

## Model Rekomendasi Jurnal dengan Algoritma *Jaccard Similarity* dan Protokol *OAI-PMH*

M. Miftakul Amin<sup>\*1</sup>, Ali Firdaus<sup>2</sup>, Yevi Dwitayanti<sup>3</sup>

<sup>1,2</sup>Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya, Palembang, Indonesia

<sup>3</sup>urusan Akuntansi, Politeknik Negeri Sriwijaya, Palembang, Indonesia

Email: <sup>1</sup>[miftakul\\_a@polsri.ac.id](mailto:miftakul_a@polsri.ac.id), <sup>2</sup>[alifirdaus@polsri.ac.id](mailto:alifirdaus@polsri.ac.id), <sup>3</sup>[yevi\\_dwitayanti@polsri.ac.id](mailto:yevi_dwitayanti@polsri.ac.id)

### Abstrak

Kesulitan peneliti dalam menemukan jurnal yang sesuai untuk publikasi artikel ilmiah menjadi salah satu hambatan dalam proses diseminasi ilmu pengetahuan. Masalah ini diperburuk oleh kurangnya integrasi antara sistem manajemen jurnal seperti *Open Journal System* (OJS) dengan layanan pencarian yang efektif. Penelitian ini bertujuan untuk merancang dan mengembangkan sebuah aplikasi *journal finder* yang mampu merekomendasikan jurnal berdasarkan kecocokan antara judul artikel dan abstrak. Metode yang digunakan adalah dengan mengimplementasikan algoritma *jaccard similarity* untuk menghitung kesamaan antara judul dan abstrak artikel dari input pengguna dengan metadata jurnal yang tersimpan dalam basis data, serta protokol *Open Archives Initiative Protocol for Metadata Harvesting* (OAI-PMH) untuk mengumpulkan metadata artikel dari repositori jurnal yang berbasis OJS. Penelitian ini telah berhasil menghimpun sebanyak 59 *repository platform* OJS, dan 3.321 metadata artikel ilmiah yang disimpan dalam sebuah basis data jurnal. Hasil penelitian menunjukkan bahwa aplikasi *journal finder* berhasil merekomendasikan jurnal dengan tingkat kemiripan tertentu berdasarkan validasi data uji. Penelitian ini diharapkan dapat membantu para peneliti menemukan jurnal yang relevan secara lebih cepat dan meningkatkan efisiensi proses publikasi artikel ilmiah.

**Kata kunci:** *Open Journal System, OAI-PMH, Journal Finder*

## *Journal Recommendation Model with Jaccard Similarity Algorithm and OAI-PMH Protocol*

### Abstract

The difficulty of researchers in finding suitable journals for the publication of scientific articles is one of the obstacles in the process of disseminating knowledge. This problem is exacerbated by the lack of integration between journal management systems such as *Open Journal System* (OJS) and effective search services. This research aims to design and develop a *journal finder* application that is able to recommend journals based on matches between article titles and abstracts. The method used is to implement the *Jaccard similarity* algorithm to calculate the similarity between article titles and abstracts from user input and journal metadata stored in the database, as well as the *Open Archives Initiative Protocol for Metadata Harvesting* (OAI-PMH) protocol to collect article metadata from OJS-based journal repositories. This research has successfully collected 59 OJS platform repositories, and 3,321 scientific article metadata stored in a journal database. The results showed that the *journal finder* application successfully recommended journals with a certain level of similarity based on the validation of test data. This research is expected to help researchers find relevant journals faster and improve the efficiency of the scientific article publication process.

**Keywords:** *Open Journal System, OAI-PMH, Journal Finder.*

## 1. PENDAHULUAN

Artikel ilmiah memiliki peran penting dalam mendukung pengembangan ilmu pengetahuan, dan keberadaannya sering dikelola melalui *platform seperti Open Journal System* (OJS), sebuah sistem manajemen jurnal elektronik yang memfasilitasi proses penerbitan artikel secara terstruktur. Artikel yang diterbitkan melalui OJS selanjutnya dapat diindeks oleh *platform seperti GARUDA* (Garba Rujukan Digital), yang merupakan layanan nasional untuk mengumpulkan, mengelola, dan menyediakan referensi digital dari berbagai artikel ilmiah Indonesia. GARUDA juga terhubung dengan SINTA (*Science and Technology Index*), sebuah indeks nasional yang mengukur kinerja publikasi ilmiah berdasarkan jumlah sitasi dan kualitasnya, serta memberikan

peringkat bagi penulis, institusi, dan jurnal [1]. Dengan hubungan ini, OJS bertindak sebagai pengelola publikasi, sementara GARUDA dan SINTA menjadi alat penting untuk *aggregator*, mengindeks, menyebarluaskan, dan mengevaluasi dampak ilmiah artikel di tingkat nasional maupun internasional.

Belum adanya layanan *journal finder* yang terintegrasi menjadi tantangan bagi para peneliti dalam menemukan jurnal yang sesuai untuk publikasi karya ilmiah yang tehal dihasilkan. Tanpa menggunakan sistem rekomendasi jurnal, langkah ini akan menjadi tugas yang sangat memakan waktu [2]. Dengan jumlah jurnal nasional yang terus meningkat, baik yang telah terakreditasi atau belum, para dosen dan peneliti sering menghadapi kesulitan dalam memilih jurnal yang relevan dengan topik penelitian, memiliki reputasi yang baik, serta sesuai dengan kategori bidang ilmu. Masalah ini diperparah oleh keterbatasan akses dan kurangnya integrasi antara sistem manajemen jurnal, seperti *Open Journal System* (OJS), dengan layanan pencarian yang cerdas. Akibatnya, proses pencarian jurnal menjadi memakan waktu dan beresiko mengirimkan artikel ke jurnal yang kurang sesuai, sehingga dapat menghambat proses publikasi. Oleh karena itu, diperlukan pengembangan layanan *journal finder* yang efektif untuk membantu para dosen dan peneliti dalam menghemat waktu, meningkatkan peluang publikasi, dan mendukung pengelolaan jurnal ilmiah di Indonesia secara lebih terorganisir.

Menurut Vara et al. [3], *journal finder* merupakan salah satu implementasi dari topik *recommender system*. Aplikasi dari *recommender system* ini cukup beragam, seperti aplikasi pada rekomendasi berita [4], *e-commerce* [5], transportasi [6], *e-learning* [7][8], *internet of things* (IoT) [9], sistem informasi geografis [10], *e-tourism* [11][12], *e-health* [13], dan tentunya dapat diaplikasikan dalam domain yang lainnya. Disamping itu *recommender system* ditinjau dari pendekatan yang digunakan, terdapat beberapa metode yang dapat dijadikan sebagai algoritma untuk membangun aplikasinya seperti *collaborative filtering* [14], *knowledge based filtering* [15], *hybrid model* [16], *machine learning* [17], integrasi media sosial [18], *fuzzy logic* [19], algoritma genetika [20], K-Means dan K-NN [21], dan tentunya masih banyak metode dan pendekatan lainnya.

