

## Sistem Temu Kembali Informasi Jurnal Ilmiah Unisbank Dengan Metode *Cosine Similarity*

Nanang Setiawan<sup>a</sup>, Fatkhul Amin<sup>b</sup>

<sup>a</sup>Teknik Informatika, Teknologi Informasi dan Industri, Unisbank, nanangsetiawaan@yahoo.com

<sup>b</sup>Teknik Informatika, Teknologi Informasi dan Industri, Unisbank, fatkhulamin@edu.unisbank.ac.id

Submitted: 09-05-2024, Reviewed: 28-05-2024, Accepted 12-06-2024

<https://doi.org/10.47233/jteksis.v6i3.1391>

### Abstract

The role of Unisbank scientific journals is very important as a source of knowledge and information. As time goes by, the number of existing journals increases, so that searching for relevant journals quickly and efficiently is certainly difficult. Manually, searching for scientific journals takes a lot of time, so if you want to find relevant scientific journals as quickly as possible, users only search based on similarities in journal titles. As a result, the scientific journals obtained are not relevant to the user's expectations. The aim of this research is to build a system that can find Unisbank scientific journals that are relevant to users by applying the Cosine Similarity method. The way the Cosine Similarity method works is to calculate the similarity between scientific journals and keywords entered by users. This research will also apply the TF-IDF algorithm and apply text-preprocessing so that the data produced before the Cosine Similarity calculation process is more structured. The results of this research are web-based Unisbank scientific journal search system with system testing on 30 Unisbank scientific journals resulting in the system being able to provide an average recall value of 0.4 and precision of 0.8 at a threshold of 0.05.

**Keywords:** Information Retrieval System, Cosine Similarity, Scientific Journal

### Abstrak

Peran jurnal ilmiah Unisbank sangat penting sebagai sumber pengetahuan dan informasi. Seiring berjalannya waktu, jumlah jurnal yang ada semakin banyak sehingga untuk melakukan pencarian jurnal yang relevan dengan cepat dan efisien pastinya sulit dilakukan. Pencarian jurnal ilmiah secara manual membutuhkan banyak waktu sehingga apabila ingin mendapatkan jurnal ilmiah yang relevan secepatnya, pengguna hanya akan mencari berdasarkan kemiripan pada judul jurnal, akibatnya jurnal ilmiah yang didapatkan tidak relevan dengan harapan pengguna. Tujuan penelitian ini adalah membangun sistem yang dapat menemukan jurnal ilmiah Unisbank yang relevan dengan dengan pengguna dengan menerapkan metode *Cosine Similarity*. Cara kerja metode *Cosine Similarity* adalah menghitung kemiripan antara jurnal ilmiah dan kata kunci yang dimasukkan pengguna. Penelitian ini juga akan menerapkan algoritma *TF-IDF* untuk pembobotan kata dan menerapkan *text-preprocessing* sehingga data yang dihasilkan sebelum proses perhitungan *Cosine Similarity* lebih terstruktur. Hasil penelitian ini berupa sistem pencarian jurnal ilmiah unisbank berbasis web dengan uji coba sistem pada 30 jurnal ilmiah Unisbank menghasilkan sistem dapat memberikan rata-rata nilai *recall* 0,4 dan *precision* 0,8 pada ambang batas 0,05.

**Keywords:** Sistem Temu Kembali Informasi, Cosine Similarity, Jurnal Ilmiah

This work is licensed under Creative Commons Attribution License 4.0 CC-BY International license



### PENDAHULUAN

Dewasa ini, informasi tumbuh dengan cepat dalam berbagai basis konten seperti text, video, audio, gambar, dan sebagainya. Jumlah informasi semakin bertambah dan beragam, terutama karena semakin mudahnya memperoleh informasi melalui internet. Informasi tidak ada gunanya apabila informasi tersebut tidak dapat dikembalikan untuk memenuhi kebutuhan informasi penggunaannya. Hal ini meningkatkan kebutuhan pengguna akan alat pencarian informasi yang efisien dan efektif dalam menghasilkan informasi yang paling relevan dengan pencarian pengguna.

Perangkat pencarian informasi yang saat ini sudah sering digunakan dalam pencarian adalah *Information Retrieval System* atau bisa disebut dengan Sistem Temu Kembali Informasi, merupakan alat pencari dokumen menggunakan kumpulan teks yang tidak terorganisir dari kumpulan database yang di simpan pada komputer [1]. Tujuan *Information Retrieval System* adalah untuk menemukan dokumen yang berisi dengan informasi relevan dengan kata kunci yang dimasukkan pengguna. Perbedaan *Information Retrieval* dengan pencarian data adalah, *Information Retrieval* terutama yang berkaitan

dengan pencarian informasi yang tidak terstruktur dan kemudian menemukan informasi yang terkait dengan query [2].

Jurnal ilmiah memiliki peran yang sangat penting untuk Mahasiswa. Peran jurnal ilmiah sangat penting sebagai sumber pengetahuan dan informasi, dan sebagai media komunikasi untuk publikasi penelitian dll. Di lingkungan kampus, Jurnal ilmiah sangat membantu bagi Mahasiswa yang sedang menyelesaikan tugas akhir sebagai referensi untuk tugas akhir mereka. Namun untuk melakukan pencarian Jurnal ilmiah yang relevan sesuai keinginan dengan cepat, tepat, dan efisien pastinya sangat sulit dilakukan, karena jumlah Jurnal ilmiah yang ada saat ini sangat banyak. Pencarian jurnal ilmiah secara manual membutuhkan banyak waktu dan tenaga, sehingga apabila Mahasiswa ingin mendapatkan jurnal ilmiah yang relevan secepatnya, Mahasiswa akan mencari jurnal ilmiah hanya berdasarkan kemiripan judul pada jurnal. Akibatnya jurnal ilmiah yang didapatkan tidak relevan dengan harapan Mahasiswa.

