

# Development of a Semantic Web Based Scientific Journal Search Application

Pengembangan Aplikasi Penelusuran Jurnal Ilmiah Berbasis Web Semantik

Vaiye E. J. Karame, Agustinus Jacobus, Dirko G. S. Ruindungan

Dept. of Electrical Engineering, Sam Ratulangi University Manado, Kampus Bahu St., 95115, Indonesia

e-mails: [efraimkaram@gmail.com](mailto:efraimkaram@gmail.com), [a.jacobus@unsrat.ac.id](mailto:a.jacobus@unsrat.ac.id), [dirko@unsrat.ac.id](mailto:dirko@unsrat.ac.id)

Received: 14 February 2024; revised: 19 July 2024; accepted: 25 September 2024

**Abstract** — *Semantic Web can enhance the relevance of search results across various fields. This technology enables search engines to understand the context and meaning of keywords, thereby producing more accurate and user-specific search results. This research aims to develop a scientific journal search application based on the Semantic Web. The development stages include Requirement Analysis, System Design, Implementation, and Testing. During the Requirement Analysis stage, we identify existing ontologies and utilize them in the system to be built. In the design stage, the existing ontology system will be further developed. Additionally, a user interface and necessary functionalities will be designed to ensure the system operates according to requirements. In the implementation stage, we use Apache Jena as the framework for developing the semantic web application. Testing is conducted to evaluate the application's performance in producing relevant search results. The test results show a Recall value of 0.8, indicating a high level of relevance, and a Precision value of 0.8, demonstrating the application's effectiveness in achieving high relevance. The F1 score, combining recall and precision, is also 0.8. Based on these results, the developed system performs well.*

**Key words:** Semantic web; journal search; application; ontology

**Abstrak** — *Web Semantik dapat meningkatkan relevansi hasil pencarian di berbagai bidang. Teknologi ini memungkinkan mesin pencari untuk memahami konteks dan makna dari kata kunci, sehingga menghasilkan hasil pencarian yang lebih tepat dan sesuai dengan kebutuhan pengguna. Penelitian ini bertujuan mengembangkan aplikasi penelusuran jurnal ilmiah berbasis web semantik. Tahapan pengembangan meliputi Analisis Kebutuhan, Perancangan Sistem, Implementasi, dan Pengujian. Dalam tahap Analisis kebutuhan, kami mengidentifikasi ontologi yang sudah ada kemudian memanfaatkannya dalam sistem yang akan dibangun. Pada tahap perancangan, sistem ontologi yang telah ada akan dikembangkan lebih lanjut. Selain itu, akan dirancang antarmuka pengguna serta fungsionalitas yang diperlukan untuk memastikan sistem bekerja sesuai dengan kebutuhan. Pada tahap implementasi, kami menggunakan Apache Jena sebagai framework pengembangan aplikasi web semantik. Tahap Pengujian dilakukan untuk melihat kinerja aplikasi apakah telah menghasilkan hasil penelusuran yang relevan. Hasil pengujian nilai Recall menunjukkan nilai yang cukup relevan yaitu 0.8 dan Precision untuk kinerja aplikasi dengan tingkat relevansi tinggi, cukup bekerja dengan baik yaitu 0.8 dan nilai F1 score dari nilai recall dan precision yaitu 0.8. Dari hasil yang dapatkan, sistem yang dikembangkan berkerja dengan baik .*

**Kata kunci:** web semantic; pencarian jurnal; aplikasi; ontology

## I. PENDAHULUAN

Penelitian sebelumnya telah menunjukkan bahwa sistem pencarian jurnal ilmiah sering kali kesulitan dalam memahami konteks topik yang kompleks, sehingga menghasilkan hasil pencarian yang kurang relevan. Studi [1]. Menggaris bawahi pentingnya pemahaman konteks dalam meningkatkan relevansi hasil pencarian. Penelitian ini kemudian dilanjutkan oleh [2]. yang mengeksplorasi variasi makna topik dalam konteks yang berbeda.Untuk mengatasi permasalahan ini, [3] mengusulkan pemanfaatan teknologi Web Semantik. Penelitian mereka menunjukkan bahwa dengan menambahkan lapisan semantik pada data, sistem pencarian dapat lebih memahami konteks dan menghasilkan hasil yang lebih relevan.

