserta informasi detail dari jurnal tersebut.



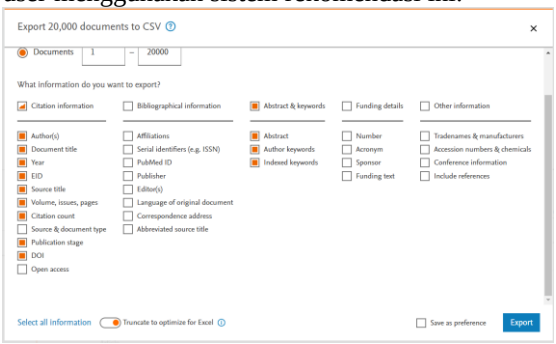
Gambar 5 Web Scrapping

Penulis menggunakan keyword spesifik, yakni Business Digital, dengan rentang waktu dari tahun 2019-2024 (5 tahun terakhir). Kemudian, akan dilakukan filtering subject area. Tujuan dari filtering subject area ini ialah untuk membatasi hasil jurnal yang muncul agar lebih spesifik dalam topik bisnis digital. Ada 8 subject area yang penulis pilih, Computer Science; Social Sciences; Engineering; Business, Management and Accounting; Mathematics; Economics, Econometrics and Finance; Decision Sciences; Art and Humanities. Diharapkan dengan filtering subject area ini akan memberikan dataset yang relevan dengan prodi Bisnis Digital Unesa.



Gambar 6 Filtering Subject Area

Setelah filtering subject area dilakukan, makan akan dilakukan Export data dalam bentuk file eksentensi CSV. Pada tahap ini, penulis akan di-export 20.000 data judul jurnal. Kemudian penulis, melakukan filtering detail informasi yang dibutuhkan untuk diexport. Informasi ini perlu untuk tahap TF-IDF dan untuk sistem agar dapat memberikan informasi detail kepada user pada saat user menggunakan sistem rekomendasi ini.



Gambar 7 Export CSV

B. Preprocessing Dataset

Tahap preprocessing data menjadi sebuah fase yang krusial. Fase ini melibatkan serangkaian teknik transformasi yang mengubah data mentah (raw data) menjadi struktur data yang lebih bersih, terstruktur dan siap digunakan. Kualitas data yang dihasilkan dari tahap preprocessing ini memiliki dampak langsung terhadap performa sistem rekomendasi yang akan dikembangkan. Data yang telah melalui tahap preprocessing ini akan menjadi bahan baku dalam proses pelatihan model TF-IDF serta cosine similarity yang menjadi inti dari sistem rekomendasi jurnal ini. Tahap preprocessing dataset dilakukan secara terpisah melalui Google Colab, hasil

preprocessing kemudian akan disimpan dan akan "dipanggil" melalui code program, untuk selanjutnya dilakukan merger antara hasil preprocessing dataset dan hasil preprocessing keyword ke dalam satu list.

1. Case Folding

   Case Folding dilakukan untuk mengubah seluruh isi teks menjadi huruf kecil untuk mengurangi kompleksitas dan variasi dalam data teks. Tahap case folding dilakukan pada string teks isi dataset, dengan menggunakan metode .lower() sebelum data diproses dengan spaCy.

```
1.  !pip install spacy
2.  !python -m spacy download en_core_web_sm
3.
4.  import spacy
5.  import pandas as pd
6.
7.  # Muat model spaCy
8.  nlp = spacy.load("en_core_web_sm")
9.
10. # Fungsi untuk melakukan preprocessing
    teks
11. def preprocess_text(text):
12.     text = str(text).lower()
13.     doc = nlp(text)
14.     processed_text          =          "
    ".join([token.lemma_ for token in doc if
    not token.is_stop and not token.is_punct
    and not token.is_space])
15. return processed_text
```