Berdasarkan permasalahan di atas, diperlukan adanya suatu sistem yang dapat mencari jurnal ilmiah yang relevan dengan keinginan menggunakan teknologi Sistem Temu Kembali Informasi. Sistem Temu Kembali Informasi adalah suatu sistem yang secara otomatis mencari informasi yang relevan dengan kebutuhan pengguna dari kumpulan informasi yang umumnya tersimpan dalam database komputer. Tidak seperti pencarian menggunakan database yang hanya dapat mencari judul sesuai dengan query yang dimasukkan, misalnya pencarian Jurnal Ilmiah dengan kata kunci "Sistem Temu Kembali Informasi", pencarian pada database akan menampilkan semua Jurnal Ilmiah yang mengandung kata "Sistem Temu Kembali Informasi", tetapi sistem tidak bisa mengukur Jurnal Ilmiah mana yang paling mirip dengan query (kata kunci). Digunakan Algoritma *Term Frequency Inverse Document Frequency (TF-IDF)* dan Metode *Cosine Similarity* pada penelitian ini. Metode *Cosine Similarity* digunakan untuk menghitung tingkat kesamaan antar dokumen Sedangkan pembobotan *TF-IDF* digunakan untuk mengetahui bobot setiap kata pada dokumen[3].

Adapun penerapan metode *cosine similarity* pada penelitian sebelumnya yang telah menghasilkan berbagai sistem yang inovatif, sebagai contoh penelitian yang dilakukan oleh Muhtar Yusuf dan Anis Charid[4] menghasilkan sebuah sistem yang dapat memberikan rekomendasi seminar kepada pengguna. Penelitian lain yang dilakukan oleh Wahyuni, dkk [5] menghasilkan sistem yang mampu mengklasifikasikan dokumen skripsi secara

otomatis ke dalam folder yang berbeda berdasarkan kategori.

Perbedaan penelitian ini dengan penelitian sebelumnya terletak pada proses *text-preprocessing*, dimana pada penelitian ini akan dilakukan *stemming* pada kata untuk menghasilkan kata dasar yang tepat dan sesuai kamus bahasa, dimana proses tersebut berguna untuk mendapatkan informasi mengenai jumlah kata yang muncul pada suatu dokumen sehingga menambah jumlah dokumen relevan terambil.

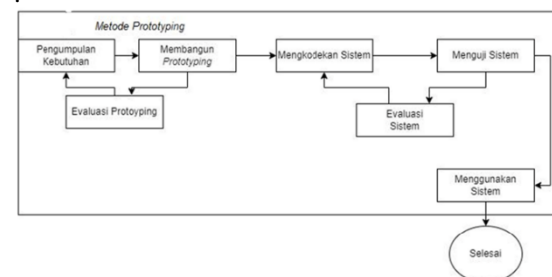
## METODE PENELITIAN

### 2.1. Metode Pengumpulan Data

Data dan informasi yang relevan diperlukan pada penelitian ini, selanjutnya akan dijadikan acuan untuk melakukan penelitian dan mendukung pembahasan dalam laporan. Metode pengumpulan data pada penelitian ini menggunakan metode studi kepustakaan dengan cara pengumpulan data jurnal ilmiah yang berkaitan dengan penelitian di internet dan pencarian langsung jurnal ilmiah di perpustakaan Unisbank Semarang.

### 2.2 Metode Pengembangan Sistem

Metode *prototype* dipilih dengan alasan metode pengembangan sistem *prototype* menggunakan *prototype* untuk menggambarkan sistem sehingga ada gambaran jelas pada sistem yang akan dibangun. Gambar 1 menggambarkan alur dari tahapan *prototype* pada penelitian ini[6].



Gambar 1. Tahap *prototype*

#### A. Pengumpulan Kebutuhan

Tahap ini mengidentifikasi kebutuhan dalam pembuatan sistem temu kembali informasi untuk pencarian jurnal ilmiah Unisbank dengan metode cosine similarity diantaranya kebutuhan *software* dan *hardware*.

#### B. Membangun *Prototyping*

Membangun *Prototyping* adalah tahap di mana perencanaan sistem dilakukan dengan Unified Modeling Language (UML).

#### C. Evaluasi *Prototyping*

Evaluasi *prototyping* adalah tahap dilakukan pengecekan terhadap model *prototyping* sementara yang dibuat pada fase sebelumnya untuk memastikan sistem yang telah didesain dapat memenuhi tujuan. Jika masih belum

sesuai akan dievaluasi serta perbaikan dengan kembali ke tahap sebelumnya.

#### D. Mengkodekan Sistem

Prototyping yang telah dibuat sesuai dengan tujuan kemudian diterjemahkan ke dalam bahasa pemrograman PHP (*Hypertext Preprocessor*) yang diintegrasikan menggunakan database *MySQL*.

#### E. Menguji Sistem

*Prototyping* yang sudah diterjemahkan dan telah menjadi software akan diuji terlebih dahulu untuk menentukan apakah *software* sudah layak digunakan atau tidak. Pengujian ini diantaranya dengan menguji nilai *recall* dan *precision*.

#### F. Evaluasi Sistem

Pada tahap ini memverifikasi apakah software atau sistem telah dirancang sebagaimana mestinya. Jika demikian, maka sistem sudah dapat digunakan, namun jika belum pengembangan harus dikembalikan ke tahap sebelumnya untuk memperbaiki ketidaksesuaian.

#### G. Menggunakan Sistem

Sistem yang telah lolos pada tahap evaluasi, maka sistem dapat digunakan.

### 2.3 Sistem Temu Kembali Informasi

Sistem temu kembali informasi adalah ilmu komputer tentang pengambilan informasi dan dokumen berdasarkan konteks dan isinya (biasanya berupa dokumen) dari kumpulan dokumen yang ada berdasarkan kata kunci yang bertujuan dapat memuaskan kebutuhan pengguna. [7][8].

Prinsip kerja sistem temu kembali informasi adalah terdapat kumpulan dokumen dan pengguna melakukan permintaan. Permintaan ini dijawab dengan mengumpulkan dokumen terkait dan menghilangkan dokumen yang tidak relevan[7].

### 2.4 Text-Preprocessing

*Text-preprocessing* biasa disebut dengan proses pengindeksan merupakan tahap awal dalam proses penyajian kumpulan dokumen ke beberapa bentuk sehingga proses penemuan dan pencarian dokumen lebih mudah dan cepat. Mengembangkan indeks dari koleksi dokumen adalah tugas utama dalam tahapan preprocessing di sistem temu kembali informasi[9].

*Text preprocessing* memiliki tahapan dimana sistem menyeleksi data yang akan diproses pada setiap dokumen. Proses *text-preprocessing* meliputi[9] :

#### 1. Case Folding

Gambar 2 menjelaskan *Case folding* adalah proses mengubah huruf kapital dari kata menjadi huruf kecil[10] dan juga menghapus

karakter non-huruf seperti tanda baca dan angka.