Studi [4] memperluas penelitian ini dengan menunjukkan bahwa integrasi data yang lebih baik melalui Web Semantik memungkinkan pemrosesan data yang lebih efisien dan kontekstual.Penelitian [5] mengidentifikasi bahwa penggunaan RDF dan OWL dapat meningkatkan akurasi hasil pencarian dengan memberikan informasi yang lebih terstruktur.

Studi [6] kemudian mengkaji implementasi teknis dari Web Semantik dalam sistem pencarian dan menemukan bahwa SPARQL memungkinkan pencarian yang lebih cerdas dan relevan dengan memahami konteks dari data yang dicari.Meskipun demikian, penerapan Web Semantik masih menghadapi beberapa tantangan, seperti integrasi data yang kompleks dan pemrosesan data yang intensif. Penelitian [7] membahas strategi untuk mengatasi tantangan ini. Studi [8] menunjukkan bahwa penambahan metadata dan konteks dalam data dapat secara signifikan meningkatkan relevansi hasil pencarian di berbagai domain.

## II. METODE

Tahapan metodologi yang dilalui mencakup analisis kebutuhan, perancangan sistem, implementasi dan pengujian. Masing – masing tahapan tersebut diuraikan sebagai berikut.

### 1. Analisis Kebutuhan

Mengidentifikasi kebutuhan aplikasi penelusuran jurnal ilmiah.Menggunakan ontologi yang sudah ada yaitu biblio.owl [2]. untuk mengintegrasikan data. Studi literatur dan interaksi

dengan pakar untuk memahami domain aplikasi.spesifikasi penggunaan ontologi untuk meningkatkan akurasi pencarian.

## 2. Perancangan Sistem

Pada tahap ini merancang arsitektur aplikasi dan antarmuka pengguna melibatkan beberapa langkah penting. Pertama, merancang diagram flowchart untuk memetakan proses pencarian. Diagram ini membantu dalam memahami alur kerja dan mengidentifikasi titik-titik penting dalam proses tersebut. Selanjutnya, mendesain antarmuka pengguna yang intuitif adalah kunci untuk memastikan pengalaman pengguna yang baik. Antarmuka yang intuitif memudahkan pengguna dalam berinteraksi dengan aplikasi, mengurangi kurva belajar, dan meningkatkan efisiensi penggunaan.

Perancangan Sistem adalah proses sistematis yang digunakan untuk mengembangkan, mengimplementasikan, dan memelihara sistem memberikan kerangka kerja yang jelas untuk setiap tahap pengembangan, mulai dari perencanaan hingga pemeliharaan.

### 2.1. Use Case diagram

Use Case Diagram adalah gambaran graphical dari beberapa atau semua actor, use case dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun dan Gambaran dari sistem aplikasi penelusuran. Diagram use case yang disajikan pada Gambar 1 menggambarkan interaksi pengguna dengan sistem aplikasi penelusuran jurnal. Pengguna, yang dapat berupa peneliti, mahasiswa, atau siapa saja yang membutuhkan informasi ilmiah, berperan sebagai aktor utama dalam sistem ini.

Pengguna dapat berinteraksi dengan sistem melalui tiga fungsi utama. Pertama, pengguna dapat mencari artikel dengan memasukkan kata kunci atau frasa pencarian yang relevan. Kedua, setelah melakukan pencarian, pengguna akan disajikan dengan daftar artikel yang sesuai dengan kriteria pencarinya. Terakhir, pengguna dapat melihat detail artikel yang diinginkan secara lengkap dari daftar hasil pencarian tersebut.

### 2.2. Rancangan Antarmuka

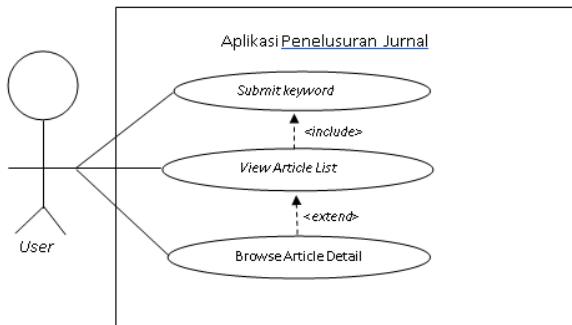
Desain antarmuka adalah proses menciptakan tampilan dan interaksi antara pengguna dengan sistem. Tujuan utama desain antarmuka adalah untuk membuat pengalaman pengguna yang intuitif, efisien, dan menyenangkan. pada gambar 2 di tunjukan prototipe desain antar muka form input dan pada gambar 3 ditunjukan desain prototipe antarmuka hasil dari pencarian.

