IDENTIFIKASI KATA KUNCI PADA KONTEN PUBLIKASI JURNAL ILMIAH UNTUK STUDI KASUS PENCARIAN PUBLIKASI ONLINE ITS (POMITS)

Abdul Munif[1)], Nurul Fajrin Ariyani[2)], dan Khairunnisa' Rahma Mardiyani[3)]
[1, 2, 3)]Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember
Jl. Teknik Kimia, Surabaya, Jawa Timur, Indonesia
e-mail: munif@if.its.ac.id[1)], nurulfajrin@if.its.ac.id[2)], mardiyani16@mhs.if.its.ac.id[3)]

ABSTRAK

Publikasi Online ITS (POMITS) adalah jurnal yang diperuntukkan sebagai jurnal publikasi bagi mahasiswa program sarjana ITS. Artikel yang terbit di dalamnya sudah cukup banyak dan seringkali diperlukan sebagai bahan referensi untuk penelitian mahasiswa lainnya. Proses pencarian yang ada saat ini masih berdasarkan judul, abstrak, nama penulis, dan kata kunci. Data-data tersebut masih dimasukkan secara manual oleh penulis. Proses ini memungkinkan adanya pemilihan kata kunci yang kurang sesuai. Sehingga diperlukan suatu upaya agar pemilihan kata kunci tersebut bisa lebih tepat dan merepresentasikan artikel tersebut.

Tujuan dari penelitian ini adalah melakukan identifikasi kata kunci dalam artikel secara otomatis. Kata kunci tersebut dibedakan menjadi perangkat lunak yang digunakan, metode, dan kata kunci lain yang representatif. Dengan adanya identifikasi ini, pencarian artikel dapat mengembalikan hasil pencarian yang lebih tepat. Masalah ini dapat diatasi dengan menggunakan Named Entity Recognition (NER). Namun, model NER bahasa Indonesia yang dimiliki SpaCy masih belum tersedia, maka diperlukan pembangunan model NER tersebut.

Dalam penelitian ini, identifikasi setiap anotasi kata kunci pada konten POMITS menjadi metadata dilakukan dengan mendeteksi named entity berupa perangkat lunak, metode, dan kata kunci representatif menggunakan model NER. Hasil anotasi NER disimpan dalam bentuk pasangan triplets pada triple store Apache Jena Fuseki. Selanjutnya, triple store tersebut dapat digunakan untuk menjawab pencarian tentang perangkat lunak, metode, dan kata kunci. Berdasarkan hasil pengujian, sistem berhasil mendeteksi entitas NER serta menyimpan anotasi dalam bentuk pasangan triplets pada Apache Jena Fuseki. Identifikasi kata kunci menghasilkan rata-rata nilai presisi 84,76% dan recall 63.59%.

***Kata Kunci**: anotasi, Named Entity Recognition (NER), triple store, Apache Jena Fuseki.*

KEYWORD IDENTIFICATION IN SCIENTIFIC JOURNAL PUBLICATION CONTENT FOR CASE STUDY ITS ONLINE PUBLICATION (POMITS) SEARCHING

Abdul Munif[1)], Nurul Fajrin Ariyani[2)], and Khairunnisa' Rahma Mardiyani[3)]
[1, 2, 3)]Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember
Jl. Teknik Kimia, Surabaya, Jawa Timur, Indonesia
e-mail: munif@if.its.ac.id[1)], nurulfajrin@if.its.ac.id[2)], mardiyani16@mhs.if.its.ac.id[3)]

ABSTRACT

ITS Online Publication (POMITS) is a publication journal for ITS undergraduate students. Many articles are published in it, and they are often needed as reference material for other student research. The search process is still based on title, abstract, author's name, and keywords. The data is still entered manually by the author. This process allows the selection of less appropriate keywords. So an effort is needed so that the choice of these keywords can be more precise and represent the article.

The purpose of this research is to identify keywords in articles automatically. These keywords are distinguished into the software used, methods, and other representative keywords. With this identification, article searches can return more precise search results. This problem can be solved by using Named Entity Recognition (NER). However, the Indonesian language NER model owned by SpaCy is still not available, so it is necessary to develop the NER model.

This study identifies each keyword annotation in POMITS content into metadata by detecting named entities in the form of software, methods, and representative keywords using the NER model. The NER annotation results are stored as triplet pairs in the Apache Jena Fuseki triple store. Furthermore, the triple store can answer searches about software, methods, and keywords. Based on the test results, the system successfully detects NER entities and saves annotations as triplet pairs on Apache Jena Fuseki. Keywords identification produce an average value of 84.76% precision and 63.59% recall.

***Keywords**: annotation, Named Entity Recognition (NER), triple store, Apache Jena Fuseki.*

http://dx.doi.org/10.12962/j24068535.v21i1.a1

I. PENDAHULUAN

Di era modern ini, kebutuhan dalam bidang ilmiah semakin meningkat, jumlah publikasi ilmiah semakin lama semakin banyak. Karena dalam publikasi ilmiah, ilmuwan dan praktisi saling berbagi dan mencari informasi tentang sumber data, metode pemrosesan, dan implementasinya [1]. Hal ini pun yang dialami oleh mahasiswa ITS saat sedang mencari referensi publikasi ilmiah yang sesuai. Namun, sejauh ini fitur pencarian yang ada masih berdasarkan teks untuk mencari berdasarkan judul jurnal publikasi ilmiah saja. Padahal mahasiswa akan lebih merasa terbantu apabila fitur pencarian tidak hanya sebatas pencarian berdasarkan judul saja.

Ada beberapa penelitian yang mengatasi identifikasi dan anotasi pada artikel online. Penelitian [2] menunjukkan proses ekstraksi konten pembelajaran otomatis menggunakan klasifikasi pada *Massive Open Online Course* (MOOC). Konten pembelajaran selanjutnya diberikan anotasi secara otomatis. Metode yang digunakan antara lain: *rule-based* tanpa menggunakan *machine learning*, penggunaan *machine learning* dengan algoritma Random Forest, Support Vector Machine, dan Naïve Bayes. Pada penelitian [3] digunakan kombinasi metode ekstraksi data bibliografi dan ekstraksi term multi-level. Kedua metode ini menunjukkan sinergi dan dapat memberikan keterangan tambahan pada artikel.

Beberapa penelitian juga menyebutkan bagaimana proses identifikasi secara manual terbukti menyita banyak waktu dan memerlukan adanya pakar dalam domain artikel ilmiah tersebut. Penelitian [4] mengusulkan penggunakan model semi otomatis untuk menemukan referensi ke dataset artikel yang sudah ada. Keunggulan pendekatan ini adalah tidak memerlukan korpus tertentu dan dapat berjalan dengan baik untuk dataset kecil. Sedangkan pada penelitian [5] menggabungkan *Natural Language Processing* (NLP), *machine learning*, dan teknologi semantik dalam melakukan ekstraksi data artikel. Hasil akhir dari penelitian ini adalah pembuatan ontologi OWL yang mendeskripsikan relasi dengan entitas artikel lainnya.

