

Pengembangan Sistem *Frequently Asked Questions* (FAQ) pada Universitas Udayana Berbasis *Ontology*

Ngurah Kelvin Febryanta Lila Ananda^{a1}, Anak Agung Istri Ngurah Eka Karyawati^{a2}, Cokorda Rai Adi
Pramartha^{a3}, I Gusti Agung Gede Arya Kadyanan^{a4}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana

Jalan Raya Kampus Unud, Jimbaran, Bali, 80361, Indonesia

¹ngurahkelvin27@gmail.com

²eka.karyawati@unud.ac.id

³cokorda @unud.ac.id

⁴gungde@unud.ac.id

Abstract

Digital transformation in public services is a strategic step to enhance organizational efficiency and responsiveness in meeting user needs. This study aims to develop an ontology-based Frequently Asked Questions (FAQ) system to provide accurate, relevant, and easily accessible information at Udayana University. The system is designed using an ontology-based approach that represents relationships between concepts in a structured manner, combined with the Sentence Transformer (Sentence-BERT) method to understand variations in language and sentence structures in user queries. The study results show that the development of the ontology-based FAQ system successfully represents information in a structured and relevant manner according to user needs. The evaluation indicates a usability score of 90% and a usefulness score of 88.57%, reflecting a high level of user acceptance and system benefits. Out of 20 questions tested, the system correctly answered 16 questions, achieving an accuracy rate of 80%. Overall, this ontology-based FAQ system effectively meets user needs by providing relevant, accurate, and efficient answers while reducing the workload of administrative staff. The system has significant potential to support more comprehensive information services at Udayana University.

Keywords: *Ontology, FAQ System, Sentence-BERT, User Evaluation, Udayana University.*

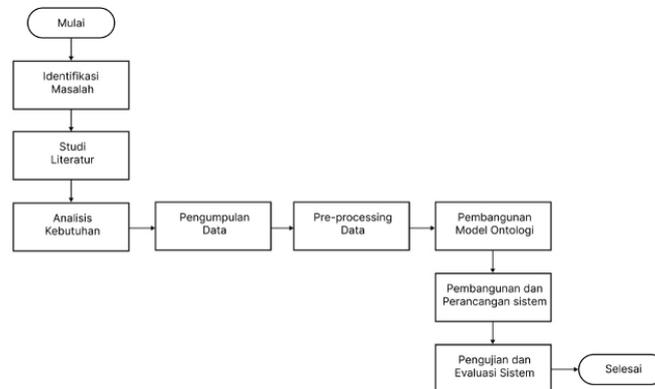
1. Pendahuluan

Transformasi digital layanan publik sangat penting untuk meningkatkan efisiensi dan responsivitas [1]. Organisasi yang sukses menyediakan informasi yang akurat, tepat waktu, dan relevan untuk memenuhi kebutuhan pengguna. Sistem FAQ (*Frequently Asked Question*) menangani hal ini dengan memberikan jawaban atas pertanyaan umum tanpa perlu interaksi langsung dengan staf [2]. Universitas Udayana, perguruan tinggi terkemuka di Bali, menghadapi tantangan dalam menyediakan informasi yang tepat waktu kepada penggunanya, seperti mahasiswa, staf, dan masyarakat. Saat ini, pertanyaan ditangani melalui interaksi dengan staf yang memakan waktu. Untuk mengatasi hal ini, diusulkan sistem informasi berbasis FAQ. Sistem ini berfungsi sebagai *Knowledge Management System* (KMS) bagi staf untuk mendokumentasikan pengetahuan organisasi dan juga sebagai alat untuk mengurangi pertanyaan yang berulang.

Dengan mengintegrasikan teknologi web semantik, sistem ini akan meningkatkan akurasi dan relevansi pencarian, mendukung penggunaan *ontology* untuk mengorganisasi pengetahuan. Penggunaan *Sentence-BERT* akan lebih meningkatkan sistem dengan memberikan pemahaman semantik, memungkinkan penanganan variasi bahasa alami dalam pertanyaan pengguna. Kombinasi sistem FAQ, *ontology*, dan *Sentence-BERT* akan menciptakan platform layanan mandiri yang efisien bagi pengguna dan mengurangi beban kerja administrasi.

2. Metode Penelitian

Pada metode pelaksanaan ini menjelaskan gambaran dari langkah-langkah yang akan Metodologi penelitian ini, dijelaskan uraian tentang prosedur yang akan dilakukan dalam penelitian ini. Proses-proses tersebut dapat ditemukan pada gambar 1 yang mencakup tahapan identifikasi permasalahan, analisis kebutuhan, preprocessing-data, pengumpulan data, pembangunan ontology, pembangunan model *ontology*, pembangunan dan perancangan sistem, serta pengujian dan evaluasi.