2. Tokenizing

   Selanjutnya dilakukan proses Tokenizing untuk memotong teks atau string menjadi potongan atau unit-unit token yang lebih kecil sebelum dilakukan analisis lebih mendalam. Tokenizing terjadi secara implisit sebagai bagian dari proses pemrosesan teks dengan spaCy ketika nlp(text) dipanggil, spaCy melakukan beberapa langkah pemrosesan NLP, termasuk tokenizing. Fungsi nlp dari spaCy melakukan beberapa langkah pemrosesan, termasuk tokenizing. Pada tahap ini, teks dipecah menjadi unit-unit token. Dengan demikian, tokenizing dilakukan oleh spaCy sebagai bagian dari pemanggilan nlp(text),yang kemudian memungkinkan untuk melakukan operasi lebih lanjut pada token, seperti lemmatization dan filtering (stopwords). Setiap token dalam doc (doc = nlp(text)) merupakan hasil dari tokenisasi, misalnya lemma (token.lemma_), stop word (token.is_stop), dan tanda baca (token.is_punct).

```
16. !pip install spacy
17. !python -m spacy download en_core_web_sm
18.
19. import spacy
20. import pandas as pd
21.
22. # Muat model spaCy
23. nlp = spacy.load("en_core_web_sm")
24.
25. # Fungsi untuk melakukan preprocessing
    teks
26. def preprocess_text(text):
27.     text = str(text).lower()
28.     doc = nlp(text)
29.     processed_text          =          "
    ".join([token.lemma_ for token in doc if
    not token.is_stop and not token.is_punct
    and not token.is_space])
    return processed_text
```

3. Lemmatization

   Pada code doc = nlp(text), setelah code dijalankan, maka proses tokenizing langsung terjadi, secara teknis proses Lemmatization juga terjadi setelah proses tokenisasi terjadi. Lemmatisasi dilakukan oleh spaCy, spaCy melihat setiap token (kata atau tanda baca) yang dihasilkan dari tokenisasi dan mencoba menentukan bentuk dasar atau "lemma" dari setiap kata. Setelah doc = nlp(text) dijalankan, objek doc yang dihasilkan berisi informasi tentang setiap token, termasuk lemma-nya. Akses dan penerapan lemma dari setiap token untuk membentuk teks yang diproses terjadi di tahapan selanjutnya

4. Filtering

   Setelah dilakukan tahap tokenizing dan lemmatization, maka dilakukan filtering stopword removal termasuk tanda baca. Hal ini untuk membersihkan kata-kata yang kurang penting termasuk juga tanda baca. Dapat dilihat pada baris ke 14 di kode program untuk melakukan filtering. *Function* if not token.is_stop bertujuan untuk memfilter token yang merupakan stop words. Stop words adalah kata-kata yang sering muncul (misalnya "the", "is", "in") dan biasanya dianggap tidak memberikan makna signifikan dalam analisis teks. Dengan mengeliminasi stop words, jumlah

token yang perlu diproses lebih lanjut dapat dikurangi, sehingga meningkatkan fokus pada kata-kata yang memiliki potensi untuk membawa makna penting.

Mirip dengan stop words, tanda baca juga dihilangkan dengan menggunakan if not token.is_punct. Tanda baca seringkali tidak diperlukan dalam banyak analisis teks dan dapat dianggap sebagai noise. Dengan melakukan stop words removal dan punctuation removal, tahap preprocessing menjadi lebih fokus pada kata-kata yang membentuk isi teks. Setelah semua langkah-langkah ini dilakukan, token yang tersisa (yang bukan stop words atau tanda baca dan telah dilemmatisasi) digabungkan Kembali menjadi string dengan menggunakan " ".join(...). Hasilnya adalah processed_text, yaitu versi teks yang telah diproses dan disaring.

```
1.  ... # Code Import Modul sebelumnya
2.
3.  # Muat model spaCy
4.  nlp = spacy.load("en_core_web_sm")
5.
6.  # Fungsi untuk melakukan preprocessing
    teks
7.          .... # Code sebelumnya
8.
9.  # Muat dataset
10. df =
    pd.read_csv("/content/drive/MyDrive/data
    set/datasetbisdig.csv")
11. df.columns = [col.lower() for col in
    df.columns]
12.
13. # Pastikan teks tidak null
14. df['title'] = df['title'].fillna('')
15. df['source title'] = df['source
    title'].fillna('')
16. df['abstract'] =
    df['abstract'].fillna('')
17. df['author keywords'] = df['author
    keywords'].fillna('')
18.
19. # Gabungkan kolom title, source title,
    abstract dan authors keyword untuk
    preprocessing
20. df['combined_text'] = df['title'] + ' '
    + df['source title'] + ' ' +
    df['abstract']+ ' ' + df['author
    keywords']
21.
22. # Terapkan preprocessing
23. df['processed_text'] =
    df['combined_text'].apply(preprocess_tex
    t)
24.
25. # Sekarang df['processed_text'] berisi
    teks yang sudah dipreprocess
26.
27. df.to_csv("dataset_preprocessed.csv",
    index=False)
28.
29. # Dowmload dataset
30. files.download("PoPCites_preprocessed.cs
    v")
```