Pada penelitian ini akan dibuat suatu aplikasi yang dapat membantu mahasiswa ITS dalam melakukan pencarian Publikasi Online ITS (POMITS) [6] yang sesuai. Tidak hanya berdasarkan judul saja, tetapi juga berdasarkan perangkat lunak, metode, dan kata kunci yang digunakan. Pembuatan aplikasi dilakukan dengan menambahkan anotasi pada konten POMITS. Penambahan anotasi pada konten POMITS akan memudahkan ITS dalam mencari POMITS yang sesuai dengan kebutuhannya.

Anotasi yang berhasil diekstraksi selanjutnya akan disimpan dalam bentuk *triple-store* pada Apache Jena Fuseki [7]. Penyimpanan dalam model semantik ini memungkinkan pencarian dapat dilakukan lebih tepat berdasarkan makna semantiknya. Selain itu, penyimpanan ini juga memudahkan dalam menjawab *query* tentang perangkat lunak, metode, dan kata kunci yang digunakan dalam artikel POMITS.

Artikel ini terdiri dari Pendahuluan yang menjelaskan tentang latar belakang identifikasi kata kunci dalam artikel ilmiah serta beberapa penelitian terkait proses otomatisasi identifikasi. Pada Studi Literatur dijelaskan mengenai beberapa penelitian sebelumnya yang berhubungan serta beberapa perangkat lunak yang digunakan. Kemudian pada bagian Metode yang Diusulkan dibahas mengenai motede yang diusulkan dalam penelitian ini. Hasil dan Pembahasan akan menyajikan hasil eksperimen sesuai dengan metode yang diusulkan. Artikel ditutup dengan Kesimpulan yang berisi rangkuman terkait hasil dari penelitian ini.

II. STUDI LITERATUR

Pada studi literatur ini akan dijelaskan mengenai penelitian sebelumnya, proses NLP secara umum, serta beberapa perangkat lunak yang digunakan dalam penelitian.

A. Penelitian Terkait

Penelitian terkait yang sudah dilakukan sebelumnya berfokus pada pencarian berdasarkan kedekatan makna kata menggunakan *Fasttext* dan metode *Word Mover's Distance* [8]. Selain itu dataset yang digunakan pada tahapan uji coba adalah dataset karya ilmiah yang mengandung sudah pasti mengandung query COCOMO, Process Mining, Semantic Search, dan Mixed Reality. Batasan query ini ditentukan menggunakan ground truth dari data yang telah dikumpulkan secara manual mengenai salah satu query tersebut yaitu COCOMO dan *systematic literature review*.

Kaitan penelitian tersebut dengan penelitian ini adalah pada tipe dataset yang digunakan, yaitu jurnal ilmiah. Hanya saja pada penelitian ini menggunakan ontologi yang dibangun untuk menyimpan kelas jurnal yang nantinya digunakan dalam proses *query* atau pencarian jurnal. Namun, pengguna pada penelitian tersebut terbatas hanya dapat melakukan pencarian yang mengandung *query* COCOMO, Process Mining, Semantic Search, dan Mixed Reality. Sedangkan pada penelitian ini *query* tidak hanya terbatas pada topik tersebut.

B. Publikasi Online ITS (POMITS)

JURNAL SAINS DAN SENI ITS Vol. X, No. Y (TAHUN), 2337-3520 (2301-928X Print) 1

Implementasi *Metasearch* dan Sistem Ranking Rekomendasi Hasil Pencarian Berdasarkan Profil Pengguna

Desy Nilasari, Abdul Munif, dan Nurul Fajrin Ariyani
Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: desy.nilasari16@mhs.if.its.ac.id, munif@if.its.ac.id, nurulfajrin@if.its.ac.id

***Abstrak*—Tujuan dari sistem yang dibuat pada penelitian ini adalah mengatasi keterbatasan pencarian konvensional dan sistem perluasan hasil pencarian melalui pembangunan profil pengguna. Profil pengguna dibentuk dengan menggunakan riwayat penelusuran yang dipersonalisasi dan telah diolah sedemikian rupa. Dengan menggunakan riwayat penelusuran diharapkan dapat memodelkan minat dan minat pengguna saat melakukan pencarian untuk membeli sebuah produk secara *online*. Dalam pembangunan sistem diperlukan pendekatan kata dan frasa penting pada riwayat penelusuran. Masalah ini dapat diatasi dengan menggunakan *Named Entity Recognition (NER), Part-of-Speech (POS) tagger*, dan *rule-based matcher*. Namun dikarenakan model *NER* dan *POS tagging* Bahasa Indonesia milik SpaCy belum tersedia maka dalam penelitian ini juga dibangun model *NER* dan *POS tagging* Bahasa Indonesia baru dengan Prodigy sebagai alat bantu anotasinya. Berdasarkan hasil evaluasi, metode yang digunakan peneliti memberikan hasil yang bagus. Hal ini dibuktikan dari hasil kuisioner yang menyatakan bahwa kesesuaian hasil yang diberikan sistem memiliki persentase 87.5% dan rentang waktu sukses dan sesungga lebih bermanfaat dibandingkan rentang waktu sebelum dan setelah.**

***Kata Kunci*—*NER, POS Tagger*, Profil Pengguna, Riwayat Penelusuran**

I. PENDAHULUAN

PADA era modern big data ini, aktivitas berbelanja online mengalami perubahan yang sangat besar dikarenakan pertumbuhan *E-Tailers* yang sangat dahsyat pada website perbelanjaan [1]. Pembeli pada zaman sekarang lebih memilih untuk berbelanja menggunakan *online shopping* dikarenakan beberapa alasan seperti adanya cashback yang menguntungkan bagi pembeli, lebih menghemat waktu, menjadi kemacetan yang akan terjadi ketika menuju pusat perbelanjaan, beberapa penjual memberi penawaran tanpa biaya ongkos pengiriman, maupun alasan yang lainnya.