### 3. Implementasi.

Mengembangkan aplikasi berdasarkan desain Memilih bahasa pemrograman Java untuk backend dan HTML untuk frontend. Mengembangkan kode program untuk fitur pencarian dan integrasi dengan basis data ontology dengan framework Apache Jena, menguji kode program untuk memastikan kualitas dan kesesuaian fungsionalitas.

## 4. Pengujian

Pada tahap pengujian ini memastikan aplikasi berfungsi dengan baik.Melakukan pengujian unit, integrasi, dan sistem.gunakan pengujian *Recall and Precision* untuk menghitung seberapa besar akurasi dan presisi system dapat mengembalikan data .



Gambar 1. Use Case diagram

Gambar 2. Prototipe Tampilan Form Input

Gambar 3. Implementasi Tampilan Hasil

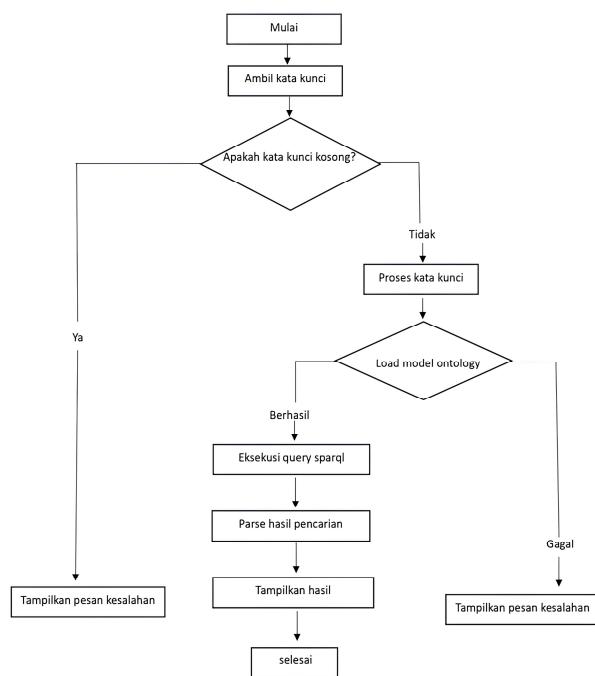
### 1. FlowChart diagram

Diagram Flowchart adalah jenis *diagram Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur kerja atau aktivitas operasional dalam suatu sistem, proses bisnis, atau kegiatan lainnya. Tujuan utama dari diagram Flowchart adalah memberikan pemahaman visual tentang langkah-langkah yang terlibat

dalam suatu aktivitas atau proses Diagram *Flowchart* di atas memberikan gambaran visual mengenai tahapan-tahapan yang dilalui dalam proses pengambilan dan pemrosesan kata kunci dalam suatu sistem. Secara garis besar, sistem ini bekerja dengan cara sebagai berikut:

Pada tahap awal, sistem akan menginisiasi proses pengambilan kata kunci. Kata kunci yang menjadi input utama ini kemudian akan dievaluasi untuk memastikan tidak kosong. Jika kata kunci yang dimasukkan kosong, sistem akan menghentikan proses dan memberikan pesan kesalahan kepada pengguna, namun, jika kata kunci valid, sistem akan melanjutkan dengan tahap pemrosesan kata kunci. Proses ini melibatkan teknik-teknik seperti *stemming* dan *stopword removal* untuk membersihkan dan menyederhanakan kata kunci. Setelah kata kunci diproses, sistem akan memuat model ontologi yang relevan. Model ontologi ini berperan sebagai basis pengetahuan yang berisi representasi informasi mengenai domain tertentu. Dengan kata lain, model ini berfungsi sebagai kamus yang memahami makna dari kata-kata yang telah diproses. Selanjutnya, sistem akan menjalankan query SPARQL. Query SPARQL ini merupakan bahasa khusus yang digunakan untuk mengajukan pertanyaan terhadap data yang terstruktur dalam bentuk *Resource Description Framework* (RDF). Pertanyaan yang diajukan didasarkan pada kata kunci yang telah diproses dan informasi yang terdapat dalam model ontologi.