Priya et al. [22] menyebut bahwa terdapat fakta banyak peneliti mengalami kesulitan untuk mempublikasikan karya ilmiahnya di jurnal yang sesuai dan mengalami keterlambatan publikasi. Dengan banyaknya jurnal yang tersedia, peneliti sering menghadapi tantangan dalam memilih jurnal yang relevan dengan topik, memiliki reputasi baik, dan sesuai dengan cakupan penelitian mereka. Aplikasi ini dapat mempercepat proses pencarian dengan merekomendasikan jurnal berdasarkan kesesuaian tema, kata kunci, atau abstrak artikel, sehingga menghemat waktu dan memastikan artikel diterbitkan di jurnal yang tepat, meningkatkan dampak dan keterbacaan penelitian mereka.

Penelitian yang dilakukan oleh Wang et al. [23], telah mengembangkan sebuah sistem rekomendasi *venue* yang terdiri dari jurnal dan prosiding pada bidang ilmu komputer menggunakan *web services*. Model rekomendasi menggunakan informasi abstrak dari artikel ilmiah. Dalam membangun data *training* dan *learning* data dikumpulkan menggunakan teknologi *web crawler*, dan seleksi fitur menggunakan metode *hybrid* berupa pendekatan *chi-square* dan *softmax regression*. Penelitian tersebut telah berhasil dievaluasi dengan tingkat akurasi sebesar 61,37%. Demikian juga penelitian yang telah dilakukan oleh Magara et al. [24], telah mengembangkan sebuah model pencarian artikel ilmiah dengan menggunakan pendekatan *jaccard similarity* dan *Bisociative Information Networks (BisoNets)*. Namun penelitian yang telah dilakukan tersebut belum secara spesifik diimplementasikan dalam sebuah lingkungan aplikasi yang secara langsung dapat digunakan oleh pengguna.

Tujuan dari penelitian yang dilakukan ini adalah untuk mengembangkan model aplikasi *journal finder* dengan menggunakan algoritma *Jaccard Similarity* dan protokol *Open Archives Initiative Protocol for Metadata Harvesting* (OAI-PMH) yang dapat menyediakan solusi cerdas dalam merekomendasikan jurnal yang relevan bagi peneliti. Algoritma *Jaccard Similarity* dipilih sebagai pendekatan dalam penelitian ini karena memungkinkan pengukuran kesesuaian antara artikel yang akan dipublikasikan dengan metadata jurnal berdasarkan persentase kemiripan judul artikel dan abstrak, sehingga rekomendasi menjadi lebih akurat. Sementara itu, protokol OAI-PMH digunakan untuk mengumpulkan metadata dari berbagai *repository* jurnal yang menggunakan platform *Open Journal System* (OJS) secara efisien, memastikan data yang diperoleh selalu terkini dan relevan. Dengan integrasi ini, aplikasi dapat membantu peneliti dalam memilih jurnal dengan cepat dan tepat, meningkatkan peluang publikasi, serta mendukung penyebaran ilmu pengetahuan yang lebih luas.

Model dalam penelitian ini menggunakan pendekatan yang lebih sederhana, yaitu menggunakan algoritma *Jaccard Similarity*. Dari aspek komputasi, algoritma ini hanya membutuhkan operasi dasar seperti *intersection* (irisan) dan *union* (gabungan) dari dua himpunan. Operasi ini sangat mudah diimplementasikan dengan struktur data dalam bahasa pemrograman. Algoritma *Jaccard Similarity* bekerja sangat baik untuk *dataset* besar yang direpresentasikan sebagai *data sparse* (berisi banyak elemen nol). Berbeda dengan beberapa metrik lainnya seperti *Cosine Similarity*, *Jaccard Similarity* tidak memerlukan vektor data yang dinormalisasi karena hanya mempertimbangkan keberadaan elemen (0 atau 1). Secara umum, kelebihan *Jaccard Similarity* dari aspek komputasi mencakup kesederhanaan implementasi, efisiensi dalam memori dan waktu komputasi, kemampuan

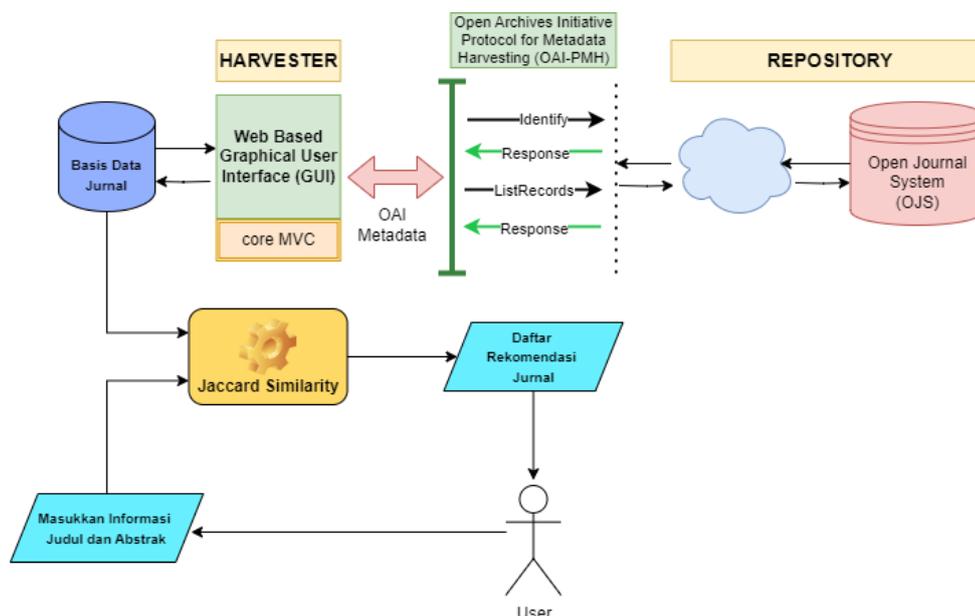
menangani dataset besar dengan *elemen sparse*, dan fleksibilitas untuk dioptimalkan menggunakan teknik skala besar.

Pada aspek akuisisi data, penggunaan *Open Archives Initiative Protocol for Metadata Harvesting* (OAI-PMH) memiliki keunggulan utama dalam kemampuannya untuk memfasilitasi interoperabilitas dan pertukaran *metadata* secara efisien di antara repositori digital. Protokol ini memungkinkan pengumpulan *metadata* dari berbagai sumber melalui mekanisme yang sederhana namun terstandarisasi, menggunakan protokol berbasis HTTP dan XML. Dengan arsitektur yang ringan, OAI-PMH mendukung pengindeksan skala besar oleh layanan pihak ketiga tanpa memerlukan akses langsung ke konten lengkap, sehingga ideal untuk membangun layanan pencarian lintas-repositori. Selain itu, fleksibilitasnya dalam mendukung berbagai format *metadata*, seperti Dublin Core, menjadikannya solusi serbaguna untuk manajemen dan pengelolaan data ilmiah, akademik, dan arsip digital secara global.