Ketika pembeli berbelanja *online*, pembeli harus melakukan pencarian terhadap banyaknya data yang tersimpan pada mesin pencarian. Mayoritas dari mereka akan menghabiskan banyak waktu dalam melakukan pencarian barang yang sesuai dengan preferensinya. Hal ini dikarenakan pada mesin pencarian konvensional seperti Google dan Bing hanya akan memberikan hasil pencarian sesuai dengan *Keyword* yang dimasukkan oleh pengguna [1]. Padahal setiap pembeli memiliki preferensi yang berbeda-beda. Hal ini akan terlihat mengakan ketika beberapa pembeli melakukan pencarian dengan *keyword* yang sama kemudian menghasilkan hasil pencarian yang sama pula. Padahal pada kenyataannya, *keyword* yang dimasukan oleh satu pengguna dengan penggguna lainnya mewakili benda yang berbeda. Permasalahan tersebut terjadi dikarenakan mesin pencarian tidak memiliki keisambungan dengan preferensi individu pelanggan. Melainkan hanya mengeluarkan hasil pencarian sesuai dengan *keyword* yang dimasukkan. Oleh karena itu pada penelitian ini akan dibuat sebuah sistem yang dapat memodelkan profil penggguna dengan menyesuaikan hasil pencarian dengan preferensi penggguna menggunakan metode pendekatan dengan *NER* dan *rule-based matcher*. Sistem diharapkan dapat memberikan rekomendasi untuk penggguna. Rekomendasi diambil berdasarkan *keyword search* secara dari masing-masing profil penggguna. *Dataset* yang digunakan untuk membangun model berasal dari riwayat pencarian. Proses pengerjaan tugas akhir ini meliputi pembangunan model *NER, POS tagging*, dan *rule-base matcher*.

II. TINJAUAN PUSTAKA

A. Profil Pengguna

Profil penggguna adalah sebuah bentuk pemodelan yang berisi preferensi pencarian tiap-tiap penggguna [2]. Sistem dengan profil penggguna diharapkan dapat dengan mudah melacak preferensi penggguna. Sistem dengan profil penggguna erat kaitannya dengan memantau riwayat pesalejahan pelanggan kemudian sistem akan secara otomatis memperbarui profil pelanggan tsapa memerlukan tombahan upaya apa pun dari pelanggan [1].

B. History Browser

History browser atau riwayat penelusuran adalah daftar halaman web yang pernah dikunjungi penggguna, serta informasi terkait seperti judul halaman dan waktu kunjungan, yang dicatat oleh perangkat lunak browser web [3]. Pada tugas akhir ini, penggguna menggunakan *history browser* yang berasal dari Google Chrome. Alasannya pada Google Chrome terdapat ekstensi yang berfungsi untuk men-*download history browser*. Hasilnya berupa file JSON yang berisi informasi seperti id, waktu penelusuran (*last visit time local* dan *last visit time UTC*), *title* /*keyword* yang dimasukkan penggguna), *url*, *typed count*, *visit count*. Pada penelitian ini penulis menggukan atribut *last visit time local* untuk dipasakan antara jam dan harinya.

Gambar 1. Contoh Artikel POMITS

Publikasi Online ITS (POMITS) merupakan media online untuk mempublikasikan karya-karya ilmiah dalam bentuk penerbitan berkala. Karya ilmiah yang dipublikasikan merupakan hasil dari penelitian ilmiah di bidang-bidang yang menjadi keunggulan ITS. Masing-masing bidang-bidang tersebut dikelompokkan menjadi dua, yaitu Jurnal Teknik dan Jurnal Sains dan Seni [6]. Pada penelitian ini, POMITS yang digunakan adalah POMITS mahasiswa departemen Teknik Informatika saja dengan total POMITS yang terkumpul sebanyak 224 artikel. Gambar 1 menunjukkan contoh artikel POMITS yang telah terbit.

C. Korpus

Korpus merupakan sekumpulan berupa teks yang menjadi dasar analisis linguistik [4]. Berdasarkan pengertian tersebut, dapat dikatakan bahwa korpus terutama muncul dalam area NLP (Natural Language Processing) atau domain aplikasi yang berkaitan dengan teks atau dokumen. Pada penelitian ini, korpus yang digunakan adalah korpus data Publikasi Online ITS (POMITS).

D. Natural Language Processing (NLP)

NLP (*Natural Language Processing*) merupakan salah satu cabang ilmu kecerdasan buatan (*artificial intelligence*) yang berfokus pada pengolahan bahasa natural. Bahasa natural adalah bahasa yang secara umum digunakan oleh manusia dalam berkomunikasi satu sama lain. Bahasa yang diterima oleh komputer butuh untuk diproses dan dipahami terlebih dahulu supaya maksud dari user bisa dipahami dengan baik oleh komputer.

Beberapa bidang penerapan NLP antara lain penjawab pertanyaan (*question answering*), ekstraksi informasi (*information extraction*), analisis sentimen (*sentiment analysis*), penerjemahan mesin (*machine translation*), pemerolehan informasi (*information retrival*), perangkuman otomatis (*automatic summarization*), dan pengenalan wicara (*speech recognition*) [9]. Bidang penerapan NLP yang digunakan pada penelitian ini adalah information extraction dimana program dapat mengekstrak data menjadi informasi yang dapat dimanfaatkan.

E. Named Entity Recognition (NER)

Named Entity Recognition (NER) merupakan bagian dari ekstraksi informasi yang bertugas untuk mengklasifikasikan teks dari sebuah dokumen atau korpus yang dikategorikan seperti nama orang, lokasi, organisasi, tanggal, waktu, dan sebagainya. NER diimplementasikan dalam banyak bidang, antara lain dalam machine translation, question-answering machine system, indexing pada information retrieval, klasifikasi, dan juga dalam automatic summarization. Tujuan yang diharapkan dari proses dalam NER adalah untuk melakukan ekstraksi dan klasifikasi nama ke dalam beberapa kategori dengan mengacu kepada makna yang tepat [9]. Pada penelitian ini NER digunakan untuk mengenali entitas software, metode, dan keyword yang digunakan pada POMITS. Contoh NER dapat dilihat pada Tabel 1.

Tabel 1. Contoh Hasil NER

Teks Awal	Hasil NER
Pada tugas jurnal ini kami membuat suatu alat untuk medeteksi kandungan kadar alkohol dari suatu produk makanan, minuman atau obat-obatan, dengan menggunakan modul sensor MQ3 dan juga MQ135 sebagai pembanding yang dijalankan pada sebuah board Arduino Uno sebagai Microcontroller nya.	MQ3 KEYWOR MQ135 KEYWORD Arduino Uno KEYWORD

F. SpaCy

SpaCy adalah pustaka open-source gratis dalam bahasa Python yang digunakan untuk *Natural Language Processing* (NLP). SpaCy dapat digunakan untuk membangun ekstraksi informasi atau praproses teks untuk *deep learning*. Beberapa fitur yang didukung spaCy antara lain tokenisasi, POS Tagging, Named Entity Recognition, dan Rule-based Matching [10]. Pada penelitian ini spaCy digunakan untuk melakukan anotasi NER.

G. Prodigy

Prodigy merupakan alat anotasi yang efisien sehingga penggunanya dapat melakukan anotasi sendiri. Pengguna dapat bekerja pada pengenalan entitas, dan Prodigy dapat membantu pengguna dalam melatih dan mengevaluasi model dengan lebih cepat. Fitur yang ada dalam Prodigy yaitu *Named Entity Recognition*, klasifikasi teks, dan visi komputer [11]. Pada penelitian ini Prodigy digunakan sebagai alat bantu untuk melakukan anotasi NER pada konten POMITS.