Gambar 2. Case Folding

#### 2. Tokenizing

Gambar 3 menjelaskan proses *tokenizing* yang dimana kalimat pada dokumen akan dipecah menjadi unsur terkecil yang disebut dengan token[11].



Gambar 3. Tokenizing

#### 3. Filtering

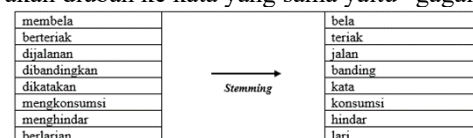
Gambar 4 menjelaskan proses *filtering* menghilangkan beberapa kata yang tidak penting pada kalimat dengan menggunakan metode stopwords. Kata yang dihilangkan adalah kata ganti, kata sambung, dan lain-lain yang disimpan pada stoplist. Contoh kata-kata tersebut adalah “yang”, “jika”, “pun”, “maka”, “ada”, “namun”, “kemudian”, “seperti”, “tersebut”, dan lain-lain [12].



Gambar 4. Filtering

#### 4. Stemming

Gambar 5 menjelaskan proses *stemming* pencarian kata dasar dari hasil dari proses filtering. Pencarian kata dasar dilakukan dengan menghilangkan imbuhan baik awalan, akhiran, sisipan, dan kombinasi awalan-akhiran[13]. Contohnya adalah kata “menggagalkan”, “kegagalan”, “digagalkan” akan diubah ke kata yang sama yaitu “gagal”.



Gambar 5. Stemming

### 2.5. Pembobotan TF-IDF (*Term Frequency Inverse Document Frequency*)

Metode *TF-IDF* merupakan teknik pembobotan suatu kata (term) dalam suatu dokumen. Dua ide perhitungan bobot digabungkan pada metode ini, yaitu banyaknya kemunculan kata pada suatu dokumen dan membalikkan frekuensi dokumen yang terkait kata tersebut. Banyaknya kata pada file dokumen yang disediakan menunjukkan berapa banyak kata-kata penting yang terkandung pada dokumen tersebut. Frekuensi dokumen yang mengandung istilah menunjukkan seberapa sering kata tersebut muncul. Hubungan

antar kata dan dokumen akan menghasilkan bobot tinggi jika suatu kata mempunyai kemunculan yang tinggi pada dokumen dan dokumen yang mengandung kata tersebut pada kelompok dokumen mempunyai frekuensi yang rendah[14]. TF-IDF dirumuskan sebagai berikut:

$$IDF = \log \left( \frac{D}{df_j} \right) \quad (1)$$

$$W_{d,t} = tf_{d,t} \times IDF_{d,t} \quad (2)$$

### 2.6. Cosine Similarity

Satu diantara banyak metode yang dapat digunakan sebagai metode pencarian data pada data mining dan sering digunakan dalam deteksi sekumpulan dokumen sejenis adalah *Cosine Similarity*. *Cosine similarity* akan menghitung kesamaan antara dua lebih dari objek yang dinyatakan dalam vektor jumlahnya ada dua vektor dengan menggunakan *query* (cosine). Jadi *cosine similarity* memungkinkan untuk menemukan kesamaan antar dokumen dalam kumpulan data besar dan menemukan dokumen yang dicari dengan lebih cepat dan akurat[15]. Metode *cosine similarity* dirumuskan sebagai berikut:

$$\cos(\theta) = \frac{Q \cdot D}{|Q||D|} = \frac{\sum_{i=1}^n (wq_i \times wd_{ij})}{\sqrt{\sum_{i=1}^n (wq_i)^2} \times \sqrt{\sum_{i=1}^n (wd_{ij})^2}} \quad (3)$$

## HASIL DAN PEMBAHASAN

### 3.1. Analisis Sistem

Analisis sistem merupakan fase yang mengkaji sistem yang ada saat ini dan mengimplementasikan berbagai kebutuhan sistem dari *stakeholder* untuk memberikan dasar dalam membangun sistem baru[16].

Dalam perancangan sistem ini, akan terbagi pada tiga proses yaitu proses text preprocessing pada dokumen jurnal dan *query*, proses pembobotan kata dengan algoritma *TF-IDF* dan proses perangkikan dokumen jurnal menggunakan metode *Cosine Similarity*. Penjelasan proses diatas sebagai berikut :

#### 1. Text Preprocessing

Proses ini dilakukan untuk membuat data teks menjadi lebih terstruktur. Pada *text-preprocessing* mengharuskan menyimpan seluruh data dokumen jurnal dan *query* yang akan diproses ke dalam database *MySQL*.

Tabel 1. Tabel Dokumen

Dokumen ID	Dokumen
query	Metode cosine similarity

Implementasi Cosine Similarity untuk Peningkatan Akurasi Pengukuran Kesamaan Dokumen pada Klasifikasi Dokumen Berita dengan K Nearest Neighbour

**D1** Tujuan penelitian ini adalah mengimplementasikan Cosine Similarity untuk meningkatkan akurasi pengukuran kesamaan dokumen pada klasifikasi dokumen berita dengan metode K-Nearest Neighbor

PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK TEXT MINING

**D2** Algoritma Term Frequency Inverse-Document Frequency merupakan suatu algoritma yang menggalkan antara Term frequency dengan Inverse Document Frequency.

Langkah awal dari *text-preprocessing* adalah *Case Folding*, dimana proses ini akan menghilangkan tanda baca pada dokumen dan mengubah huruf dokumen menjadi kecil. Hasil proses *Case Folding* terhadap dokumen pada tabel 1 ditunjukkan pada tabel 2.

Tabel 2. Hasil Case Folding

Case Folding	
Dokumen ID	Dokumen
query	metode cosine similarity cosine similarity untuk peningkatan akurasi pengukuran kesamaan dokumen pada klasifikasi dokumen berita dengan k nearest neighbour
<b>D1</b>	tujuan penelitian ini adalah mengimplementasikan cosine similarity untuk meningkatkan akurasi pengukuran kesamaan dokumen pada klasifikasi dokumen berita dengan metode k nearest neighbor
	penerapan algoritma term frequency inverse document frequency tf idf untuk text mining
<b>D2</b>	algoritma term frequency inverse document frequency merupakan suatu algoritma yang menggalkan antara term frequency dengan inverse document frequency

Langkah selanjutnya adalah *Tokenizing*. Pada tahap ini, kalimat pada dokumen akan dipecah menjadi kata. Hasil *tokenizing* terhadap dokumen yang telah diproses *case folding* pada tabel 2 ditunjukkan pada tabel 3.