## 2. METODE PENELITIAN

### 2.1. Arsitektur Model

Penelitian ini menggunakan arsitektur model, seperti dapat dilihat pada Gambar 1. Basis data jurnal yang berisi *metadata* artikel ilmiah, seperti judul, penulis, abstrak, dan informasi penting lainnya diperoleh dari *Open Journal System* (OJS). Melalui *protocol Open Archives Initiative Protocol for Metadata Harvesting* (OAI-PMH) *metadata* artikel ilmiah dihimpun ke dalam basis data jurnal dengan menjalankan *method* atau *query Identify* dan *ListRecords* [25]. Kedua *method* ini akan menghasilkan dua buah *dataset* berupa informasi *repository* dan *metadata* setiap artikel ilmiah, yang selanjutnya disimpan dalam basis data jurnal ilmiah. Algoritma *Jaccard Similarity* digunakan untuk membandingkan *metadata* yang terdapat pada basis data jurnal, dengan input data yang dilakukan oleh user. Data yang dibandingkan adalah input data yang dilakukan oleh user berupa data judul dan abstrak, untuk dicari tingkat kemiripannya dengan informasi yang tersimpan dalam basis data jurnal.



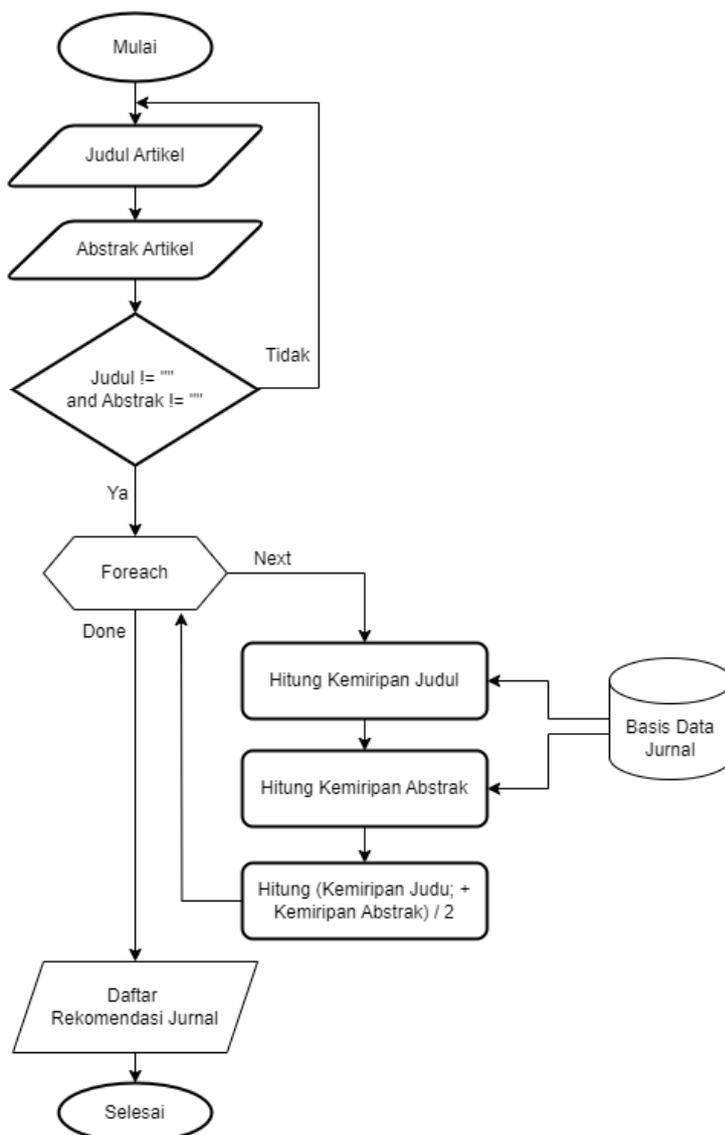
Gambar 1. Model Arsitektur Aplikasi

Sistem yang dikembangkan ini menggunakan *framework Codeigniter* untuk mengimplementasikan arsitektur *MVC*. Aplikasi dikembangkan menggunakan PHP dan MySQL sebagai penyimpanan persisten. PHP, MySQL, *framework CodeIgniter*, dan konsep *MVC* memiliki hubungan erat dalam pengembangan aplikasi web modern. PHP adalah bahasa pemrograman sisi *server* yang digunakan untuk membangun logika aplikasi web, sementara MySQL berfungsi sebagai sistem manajemen basis data relasional yang menyimpan data yang digunakan aplikasi. *CodeIgniter*, sebuah *framework PHP* yang ringan dan cepat, mempermudah pengembangan aplikasi dengan menyediakan struktur terorganisir berdasarkan konsep *Model-View-Controller* (*MVC*). Dalam *MVC*, *Model* menangani interaksi dengan database (sering menggunakan MySQL untuk mengelola data), *View* bertanggung jawab untuk menampilkan antarmuka pengguna, dan *Controller* menjadi penghubung antara *Model* dan *View* dengan menangani logika bisnis serta alur aplikasi. Kombinasi ini memberikan arsitektur yang terstruktur, meningkatkan pemeliharaan kode, dan mempercepat pengembangan aplikasi web yang dinamis.

### 2.2. Rancangan Flowchart Program

Gambar 2 memperlihatkan *flowchart program* yang memberikan gambaran, bagaimana proses rekomendasi jurnal dihasilkan. Pada tahap awal, pengguna memasukkan 2 buah informasi berupa judul dan abstrak dari artikel yang akan dicarikan rekomendasi alternatif jurnal oleh sistem. Kemudian akan diperiksa, apakah informasi judul dan abstrak artikel telah diinput semua? Jika masih terdapat data yang belum terisi, maka tahapan berikutnya akan meminta user mengisi datanya dengan lengkap. Selanjutnya, jika judul dan abstrak artikel telah terisi, maka dapat dilanjutkan ke tahap berikutnya dengan melakukan perulangan pada semua record yang ada dalam basis data yang telah berisi metadata artikel ilmiah.

Pada proses perulangan, setiap *record* dari basis data yang berisi metadata artikel ilmiah, akan dibandingkan tingkat kemiripannya antara judul dan abstrak yang telah diinput oleh pengguna. Kemudian dilakukan perhitungan untuk mendapatkan nilai agregasi tingkat kemiripan dengan mengambil nilai rata-rata dari hasil perhitungan dari judul dan abstrak. Setelah proses perulangan untuk setiap record dalam basis data selesai dilakukan, maka sistem akan menampilkan daftar rekomendasi jurnal yang dihasilkan. Daftar ini diurutkan dari data yang memiliki tingkat kemiripan paling tinggi sampai tingkat kemiripan yang paling rendah.