Hasil dari eksekusi query SPARQL kemudian akan diurai atau diparsing untuk mendapatkan informasi yang relevan dengan pertanyaan yang diajukan. Jika proses parsing berhasil, sistem akan menampilkan hasil pencarian kepada pengguna. Namun, jika terjadi kesalahan dalam proses parsing atau pada tahap sebelumnya, sistem akan memberikan pesan kesalahan. Dapat dilihat pada gambar 4.



Gambar 4. Flowchart penelusuran semantic

### III. HASIL DAN PEMBAHASAN

Proses pembangunan ontologi hanyadilakukan proses penambahan *terms* pada ontologi jurnal yang telah ada untuk memenuhi kebutuhan sistem pencarian. Pembangunan ontologi dibangun menggunakan Protégé 5 dimana setelah ontologi selesai dibangun konsistensi ontology diperiksa menggunakan *Pellet Reasoner*. *Pellet* merupakan sebuah *tools reasoner* untuk menguji konsistensi deskripsi logika yang telah didefinisikan. *Pellet* akan memberikan peringatan apabila terdapat deskripsi logika yang mengalami inkonsistensi.

#### a. Class ontology

Ontologi biblio memiliki 4 *class* yaitu Organization, Person, Publication dan Works dimana tiap *class* dibagi lagi menjadi beberapa *subclass* dan dari kelas-kelas ini mempunyai hubungan dengan *data properties* dan *object properties*. Gambar 5 menampilkan sebuah hierarki ontologi atau klasifikasi konseptual yang menggambarkan struktur dan hubungan antar kelas dalam domain “Artikel Jurnal Ilmiah”. Hierarki ini dibangun menggunakan logika deskriptif dan direpresentasikan dengan *Web Ontology Language* (OWL).

Hierarki ontologi ini dimulai dari kelas paling atas yang disebut owl:Thing. Kelas ini berfungsi sebagai akar dari semua kelas dalam ontologi, menunjukkan bahwa setiap konsep dalam hierarki ini adalah jenis dari “sesuatu”.

Terdapat beberapa kelas yang merepresentasikan entitas organisasi yang terkait dengan publikasi artikel. Kelas-kelas ini termasuk Organization, Institution, dan Publisher. Mereka menggambarkan lembaga, institusi, atau penerbit yang berperan dalam proses publikasi. Selanjutnya, ada kelas Person dan Author yang mewakili individu, khususnya penulis artikel.

Penulis adalah elemen penting dalam dunia publikasi ilmiah, karena mereka adalah pencipta konten. Kelas Publication berfungsi sebagai induk untuk semua jenis publikasi, termasuk artikel jurnal. Di dalamnya, terdapat atribut-atribut seperti ISSN, Journal, Volume, dan Year yang terkait dengan publikasi berkala. Nomor ISSN, nama jurnal, volume, dan tahun terbit adalah informasi penting yang membantu mengidentifikasi dan mengkategorikan publikasi.

Kelas Works adalah kelas induk untuk semua karya ilmiah. Di bawahnya, terdapat kelas Abstract dan Article. Abstract adalah ringkasan dari karya ilmiah, sedangkan Article adalah fokus utama dari hierarki ini, mewakili sebuah artikel jurnal lengkap.

Terakhir, ada atribut-atribut yang terkait dengan artikel seperti *Digital Object Identifier* (DOI), Keyword, dan URL. DOI adalah kode unik yang diberikan kepada setiap artikel untuk memudahkan identifikasi dan akses. Kata kunci membantu dalam pencarian dan pengindeksan artikel, sementara URL menyediakan alamat web di mana artikel dapat diakses. Hubungan antar kelas dalam hierarki ini ditunjukkan oleh garis-garis yang menghubungkan satu kelas dengan kelas lainnya. Garis-garis ini menunjukkan hubungan “*is-a*” atau “*kind-of*”, yang berarti kelas anak merupakan spesialisasi dari kelas induknya. Misalnya, kelas Article

adalah spesialisasi dari kelas Works, yang berarti setiap artikel adalah jenis karya ilmiah.

#### b. Data properties

Ontologi bibliografi mencakup 14 properti tipe data. *Data Properties* dalam ontologi bibliografi ini adalah properti yang berisi nilai literal, seperti angka, teks, atau tanggal. Data tersebut dapat dilihat pada Gambar 6. Properti tipe data ini penting karena memungkinkan representasi informasi yang lebih terstruktur dan terperinci dalam ontologi.