Tabel 3. Tokenizing

Tokenizing	
Dokumen ID	Kata
query	penerapan
query	cosine
query	similarity
d1	cosine
d1	similarity
d1	untuk
d1	peningkatan
d1	akurasi
d1	pengukuran



d1	kesamaan
d1	dokumen
d1	pada
d1	klasifikasi
d1	dokumen
d1	berita
d1	dengan
d1	k
...	...
d2	frequency

Langkah selanjutnya adalah *filtering*. Pada tahap ini, kata yang masuk pada daftar *stopword* akan dihilangkan. Hasil *filtering* terhadap dokumen yang telah diproses *tokenizing* pada tabel 3 ditunjukkan pada tabel 4.

**Tabel 4. Filtering**

Dokumen ID	Kata
query	penerapan
query	cosine
query	similarity
d1	cosine
d1	similarity
d1	peningkatan
d1	akurasi
d1	pengukuran
d1	kesamaan
d1	dokumen
d1	klasifikasi
d1	dokumen
d1	berita
d1	k
...	...
d2	frequency

Langkah terakhir adalah *stemming*. Pada tahap ini, kata akan dikembalikan ke dalam bentuk kata dasarnya. Hasil *stemming* terhadap dokumen yang telah diproses *filtering* pada tabel 4 ditunjukkan pada tabel 5.

**Tabel 5. Stemming**

Dokumen ID	Kata	Stemming
query	penerapan	terap
query	cosine	cosine
query	similarity	similarity
d1	cosine	cosine
d1	similarity	similarity
d1	peningkatan	tingkat
d1	akurasi	akurasi
d1	pengukuran	ukur
d1	kesamaan	sama
d1	dokumen	dokumen
d1	klasifikasi	klasifikasi
d1	dokumen	dokumen
d1	berita	berita
d1	k	k
...	...	...
d2	frequency	frequency

## 2. Indexing

Indexing merupakan tahapan untuk menghitung bobot menggunakan metode TF-IDF untuk setiap kata, dimana setiap kata hasil dari proses *text preprocessing* dilakukan perhitungan untuk

mengetahui jumlah kemunculan kata (TF) pada tiap dokumen. Kemudian menghitung jumlah dokumen yang mengandung kata tersebut (DF), dan selanjutnya menghitung nilai IDF dengan persamaan (1). Hasil perhitungan *IDF* ditunjukkan pada tabel 6.

**Tabel 6. IDF**

Kata	TF			df	idf
	q	d1	d2		
implementasi		1		1	0,4771
cosine	1	1		2	0,1761
Similarity	1	1		2	0,1761
tingkat		2		1	0,4771
akurasi		2		1	0,4771
ukur		2		1	0,4771
sama		2		1	0,4771
dokumen		4		1	0,4771
klasifikasi		2		1	0,4771
berita		2		1	0,4771
K		2		1	0,4771
nearest		2		1	0,4771
neighbour		2		1	0,4771
tuju		1		1	0,4771
teliti		1		1	0,4771
metode		1		1	0,4771
terap	1		1	2	0,1761
algoritma			3	1	0,4771
term			3	1	0,4771
frequency			6	1	0,4771
inverse			3	1	0,4771
document			3	1	0,4771
Tf			1	1	0,4771
idf			1	1	0,4771
text			1	1	0,4771
mining			1	1	0,4771
gali			1	1	0,4771

Setelah mendapat nilai *IDF*, selanjutnya adalah melakukan pembobotan *TF-IDF* dengan persamaan (2). Hasil *TF-IDF* ditunjukkan pada tabel 6.

**Tabel 6. TFxIDF**

Kata	Tf x idf		
	q	d1	d2
implementasi	0	0,4771	0
cosine	0,1761	0,1761	0
Similarity	0,1761	0,1761	0
tingkat	0	0,9542	0
akurasi	0	0,9542	0
ukur	0	0,9542	0
sama	0	0,9542	0
dokumen	0	1,9085	0
klasifikasi	0	0,9542	0
berita	0	0,9542	0
K	0	0,9542	0
nearest	0	0,9542	0
neighbour	0	0,9542	0
tuju	0	0,4771	0
teliti	0	0,4771	0
metode	0	0,4771	0
terap	0,1761	0	0,1761
algoritma	0	0	1,4314
term	0	0	1,4314
frequency	0	0	2,8627
inverse	0	0	1,4314
document	0	0	1,4314



			14			88
tf	0	0	0,47	0	0	0,22
			71			76
idf	0	0	0,47	0	0	0,22
			71			76
text	0	0	0,47	0	0	0,22
			71			76
mining	0	0	0,47	0	0	0,22
			71			76
gali	0	0	0,47	0	0	0,22
			71			76

Cross product (query) = 0 + 0,0310 + 0,0310 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0,0310 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 0,0930

Cross product (d1) = 0,2276 + 0,0310 + 0,0310 + 0,9106 + 0,9106 + 0,9106 + 0,9106 + 3,6432 + 0,9106 + 0,9106 + 0,9106 + 0,9106 + 0,9106 + 0,2276 + 0,2276 + 0,2276 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 12,8101

Cross product (d2) = 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0,0310 + 2,0488 + 2,0488 + 8,1952 + 2,0488 + 2,0488 + 0,2276 + 0,2276 + 0,2276 + 0,2276 = 17,5596

Setelah nilai *dot product* dan *cross product* diketahui, selanjutnya adalah mengimplementasikan perhitungan *cosine similarity* pada persamaan (3).

$$\begin{aligned} \text{Cos}(Q, D1) &= \frac{0,0620}{\sqrt{0,0930} \sqrt{12,8101}} = \frac{0,0620}{0,3049 \cdot 3,5791} \\ &= \frac{0,0620}{1,0912} = 0,0568 \end{aligned}$$