Gambar 2. Flowchart Program Sistem

### 2.3. Algoritma Jaccard Similarity

*Jaccard Similarity* adalah sebuah metode dalam teori himpunan yang digunakan untuk mengukur tingkat kesamaan antara dua buah himpunan. Nilai *Jaccard* dihitung dengan membandingkan ukuran irisan

(*intersection*) dari dua himpunan terhadap ukuran gabungannya (*union*). Secara matematis, *Jaccard Similarity* dirumuskan sebagai  $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$ , di mana A dan B adalah dua himpunan yang dibandingkan [26]. Nilai hasil perhitungan berkisar antara 0 hingga 1, di mana 0 menunjukkan bahwa kedua himpunan tidak memiliki elemen yang sama, sementara 1 menunjukkan bahwa kedua himpunan identik. Dalam konteks aplikasi komputer, algoritma ini sering digunakan untuk mengukur kesamaan dokumen, rekomendasi konten, atau analisis data berbasis teks, karena keunggulannya yang sederhana namun efektif dalam menghitung kesamaan elemen berbasis himpunan. Penelitian ini menggunakan algoritma *similarity* menggunakan *jaccard similarity* seperti dituangkan dalam *pseudocode* sebagai berikut:

```

ALGORITHM JaccardSimilarity
INPUT: Set A, Set B
OUTPUT: Jaccard Similarity between Set A and Set B

BEGIN
  // Step 1: Find the intersection of Set A and Set B
  Intersection <- A ∩ B

  // Step 2: Find the union of Set A and Set B
  Union <- A ∪ B

  // Step 3: Calculate Jaccard Similarity
  IF size(Union) != 0 THEN
    JaccardSimilarity <- size(Intersection) / size(Union)
  ELSE
    JaccardSimilarity <- 0
  END IF

  // Step 4: Return the result
  RETURN JaccardSimilarity
END
    
```

Sebagai contoh implementasi dari algoritma *Jaccard Similarity* ini adalah jika diberikan 2 buah teks, kemudian dilakukan beberapa tahapan perhitungan, sehingga diperoleh nilai tingkat kemiripan.

Teks 1 = "Augmented Reality untuk Media Pembelajaran Komponen pada Laptop";  
 Teks 2 = "Model Media Pembelajaran Berbasis Augmented Reality";

Jika diurai lebih jauh, bagaimana algoritma *Jaccard Similarity* ini bekerja dengan beberapa tahapan yang dapat dilakukan. Tahapan terdiri dari tokenisasi, menghitung irisan, menghitung gabungan, dan menghitung tingkat kemiripan. Secara lebih detail, berikut tahapan algoritma *Jaccard Similarity* bekerja:

1. Tokenisasi, pada tahapan ini teks akan dipecah menjadi kata-kata yang unik.  
 Teks 1 = A = {"Augmented", "Reality", "untuk", "Media", "Pembelajaran", "Komponen", "pada", "Laptop"}  
 Teks 2 = B = {"Model", "Media", "Pembelajaran", "Berbasis", "Augmented", "Reality"}
2. Hitung Irisan ( $A \cap B$ ), hasil irisan ini merupakan kata-kata yang terdapat pada kedua teks.  
 $A \cap B = \{\text{"Augmented", "Reality", "Media", "Pembelajaran"}\}$ ;  
 sehingga diperoleh jumlah irisan sebanyak  $|A \cap B| = 4$ .
3. Hitung Gabungan ( $A \cup B$ ), Gabungan adalah semua kata unik dari kedua teks, sehingga diperoleh:  
 $A \cup B = \{\text{"Augmented", "Reality", "untuk", "Media", "Pembelajaran", "Komponen", "pada", "Laptop", "Model", "Berbasis"}\}$ ;  
 Sehingga jumlah elemen himpunan adalah  $|A \cup B| = 10$ .
4. Hitung Tingkat Kemiripan, dengan menggunakan rumus seperti pada persamaan (1)

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

Sehingga memperoleh nilai tingkat kemiripan sebesar 40%.

$$J(A, B) = \frac{4}{10} = 0,40 = 40\%$$

Berdasarkan contoh perhitungan algoritma *Jaccard Similarity*, maka akan diperoleh tingkat kemiripan sebesar 0,40. Nilai 0,40 ini juga merupakan nilai akurasi yang diperoleh dari hasil perhitungan. Sehingga tingkat kepercayaan tingkat kemiripan dari dua buah teks tersebut, sebagai hasil rekomendasi adalah 40%.

### 3. HASIL DAN PEMBAHASAN

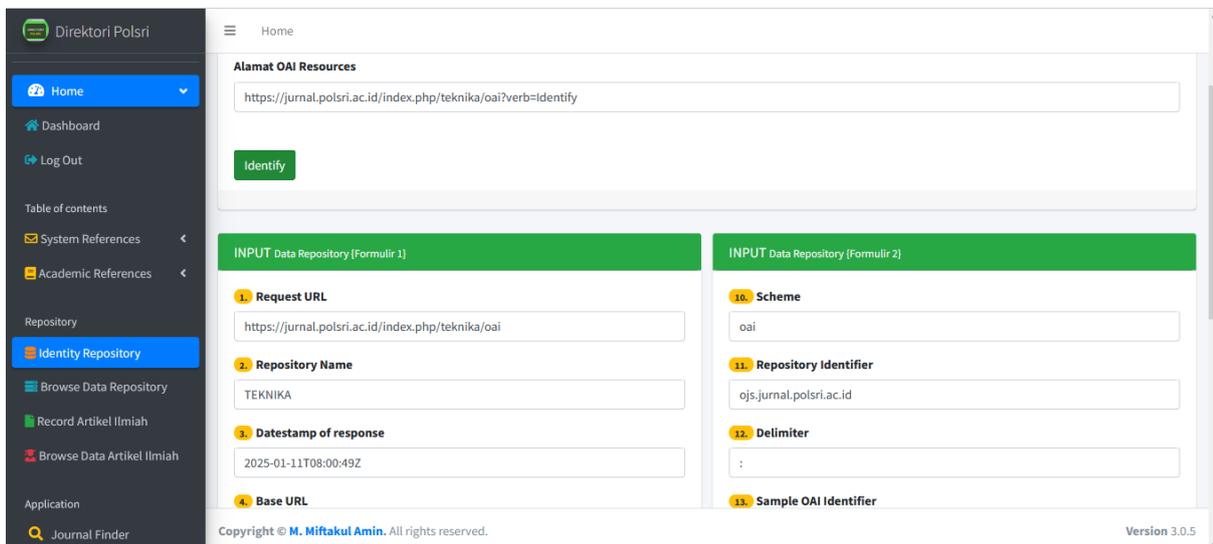
#### 3.1. Hasil

Penelitian ini telah menghasilkan sebuah aplikasi *journal finder* yang dioperasikan dalam lingkungan aplikasi berbasis web. Aplikasi dikembangkan menggunakan konsep *Model View Controller (MVC)* menggunakan framework Codeigniter. Model MVC adalah pola desain perangkat lunak yang membagi aplikasi menjadi tiga komponen utama, yaitu *Model*, *View*, dan *Controller*, untuk memisahkan logika bisnis, antarmuka pengguna, dan kontrol alur aplikasi. *Model* bertanggung jawab atas pengelolaan data dan logika aplikasi, *View* berfungsi menampilkan antarmuka kepada pengguna berdasarkan data dari *Model*, sedangkan *Controller* bertindak sebagai penghubung antara *Model* dan *View*, memproses input dari pengguna, dan mengatur aliran data. Pola ini mempermudah pengembangan aplikasi dengan meningkatkan modularitas, memisahkan tanggung jawab, dan memudahkan pemeliharaan [27].