Gambar 1. Metode Pelaksanaan

Penelitian ini mengembangkan sistem FAQ berbasis ontology untuk meningkatkan akses informasi akademik di Universitas Udayana. Masalah utama yang diidentifikasi adalah keterlambatan respons sistem yang masih bergantung pada admin USDI Care. Dengan metode Methontology, sistem dirancang melalui tahapan spesifikasi, akuisisi pengetahuan, konseptualisasi, hingga implementasi [3]. Teknologi SBERT digunakan untuk mendeteksi kata kunci dalam pertanyaan, yang kemudian diterjemahkan ke dalam query SPARQL untuk mengambil jawaban dari database ontology. Pengujian menggunakan Black Box Testing dan Confusion Matrix menunjukkan bahwa sistem mampu meningkatkan efisiensi pencarian informasi, sementara evaluasi TAM mengonfirmasi kemudahan penggunaan serta manfaatnya bagi mahasiswa.

3. Hasil dan Diskusi

Hasil utama dari penelitian ini adalah menghasilkan sebuah website sistem *Frequently Asked and Questions* (FAQ) berbasis *ontology* yang dibangun dengan Langkah-langkah sebagai berikut.

3.1. Identifikasi Masalah

Tahapan pertama dalam penelitian ini yaitu mengidentifikasi dari permasalahan yang diangkat. Dimana tujuan dari dilakukannya penelitian ini adalah untuk menyelesaikan masalah yang ada. Permasalahan didalam penelitian ini adalah mahasiswa masih harus menunggu waktu lama untuk menanyakan suatu hal terkait akademik yang dimana, *system* itu sendiri belum akurat dan relevan. Walaupun sudah ada sistem yang dapat membantu menjawab pertanyaan yang dimana harus menunggu admin USDI *care* menjawabnya, dan belum adanya sistem FAQ bagi mahasiswa untuk dapat memudahkan mahasiswa dalam memberikan jawaban terkait Uang Kuliah Tunggal (UKT).

3.2. Analisis Kebutuhan

Tahap kedua dalam penelitian ini adalah menjelaskan mengenai analisis kebutuhan sistem. Analisis ini terdiri dari analisis kebutuhan fungsional dan analisis non-fungsional sebagai berikut.

Tabel 1. Analisis Kebutuhan Fungsional dan Non-Fungsional

No.	Kebutuhan Fungsional	Kebutuhan Non-Fungsional
1	Sistem yang dibangun memungkinkan pengguna untuk mengetahui pertanyaan yang sering ditanyakan oleh pengguna lainnya.	Hardware Requirement : Perangkat keras yang dibutuhkan yaitu laptop atau komputer yang dapat digunakan untuk membangun ontology serta sistem berbasis web.
2	Sistem yang dibangun dengan adanya fitur pencarian (<i>searching</i>) FAQ, sistem memiliki kemampuan untuk melakukan pencarian informasi yang lebih terperinci sesuai dengan keterkaitan dan makna antara konsep-konsep dalam domain FAQ.	Software Requirement : <ul style="list-style-type: none"> • Protégé 5.5.0 Build Beta-9 dan webprotege.stanford.edu untuk membuat, mengedit, dan menyimpan model ontology. • Apache Jena Fuseki sebagai server untuk mengeksekusi SPARQL dalam mengolah data RDF. • Visual Studio Code sebagai text editor dalam pembangunan program berbasis web.

3.3. Pengumpulan Data

Proses pengumpulan data dalam sistem ini terdiri dari dua tahap, yaitu pengumpulan data untuk ontology dan pengumpulan data untuk pengujian serta evaluasi. Berikut ini adalah penjelasan mengenai proses pengumpulan data yang dilakukan dalam penelitian ini.

a. Pengumpulan Data Ontology

Mengumpulkan data untuk membangun model ontology pada domain *Frequently Asked and Questions* (FAQ). Dalam pengumpulan data ontology, peneliti mendapatkan data dengan melakukan pengajuan permohonan permintaan data pertanyaan serta jawaban kepada petugas Unit Sumber Daya Informasi (USDI) dengan mengajukan surat permohonan. Data yang didapat peneliti sebanyak 2.442 data pertanyaan dan jawaban. Berdasarkan data yang diperoleh, mulai dari tahap pengumpulan data awal hingga tahap pengumpulan data untuk pengembangan model ontology, peneliti menetapkan sejumlah kriteria yang akan digunakan dalam sistem ini. Kriteria tersebut meliputi beberapa aspek berikut.

1. Jawaban.
2. Pertanyaan.
3. Kategori.
4. *Keywords*.

b. Pengumpulan Data Evaluasi

Pengumpulan data untuk evaluasi sistem FAQ dilakukan melalui media sosial dengan menggunakan kuesioner Google Form. Kuesioner tersebut dirancang berdasarkan konsep *Technology Acceptance Model* (TAM) dan mencakup aspek Persepsi Kemanfaatan (*Perceived Usefulness*) serta Persepsi Kemudahan Penggunaan (*Perceived Ease of Use*).