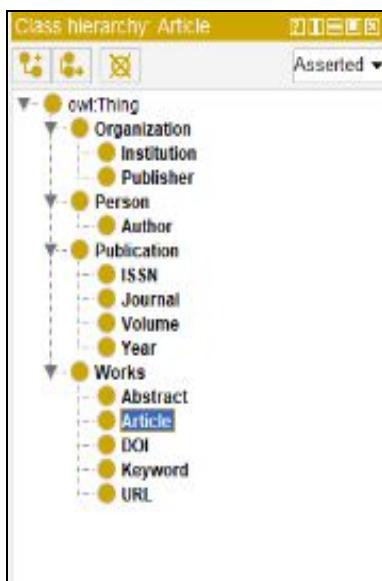
#### c. Object properties

*Object properties* adalah komponen yang menghubungkan kelas-kelas dalam sebuah ontologi. Melalui object properties ini, data properties dapat dihubungkan berdasarkan instance pada kelas tersebut. *Object properties* dalam ontologi bibliografi dapat dilihat pada Gambar 7.

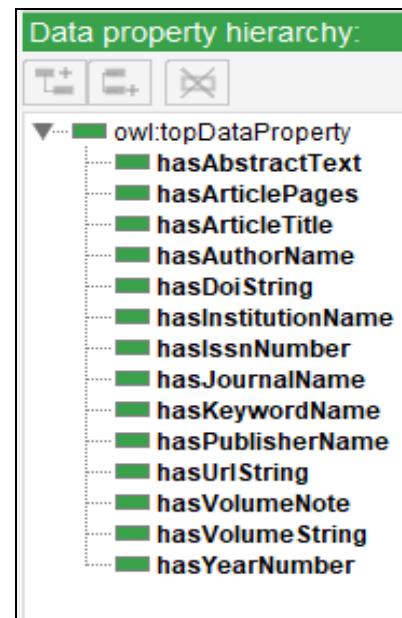
*Object properties* memungkinkan hubungan antara entitas yang berbeda dalam ontologi, seperti menghubungkan penulis dengan buku yang mereka tulis atau penerbit dengan buku yang mereka terbitkan. Ini membantu dalam menciptakan struktur data yang lebih kompleks dan bermakna, memungkinkan analisis dan pencarian informasi yang lebih efektif.

#### d. Implementasi

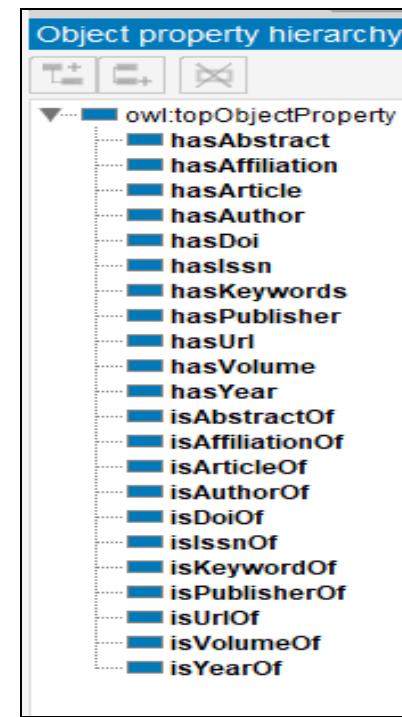
Dari prototipe yang ada di lakukan implementasi dengan penambahan teknologi yang ada dari struktur , Bahasa program JSP yaitu peng gabungan java dan html, framework serta database OWL dan juga desain style atau tampilan dan sebagainya menjadi sebuah aplikasi web yang berfungsi implementasi dari prototipe yang ada ditunjukan pada gambar 8 dan 9 yaitu tampilan form input dan tampilan hasil yang sudah berhasil diimplementasi.