##### 3.1.1. Halaman Identify

Gambar 3 memperlihatkan halaman *Identify Repository* yang digunakan untuk mengidentifikasi informasi metadata dari *Open Journal System (OJS)* sesuai dengan alamat *OAI Resources* yang diinput. Informasi ini berupa alamat URL yang merupakan alamat halaman OJS dari sebuah jurnal. Dengan menambahkan *verb=Identify* diakhir URL, maka aplikasi *journal finder* ini dapat mendeteksi informasi *metadata* penting seperti alamat URL, nama jurnal, versi protocol, dan informasi penting lainnya.

Pada halaman ini akan menghasilkan identifikasi sebanyak 18 buah informasi penting yang dihasilkan dari protokol *OAI-PMH*, meliputi *Request URL*, *Repository Name*, *Datestamp of response*, *Base URL*, *Protocol Version*, *Earliest Datestamp*, *Deleted Record Policy*, *Granularity*, *Admin Email*, *Scheme*, *Repository Identifier*, *Delimiter*, *Sample OAI Identifier*, *Title*, *Name*, *Email*, *Version*, dan *Url*. Setelah seluruh informasi penting dari *platform OJS* ini berhasil diidentifikasi, maka tahapan berikutnya yang dapat dilakukan oleh user adalah menyimpan 18 informasi tersebut ke dalam basis data.

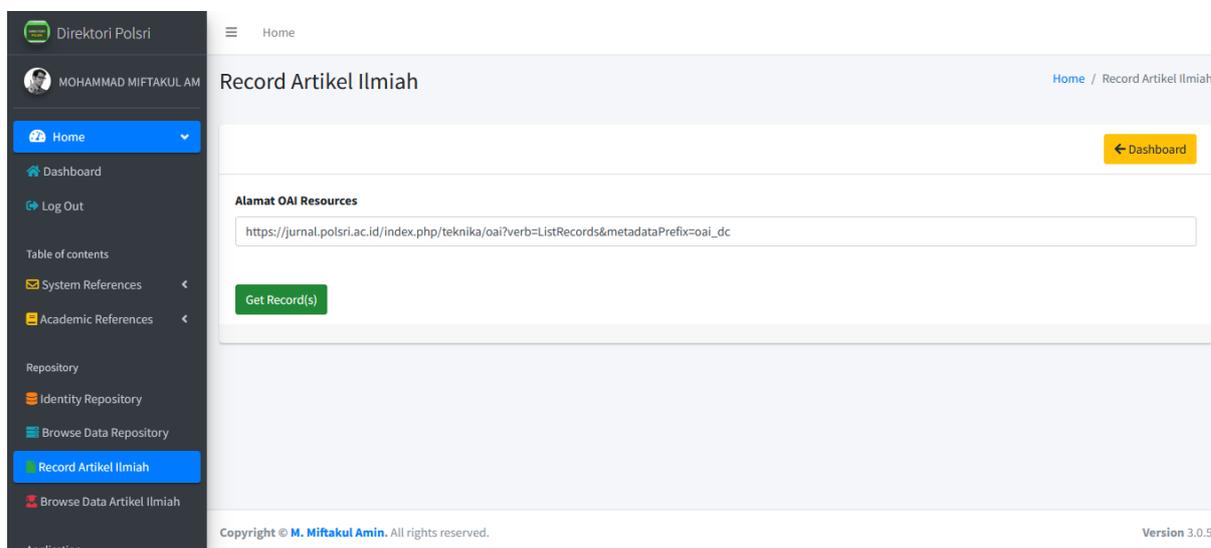


Gambar 3. Halaman *Identify Repository*

##### 3.1.2. Halaman Record Artikel Ilmiah

Menu record artikel ilmiah seperti diperlihatkan pada Gambar 4 ini digunakan untuk mengambil informasi metadata setiap artikel ilmiah yang telah berhasil diidentifikasi *repository* melalui halaman *Identify Repository* seperti Gambar 3. Pada tahapan ini *user* dapat memasukkan sebuah alamat *OAI Resources* untuk menentukan *repository* yang akan ditarik datanya. Sebagai contoh seperti yang diperlihatkan pada Gambar 3 terdapat alamat [https://jurnal.polsri.ac.id/index.php/teknika/oai?verb=ListRecords&metadataPrefix=oai\\_dc](https://jurnal.polsri.ac.id/index.php/teknika/oai?verb=ListRecords&metadataPrefix=oai_dc). Setelah alamat *OAI*

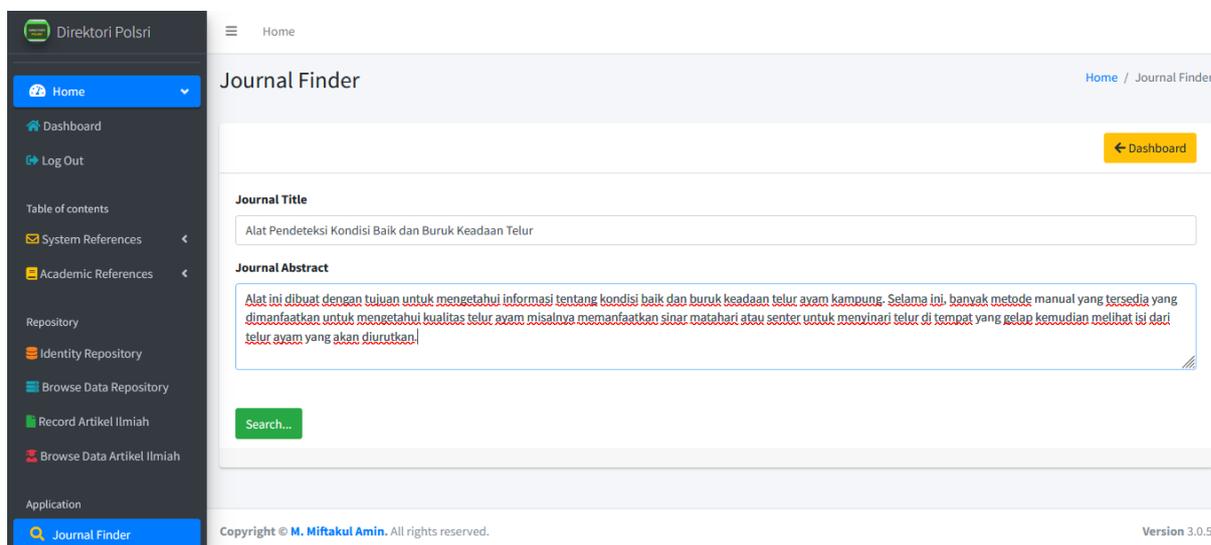
*Resources* ditentukan, maka *user* dapat mengklik tombol *Get Records* untuk menyimpan informasi metadata artikel ke dalam basis data.