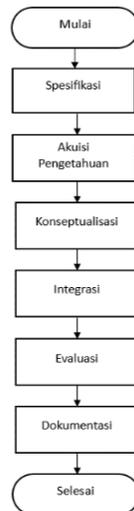
3.4. Pre – Processing

Data yang diperoleh sejumlah 1334 kemudian dilakukan pre-processing meliputi penghapusan data duplikat dan penghapusan data yang mengandung nilai kosong, *lowercasing* data guna untuk meningkatkan akurasi dalam model, lalu pembersihan karakter disetiap kalimat. Tujuannya adalah untuk membuat representasi identik dari kata-kata yang terkait dengan kapitalisasi yang berbeda, seperti “UKT” dan “ukt”, sehingga akurasi model dalam memproses teks bisa lebih konsisten. Selain itu, teknik pembersihan karakter menggunakan *regular expression* untuk menghapus elemen-elemen seperti tanda baca, simbol, atau tag html, sehingga meningkatkan performa model dalam mengelola data. Idenya adalah mengurangi gangguan dalam analisis, memudahkan proses tokenisasi, dan memungkinkan model berkerja secara lebih baik dengan data yang diberikan. Serta dilakukan

tokenisasi, penyaringan data, klasifikasi kategori, dan ekstraksi kata kunci untuk memecah kalimat menjadi potongan-potongan kata, sehingga mempermudah proses analisis.

3.5. Membangun Model Ontology

Bagian ini membahas proses implementasi pembangunan ontology pada domain FAQ menggunakan metode *Methontology* dan metode SBERT untuk klasifikasi pertanyaan. *Methontology* memiliki kemampuan untuk melakukan life cycle ontology yang didasarkan pada pengembangan prototype yang mengijinkan untuk melakukan penambahan, perubahan, dan penghapusan terms pada tiap versi terbarunya[4].



Gambar 2. Diagram alur Metode *Methontology*

a. Pengembangan Model Ontology Dengan Methontology

1. Spesifikasi

Pada tahapan ini sudah dihasilkan spesifikasi ontology domain buku dengan deskripsi sebagai berikut:

- Domain : *Frequently Aksed and Question (FAQ)*
- Tanggal : 27 November 2024
- Dirancang Oleh : Ngurah Kelvin Febryanta Lila Ananda
- Diimplementasikan oleh : Ngurah Kelvin Febryanta Lila Ananda
- Tujuan : Membangun model *ontology* sebagai dasar Pengembangan Sistem *Frequeuntly Asked and Quenstion (FAQ)* Pada Universitas Udayana Berbasis Ontology
- Tingkat formalitas : Formal
- Ruang lingkup : Universitas Udayana
- Sumber pengetahuan : Unit Sumber Daya Informasi (USDI) Universitas Udayana

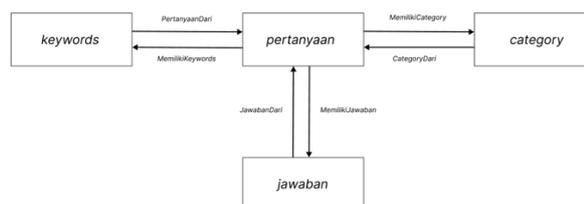
2. Akuisisi Pengetahuan

Pada tahap ini, proses akuisisi pengetahuan untuk *ontology* pada domain FAQ yang dilakukan oleh peneliti mencakup beberapa langkah berikut:

- Berdiskusi dengan dosen pembimbing untuk menyusun dokumen draft awal dokumen spesifikasi persyaratan
- Melakukan analisis terhadap teks formal dengan cara mengidentifikasi struktur yang dicari serta mengelompokkan informasi yang terkandung dalam setiap konsep, atribut, nilai, dan hubungan yang terkait.

3. Konseptualisasi

Tahap konseptualisasi ontology bertujuan untuk mengorganisasi dan mengelola pengetahuan yang diperoleh selama proses akuisisi pengetahuan [5] yang Dimana memanfaatkan konsep *triplestore*. Setelah model konseptual selesai dibuat, metodologi ini akan mengubahnya menjadi model formal.



Gambar 3. Konseptualisasi Pada Ontology FAQ

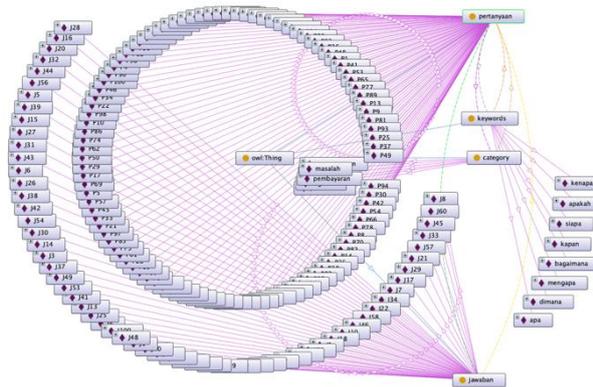
4. Integrasi

Dalam pembangunan model *ontology* FAQ dengan studi kasus pertanyaan di Universitas Udayana, tahap integrasi tidak dilakukan karena ini merupakan pertama kalinya *ontology* FAQ dikembangkan dan belum ada model *ontology* serupa yang relevan.