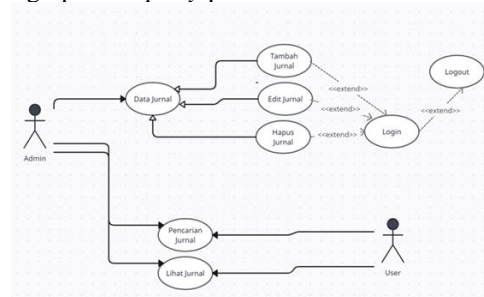
$$\begin{aligned} \text{Cos}(Q, D2) &= \frac{0,0310}{\sqrt{0,0930} \sqrt{17,5596}} = \frac{0,0310}{0,3049 \cdot 4,1904} \\ &= \frac{0,0310}{1,2776} = 0,0242 \end{aligned}$$

Berdasarkan hasil dari perhitungan *cosine similarity* diatas, hasil kemiripan dokumen d1 dan query adalah 0,0568 dan hasil kemiripan dokumen d2 dan query adalah 0,0242. Maka disimpulkan dokumen dengan kemiripan terdekat dengan query adalah d1. Karena hukum *cosine similarity* semakin nilai *cosine similarity* mendekati 1 maka dokumen semakin mirip dengan query.

### 3.2. Perancangan UML

Perancangan UML membantu pengembang sistem perangkat lunak dalam mendefinisikan, memvisualisasikan, dan mendokumentasikan berbagai artefak sistem perangkat lunak[17][18]. UML mendefinisikan bermacam-macam diagram, salah satunya adalah *use-case* diagram yang digunakan pada penelitian ini. Dengan menentukan langkah-langkah yang diperlukan dari suatu sistem, *use-case* memainkan peran penting dalam memfasilitasi interaksi pengguna dengan sistem. [19].

Penjelasan jalannya sistem pada penelitian digambarkan dengan *use-case* diagram yang ditunjukkan pada gambar 2, mendefinisikan dua actor yaitu admin dan user. Interaksi admin pada sistem ini mewajibkan admin login. Setelah login, admin dapat melakukan proses diantaranya menambah jurnal, menghapus jurnal, dan mengedit metadata jurnal. Sedangkan aktor user dapat melakukan operasi pencarian jurnal dengan menginputkan *query* pada sistem.

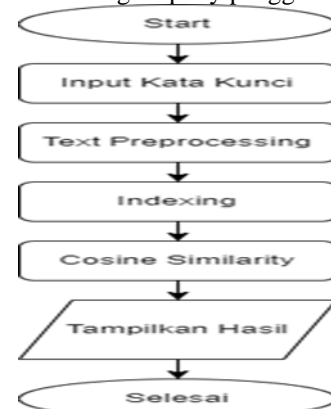


Gambar 2. Use Case Sistem Berjalan

### 3.3. Perancangan Flowchart

*Flowchart* berguna untuk menggambarkan alur kerja dari proses sistem temu kembali informasi dan menggambarkan hubungan antara satu proses dengan lainnya secara detail[20].

Penjelasan prosedur sistem temu kembali informasi dapat digambarkan dengan menggunakan *flowchart* yang dapat dilihat pada gambar 3. Proses dimulai dengan memasukkan kata kunci oleh user, kemudian dilakukan proses *text preprocessing* pada jurnal hingga menghasilkan kata dasar yang akan diberikan bobot pada proses indexing, diteruskan dengan menghitung kesamaan similaritas antara kata kunci dan jurnal hingga menampilkan daftar jurnal yang terkait dengan query pengguna.



Gambar 3. Flowchart Proses STKI

### 3.4. Perancangan Database

Database pada sistem ini dibangun dengan nama stki yang didalamnya terdapat beberapa tabel, diantaranya *tb\_jurnal*, *jurnal\_indexing*, *result\_cosine\_similarity*. Adapun rancangan dijelaskan pada berikut :

1. Tabel tb\_jurnal

Tabel ini berfungsi untuk menyimpan koleksi jurnal ilmiah unisbank. Tabel 1 menjelaskan bahwa id pada tabel berfungsi untuk identitas dari jurnal, title digunakan untuk menyimpan judul jurnal, pengarang digunakan untuk menyimpan Nama pengarang jurnal, tahun\_terbit digunakan untuk menyimpan tahun penerbitan jurnal, abstrak digunakan untuk menyimpan abstrak pada jurnal dan kd\_jurnal digunakan untuk menyimpan kode jurnal.

Tabel 9. Tabel tb\_jurnal

tb_jurnal		
Nama Field	Tipe Data	size
id	Int	11
Title	Varchar	255
pengarang	Varchar	255
tahun_terbit	Date	-
abstrak	Text	-
link	text	-
kd_jurnal	Varchar	255

2. Tabel Jurnal\_indexing

Tabel jurnal\_indexing berguna untuk menyimpan data hasil dari preprocessing dan indexing. Dari table 2 dapat diketahui bahwa id digunakan untuk menyimpan nilai unik term, jurnal\_id digunakan untuk menyimpan identitas dari jurnal, term digunakan untuk menyimpan kata hasil proses text-preprocessing, jumlah digunakan untuk menyimpan frekuensi kemunculan kata pada jurnal, dan bobot digunakan untuk menyimpan bobot setiap term dari hasil perhitungan TF-IDF.

Tabel 10. Tabel jurnal\_indexing

Jurnal_indexing		
Nama Field	Tipe Data	Size
id	int	11
jurnal_id	int	11
term	text	-
Jumlah	int	11
Bobot	float	-

3. Tabel result\_cosine\_similarity

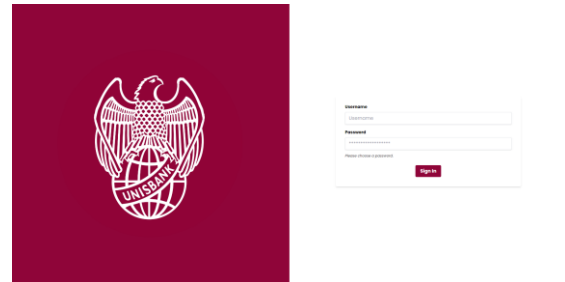
Tabel result\_cosine\_similarity berguna untuk menyimpan data hasil dari perhitungan cosine similarity. Dari tabel 3 dapat diketahui bahwa id digunakan untuk menyimpan nilai unik setiap hasil cosine similarity, jurnal\_id digunakan untuk menyimpan identitas jurnal, cosine\_similarity digunakan untuk menyimpan nilai cosine similarity.