Gambar 4. Halaman *Record Artikel Ilmiah*

### 3.1.3. Implementasi Aplikasi Web

Seperti diperlihatkan pada Gambar 5, bahwa dalam operasional *journal finder* terdapat 2 buah input yang dapat diisi oleh *user*, yaitu informasi judul dan abstrak dari artikel ilmiah. Sistem *journal finder* yang dikembangkan ini akan melakukan pengujian tingkat kemiripan antara input data, dengan informasi yang telah tersimpan pada basis data. Setelah *user* mengisikan data judul dan abstrak, maka dapat dilanjutkan dengan mengklik tombol *Search* untuk mulai melakukan pencarian data.

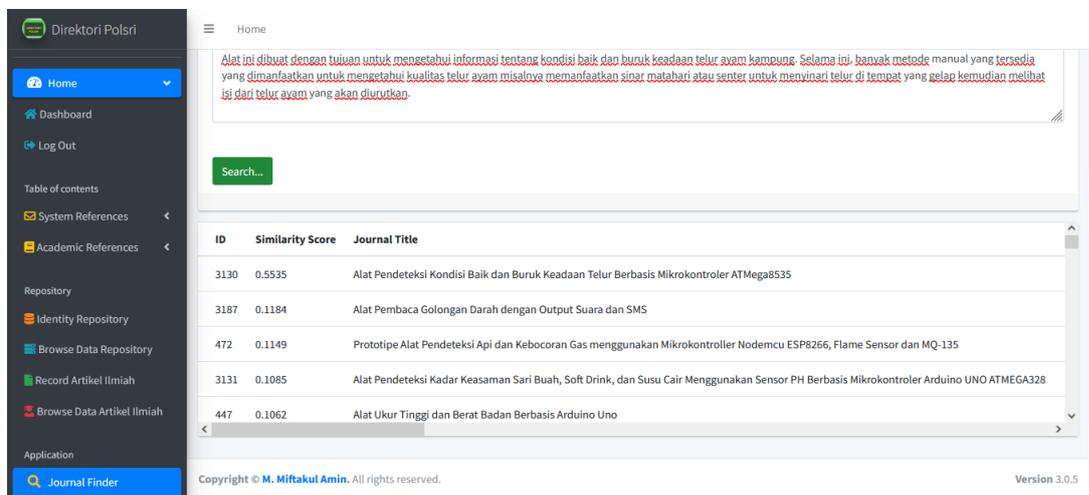


Gambar 5. Halaman Pengujian *Journal Finder*

Selanjutnya pada Gambar 6 ditampilkan hasil pencarian yang dilakukan oleh sistem, dengan menampilkan daftar rekomendasi jurnal ilmiah yang memiliki kesamaan dengan input data oleh *user*. Pada halaman hasil pencarian ditampilkan informasi ID, nilai tingkat kemiripan, dan judul yang memiliki tingkat kemiripan paling besar yang mengindikasikan hasil yang paling direkomendasikan oleh sistem *journal finder* ini.

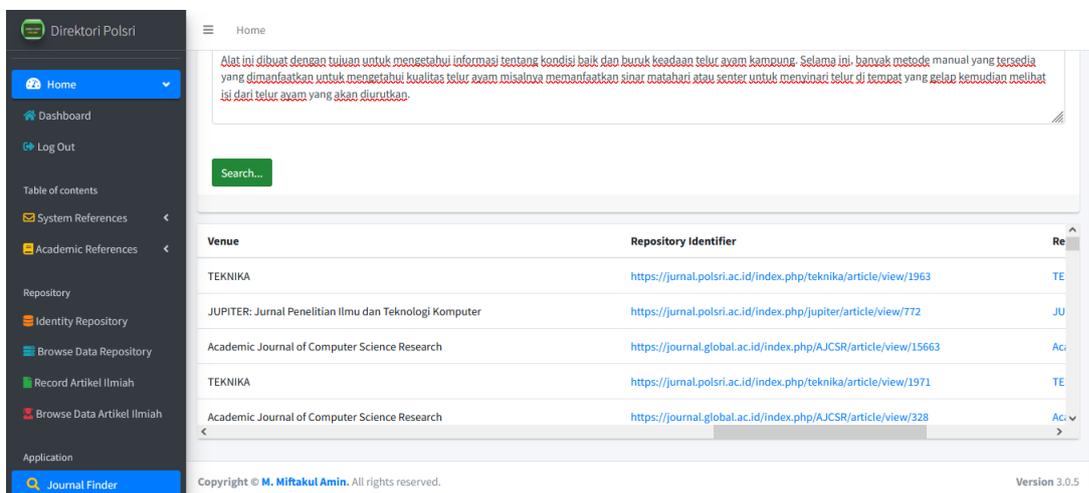
Disamping informasi rekomendasi yang disajikan pada Gambar 6, aplikasi ini juga menyajikan tampilan seperti pada Gambar 7. Hasil pencarian yang dilakukan oleh *journal finder* ini dapat memberikan informasi berupa *venue* yang merupakan nama jurnal atau prosiding pada *platform OJS*. Nama *venue* akan ditampilkan disertai dengan informasi *URL OJS* yang dapat dengan mudah diakses oleh user. Jika informasi *URL OJS* ini

diklik oleh user, maka akan ditampilkan informasi halaman *web OJS* dari link yang bersesuaian. Sehingga hal ini akan memudahkan bagi para peneliti dan dosen mendapatkan informasi secara komprehensif terkait dengan artikel ilmiah yang telah mereka hasilkan, untuk mempublikasikan artikelnya pada jurnal yang relevan.



Gambar 6. Hasil Rekomendasi Berdasarkan Judul Artikel

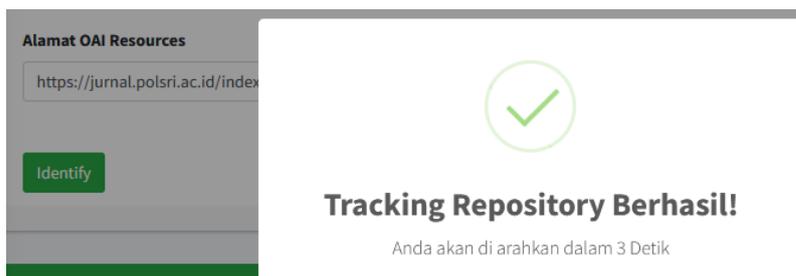
Nilai *Similarity Score* seperti yang tersaji pada Gambar 6, juga merupakan tingkat akurasi dari hasil rekomendasi yang dihasilkan oleh sistem. Sebagai contoh hasil rekomendasi dengan nilai 0,5535 merupakan hasil tingkat kemiripan antara judul dan abstrak artikel ilmiah, dengan metadata yang terdapat pada basis data jurnal. Nilai 0,5535 atau 55,35% ini sekaligus juga merupakan tingkat akurasi yang dihasilkan oleh sistem. Nilai ini merupakan hasil agregasi dari nilai rata-rata yang dihasilkan dari tingkat kemiripan judul dan abstrak artikel ilmiah.