Gambar 5. Class pada Ontology



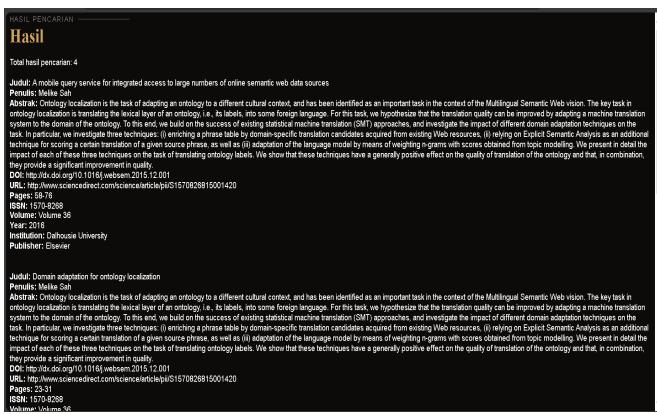
Gambar 6. Data properties pada Ontology



Gambar 7. Object properties pada Ontology



Gambar 8. Implementasi Tampilan Form Input



Gambar 9. Implementasi Tampilan Hasil

#### d. Tahapan Pengujian

Area pengujian mengambil bagian pengujian dari sistem pencarian informasi, yaitu memori dan pemantauan. Recall menggambarkan pengambilan informasi oleh sistem dan akurasi menggambarkan keakuratan informasi yang relevan. Kedua penilaian ini merupakan parameter yang dapat digunakan untuk menunjukkan relevansi sistem temu kembali informasi.

Relevansi menunjukkan kemampuan proses pencarian informasi, dan sistem dapat memberikan informasi yang sesuai dengan sudut pandang pengguna. Nilai recall (R) adalah nilai yang mewakili tingkat hasil yang dikembalikan oleh suatu sistem. Nilai recall (R) diperoleh dengan membandingkan jumlah data yang ditampilkan dengan jumlah data pada ontologi.

Nilai ini diperoleh seperti pada persamaan (1). Nilai *Recall* yang lebih tinggi tidak dapat menunjukkan apakah sistem tersebut baik atau tidak. Nilai *Recall*(R) maksimum adalah 1 yang menunjukkan bahwa semua dokumen dalam koleksi telah ditemukan. Nilai *Precision* (P) menunjukkan tingkat sistem yang tepat untuk mengembalikan informasi yang relevan kepada pengguna. Nilai *Precision* (P) diperoleh dengan membandingkan jumlah item yang dikembalikan sesuai dengan jumlah item yang dikembalikan. Nilai ini diperoleh seperti pada persamaan (2). semakin tinggi nilai akurasi suatu sistem maka dapat dikatakan sistem tersebut semakin baik.

Nilai F1score adalah ukuran evaluasi yang menggabungkan nilai-nilai yang benar dan mengingat satu nilai yang menggambarkan kualitas Model klasifikasi F1score berguna saat kita membutuhkannya. mencari keseimbangan antara *Precision* dan *Recall*, terutama jika kelas-kelas yang dihasilkan oleh model tidak seimbang Seperti pada persamaan (3)

$$R = \frac{\text{Jumlah dokumen relevan yang dikembalikan}}{\text{Jumlah semua dokumen relevan}} \times 100 \quad (1)$$

$$P = \frac{\text{Jumlah dokumen relevan yang dikembalikan}}{\text{Jumlah semua dokumen relevan}} \times 100 \quad (2)$$

$$F1 Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

Dari rumus perhitungan diatas didapatkanlah hasil sebagai berikut pada tabel I dan tabel II.

TABEL I  
PERHITUNGAN RECALL

No	Keyword	Data yang ditampilkan	Total data relevan pada ontologi	R
1	Judul	4	4	1
2	Penulis	2	2	1
3	Judul dan Penulis	4	2	0.5
<b>Rata -Rata R</b>				0.8

TABEL II  
PERHITUNGAN PRECISION

No	Keyword	Data yang ditampilkan	Total data relevan yang ditampilkan	p
1	Judul	4	4	1
2	Penulis	2	2	1
3	Judul dan Penulis	4	2	0.5
<b>Rata -rata P</b>				0.8

#### e. Pembahasan

Hasil uji *Recall* (R) menunjukkan nilai R yang diperoleh masih jauh dari 1 yaitu 0,8. Hal ini menunjukkan bahwa, bergantung pada jumlah informasi relevan yang disimpan

dalam ontologi jurnal, sistem cukup menemukan hampir seluruh rangkaian informasi relevan. Nilai *Precision* (P) Hasil pengujian menunjukkan nilai P adalah 0.8. Nilai presisi yang diperoleh menunjukkan bahwa seluruh informasi relevan telah cukup dikembalikan oleh sistem dari penjumlahan informasi relevan.

Hasil pengujian nilai *Recall* dan *Precision* menunjukkan relevansi sistem penelusuran jurnal ilmiah cukup tinggi. Sistem ini sangat efektif untuk memastikan bahwa pengguna dan sistem mengartikan kata kunci yang dimasukkan di kotak pencarian memiliki arti yang sama.