H. Flask

Flask merupakan salah satu kerangka kerja aplikasi web Python yang paling populer. Flask memberikan kebebasan kepada pengembang untuk memilih alat dan pustaka yang ingin digunakan. Serta ada banyak ekstensi yang disediakan sehingga penambahan fungsionalitas baru menjadi lebih mudah [12]. Pada penelitian ini Flask digunakan untuk membuat aplikasi web.

I. SPARQL

SPARQL merupakan protokol bahasa query RDF yang berfungsi untuk mengambil dan memanipulasi data dari sebuah triple store [13]. RDF (*Resource Description Framework*) merupakan jenis file yang terdiri dari statement yang memiliki 3 variabel sebagai subjek, predikat, dan objek, yang dikenal dengan sebutan triplets. Pada penelitian ini, SPARQL query digunakan untuk mencari data POMITS yang tersimpan dalam triple store Apache Jena Fuseki dan untuk meng-input data POMITS beserta anotasinya pada triple store Apache Jena Fuseki.

J. Apache Jena Fuseki

Apache Jena Fuseki adalah server SPARQL (bahasa query semantik). Apache Jena Fuseki memiliki antarmuka pengguna untuk pemantauan dan administrasi server. Selain itu, Apache Jena Fuseki juga menyediakan mesin protokol untuk sistem penyimpanan dan query RDF (Resource Description Framework) [7]. Pada penelitian ini Apache Jena Fuseki bertindak sebagai basis data triple store yang bisa diakses melalui request HTTP.

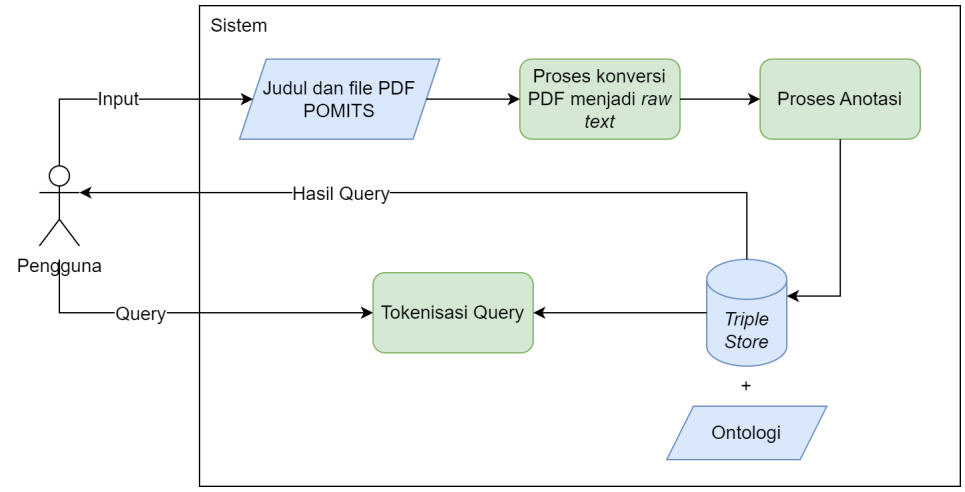
Apache Jena Fuseki menyediakan beberapa API. Beberapa fungsi API tersebut antara lain untuk endpoint pengunggahan file RDF (/upload), membaca data (/get), query SPARQL (/query atau /sparql), dan memperbarui data (/update).

III. METODE YANG DIUSULKAN

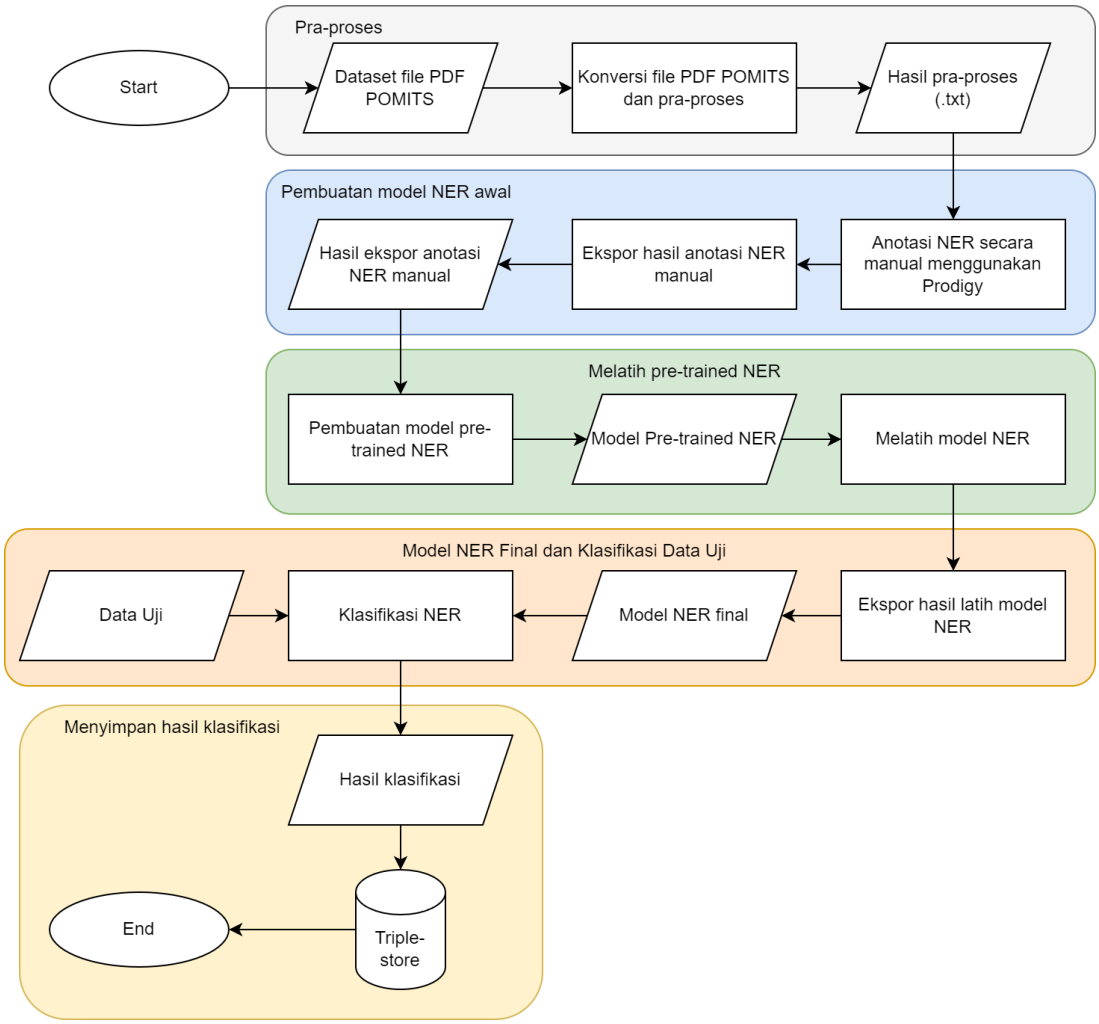
Penelitian ini terdiri dari dua bagian dalam pembuatan identifikasi kata kunci pada konten publikasi jurnal ilmiah. Bagian pertama berisi arsitektur sistem secara umum dan bagaimana pengguna dapat menggunakan sistem untuk melakukan *query*/pencarian data publikasi POMITS. Bagian kedua berisi tahapan pembangunan model NER yang akan disimpan pada *triple-store*.