C. Preprocessing Keyword

Pada preprocessing keyword, tahap ini langsung dilakukan oleh sistem itu sendiri disaat user menginput keyword. Fungsi (function) yang digunakan untuk preprocessing keyword menggunakan fungsi yang sama untuk melakukan preprocessing pada dataset. Fungsi preprocessing akan “dipanggil” melalui code seperti code pada baris 7. Selanjutnya, melalu function index (def_index) dataset dan keyword yang sudah diprocessing digabungkan ke dalam satu list Bernama all_text.

```
1.  def index():
2.      keyword = ''
3.  ..... # Code sebelumnya
4.
5.
6.  # Preproccesing Keyword dengan spaCy
7.  processed_keyword                    =
    preprocess_text(keyword)
8.
9.  #  Menggabungkan  Dataset  yang  Sudah
    Dipreproses  dengan  Kata  Kunci  yang
    Dipreproses ke dalam Satu Daftar
10. all_texts                            =
    data['processed_text'].tolist()      +
    [processed_keyword]
```

D. Training Model (TF-IDF dan Cosine Similarity)

Tahap training model, yakni Term Frequency (TF) dan Inverse Document Frequency (IDF) serta Cosine Similarity merupakan tahap inti dalam sistem rekomendasi yang menggunakan content based filtering. Pada tahap ini, akan dilakukan pembobotan TF-IDF terhadap seluruh data jurnal, kemudian akan dilakukan penghitungan kemiripan *Cosine Similarity*.

1. Pembobotan Menggunakan TF-IDF

   TF-IDF mengukur signifikansi atau seberapa pentingnya suatu kata dalam sebuah dokumen relatif terhadap seluruh kumpulan dokumen (corpus). Vektor TF-IDF dihitung menggunakan TfidfVectorizer dari modul sklearn.feature_extraction.text (scikit-learn). TfidfVectorizer digunakan untuk mengubah teks menjadi representasi fitur numerik (vektor). Kemudian method fit_transform() digunakan pada objek vectorizer dengan input berupa koleksi dokumen teks. Method ini akan melakukan fitting. model dan transformasi teks menjadi matriks TF-IDF. Matriks ini dibangun dengan menggabungkan nilai TF dan IDF. Setiap baris dalam matriks mewakili dokumen, dan setiap kolom mewakili kata unik di seluruh dokumen (corpus). Nilai dalam matriks ini adalah hasil kali TF dan IDF untuk setiap kata dalam setiap dokumen. Nilai dalam matriks ini mencerminkan bobot TF-IDF dari kata-kata tersebut dalam dokumen.

```
# Menghitung matriks TF-IDF
vectorizer = TfidfVectorizer()
tfidf_matrix =
vectorizer.fit_transform(all_texts)
```

2. Cosine Similarity

   Pada tahap cosine similarity menggunakan library scikit-learn dari modul sklearn.metrics.pairwise. Selanjutnya akan dilakukan ekstraksi vector, yakni vector keyword dan vector article (dataset jurnal). Vektor untuk kata kunci (baris terakhir dari matriks TF-IDF dan vector untuk semua artikel (semua baris kecuali baris terakhir). Vektor kata kunci berada pada baris terahkir karena hasil preproccesing keyword merupakan hasil penambahan ke list. Nilai cosine similarity berkisar antara -1 dan 1, di mana nilai 1 menunjukkan bahwa dua vektor memiliki arah yang sama (sudut 0 derajat), nilai 0 menunjukkan bahwa dua vector orthogonal (sudut 90 derajat), dan nilai -1 menggambarkan bahwa dua vektor memiliki arah yang berlawanan (sudut 180 derajat).