Tabel 11. Tabel result\_cosine\_similarity

Result_cosine_similarity		
Nama Field	Tipe Data	size
id	Int	11
jurnal_id	Int	11
cosine_similarity	Float	-

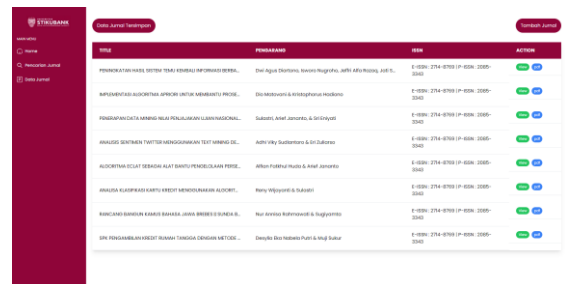
3.5. Implementasi Sistem

Interface berikut adalah hasil rancangan yang telah dibuat.

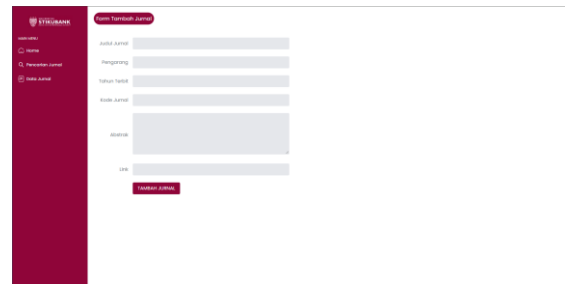


Gambar 4. Halaman Login

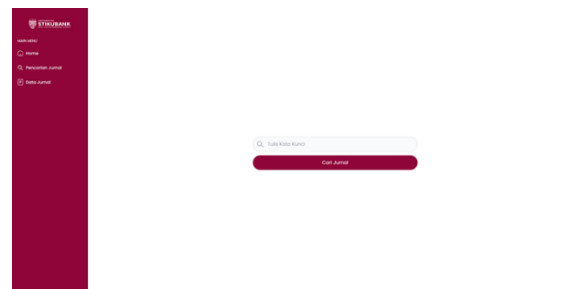
Halaman login pada gambar 4 merupakan halaman admin masuk ke dalam sistem dengan memasukkan username dan kata sandi. jika data yang dimasukkan benar, admin akan di arahkan pada halaman kelola jurnal pada gambar 5 yang berfungsi untuk admin mengelola jurnal yang ada pada sistem. Admin dapat menambah list jurnal dengan menekan tombol “tambah jurnal” yang nanti akan diarahkan pada halaman form tambah jurnal seperti gambar 6.



Gambar 5. Halaman Admin Kelola Jurnal



Gambar 6. Halaman Admin Tambah Jurnal

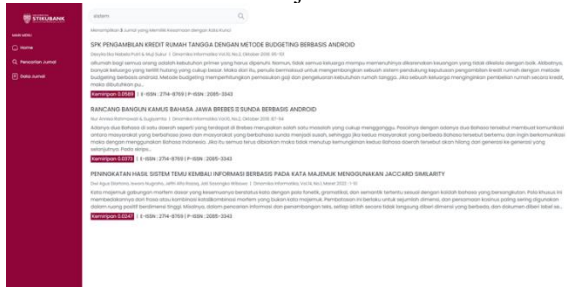


Gambar 7. Halaman Pencarian Jurnal

Halaman pencarian pada gambar 7 dapat di akses tanpa login yang artinya semua pengguna dapat



mengakses halaman ini. Untuk melakukan pencarian jurnal, pengguna hanya perlu memasukkan kata kunci pada form yang tersedia lalu menekan tombol “cari jurnal”.



Gambar 8. Halaman hasil pencarian

Setelah sistem memproses inputan kata kunci dari pengguna, sistem akan menampilkan data jurnal ilmiah Unisbank yang telah selesai diproses menggunakan metode *Cosine Similarity*, sehingga list jurnal yang ditampilkan relevan dengan keinginan pengguna. Gambar 8 menunjukkan hasil pencarian pengguna dengan kata kunci “sistem” menampilkan 3 jurnal yang relevan dimana jurnal dengan judul “Sistem Pengambilan Kredit Rumah Tangga Dengan Metode Budgeting Berbasis Android” ditampilkan paling atas karena memiliki nilai kemiripan paling tinggi.

### 3.6. Pengujian

Sebanyak 30 jurnal ilmiah unisbank dari berbagai prodi digunakan pada pengujian. Berikut merupakan hasil pengujian dari penelitian ini :

#### 1. Pengujian Waktu *Indexing*

Pengujian dilakukan untuk melihat seberapa lama sistem mampu memproses waktu *indexing*. Pengujian ini dilakukan dengan menghitung waktu proses index setiap 5 jurnal. pada tabel 4 menjelaskan waktu proses *indexing* semakin meningkat tergantung banyaknya jumlah jurnal dan banyaknya kata yang di index. Software dan hardware yang berbeda memungkinkan mendapat hasil yang berbeda.

Tabel 4. Pengujian Waktu Indexing

Pengujian Waktu Indexing		
Banyaknya Jurnal	Waktu / Detik	Banyaknya Kata
5	10 detik	876
10	21 detik	1793
15	37 detik	2838
20	50 detik	3735
25	61 detik	4471

#### 2. Pengujian *Text Preprocessing*

Pengujian *text-preprocessing* dilakukan untuk mengetahui ketepatan sistem dalam mencari kata dasar setiap term setelah melewati proses *case folding*, *tokenizing*, *filtering*, dan *stemming*. Dapat dilihat pada tabel 5, hasil *text-preprocessing* terbilang

cukup baik. Term dengan awalan “di-” dan akhiran “-kan” pada term “dimanfaatkan” mampu ditemukan kata dasarnya yaitu kata manfaat. Term panjang dengan awalan “memper-” dan akhiran “-kannya” pada term “memperkenalkannya” pun dapat ditemukan kata dasarnya yaitu “kenal”.

Tabel 12. Pengujian *Text-Preprocessing*

Pengujian Text-Preprocessing	
TERM	Hasil <i>text-preprocessing</i>
Menghasilkan	Hasil
Kemiripan	Mirip
Mengandung	Kandung
Penambahan	Tambah
Membantu	Bantu
Dimanfaatkan	Manfaat
Memperkenalkannya	Kenal

#### 3. Pengujian *Indexing*

Pengujian *indexing* adalah pemberian bobot tiap kata pada jurnal dimana perhitungan pembobotan menggunakan algoritma TF-IDF. Hasil dari proses *indexing* ditunjukkan tabel 13.