A. Sistem untuk Melakukan Query/Pencarian

Pada arsitektur sistem yang ditunjukkan oleh Gambar 2, pengguna dapat memasukkan input artikel ke dalam sistem. Input yang dimasukkan adalah judul dan file PDF dari artikel POMITS. Kemudian file ini akan diproses menjadi *raw text* agar dapat dilakukan proses anotasi. Selanjutnya, hasil anotasi disimpan dalam *triple-store*. Selanjutnya user dapat melakukan *query* atau pencarian artikel POMITS. Pencarian dilakukan dengan memotong tiap kata pada query lalu mengambil data yang sesuai dengan *triple-store*. Model data *triple-store* atau *triplets* terdiri dari pasangan *subject-predicate-object*.



Gambar 2. Sistem untuk Melakukan Query/Pencarian



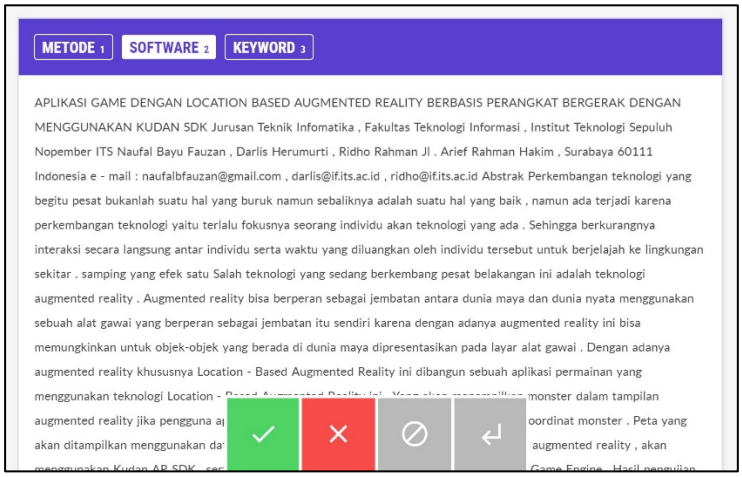
Gambar 3. Sistem Pembangunan NER

B. Sistem Pembangunan Model NER

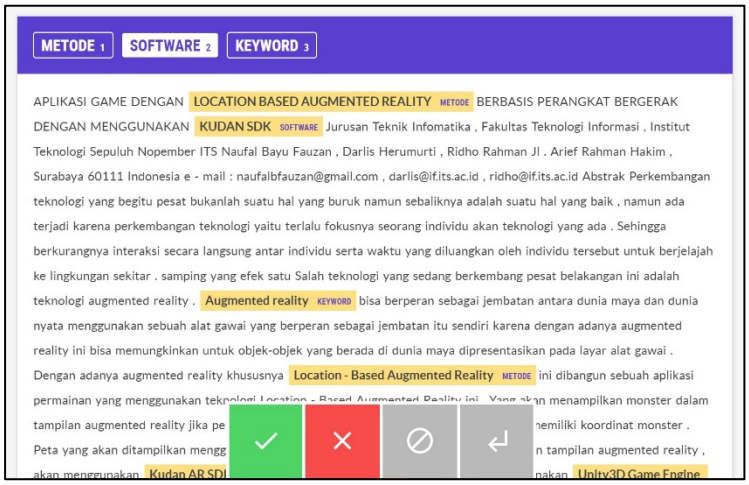
Gambar 3 menunjukkan alur pembangunan model NER secara keseluruhan. Model NER dibuat untuk melakukan proses identifikasi kata kunci secara otomatis. Adapun tahapan-tahapan pembuatan model NER adalah sebagai berikut.

1) Pra-proses

Artikel POMITS yang dimasukkan ke dalam sistem berformat PDF. File-file ini nantinya akan dikonversi ke dalam bentuk PDF dan dilakukan pra-proses. Pra-proses meliputi penghapusan tanda baca, *stopwords*, dan



Gambar 4. Tampilan Prodigy sebelum dilakukan anotasi NER



Gambar 5. Tampilan Prodigy setelah dilakukan anotasi NER

karakter-karakter yang tidak diperlukan. Hasil akhir dari tahapan pra-proses ini adalah file-file bertipe teks (.txt) yang dapat digunakan untuk proses selanjutnya.

2) *Pembuatan model NER awal*

Dalam tahap ini dilakukan proses anotasi NER secara manual menggunakan Prodigy. Proses anotasi manual ini dilakukan karena tidak adanya model NER awal untuk dataset artikel ilmiah berbahasa Indonesia. Setelah proses NER selesai dilakukan, hasil model NER akan diekspor sebagai input pembuatan model *pre-trained* NER. Gambar 4 dan Gambar 5 menunjukkan proses anotasi NER manual menggunakan Prodigy. Daftar NER yang digunakan sebagai label dapat dilihat pada Tabel 2.

Tabel 2. Daftar Label NER

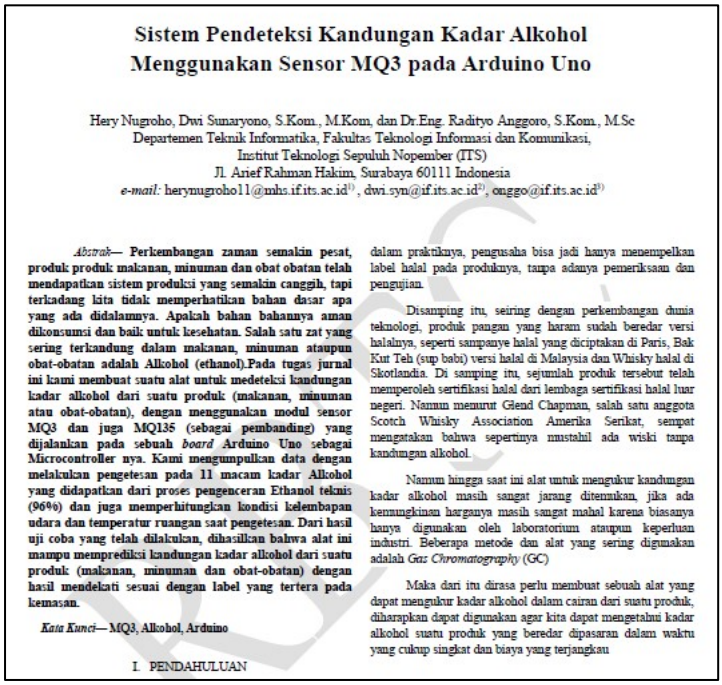
Nama Label	Keterangan
SOFTWARE	*Software* yang digunakan dan disebutkan di dalam konten *file* pdf POMITS
METODE	Metode yang digunakan dan disebutkan di dalam konten *file* pdf POMITS
KEYWORD	*Keyword* atau kata-kata penting yang disebutkan pada POMITS bagian abstrak yang bukan merupakan nama *software* dan nama metode