Gambar 7. Hasil Rekomendasi Berdasarkan Informasi Nama Jurnal dan Alamat URL Jurnal

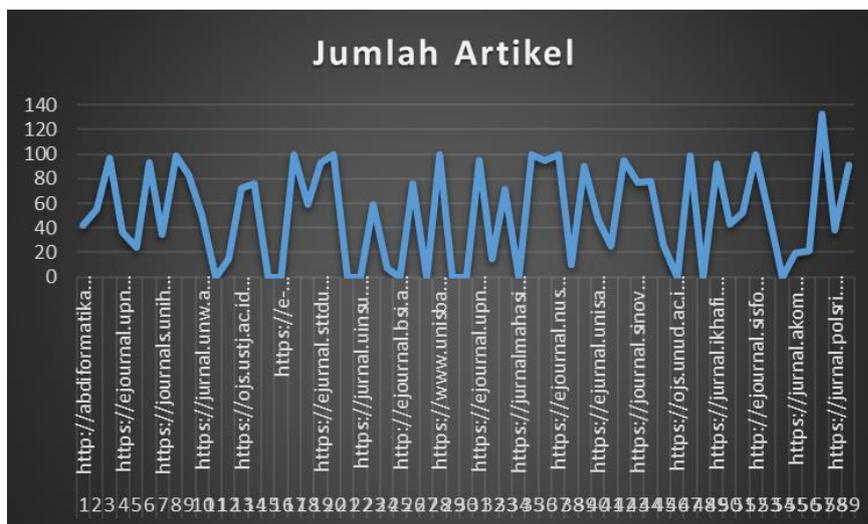
### 3.2. Pembahasan

Berdasarkan pengujian yang telah dilakukan, dengan menjalankan beberapa layanan dan fungsionalitas yang ada dalam aplikasi *journal finder* berbasis web ini telah mampu menyajikan rekomendasi berupa informasi alamat *OJS* jurnal yang relevan sesuai dengan input data berupa judul dan abstrak dari sebuah artikel. Dari hasil penarikan data menggunakan protokol *OAI-PMH* pada platform *OJS*, diperoleh sejumlah 59 *repository* ilmiah. Selanjutnya dari proses penarikan data dari setiap *repository*, telah terhimpun metadata artikel ilmiah sebanyak 3.321 *record*. Informasi metadata artikel ilmiah ini yang nantinya digunakan sebagai data pengujian pada aplikasi *journal finder*. Rata-rata waktu akses yang dibutuhkan untuk menghimpun metadata artikel adalah 3 detik, yang ditampilkan pada saat selesai menjalankan proses identifikasi seperti diperlihatkan pada Gambar 8.



Gambar 8. Waktu Eksekusi Aplikasi *Journal Finder*

Dari beberapa proses pengujian yang dilakukan, terdapat beberapa *repository* ilmiah dalam *platform* OJS yang tidak dapat memperoleh metadata artikel sama sekali sebanyak 13 buah *repository*, seperti *repository* nomor ID 11, 15, 16, 21, 22, 25, 27, 29, 30, 34, 46, 48, 54 yang tersaji pada Gambar 9. Hal ini dikarenakan pada beberapa *platform* OJS tersebut telah ditambahkan mekanisme keamanan sistem, sehingga tidak dapat secara langsung dilakukan penarikan data melalui aplikasi *jurnal finder* ini. Demikian juga terdapat jumlah metadata artikel sebanyak 7, 10, 15 buah saja, dikarenakan merupakan *repository* OJS baru yang berusia kurang dari 1 tahun. Sedangkan dalam 1 kali proses pengambilan metadata artikel, akan diperoleh maksimal 100 record, jika dalam sebuah *repository* ilmiah dalam *platform* OJS tersebut terdapat lebih dari 100 record.



Gambar 9 Distribusi Artikel Ilmiah Setiap *Repository* OJS

#### 4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka dihasilkan sebuah aplikasi *journal finder* menggunakan algoritma *jaccard similarity*. Nilai kemiripan yang diperoleh berada dalam rentang 0,00 yang berarti tidak terdapat kesamaan, sampai dengan 1,00 yang mengindikasikan perbandingan sama persis atau identik. *Metadata* artikel ilmiah yang digunakan dalam sistem berjumlah 3.321 buah, yang berasal dari 59 *repository* jurnal ilmiah dari *platform open journal system* (OJS). Data artikel ilmiah ini dihimpun menggunakan *protocol Open Archives Initiative Protocol for Metadata Harvesting* (OAI-PMH) dengan menggunakan dua buah *method/query* utama, yaitu *Identify* dan *ListRecords*.

#### 5. SARAN

Penelitian lanjutan yang dapat dilakukan dari penelitian ini diantaranya, mengeksplorasi beberapa *repository* ilmiah yang tidak dapat ditarik metadata artikelnya dikarenakan adanya tambahan mekanisme keamanan data. Demikian juga dalam penelitian lebih lanjut dapat menggunakan pendekatan yang lebih mendalam seperti *machine learning*, *deep learning*, dan dari aspek *dataset* dapat menggunakan data dengan volume yang lebih besar. Demikian juga dari sisi kinerja dari model, dapat melakukan pengukuran kecepatan proses, *payload*, dan aspek komputasi lainnya.