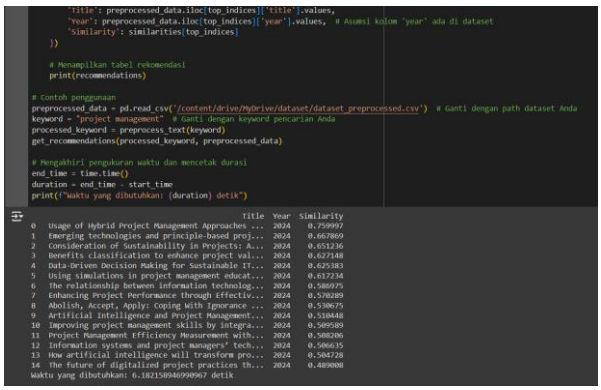
```
# Mengambil baris akhir
keyword_vector = tfidf_matrix[-1]
# Mengambil semua baris kecuali baris
terakhir (keyword)
article_vectors = tfidf_matrix[:-1]

similarities =
cosine_similarity(keyword_vector,
article_vectors)[0]
data['similarity'] = similarities
```

Dalam hal ini, *cosine_similarity(keyword_vector, article_vectors)[0]* menghitung cosine dari sudut antara vektor kata kunci (keyword_vector) dan setiap vektor artikel dalam article_vector. Hasilnya adalah array dari nilai kesamaan, yang menunjukkan seberapa dekat atau relevan setiap artikel terhadap kata kunci berdasarkan sudut antara kedua vector.

E. Hasil Rekomendasi

Setelah melalui tahap Preprocessing, TD-IDF, dan Cosine Similarity, selanjutnya penulis melakukan uji coba hasil Rekomendasi Jurnal melalui Google Colab, untuk memberi gambaran umum bagaimana sistem bekerja. Penulis mencoba menyesuaikan kode pemrograman agar dapat di uji coba di Google Colab untuk melihat implementasi content based filtering apakah berjalan dengan baik dan untuk memberikan gambaran umum bagaimana sistem bekerja. Pada gambar 8 diatas, penulis mencoba menggunakan keyword “project management”, dan berhasil menampilkan rekomendasi jurnal yang sesuai dengan keyword tersebut dengan nilai cosine similarity yang terlihat pada gambar 8.



Gambar 8 Hasil Uji Coba

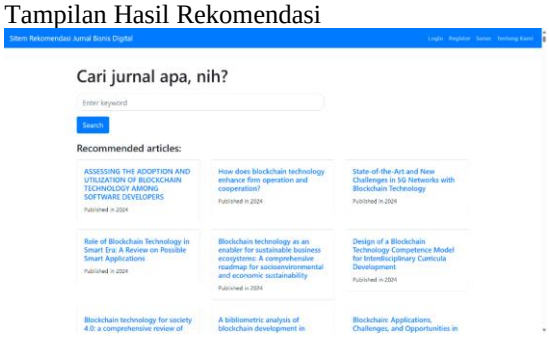
F. Implementasi GUI

Sistem rekomendasi jurnal ini kemudian akan di implementasikan dalam bentuk sebuah website. Dikarenakan preprocessing dataset sudah dilakukan secara terpisah, tahap preprocessing keyword, penggabungan hasil preprocessing, penghitungan TF-IDF, dan cosine similarty akan langsung dilakukan oleh sistem.