3) *Melatih Pre-trained NER*

Setelah diperoleh model awal hasil anotasi NER manual, langkah selanjutnya adalah melatih pre-trained NER menggunakan sisa data yang ada. Hal ini bertujuan agar NER dapat mengenali entitas-entitas lainnya, Sehingga model dapat melakukan identifikasi dngan baik. Proses ini di dalam Prodigy disebut *ner.teach*.

4) *Model NER Final dan Klasifikasi Data Uji*

Setelah proses pelatihan menggunakan semua data telah selesai, maka akan dilakukan ekspor model NER tersebut. Di dalam Prodigy model ini disebut sebagai *ner.make-gold*. Hasil model NER ini kemudian dapat



Sistem Pendeteksi Kandungan Kadar Alkohol Menggunakan Sensor MQ3 pada Arduino Uno

Hery Nugroho, Dwi Sunaryono, S.Kom., M.Kom, dan Dr.Eng. Radityo Anggoro, S.Kom., M.Sc
Departemen Teknik Informatika, Fakultas Teknologi Informasi dan Komunikasi,
Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: herynugroho11@mhs.if.its.ac.id[1)] , dwi.syn@if.its.ac.id[2)], onggo@if.its.ac.id[3)]

***Abstrak*— Perkembangan zaman semakin pesat, produk produk makanan, minuman dan obat obatan telah mendapatkan sistem produksi yang semakin canggih, tapi terkadang kita tidak memperhatikan bahan dasar apa yang ada didalamnya. Apakah bahan bahannya aman dikonsumsi dan baik untuk kesehatan. Salah satu zat yang sering terkandung dalam makanan, minuman ataupun obat-obatan adalah Alkohol (ethanol).Pada tugas jurnal ini kami membuat suatu alat untuk mendeteksi kandungan kadar alkohol dari suatu produk (makanan, minuman atau obat-obatan), dengan menggunakan modul sensor MQ3 dan juga MQ135 (sebagai pembanding) yang dijalankan pada sebuah *board* Arduino Uno sebagai Microcontroller nya. Kami mengumpulkan data dengan melakukan pengetesan pada 11 macam kadar Alkohol yang didapatkan dari proses pengenceran Ethanol teknis (96%) dan juga memperhitungkan kondisi kelembapan udara dan temperatur ruangan saat pengetesan. Dari hasil uji coba yang telah dilakukan, dihasilkan bahwa alat ini mampu memprediksi kandungan kadar alkohol dari suatu produk (makanan, minuman dan obat-obatan) dengan hasil mendekati sesuai dengan label yang tertera pada kemasan.**

***Kata Kunci*— MQ3, Alkohol, Arduino**

I. PENDAHULUAN

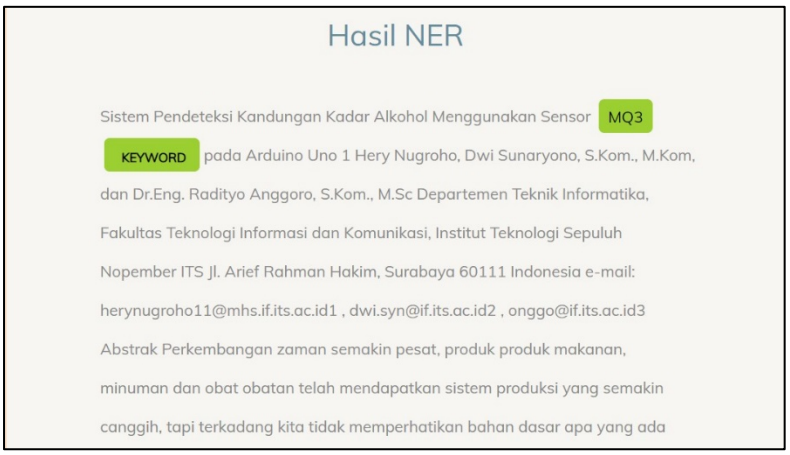
dalam praktiknya, pengusaha bisa jadi hanya menempelkan label halal pada produknya, tanpa adanya pemeriksaan dan pengujian.

Disamping itu, seiring dengan perkembangan dunia teknologi, produk pangan yang haram sudah beredar versi halalnya, seperti sampanye halal yang diciptakan di Paris, Bak Kut Teh (sup babi) versi halal di Malaysia dan Whisky halal di Skotlandia. Di samping itu, sejumlah produk tersebut telah memperoleh sertifikasi halal dari lembaga sertifikasi halal luar negeri. Namun menurut Glend Chapman, salah satu anggota Scotch Whisky Association Amerika Serikat, sempat mengatakan bahwa sepertinya mustahil ada wiski tanpa kandungan alkohol.

Namun hingga saat ini alat untuk mengukur kandungan kadar alkohol masih sangat jarang ditemukan, jika ada kemungkinan harganya masih sangat mahal karena biasanya hanya digunakan oleh laboratorium ataupun keperluan industri. Beberapa metode dan alat yang sering digunakan adalah *Gas Chromatography* (GC)

Maka dari itu dirasa perlu membuat sebuah alat yang dapat mengukur kadar alkohol dalam cairan dari suatu produk, diharapkan dapat digunakan agar kita dapat mengetahui kadar alkohol suatu produk yang beredar dipasaran dalam waktu yang cukup singkat dan biaya yang terjangkau

Gambar 6. Contoh Artikel untuk Identifikasi Kata Kunci



Gambar 7. Hasil Pemrosesan NER

digunakan untuk melakukan proses klasifikasi atau identifikasi kata kunci dari artikel-artikel yang ada pada data uji. Contoh proses klasifikasi data uji ditunjukkan pada Gambar 6 dan Gambar 7.