5. Implementasi

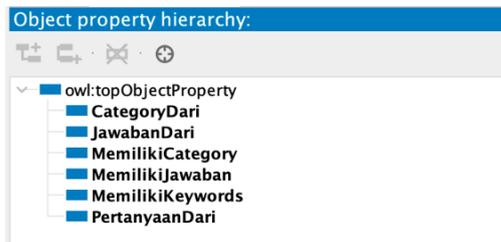
Pada tahap implementasi merupakan proses penerapan model *ontology* yang telah dikembangkan pada tahap konseptualisasi. Setiap elemen *ontology* dirancang berdasarkan langkah-langkah Methontology, di mana kelas mewakili konsep, object properties mendefinisikan hubungan antar konsep, datatype properties digunakan untuk atribut, dan individu merepresentasikan instance. Class merupakan semua entitas yang

tergolong dalam kelompok atau keluarga objek. Model ontology ini kemudian diimplementasikan menggunakan Protégé untuk domain FAQ.



Gambar 4. Ontograp pada *ontology* FAQ

Pada gambar 4 terdapat 4 *class* utama yang digunakan dalam *ontology* ini yaitu pertanyaan, kategori, *keywords* dan jawaban yang divisualisasikan dengan ontograp.



Gambar 5. *Object property* pada *ontology* FAQ

Gambar 5 menampilkan *object property* dalam ontology domain FAQ, di mana individu atau instance direpresentasikan berdasarkan pola *triplestore*. *Triplestore* terdiri dari tiga elemen utama: Subjek, Predikat, dan Objek, yang berfungsi sebagai pernyataan tetap dalam ontology. Predikat berperan sebagai penghubung antar objek dalam sistem.

6. Evaluasi
Proses evaluasi *ontology* melibatkan verifikasi untuk memastikan kecocokan dengan kerangka acuan dan validasi untuk memastikan kesesuaian dengan spesifikasi sistem. Pada aplikasi Protégé, alat Reasoner HermiT versi 1.3.6 digunakan untuk memeriksa konsistensi ontology.
7. Dokumentasi
Tidak terdapat pedoman baku mengenai cara mendokumentasikan *ontology*. Dalam praktiknya, dokumentasi *ontology* seringkali hanya tersedia dalam bentuk kode *ontology*,

deskripsi dalam bahasa alami, serta publikasi seperti makalah dalam prosiding konferensi atau jurnal ilmiah.

Ontology metrics:	
Metrics	
Axiom	926
Logical axiom count	739
Declaration axioms count	187
Class count	4
Object property count	6
Data property count	3
Individual count	174
Annotation Property count	0
Class axioms	
SubClassOf	0
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0
Object property axioms	
SubObjectPropertyOf	0
EquivalentObjectProperties	0
InverseObjectProperties	3
DisjointObjectProperties	0
FunctionalObjectProperty	0
InverseFunctionalObjectProperty	0
TransitiveObjectProperty	0
...	^

Gambar 6. Metrics Ontology Domain FAQ

3.6. Pembangunan dan perancangan system

Bagian ini menjelaskan implementasi sistem untuk penelusuran dan pencarian informasi buku yang dikembangkan menggunakan metode *Prototyping*. Proses implementasi sistem akan diuraikan berdasarkan tahapan-tahapan yang telah dirancang sebelumnya.

a. Kebutuhan system

Beberapa perangkat lunak dan *library* yang digunakan dalam pengembangan sistem pencarian dan rekomendasi buku adalah sebagai berikut.

1. MacOS Sonoma 14.2.1
2. Protege 5.6.4
3. JavaScript v20.9.0
4. Next JS 15
5. Visual Studio Code
6. Tailwind CSS
7. Apache Jena Fuseki 5.2.0
8. Apache J Meter 5.6.3
9. MYSQL
10. Google Chrome
11. Google Colab

b. Arsitektur Sistem

Arsitektur sistem (*system architecture*) merupakan susunan atau kerangka yang digunakan dalam merancang, menghubungkan, dan mengelola berbagai komponen yang membentuk suatu sistem.

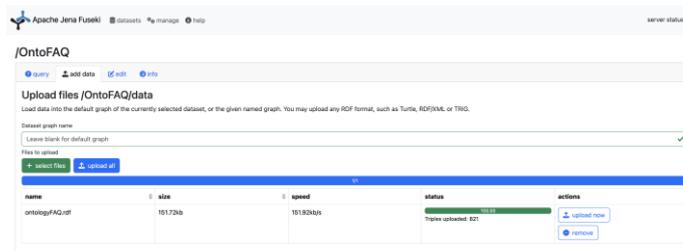


Gambar 7. Arsitektur sistem FAQ

Arsitektur sistem FAQ berbasis *ontology* dengan metode SBERT terdiri dari beberapa komponen utama, termasuk antarmuka pengguna, *ontology*, SBERT, triple store, dan web server. *Ontology* berperan sebagai basis pengetahuan, sementara SBERT bertugas mengidentifikasi serta mencocokkan pertanyaan dengan data yang tersedia. *Triple store*, seperti Apache Jena Fuseki, digunakan untuk menjalankan query SPARQL, sedangkan web server berbasis *flask* mengatur komunikasi antara UI dan *backend*. Seluruh sistem berjalan di atas sistem operasi, memastikan informasi yang diberikan akurat dan sesuai dengan konteks pertanyaan pengguna.

c. Implementasi Model *Ontology* ke Sistem

Pada tahap implementasi, *ontology* domain buku diintegrasikan ke dalam sistem dengan mengunggahnya ke server Apache Jena Fuseki. Server Fuseki berfungsi untuk mengelola dan menyimpan data *ontology*, sehingga *ontology* tersebut dapat diakses dan digunakan oleh sistem. File *ontology* yang diunggah berformat OWL. Proses pengunggahan *ontology* ke server Fuseki ditunjukkan pada Gambar 8.