1. Tampilan Beranda

Gambar 9 Tampilan Beranda

2. Tampilan Hasil Rekomendasi



Gambar 10 Tampilan Result

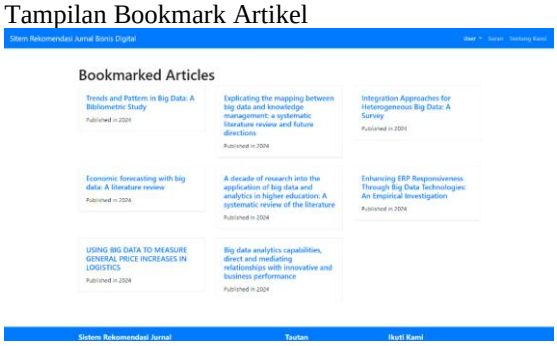
3. Tampilan “Article Not Found”

Gambar 11 Tampilan Pesan Article Not Found

4. Tampilan Detail Artikel/Jurnal

Gambar 12 Tampilan Detail Artikel

5. Tampilan Bookmark Artikel

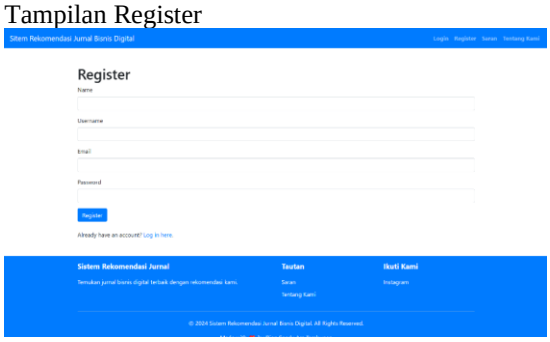


Gambar 13 Tampilan Bookmark Artikel

6. Tampilan Login

Gambar 14 Tampilan Login

7. Tampilan Register



Gambar 15 Tampilan Register

8. Tampilan Saran

Gambar 16 Tampilan Saran

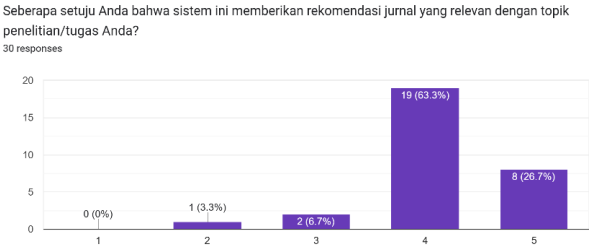
G. Pengujian Sistem oleh User

Pengujian user dilakukan untuk mendapatkan feedback dan mengukur efektivitas sistem rekomendasi jurnal berbasis web dalam membantu mahasiswa Program Studi Bisnis Digital menemukan jurnal terindeks Scopus yang relevan dengan kebutuhan mereka. Sebanyak 30 mahasiswa berpartisipasi dalam survei ini, memberikan umpan balik mengenai berbagai aspek sistem, termasuk relevansi rekomendasi, efisiensi waktu, dan kemudahan penggunaan. Survei ini dilakukan secara online melalui WhatsApp dari tanggal 21 Mei-27 Mei 2024 dengan menargetkan mahasiswa S1 Program Studi Bisnis Digital Universitas Negeri Surabaya yang dalam aktivitas perkuliahannya membutuhkan referensi jurnal untuk keperluan tugas atau penelitian. Kuesioner menggunakan skala Likert 1-5 dan terdiri dari 6 item pertanyaa. Adapaun pertanyaannya sebagai berikut :

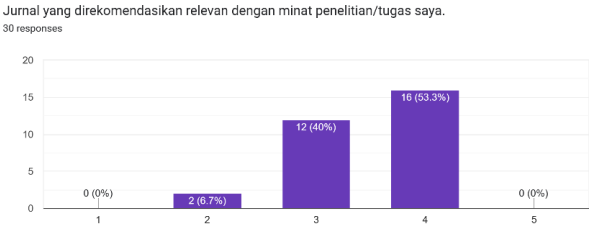
a) Seberapa setuju Anda bahwa sistem rekomendasi ini membantu Anda menemukan jurnal terindeks Scopus yang relevan dengan kebutuhan Anda?
b) Seberapa setuju Anda bahwa sistem ini memberikan rekomendasi jurnal yang relevan dengan topik penelitian/tugas Anda?
c) Jika berkaca dari cara anda mencari jurnal sebelumnya, seberapa setuju Anda bahwa sistem ini mengurangi waktu dan usaha yang Anda butuhkan untuk mencari jurnal yang sesuai? (terlepas dari lambatnya situs memuat hasil atau loading website)
d) Jurnal yang direkomendasikan relevan dengan minat penelitian/tugas saya.
e) Sistem ini membantu saya menemukan jurnal baru yang belum saya ketahui sebelumnya .
f) Saya merasa sistem ini menghemat waktu saya dalam mencari jurnal. (terlepas dari lambatnya situs memuat hasil atau loading website)