5) Menyimpan Hasil Klasifikasi

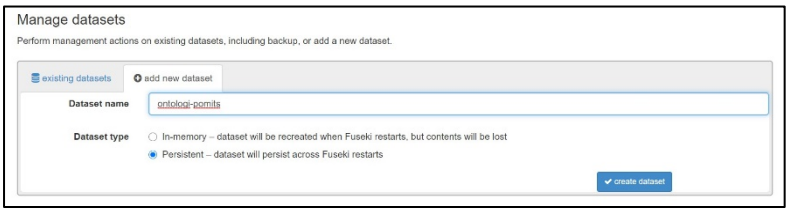
Setelah proses klasifikasi selesai, data-data yang telah berhasil diklasifikasi akan disimpan dalam bentuk *triple-store* pada Apache Jena Fuseki. Protégé digunakan untuk membantu membuat data awal berupa file RDF/XML yang nantinya akan diunggah ke Apache Jena Fuseki. Dalam prosesnya, dibuat ontologi baru yang disimpan dalam tautan http://www.semanticweb.org/khairunnisarahmam/ontologies/2020/5/ontologi-pomits#. Dalam ontologi tersebut terdapat sembilan property anotasi baru, yaitu: nama, judul, nrp, pembimbing1, pembimbing2, software, metode, keyword, dan konten. Ontologi yang dibuat kemudian disimpan ke dalam file berformat RDF/XML. Gambar 8 dan Gambar 9 menunjukkan proses pembuatan dataset baru. Sedangkan Gambar 10 dan Gambar 11 menunjukkan proses query SPARQL pada Apache Jena Fuseki beserta hasilnya.

IV. HASIL DAN PEMBAHASAN

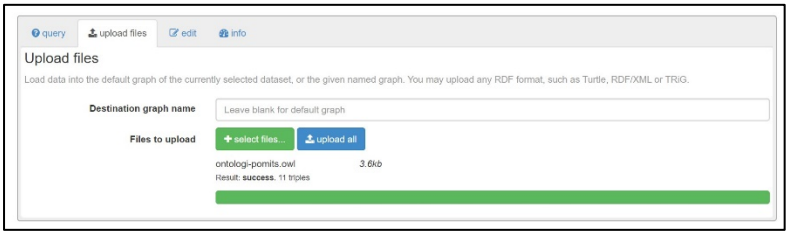
Bagian ini menjelaskan hasil uji coba dan evaluasi sistem yang telah dibuat. Pengujian dilakukan untuk mengetahui kinerja sistem dengan beberapa skenario yang berbeda.

A. Data Uji Coba

Data uji coba yang digunakan untuk evaluasi model NER merupakan data masukan baru (di luar data latih dan *dataset* yang sudah ada) dengan total 55 *file* POMITS. Untuk evaluasi *input* data dan bentuk anotasi juga



Gambar 8. Pembuatan Dataset Baru pada Apache Jena Fuseki



Gambar 9. Upload File RDF/XML pada Apache Jena Fuseki

```
PREFIX ont: <http://www.semanticweb.org/khairunnisarahmam/ontologies/2020/5/ontologi-
pomits#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT *
WHERE { ?subject ?property ?object }
```

Gambar 10. Query SPARQL untuk menampilkan semua pasangan triplets pada Apache Jena Fuseki

Gambar 11. Hasil query SPARQL pada Apache Jena Fuseki

menggunakan lima data masukan baru berupa nama penulis, nrp penulis, judul POMITS, nama dosen pembimbing dan juga *file* POMITS yang berekstensi pdf. Sedangkan untuk evaluasi *query* label digunakan beberapa kata kunci pencarian yang berbeda.

B. Skenario Uji Coba

Terdapat tiga skenario uji coba yang dilakukan. Berikut penjelasan masing-masing skenario beserta hasilnya.

1) Skenario Uji Coba 1

Dalam skenario ini akan dilakukan pengujian terhadap model NER terhadap 49 data masukan baru. Evaluasi model NER dilakukan dengan menghitung nilai *precision* dan *recall* untuk setiap label NER yang dibandingkan dengan pelabelan secara manual oleh pakar. Tabel 4 dan Tabel 5 menunjukkan hasil pengujian skenario 1.

2) Skenario Uji Coba 2

Dalam skenario ini dilakukan pengujian dengan menggunakan enam data masukan baru berupa nama penulis, nrp penulis, judul POMITS, nama dosen pembimbing dan juga *file* POMITS yang berekstensi pdf ke dalam sistem. Contoh data uji coba 2 ditunjukkan pada Tabel 3. Hasil deteksi NER data uji 2 ditunjukkan pada Gambar 12, dan hasil Apache Jena Fusekinya ditunjukkan pada Gambar 13. Hasil deteksi NER data uji 3 dapat dilihat pada Gambar 14. Sedangkan hasil Apache Jena Fusekinya ditunjukkan pada Gambar 15.

Tabel 4. Hasil jumlah True Positive (TP), False Positive (FP), dan False Negative (FN)

Entitas	*True Positive* (TP)	*False Positive* (FP)	*False Negative* (FN)
SOFTWARE	46	7	32
METODE	87	15	81
KEYWORD	60	13	15

Tabel 5. Hasil evaluasi model NER pada data uji 1

Entitas	*Precision*	*Recall*
SOFTWARE	86.79%	58.97%
METODE	85.29%	51.79%
KEYWORD	82.19%	80.00%

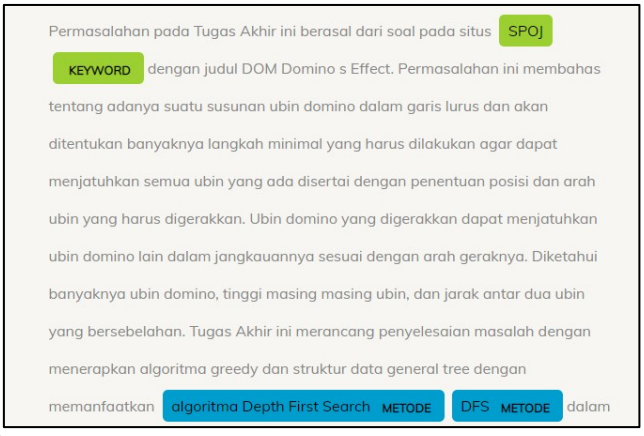
Tabel 3. Data Uji Skenario 2

Nama	Data
Data Uji 2	Nama: Safira Vanillia Putri NRP: 05111640000001 Judul: Klasifikasi Multi Label Gaya Belajar VAK Berdasarkan Perilaku Pembelajaran pada E-learning Menggunakan Metode Supervised Learning Dosen Pembimbing 1: Dini Adni Navastara, S.Kom., M.Sc. Dosen Pembimbing 2: Umi Laili Yuhana, S.Kom., M.Sc. *File* POMITS: 05111640000001-Safira_Vanillia_Putri-POMITS.pdf
Data Uji 3	Nama: Zahrah Ayu Afifah Febriani NRP: 05111640000108 Judul: Penerapan Struktur Data Tree dan Algoritma Greedy dalam Pencarian Solusi Optimal pada Permasalahan SPOJ DOM Domino's Effect Dosen Pembimbing 1: Rully Soelaiman, S.Kom., M.Kom. Dosen Pembimbing 2: Dwi Sunaryono, S.Kom., M.Kom. File POMITS: 05111640000108_Zahrah_Ayu_Afifah_Febriani-POMITS.pdf