Dari hasil kedua nilai *Recall* dan *Precision* maka akan dilakukan perhitungan F1 score untuk menambah nilai keakuratan dari hasil pencarian dan dari hasil perhitungan nilai F1 diperoleh 0.8 dimana sistem berjalan cukup baik dalam melakukan pencarian.

#### IV. KESIMPULAN DAN SARAN

Pada penelitian ini telah berhasil dikembangkan sebuah aplikasi penelusuran jurnal ilmiah berbasis web semantik. Pengguna dapat memasukan kata kunci pencarian dan system akan menampilkan hasil pencarian yang disertai fitur penelusuran terhadap detail artikel.

Hasil pengujian menunjukkan bahwa sistem dapat memberikan hasil penelusuran dengan tingkat relevansi yang cukup tinggi. nilai Recall dan Precision adalah sebesar 0,9. Ini menunjukkan bahwa sistem mampu menemukan hampir seluruh informasi relevan yang ada dalam ontologi jurnal dan mengembalikan informasi tersebut dengan tingkat akurasi yang cukup tinggi. Nilai F1 score yang juga sebesar 0,9 mengindikasikan bahwa sistem berjalan dengan baik dalam melakukan pencarian dan memastikan bahwa kata kunci yang dimasukkan oleh pengguna diartikan dengan benar oleh sistem.

Diharapkan jika penelitian ini dikembangkan dapat memperluas lagi analisis ontologi dan pengembangan fitur dan system yang lebih baik lagi.

#### V. KUTIPAN

- [1] Abburu, S. (2012). A Survey on Ontology Reasoners and Comparison. International Journal of Computer Applications, 57(17), 33–39.

- [2] Yunizar Fahmi (2017). "Studi Tentang Pemodelan Ontologi Web Semantik Dan Prospek Penerapan Pada Bibliografi Artikel Jurnal Ilmiah" Universitas Islam Indonesia.
- [3] Prasetyo Imam Nugroho, Bayu Priyambadha, N. Y. S. (2018). Rancang Bangun Sistem Pencarian Koleksi Laporan Skripsi Dan PKL Dengan Teknologi Web Semantik (Studi Kasus: Ruang Baca Fakultas Ilmu Komputer Universitas Brawijaya). Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer,
- [4] Roger S. Pressman, Ph.D. "Software Engineering A Practitioner's Approach" (2001)
- [5] Johnson, P., Smith, R., & Lee, A. (2020). Improving search relevance using semantic web technologies. Journal of Information Systems, 45(2), 135-150.
- [6] Kaur, S., & Gupta, M. (2021). Context-aware information retrieval in digital libraries: A semantic approach. International Journal of Computer Science and Information Security, 19(4), 221-229.
- [7] Chen, H., Li, M., & Wang, Y. (2022). An ontology-based framework for improving data relevance in academic journal search engines. Journal of Knowledge Management, 36(5), 298-310.
- [8] Zhang, T., & Li, X. (2021). Semantic web technologies for intelligent academic search. Journal of Semantic Web Research, 12(3), 160-177.
- [9] Singh, R., & Saini, A. (2020). A review of semantic web approaches for improving web search engines. Journal of Information Retrieval and Web Intelligence, 13(6), 525-540.
- [10] Patel, S., & Shah, K. (2022). Enhancing search accuracy in academic databases using SPARQL and OWL. International Journal of Data Science and Semantics, 25(4), 412-425.
- [11] Bordes, A., Usunier, N., & Weston, J. (2021). World Wide Web Consortium: Semantic web and its applications. Journal of Web and Semantic Systems, 27(2), 90-102.



**Vaiye Efraim Jire Karame** lahir di Manado 9 November 2000 merupakan anak ke 2 dari 3 bersaudara. Penulis menempuh Pendidikan ditaman Kanak-kanak Mawar Sharon Karalung Siau Timur Sitaro (2005 – 2006) selanjutnya penulis melanjutkan ke jenjang sekolah dasar di SD GMIST Sion Karalung (2006-2012) dan melanjutkan ke sekolah menengah pertama di SMP Negeri 1 Siau Timur (2012 – 2015) setelahnya penulis melanjutkan ke sekolah menengah atas di SMA Negeri 1 Siau Timur (2015-2018) dan melanjutkan pendidikan S1 di Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Sam Ratulangi.