Berdasarkan hasil survei kuesioner yang sudah diisi oleh 28 responden, dapat disimpulkan bahwa mayoritas responden setuju bahwa sistem rekomendasi jurnal berbasis web dengan metode content-based filtering membantu mereka dalam menemukan jurnal bisnis digital terindeks Scopus yang relevan dengan kebutuhan mereka. Hal ini terlihat dari tingginya persentase responden yang memberikan nilai 4 atau 5 pada pertanyaan-pertanyaan kunci yang relevan dengan rumusan masalah, seperti terlihat pada gambar grafik dibawah.

Gambar 17 Grafik Pertanyaan 1

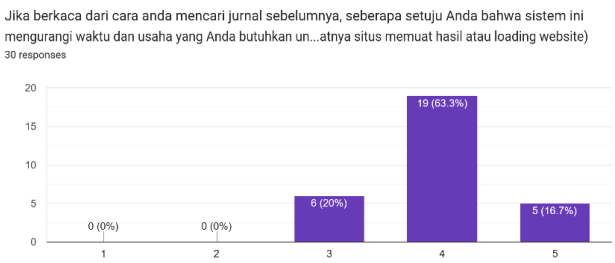


Gambar 18 Grafik Pertanyaan 2

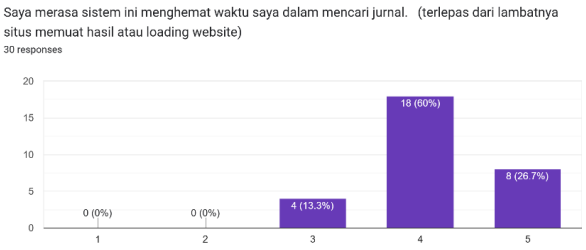


Gambar 19 Grafik Pertanyaan 3

Gambar 20 Grafik Pertanyaan 4



Gambar 21 Grafik Pertanyaan 5



Gambar 22 Grafik Pertanyaan 6

H. Pengujian Sistem

1. Precision

Pengujian Precision dilakukan dengan mengambil 20 sampel keyword yang relevan dengan topik Bisnis Digital. Setelah itu dilakukan pencarian satu per satu dengan semua keyword sampel. Lalu diambil 6 hasil rekomendasi pertama untuk dinilai relevansinya atau tidak (True Positive dan False Positive). Untuk penilaian relevansi dilakukan secara analisa manual dengan menggunakan kriteria judul dan keyword pada artikel yang mengandung keyword sample. Kemudian penulis menganalisa konteks artikel tersebut melalui informasi Abstract, untuk mendapatkan kesimpulan bahwa artikel hasil rekomendasi relevan atau tidak. Selanjutnya akan dihitung Precision setiap keyword menggunakan rumus:

$$Precision = TP / (TP + FP)$$

No	Keyword Sample	Total Data Relevan (TP)	Total Data Tidak Relevan (FP)	Jumlah Data (TP+FP)	Precision
1	Digital Marketing	6	0	6	100%
2	E-commerce	6	0	6	100%
3	Blockchain in Business	6	0	6	100%
4	Digital Transformation	6	0	6	100%
5	Online Consumer Behavior	6	0	6	100%
6	Social Media Marketing	6	0	6	100%
7	Fintech	6	0	6	100%
8	Big Data Analytics in Business	6	0	6	100%
9	Internet of Things (IoT) in Business	0	6	6	0%
10	Artificial Intelligence in Business	3	3	6	50%
11	Cloud Computing in Business	2	4	6	33.3%
12	Cybersecurity in Business	1	5	6	16.6%
13	Project Management	6	6	6	100%
14	Digital Payment Systems	5	1	6	83.3%
15	Digital Supply Chain	6	0	6	100%
16	Smart Contracts	6	0	6	100%
17	Digital Business Models	6	0	6	100%
18	Digital Advertising	4	2	6	66.6%
19	Machine Learning	6	0	6	100%
20	Customer Experience	6	0	6	100%
Rata-rata *Precision*					85.0%