Tabel 13. Pengujian TF-IDF

TERM	Pengujian TF-IDF		
	Jurnal_ID	TF	TF-IDF
status	1	1	1,0142
	6	1	1,0142
	20	1	1,0142
pola	1	2	2,0285
	2	1	1,0142
	22	1	1,0142
basis	1	1	0,5371
	7	2	1,0742
	8	2	1,0742
	9	1	0,5371
	13	2	1,0742
	15	1	0,5371
	20	1	0,5371
28	4	2,1485	
29	1	0,5371	

Keterangan, TERM = Kata, Jurnal\_ID = id jurnal, TF= Frekuensi Kemunculan Kata, TF-IDF = Bobot.

#### 4. Pengujian *Retrieval*

Pengujian *retrieval* adalah proses untuk mencari jurnal ilmiah yang terkait berdasarkan query. Pada tahap pengujian dilakukan, kata kunci yang dimasukkan berjumlah 1 sampai dengan 3 term. Tabel 14 menunjukkan hasil pengujian.

Tabel 14. Pengujian *Retrieval*

Jumlah Term	Kata Kunci	Pengujian Retrieval	
		Jurnal ID	Kemiripan
1 term	Model	18	0,2549
		7	0,0605
		17	0,0487
2 term	Sistem Digital	20	0,0715
		14	0,0464
		8	0,0406
		27	0,0367

	28	0,0356	
	13	0,0338	
	24	0,032	
	17	0,0312	
	7	0,022	
	1	0,0159	
	26	0,0133	
	22	0,2292	
	9	0,1361	
	18	0,1104	
	25	0,0949	
	27	0,0501	
	17	0,0491	
	14	0,0262	
	21	0,0254	
3 term	Peningkatan Kemampuan Mahasiswa	12	0,0249
		26	0,0232
		30	0,0184
		28	0,0183
		23	0,0154
		6	0,015
		10	0,0117
		29	0,0107
		1	0,0044

Keterangan, Kata Kunci = kata kunci, jurnal id = id jurnal, Kemiripan = nilai kemiripan antara jurnal dengan kata kunci

#### 5. Pengujian Recall dan Precision

Dilakukan pengujian sebanyak 5 kali dengan query berbeda-beda. Pengujian dilakukan dengan nilai batas ambang 0.05 yang apabila nilai kemiripan jurnal yang ditemukan diatas atau sama dengan 0,05, maka jurnal dinyatakan relevan. Sebaliknya, jika nilai kemiripannya kurang dari 0,05 maka jurnal yang ditemukan dianggap tidak relevan. Hasil pengujian recall dan precision ditunjukkan pada tabel 15.

Tabel 15. Pengujian Recall dan Precision

Pengujian Recall dan Precision					
Kata kunci	d1	d2	n	r	p
Rancang Bangun	3	7	3	1	0,43
Augmented Reality	1	1	1	1	1
kewirausahaan	2	2	2	1	1
Rekayasa Perangkat Lunak	2	6	2	1	0,3
Pengembangan Metode Otentikasi Keaslian Ijasah	1	24	1	1	0,04
Mahasiswa Unisbank	4	12	5	0,8	0,3
Search Engine	2	2	2	1	1
Implementasi Akuntansi Dalam Pengelolaan Persediaan Barang Dagangan	3	12	3	1	0,25
Pelatihan Publikasi Ilmiah	5	10	6	0,83	0,5
	1	1	1	1	1

Keterangan, d1 = Jumlah Jurnal Relevan Ditemukan, d2 = Jumlah Jurnal Ditemukan, N = Jumlah Jurnal Relevan dalam koleksi, r = recall, p = precision

Dapat dilihat pada tabel 15, nilai rata-rata untuk recall adalah 0,8 dan nilai precision rata-rata adalah 0,4.

#### SIMPULAN

Berdasarkan hasil penelitian yang dilakukan, Sistem temu kembali informasi pencarian jurnal ilmiah Unisbank berhasil dibangun dengan mengimplementasikan metode *cosine similarity*. Sistem yang dibuat sudah bisa dikatakan efektif, hal ini dibuktikan dengan *recall* pada pengujian mendapat nilai rata rata 0,8 yang artinya hampir seluruh jurnal ilmiah yang ada pada sistem dapat ditemukan. Sedangkan nilai precision memiliki nilai yang hampir selalu berbeda untuk setiap pengujian. Hal ini dikarenakan nilai *precision* bergantung seberapa unik query yang di uji.

#### UCAPAN TERIMAKASIH

Terima kasih disampaikan kepada pihak Universitas Stikubank Semarang sebagai tempat pelaksanaan penelitian.

#### DAFTAR PUSTAKA

- [1] Yusrawati, "Strategi pengembangan sistem temu kembali informasiberbasis image di perpustakaan perguruan tinggi," *LIBRIA*, Vol.9(1), 53-68, Juni, 2017, <https://jurnal.ar-raniry.ac.id/index.php/libria/article/download/1688/1226>
- [2] Putung, K. D., Arie, S. M. L., & Agustinus, J., "Penerapan Sistem Temu Kembali Informasi Pada Kumpulan Dokumen Skripsi," *E-Journal Teknik Informatika*, Vol 8(1), 18-21, Mei, 2016, <https://ejournal.unsrat.ac.id/index.php/informatika/article/view/12227>
- [3] Apriani, Hizbu Z., & Khairan M. "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF System Penerimaan Mahasiswa Baru pada Kampus Swasta," *Jurnal Burnigora Information Technology (BITe)*, Vol.3(1), 19-27, Juni, 2017, <https://journal.universitasbumigora.ac.id/index.php/bit/article/view/1110>
- [4] Yusuf, Muhtar, & Anis, C. "Implementasi Algoritma Cosine Similarity Dan Metode TF-IDF Berbasis PHP untuk Menghasilkan Rekomendasi Seminar," *Jurnal Ilmiah Fakultas Ilmu Komputer*, Vol 9(1), 8-16, 2020, <https://publikasi.mercubuana.ac.id/index.php/fasilkom/article/download/8830/3555>
- [5] Wahyuni, T., R., Dhidik, P., & Eko, S. "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi", *Jurnal Teknik Elektro*, Vol 9(1), 18-23, Januari – Juni, 2017, <https://journal.unnes.ac.id/nju/jte/article/view/10955/1000>
- [6] Dewi, N. L. A. M. R., Rukmi, S. H., & Yoga, D., "Penerapan Metode Prototype dalam Perancangan Sistem Informasi Penerimaan Karyawan Berbasis Website pada Berlian Agency," *Majalah Ilmiah Teknologi Elektro*, Vol.20(1), 147-152, Januari –