Gambar 8. Proses Unggah File *Ontology* ke Server Jena Fuseki

Setelah proses pengunggahan selesai, langkah berikutnya adalah menghubungkan *ontology* dengan sistem menggunakan CURL untuk berkomunikasi dengan *endpoint* SPARQL. Di bawah ini adalah kode sumber untuk proses koneksi antara *ontology* dan sistem.

d. Implementasi SBERT

Implementasi sistem FAQ berbasis *ontology* dengan metode SBERT terdiri dari beberapa tahapan utama. SBERT digunakan untuk menghasilkan *embedding* pertanyaan pengguna dan dataset, memungkinkan perbandingan semantik menggunakan *Cosine Similarity*[6]. Model ini diimplementasikan dalam kelas *QuestionClassifier*, yang bertugas memuat serta memproses dataset, menghitung *embedding* pertanyaan, dan mengklasifikasikan input berdasarkan kesamaan dengan data yang tersimpan. Ketika pengguna mengajukan pertanyaan melalui *endpoint/classify*, sistem mengidentifikasi kategori menggunakan SBERT, menghitung skor kesamaan, dan mengembalikan hasil dalam format JSON. Selanjutnya, *query* SPARQL dibuat berdasarkan kategori dan kata kunci 5W1H yang terdeteksi untuk mengambil informasi dari *ontology* yang tersimpan di server Fuseki. Berikut adalah kode SPARQL yang digunakan untuk mengekstrak informasi yang relevan:

Tabel 2. Code SPARQL Query

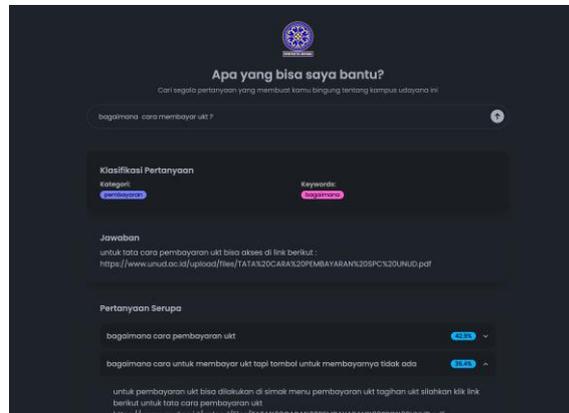
```
SELECT ?isi ?jawaban
WHERE {
  ?question rdf:type f:pertanyaan.
  ?question f:MemilikiCategory f:${res.data.predicted_category}.
  ?question f:MemilikiKeywords f:${res.data.detected_5w1h_keywords}.
  ?question f:isi ?isi.
  ?question f:jawaban ?jawaban.
}
```

Melalui proses ini, sistem mampu menyaring hasil pencarian secara efektif dan hanya menampilkan jawaban yang paling relevan. Setelah Fuseki mengeksekusi *query* SPARQL, aplikasi memproses data untuk menemukan jawaban yang paling sesuai dengan

pertanyaan pengguna, sehingga meningkatkan efisiensi dan akurasi dalam sistem FAQ berbasis *ontology*.

e. Implementasi Antarmuka Sistem

Dalam implementasi antarmuka, desain antarmuka yang telah dirancang sebelumnya akan direalisasikan menggunakan bahasa pemrograman HTML dengan *framework* Next JS serta CSS menggunakan Tailwind. Berikut ini adalah penjelasan mengenai hasil implementasi antarmuka sistem yang telah dikembangkan.



Gambar 9. Tampilan Antarmuka Sistem

Gambar diatas sistem FAQ ini. Dimana pengguna akan diminta meinputkan pertanyaan mengenai kampus udayana khususnya mengenai UKT. Setelah itu hasil dari inputan user dimana akan menampilkan klasifikasi pertanyaan berupa kategori serta *keywords*. Jawaban yang ditampilkan sesuai dengan apa yang ditanyakan oleh pengguna.

3.7. Pengujian dan Evaluasi

a. Pengujian

Pengujian mencakup fitur utama, seperti klasifikasi kategori pertanyaan dan deteksi kata kunci 5W1H, untuk memastikan kinerja sistem sesuai ekspektasi. Evaluasi mencakup pengukuran akurasi, precision, recall, dan F1-score menggunakan *Confusion Matrix* dan *Classification Report*, serta *precision*, *recall*, dan *F1-score* untuk deteksi kata kunci. Untuk mengoptimalkan kinerja dan menguji ketahanan, pengujian stress API dilakukan menggunakan Apache JMeter.