Gambar 23 Precision

Maka didapatkan rata-rata Precision dari sistem ini adalah 85.0%. Hasil tersebut menunjukkan bahwa Sistem Rekomendasi Jurnal dengan Metode Content Based Filtering mampu memberikan rekomendasi jurnal/artikel dengan baik.

2. Pengujian Sistem Menggunakan Black-box

Pengujian black-box adalah metodologi pengujian perangkat lunak yang menekankan evaluasi fungsionalitas suatu sistem tanpa memerlukan pengetahuan tentang detail implementasi internalnya. Pengujian ini dilakukan dengan memberikan input ke sistem dan mencermati output yang dihasilkan, lalu membandingkannya dengan hasil yang diharapkan.

Table 5 Blackbox Testing

No	Scenario	Actual Result	Kesimpulan
1	Sistem dapat memuat beranda	Masuk ke url, muncul tampilan beranda	Valid
2	Sistem dapat menerima input keyword	Memasukkan keyword ke kolom pencarian	Valid
3	Sistem dapat menampilkan hasil	Memasukkan keyword dan mengklik Search	Valid
4	Sistem menampilk	Memasukkan keyword	Valid

	an pesan "no result"	yang bukan topik bisnis digital	
5	Artikel hasil rekomendasi dapat dilihat detailnya	Hasil rekomendasi artikel di klik	Valid
6	Artikel hasil rekomendasi dapat dibookmark	Mengklik tombol "Bookmark" dari artikel hasil rekomendasi	Valid
7	User bisa mengedit profile	Mengedit detail profile user (username, nama, email, password) melalui halaman "User">"Profile"	Valid
8	User bisa register	Mendaftar melalui halaman Register	Valid
9	User bisa login	Login melalui halaman login	Valid
10	User bisa logout	Mengklik tombol logout	Valid
11	User bisa reset password	Masuk ke halaman login dan mengklik tombol "Forgot Password"	Valid
12	User dapat memberi Saran	Mengklik navbar "Saran" dan memberikan saran	Valid

Berdasarkan hasil pengujian blackbox testing, dapat disimpulkan bahwa fungsionalitas dari Sistem Rekomendasi Jurnal ini berjalan dengan lancer.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan temuan penelitian yang dilakukan pada pengembangan Sistem Rekomendasi Jurnal dengan metode Content-Based Filtering, maka dapat disimpulkan bahwa:

1. Sistem rekomendasi jurnal dengan menggunakan metode Content Based Filtering (CBF) dapat memberikan rekomendasi jurnal yang relevan kepada pengguna atau mahasiswa produ bisnis digital Universitas Negeri Surabaya sesuai dengan topik bisnis digital dan keyword yang dimasukkan.
2. Hasil penelitian menunjukkan bahwa akurasi precision dari sistem rekomendasi jurnal menggunakan metode content based filtering adalah sebesar 85,0%.
3. Sistem rekomendasi jurnal ini mampu menunjang pelaksanaan tugas dan penelitian mahasiswa prodi Bisnis Digital Universitas Negeri Surabaya.