$$f_{\text{baud}} = \frac{2^{\text{SMOD}}}{64} \times f_{\text{osc}} \quad (1)$$

## DAFTAR PUSTAKA

- [1] M. Miftakul Amin, D. Stiawan, Ermatita, and R. Budiarto, "Proposed threshold-based and rule-based approaches to detecting duplicates in bibliographic database," *Bull. Electr. Eng. Informatics*, vol. 13, no. 3, pp. 2036–2047, 2024, doi: 10.11591/eei.v13i3.7665.
- [2] M. Demirkan, A. Özgür, and H. Erdem, "Journal Finder for TRDIZIN: Baseline Study," in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, 2021. doi: <https://doi.org/10.1109/SIU53274.2021.9477700>.
- [3] N. Vara, F. Rahimi, and F. Danesh, "Do LIS experts select more appropriate journals than journal finders? A study about LIS journals?," *J. Librariansh. Inf. Sci.*, no. December 2024, pp. 1–13, 2023, doi: 10.1177/09610006231214562.
- [4] X. Li, S. Feng, and X. Zhang, "A Review of Methods Using Large Language Models in News Recommendation Systems," in *2024 IEEE/ACIS 27th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2024. doi: <https://doi.org/10.1109/SNPD61259.2024.10673956>.
- [5] G. Huang, "E-Commerce Intelligent Recommendation System Based on Deep Learning," in *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, 2022. doi: <https://doi.org/10.1109/IPEC54454.2022.9777500>.
- [6] Y. Zhang, Y. Li, R. Wang, M. S. Hossain, and H. Lu, "Multi-Aspect Aware Session-Based Recommendation for Intelligent Transportation Services," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4696–4705, 2021, doi: <https://doi.org/10.1109/TITS.2020.2990214>.
- [7] Y. H. Alfaifi, "Towards an Ontology-Based E-Learning Recommendation System," in *2023 3rd International Conference on Computing and Information Technology (ICCIIT)*, 2023. doi: <https://doi.org/10.1109/ICCIIT58132.2023.10273903>.
- [8] V. N. Rathod, R. H. Goudar, A. Kulkarni, D. G. M, and G. S. Hukkeri, "A Survey on E-Learning Recommendation Systems for Autistic People," *IEEE Access*, vol. 12, pp. 11723–11732, 2024, doi: <https://doi.org/10.1109/ACCESS.2024.3355589>.
- [9] P. Ahlawat and C. Rana, "A Comprehensive Insight on Machine Learning Enabled Internet of Things Recommender Systems (IoTRS)," in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 2021. doi: <https://doi.org/10.1109/ICAC3N53548.2021.9725718>.
- [10] S. Muskan, R. S, A. B, J. G, and S. Challa, "An Early Recommendation Tool to Enhance Medicinal Plant Growth based on GIS and Soil Data," in *2023 International Conference on Computer, Electronics & Electrical Engineering & their Applications (IC2E3)*, 2023. doi: <https://doi.org/10.1109/IC2E357697.2023.10262541>.
- [11] K. Al Farani, B. Aghoutane, L. Maada, F. Nafis, A. Yahyaouy, and J. Riffi, "Item Based Collaborative Filtering Tourism Recommender System Using Apache Mahout," in *2024 International Conference on Intelligent Systems and Computer Vision (ISCV)*, 2024. doi: <https://doi.org/10.1109/ISCV60512.2024.10620090>.
- [12] K. Al Farani, F. Nafis, B. Aghoutane, A. Yahyaouy, J. Riffi, and A. Sabri, "Hybrid recommender system for tourism based on big data and AI: A conceptual framework," *Big Data Min. Anal.*, vol. 4, no. 1, pp. 47–55, 2021, doi: <https://doi.org/10.26599/BDMA.2020.9020015>.
- [13] D. Mogaveera, V. Mathur, and S. Waghela, "e-Health Monitoring System with Diet and Fitness Recommendation using Machine Learning," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, 2021. doi: <https://doi.org/10.1109/ICICT50816.2021.9358605>.
- [14] S. M. Al-Ghuribi, S. A. M. Noah, M. A. Mohammed, N. Tiwary, and N. I. Y. Saat, "A Comparative Study of Sentiment-Aware Collaborative Filtering Algorithms for Arabic Recommendation Systems," *IEEE Access*, vol. 12, pp. 174441–174454, 2024, doi: <https://doi.org/10.1109/ACCESS.2024.3489658>.
- [15] F. U. D. Laseno and B. Hendradjaya, "Knowledge-Based Filtering Recommender System to Propose Design Elements of Serious Game," *Proc. Int. Conf. Electr. Eng. Informatics*, vol. 2019-July, no. July, pp. 158–163, 2019, doi: 10.1109/ICEEI47359.2019.8988797.

- 
- [16] R. M. Yusup and R. Wahdiniwaty, "Siamese Neural Networks Approach to Hybrid Recommender System Modeling for Fostering Economic Growth in Fashion Domain," in *2023 9th International Conference on Signal Processing and Intelligent Systems (ICSPIS)*, 2023. doi: <https://doi.org/10.1109/ICSPIS59665.2023.10402654>.
- [17] S. Rani, "Machine Learning Algorithms for building Recommender Systems," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, IEEE, 2019, pp. 785–790.
- [18] M. Ge, G. Pilato, F. Persia, and D. D'Auria, "Recommender System for Social Media: Research Challenges and Future Applications," in *2023 Fifth International Conference on Transdisciplinary AI (TransAI)*, 2023. doi: <https://doi.org/10.1109/TransAI60598.2023.00033>.
- [19] C. D. Casuat, A. S. M. Isira, E. D. Festijo, A. S. Alon, J. N. Mindoro, and J. A. B. Susa, "A Development of Fuzzy Logic Expert-Based Recommender System for Improving Students' Employability," in *2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC)*, 2020. doi: <https://doi.org/10.1109/ICSGRC49013.2020.9232543>.
- [20] X. Nan and D. Wu, "Design and Implementation of Japanese Smart Tourism System Based on Improved Genetic Algorithm," in *2022 International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS)*, 2022. doi: <https://doi.org/10.1109/AIARS57204.2022.00033>.
- [21] R. Ahuja, A. Solanki, and A. Nayyar, "Movie recommender system using k-means clustering and k-nearest neighbor," *Proc. 9th Int. Conf. Cloud Comput. Data Sci. Eng. Conflu. 2019*, pp. 263–268, 2019, doi: 10.1109/CONFLUENCE.2019.8776969.
- [22] S. P. Priya\* and D. M. Karthikeyan, "A Novel Automatic Journal Recommender System," *Int. J. Recent Technol. Eng.*, vol. 8, no. 6, pp. 2608–2612, 2020, doi: 10.35940/ijrte.f8598.038620.
- [23] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, "A content-based recommender system for computer science publications," *Knowledge-Based Syst.*, vol. 157, pp. 1–9, 2018, doi: 10.1016/j.knosys.2018.05.001.
- [24] M. B. Magara, S. O. Ojo, and T. Zuva, "Towards a Serendipitous Research Paper Recommender System Using Bisociative Information Networks (BisoNets)," *2018 Int. Conf. Adv. Big Data, Comput. Data Commun. Syst. icABCD 2018*, pp. 1–6, 2018, doi: 10.1109/ICABCD.2018.8465475.
- [25] R. Yuliant and N. Karna, "Knowledge Sharing Filtering on OAI-PMH," *2016 Int. Conf. Inf. Technol. Syst. Innov.*, no. October 2016, 2016, doi: 10.1109/ICITSI.2016.7858213.
- [26] W. H. Gomma and A. A. Fahmy, "A Survey of Text Similarity Approaches," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 13–18, 2013.
- [27] S.-C. Necula, "Exploring the Model-View-Controller (MVC) Architecture: A Broad Analysis of Market and Technological Applications," *Preprints*, no. Mvc, 2024, doi: 10.20944/preprints202404.1860.v1.