1. Stress Testing

Setelah sebuah sistem selesai di bangun, maka dilakukanlah sebuah pengujian stress terhadap API (*Application Programing Interface*) dengan menggunakan Apache JMeter

[7]. Testing ini dilakukan untuk lebih mengoptimalkan sistem dan memeriksa ketahanan dari sistem.

Sample #	Start Time	Thread Name	Label	Sample Time(s)	Status	Bytes	Sent Bytes	Latency	Connect Time(s)
1	15:50:20.879	Thread Group 1..	HTTP Request	2	200	17832	548	2	0
2	15:50:20.905	Thread Group 1..	HTTP Request	3	200	17832	548	2	1
3	15:50:20.719	Thread Group 1..	HTTP Request	2	200	17832	548	2	1
4	15:50:20.739	Thread Group 1..	HTTP Request	2	200	17832	548	2	1
5	15:50:20.793	Thread Group 1..	HTTP Request	3	200	17832	548	3	1
6	15:50:20.783	Thread Group 1..	HTTP Request	2	200	17832	548	2	0
7	15:50:20.800	Thread Group 1..	HTTP Request	3	200	17832	548	3	0
8	15:50:20.802	Thread Group 1..	HTTP Request	2	200	17832	548	2	0
9	15:50:20.843	Thread Group 1..	HTTP Request	2	200	17832	548	2	0
10	15:50:20.883	Thread Group 1..	HTTP Request	2	200	17832	548	2	0
11	15:50:20.883	Thread Group 1..	HTTP Request	3	200	17832	548	3	1
12	15:50:20.905	Thread Group 1..	HTTP Request	3	200	17832	548	3	1
13	15:50:20.803	Thread Group 1..	HTTP Request	4	200	17832	548	4	1
14	15:50:20.842	Thread Group 1..	HTTP Request	4	200	17832	548	4	1
15	15:50:20.809	Thread Group 1..	HTTP Request	8	200	17832	548	7	3
16	15:50:20.880	Thread Group 1..	HTTP Request	4	200	17832	548	4	1
17	15:50:21.003	Thread Group 1..	HTTP Request	4	200	17832	548	4	0
18	15:50:21.003	Thread Group 1..	HTTP Request	5	200	17832	548	5	1
19	15:50:21.043	Thread Group 1..	HTTP Request	4	200	17832	548	4	1
20	15:50:21.046	Thread Group 1..	HTTP Request	3	200	17832	548	3	0
21	15:50:21.063	Thread Group 1..	HTTP Request	3	200	17832	548	3	0
22	15:50:21.103	Thread Group 1..	HTTP Request	2	200	17832	548	2	0
23	15:50:21.053	Thread Group 1..	HTTP Request	1	200	17832	548	2	0

Gambar 10. Tampilan API Testing Respons Jmaster

2. Confusion Matrix

Hasil evaluasi model menunjukkan akurasi klasifikasi kategori sebesar 78,57, dengan beberapa kategori, seperti "lainnya" dan "uncategorized", memiliki *precision*, *recall*, dan *F1-score* rendah, kemungkinan akibat ketidak seimbangan data. Kategori "pembayaran" menunjukkan hasil sangat baik (*precision* 0.91, *recall* 0.88, *F1-score* 0.90), sementara kategori lain seperti "pengurangan" dan "tagihan" memiliki performa lebih rendah. Rata-rata akurasi makro menunjukkan *precision* 0.46, *recall* 0.39, dan *F1-score* 0.40, dipengaruhi kategori dengan hasil buruk, sementara rata-rata tertimbang lebih baik (*precision* 0.83, *recall* 0.79, *F1-score* 0.80) karena kontribusi kategori unggul seperti "pembayaran".

Tabel 3. Tampilan Precision, recall, f1-score

kategori	precision	recall	f1-score	support
keringanan	1.00	0.61	0.76	18
lainnya	0.00	0.00	0.00	0
masalah	0.67	0.57	0.62	7
pembayaran	0.91	0.88	0.90	213
pengurangan	0.50	0.29	0.36	7
tagihan	0.13	0.40	0.20	10
uncategorized	0.00	0.00	0.00	11
Accuracy	0.79			

Model mendeteksi kata kunci 5W1H dengan cukup baik, mencapai *precision*, *recall*, dan *f1-score* sebesar 0.8540. Meskipun efektif, masih ada ruang untuk perbaikan, terutama dalam meningkatkan *recall* dan *precision* pada kategori yang sulit dikenali. Secara keseluruhan, model menunjukkan kinerja yang memadai, namun perlu optimalisasi untuk meningkatkan akurasi deteksi kata kunci. Hasil selengkapnya ditampilkan pada tabel 4.