- Juni, 2021,  
<https://www.researchgate.net/publication/368005836>  
Penerapan Metode Prototype dalam Perancangan Sistem Informasi Penerimaan Karyawan Berbasis Website pada Berlian Agency
- [7] Amin, Fatkhul. "Sistem Temu Kembali Informasi dengan Metode Vector Space Model," *Jurnal Informasi Bisnis Vol 2(2)*, 78-83, 2012, <https://ejournal.undip.ac.id/index.php/jsinbis/article/view/6124>
- [8] Melita, Ria., Victor, A., Hendra, B. S., & Taslimun, D. "Penerapan Metode Term Frequency Inverse Document Frequency(Tf-Idf) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi untuk Mengetahui Syarah Hadits Berbasis Web(Studi Kasus: Syarah Umdatil Ahkam)," *Jurnal Teknik Informatika, Vol 11(2)*, Oktober, 2018, <https://www.researchgate.net/publication/331910857>  
PENERAPAN METODE TERM FREQUENCY INVERSE DOCUMENT FREQUENCY TF-IDF DAN COSINE SIMILARITY PADA SISTEM TEMU KEMBALI INFORMASI UNTUK MENGETAHUI SYARAH HADITS BERBASIS WEB STUDI KASUS HADITS SHAHIH BUKHARI
- [9] Sanjaya, Feri., "Pemanfaatan Sistem Temu Kembali Informasi dalam Pencarian Dokumen Menggunakan Metode Vector Space Model," *Journal Of Information and Technology, Vol 05(02)*, Desember, 2017, <https://jurnal.stiki.ac.id/J-INTECH/article/view/189>
- [10] Sugara, Bayu, Dody, & Donny, "Sistem Temu Kembali Informasi Pada Gejala Autisme Dengan Metode Vector Space Model," *Jurnal RESTI, Vol.3(2)*, 257-264, 2019, <https://jurnal.iaii.or.id/index.php/RESTI/article/view/1028/152>
- [11] Arifa, A. B., Gita F. F., & Ananda R. H., "Sistem Temu Kembali Informasi pada Soal Ujian dengan Rencana Pembelajaran Menggunakan Vector Space Model," *Jurnal RESTI, Vol.5(1)*, 63-68, 2021, <https://jurnal.iaii.or.id/index.php/RESTI/article/view/2739/368>
- [12] Tala, F. Z., "A Study of Stemming Effect on Information Retrieval in bahasa indonesia," In *Institute for Logic, Language and Computation Universeit Van Amsterdam*, 2003, <https://eprints.illc.uva.nl/id/eprint/740/1/MoL-2003-02.text.pdf>
- [13] Gutters, Anita, Gunawan, & Joan S., "Stemming Bahasa Tetun Menggunakan Pendekatan Rule Based," *TEKNIKA, Vol 8(2)*, November, 2019, <https://ejournal.ikado.ac.id/index.php/teknika/article/view/224>
- [14] Nurjannah, Musfiroh, Hamdani, & Inda F. A., "Penerapan Algoritma Term Frequency-Inverse Document Frequency (Tf-Idf) Untuk Text Mining," *Jurnal Informatioa Mulawarman, Vol 8(3)*, 110-114, September, 2013, <https://ejournals.unmul.ac.id/index.php/JIM/article/view/113>
- [15] Kurniadi, Dedy, Sam F. C. H., & Andika N., "Implementasi Algoritma Cosine Similarity pada sistem arsip dokumen di Universitas Islam Sultan Agung," *TRANSFORMATIKA, Vol 17(2)*, 124-132, Januari, 2020, <https://journals.usm.ac.id/index.php/transformatika/article/view/1613>
- [16] Chautie, M. R., "Perancangan Sistem Pendukung Keputusan Pemilihan Guru Terbaik Dengan Pendekatan User Centered Design", *Jurnal Teknologi Dan Sistem Informasi Bisnis, Vol 5(2)*, 132-140, April, 2023, <https://www.researchgate.net/publication/370456942>  
Perancangan Sistem Pendukung Keputusan Pemilihan Guru Terbaik Dengan Pendekatan User Centered Design
- [17] Munawaroh, Hidaratul, Rizqi, A. W. Y., & Liris. A. N., "Pengukuran Keserupaan Pada Communication Diagram Menggunakan Metode Graph Edit Distance," *Jurnal Teknologi Dan Sistem Informasi Bisnis, Vol 5(4)*, 460-467, Oktober, 2023, <http://jurnal.unidha.ac.id/index.php/jteksis/article/view/900/644>
- [18] Binangkit, C. A., Apriade V., & Nono H., "Pemanfaatan UML (Unified Modeling Language) Dalam Perencanaan Sistem Pengelolaan Sewa Alat Musik Berbasis Website," *JATI, Vol 7(2)*, 1429-1436, April, 2023, <https://ejournal.itn.ac.id/index.php/jati/article/download/6858/4077/>
- [19] Mayola, Liga, M., Hafizh, & Deri, M. P., "Perancangan Aplikasi Similarity Deteksi Kemiripan Judul Disertasi Berbasis Web", *Jurnal Teknologi Dan Sistem Informasi Bisnis, Vol 6(2)*, 452-457, April, 2024, <https://jurnal.unidha.ac.id/index.php/jteksis/article/view/1164/773>
- [20] Hasanah, N. N., & Agus, S. P., "Implementasi Data Mining Untuk Pengelompokan Buku Menggunakan Algoritma K-Means Clustering (Studi Kasus : Perpustakaan Politeknik Lpp Yogyakarta)", *Jurnal Teknologi Dan Sistem Informasi Bisnis, Vol 4(2)*, 300-311, Juli, 2022, <https://www.neliti.com/id/publications/441596/implimentasi-data-mining-untuk-pengelompokan-buku-menggunakan-algoritma-k-means>