B. Saran

Berdasarkan temuan penelitian pengembangan Sistem Rekomendasi Jurnal Terindeks Scopus dengan menggunakan metode Content Based Filtering (CBF), ada beberapa rekomendasi untuk penelitian selanjutnya yang dapat dipertimbangkan, yaitu:

1. Dataset yang digunakan dalam sistem rekomendasi ini dapat diperluas atau diperbanyak, sehingga memberikan rekomendasi hasil yang lebih baik
2. Dalam pengujian precision, agar menggunakan Ground Truth dari ahli.
3. Menambahkan fitur-fitur pendukung pada sistem, untuk mempermudah user

REFERENSI

[1] Bawden, D. & Robinson, L. (2020). Information Overload: An Overview. In: Oxford Encyclopedia of Political Decision Making. . Oxford: Oxford University Press. doi: 10.1093/acrefore/9780190228637.013.1360

[2] Da'u, A., Salim, N. (2020). Recommendation system based on deep learning methods: a systematic review and new directions. Artif Intell Rev 53, 2709–2748. https://doi.org/10.1007/s10462-019-09744-1

[3] Fayyaz, Z.; Ebrahimian, M.; Nawara, D.; Ibrahim, A.; Kashef, R. (2020). Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities. Applied Sciences 10, no. 21: 7748. https://doi.org/10.3390/app10217748

[4] Gupta, M., Thakkar, A., Aashish, V., Gupta, V., & Rathore, D. P. S. (2020). Movie recommender system using collaborative filtering. In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 415-420). IEEE. https://doi.org/10.1109/ICESC48915.2020.9155879

[5] Roy, D., Dutta, M. (2022). A systematic review and research perspective on recommender systems. J Big Data 9, 59. https://doi.org/10.1186/s40537-022-00592-5

[6] Lops, P., Jannach, D., Musto, C. et al. (2019). Trends in content-based recommendation. User Model User-Adap Inter 29, 239–249. https://doi.org/10.1007/s11257-019-09231-w

[7] Dogucu, M., & Çetinkaya-Rundel, M. (2021). Web Scraping in the Statistics and Data Science Curriculum: Challenges and Opportunities. Journal of Statistics and Data Science Education, 29(sup1), S112–S122. https://doi.org/10.1080/10691898.2020.1787116

[8] NAF'AN, M. Z., BURHANUDDIN, A., & RIYANI, A. (2019). Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen [Application of Cosine Similarity and TF-IDF Weighting to Detect Document Similarity]. *Jurnal Linguistik Komputasional*, *2*(1), 23-27. https://inacl.id/journal/index.php/jlk/article/view/17. doi:10.26418/jlk.v2i1.17

[9] Asudani, D. S., Nagwani, N. K., & Singh, P. (2023). Impact of word embedding models on text analytics in deep learning environment: a review. Artif Intell Rev 56, 10345–10425. https://doi.org/10.1007/s10462-023-10419-1

[10] Utomo, P. E. P., Manaar, Khaira, U., & Suratno, T. (2019). ANALISIS SENTIMEN ONLINE REVIEW PENGGUNA BUKALAPAK MENGGUNAKAN METODE ALGORITMA TF-DF. Jurnal Sains Dan Sistem Informasi

[11] Apriani, Zakiyudin, H., & Marzuki, K. (2021). Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF System Penerimaan Mahasiswa Baru pada Kampus Swasta. Jurnal Bumigora Information Technology

[12] Laxmi, M. D., Indriati, & Fauzi, M. A. (2019). Query Expansion Pada Sistem Temu Kembali Informasi Berbahasa Indonesia dengan Metode Pembobotan TF-IDF dan Algoritme Cosine Similarity Berbasis Wordnet. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer.

[13] Nugroho, F. A., Septian, F., Pungkastyo, D. A., & Riyanto, J. (2020). Penerapan Algoritma Cosine Similarity untuk Deteksi Kesamaan Konten pada Sistem Informasi Penelitian dan Pengabdian Kepada Masyarakat. Jurnal Informatika Universitas Pamulang, 5(4), 529-536. doi: 10.32493/informatika.v5i4.7126. Diakses dari http://openjournal.unpam.ac.id/index.php/informatika

[14] Education, I. C. (2020). What is Natural Language Processing? Dipetik Maret, 2024, dari IBM: https://www.ibm.com/cloud/learn/natural-language-processing

[15] IGLPE Prismana, DR Prehanto, DA Dermawan, AC Herlingga, & SC Wibawa. (2021). Nazief & Adriani Stemming Algorithm With Cosine Similarity Method For Integrated Telegram Chatbots With Service. IOP Conference Series: Materials Science and Engineering, 1125, 012039.