Tabel 3. Tampilan Precision, recall, f1-score

Matrik evaluasi	Nilai
Precision	0.854
Recall	0.854
f-1 score	0.854

3. Pengujian pertanyaan

Dari 20 pertanyaan yang diuji, sistem berhasil menjawab 16 dengan benar, menghasilkan tingkat keberhasilan 80%, dan kegagalan 20%. Ini menunjukkan bahwa sistem dapat

memahami dan menjawab sebagian besar pertanyaan sesuai kategori dan *keywords* yang ditentukan.

4. Blackbox Testing

Tabel 2. Pengujian Blackbox

No.	Test Case	Hasil
1	<ul style="list-style-type: none"> Skenario : pengguna membuka sistem FAQ Tes : aplikasi akan menampilkan halaman <i>landing page</i> sistem yang terdapat judul halaman serta inputan user berupa pertanyaan yang akan diajukan 	Berhasil
2	<ul style="list-style-type: none"> Skenario : pengguna mengisi pertanyaan di bar <i>search</i> Tes : pengguna dapat mengisi pertanyaan 	Berhasil
3	<ul style="list-style-type: none"> Skenario : pengguna tidak mengisi input pertanyaan Tes : sistem akan memberikan peringatan pertanyaan tidak ditemukan 	Berhasil
4	<ul style="list-style-type: none"> Skenario : pengguna mengisi pertanyaan yang valid degan topik UKT Tes : sistem akan menampilkan klasifikasi pertanyaan berupa <i>Category</i> serta <i>Keywords</i> 	Berhasil

Hasil pengujian terhadap Pengujian dilakukan dengan metode Blackbox Testing sesuai skenario dalam Tabel 2. Dapat dilihat dari pengujian ini hasil seluruh adalah berhasil atau valid dapat berjalan dengan baik.

a. Evaluasi Sistem

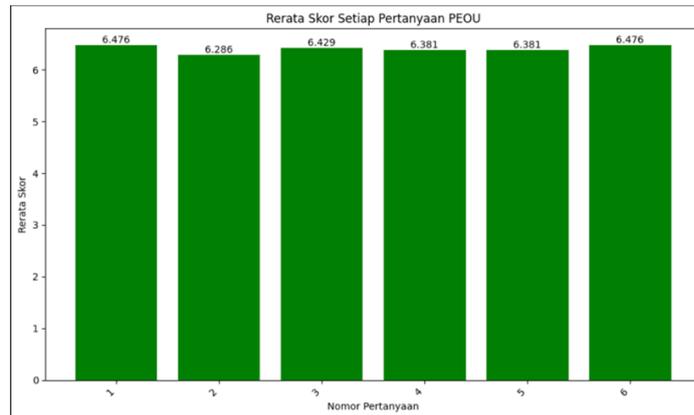
Pada tahap evaluasi sistem ini, data dari penilaian responden terhadap sistem diolah dan dianalisis. Proses ini mencakup dua aspek utama, yaitu evaluasi kemudahan penggunaan sistem dan evaluasi kegunaannya, sesuai dengan konsep *Technology Acceptance Model* (TAM) [8]. Untuk mengolah hasil kuesioner, dilakukan analisis statistik menggunakan google colab.

Setelah dilakukannya pengolahan data, didapat hasil PU dan PEOU sebagai berikut:

1. Hasil PEOU (*Perceived Ease Of Use*)

- Nilai rata-rata (*mean*): Analisis menunjukkan bahwa skor rata-rata yang diberikan oleh responden untuk seluruh pertanyaan adalah 6,4, yang setara dengan 90% jika diubah ke dalam bentuk persentase. Ini menandakan bahwa mayoritas responden menyatakan "sangat setuju" dengan kemudahan penggunaan sistem.
- Nilai tengah (*median*): Nilai median didapatkan dengan cara mengurutkan skor dan memilih angka yang ada di tengah. Dari seluruh responden, median yang didapat adalah 6, yang berarti "setuju" dalam skala *likert*. Ini menunjukkan bahwa sebagian besar peserta memiliki tingkat persetujuan yang cukup tinggi terhadap kemudahan sistem.
- Nilai terkecil (*minimum*): Skor terendah yang didapat dalam evaluasi kemudahan adalah 5, yang dalam skala Likert berarti "agak setuju." Ini menunjukkan bahwa responden yang memberikan nilai ini masih menyatakan persetujuan, meskipun tidak sekuat responden lainnya.
- Nilai terbesar (*maximum*): Hasil analisis menunjukkan bahwa skor tertinggi yang diberikan oleh responden adalah 7, yang pada skala *likert* berarti "sangat setuju."

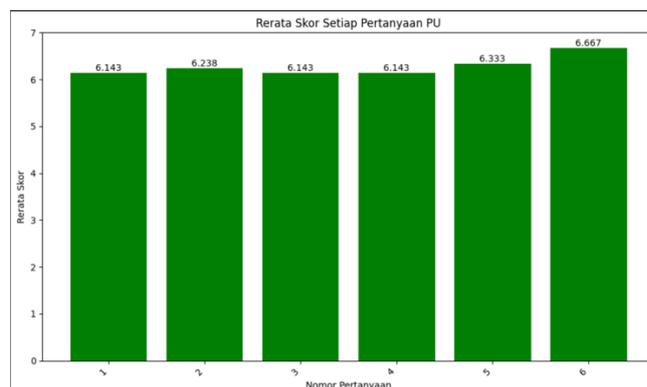
Skor ini mencerminkan tingkat persetujuan tertinggi dari peserta terkait kemudahan penggunaan sistem.



Gambar 13. Diagram Hasil PEOU

2. Hasil PU (*Perceived Usefulness*)

- Nilai rata-rata (*mean*): Berdasarkan analisis, rata-rata skor yang diberikan responden untuk semua pertanyaan adalah 6,2, yang jika diubah ke persentase setara dengan 88,57%. Ini menunjukkan bahwa mayoritas responden setuju dengan kegunaan sistem yang digunakan.
- Nilai tengah (*median*): Nilai median didapatkan dengan mengurutkan skor dan memilih angka yang ada di tengah. Dari hasil analisis, median yang diperoleh adalah 6, yang dalam skala Likert berarti "setuju." Hal ini menunjukkan bahwa sebagian besar peserta memberi penilaian positif terhadap kegunaan sistem.
- Nilai terkecil (*minimum*): Skor terendah yang didapat dalam penilaian kegunaan sistem adalah 5, yang pada skala Likert berarti "agak setuju." Ini menunjukkan bahwa tingkat persetujuan yang paling rendah dari responden terhadap kegunaan sistem masih cukup positif.
- Nilai terbesar (*maximum*): Analisis mengungkapkan bahwa skor tertinggi yang diberikan responden adalah 7, yang pada skala Likert berarti "sangat setuju." Nilai ini menunjukkan bahwa sebagian peserta sangat setuju dengan penilaian mereka terhadap kegunaan sistem.



Gambar 14. Diagram Hasil PU

4. Kesimpulan

Dalam Perancangan Sistem Informasi Buku dapat diambil kesimpulan sebagai berikut:

- a. Hasil penelitian ini merancang sistem FAQ berbasis ontologi yang menyajikan informasi secara terstruktur dan saling terhubung. Dengan ontologi sebagai kerangka pengetahuan, sistem dapat merepresentasikan hubungan antar konsep secara jelas, memastikan informasi yang disediakan sesuai dengan kebutuhan pengguna. Hasil evaluasi menunjukkan tingkat kemudahan penggunaan sebesar 90% dan kegunaan 88,57%, yang mengindikasikan bahwa sistem ini efektif dalam memberikan jawaban yang akurat. Secara keseluruhan, sistem mampu

menjawab pertanyaan dengan tepat serta menyusun informasi sesuai dengan struktur ontologi yang telah dikembangkan.

- b. Berdasarkan hasil pengujian menunjukkan bahwa sistem dapat menjawab 16 dari 20 pertanyaan dengan benar, sementara 4 lainnya belum sesuai harapan. Keberhasilan sistem dipengaruhi oleh kategori pertanyaan, penggunaan kata kunci, dan struktur kalimat. Kinerja terbaik tercapai pada kategori pembayaran dan tagihan dengan skor di atas 0.85, sedangkan kategori pengurangan UKT memerlukan perbaikan. Secara keseluruhan, sistem cukup efektif dalam menangani permasalahan UKT, meskipun masih terdapat aspek yang perlu dikembangkan.

Referensi

- [1] Budiman, A. E., & Widjaja, A. (2020). Analisis pengaruh teks preprocessing terhadap deteksi plagiarisme pada dokumen tugas akhir. *JuTISI*, 6(3), 1-10.
- [2] Yunaningsih, A., Indah, D., & Septiawan, F. E. (2021). Upaya Meningkatkan Kualitas Layanan Publik Melalui . *Jurnal ALTASIA*, 9-16.
- [3] Pramatha, C., Koten, I., Putra, I. G. N. A. C., Supriana, I. W., & Arka, I. W. (2022). Pengembangan Sistem Dokumentasi Melalui Pendekatan Ontology untuk Praktek Budaya Bali. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 11(3), 259–268. <https://doi.org/10.23887/janapati.v11i3.53939>
- [4] Purnomo, D. (2017). Model Prototyping Pada Pengembangan. *JIMP - Jurnal Informatika Merdeka Pasuruan*, 54-61.
- [5] Novianti, K. D. P., & Wibawa, M. S. (2017). Ontology Model untuk Tourist Information Retrieval. *Konferensi Nasional Sistem & Informatika 2017*, 165.
- [6] Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-*
- [7] Sabella, B., Lestari, A., Rahma Tina, R., & Achmad, F. (2024). Pengujian Aplikasi SIHARAPAN Menggunakan Metode Stress Testing. In *Jurnal Teknologi Informasi dan Terapan (J-TIT)* (Vol. 11, Issue 1). <https://doi.org/10.25047/jtit.v11i1.359>
- [8] Yani, E., Lestari, A. F., Amalia, H., & Puspita, A. (2018). Pengaruh Internet Banking Terhadap Minat Nasabah Dalam Bertransaksi Dengan Technology Acceptance Model. *JURNAL INFORMATIKA*, 34-42.

This page is intentionally left blank.