Gambar 12. Hasil Uji Coba Skenario 2 Data Uji 2

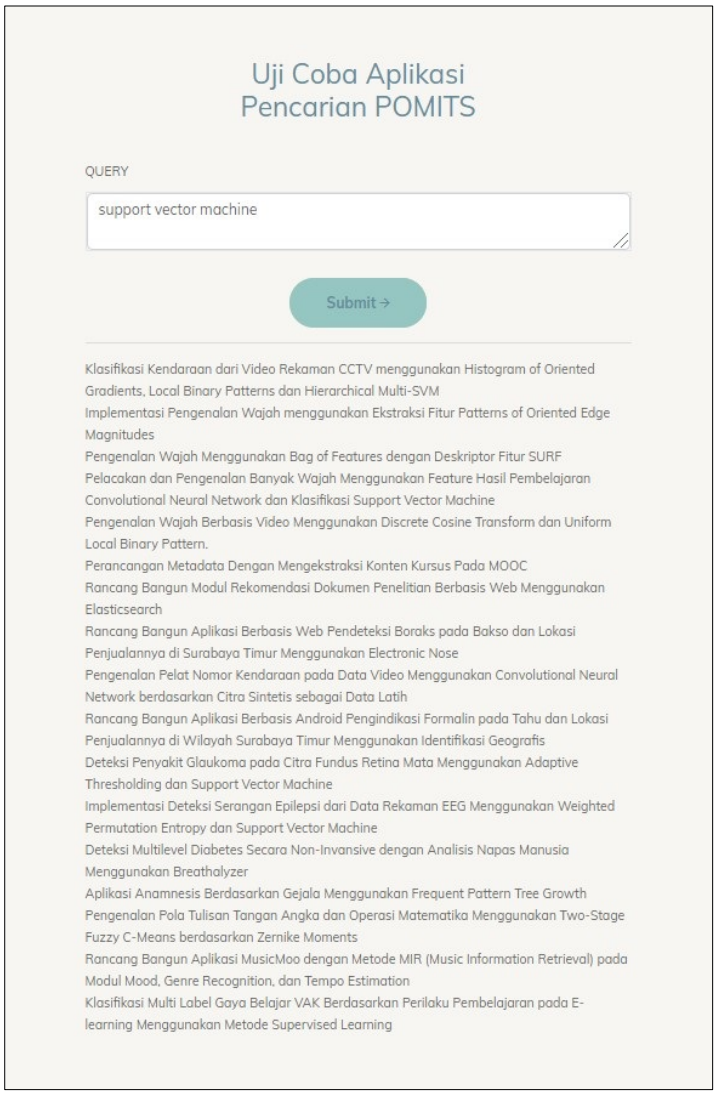
	subject	property	object
1	ont:05111640000001	rdf:type	ont:pomits
2	ont:05111640000001	ont:penulis	"Safira Vanillia Putri"
3	ont:05111640000001	ont:pembimbing2	"Umi Laili Yuhana, S.Kom., M.Sc."
4	ont:05111640000001	ont:pembimbing1	"Dini Adni Navastara, S.Kom., M.Sc."
5	ont:05111640000001	ont:nrp	"05111640000001"
6	ont:05111640000001	ont:metode	"support vector machine"
7	ont:05111640000001	ont:metode	"svm"
8	ont:05111640000001	ont:metode	"algoritma support vector machine"
9	ont:05111640000001	ont:metode	"cross-validation"
10	ont:05111640000001	ont:metode	"k-nearest neighbor"
11	ont:05111640000001	ont:metode	"k-nearest neighbors"
12	ont:05111640000001	ont:metode	"k-nn"
13	ont:05111640000001	ont:metode	"k- nn"
14	ont:05111640000001	ont:metode	"k-nearest neighbours"
15	ont:05111640000001	ont:metode	"over-sampling"

Gambar 13. Hasil Uji Coba Skenario 2 Data Uji 2 pada Apache Jena Fuseki



	subject	property	object
1	ont:05111640000108	rdf:type	ont:pomits
2	ont:05111640000108	ont:penulis	"Zahrah Ayu Afifah Febriani"
3	ont:05111640000108	ont:pembimbing2	"Dwi Sunaryono, S.Kom., M.Kom."
4	ont:05111640000108	ont:pembimbing1	"Rully Soelaiman, S.Kom., M.Kom."
5	ont:05111640000108	ont:nrp	"05111640000108"
6	ont:05111640000108	ont:metode	"algoritma greedy"
7	ont:05111640000108	ont:metode	"algoritma dfs"
8	ont:05111640000108	ont:metode	"dfs"
9	ont:05111640000108	ont:metode	"depth first search"
10	ont:05111640000108	ont:metode	"algoritma depth first search"
11	ont:05111640000108	ont:metode	"dfs iii"

Gambar 15. Hasil Uji Coba Skenario 2 Data Uji 3 pada Apache Jena Fuseki



Gambar 16. Hasil Pengujian Query

3) Skenario Uji Coba 3

Dalam skenario ini dilakukan pengujian dengan melakukan *query* label. Pengujian dilakukan dengan memasukkan kata kunci yang berupa *software*, metode, atau *keyword,* dan gabungan dua atau lebih dari ketiganya. Gambar 16 menunjukkan contoh hasil keluaran aplikasi saat dilakukan pencarian dengan keyword "*support vector machine*".

C. Evaluasi

Pada skenario pengujian 1 diperoleh hasil rata-rata nilai presisi 84.76% dan *recall* 63.59% untuk semua entitas NER. Pada skenario pengujian 2 diperoleh bahwa sistem berhasil mendeteksi entitas NER. Sistem juga berhasil menyimpan hasil anotasi NER dalam bentuk pasangan *triplets* pada Apache Jena Fuseki. Sistem otomatis menambahkan data masukan baru ke *triple-store*.

Pada skenario pengujian 3 diperoleh bahwa sistem berhasil menampilkan judul POMITS yang sesuai dengan kata kunci pencarian yang dimasukkan pengguna ke dalam sistem. Berdasarkan hasil pencarian yang ditampilkan oleh sistem masing-masing judul POMITS memiliki tingkat relevansi dengan kata kunci pencarian yang sama karena tidak dilakukan pembobotan pada sistem pencarian.

V. KESIMPULAN

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian yang dilakukan, dapat diambil kesimpulan sebagai berikut. Ekstraksi setiap anotasi pada konten POMITS menjadi sebuah metadata dilakukan dengan mendeteksi *named entity* berupa *software*, metode, dan *keyword* menggunakan model NER. Hasil anotasi NER disimpan dalam bentuk pasangan *triplets* pada *triple store* Apache Jena Fuseki. Diperoleh hasil rata-rata nilai *precision* 84.76% dan *recall* 63.59% untuk semua entitas NER. Metadata POMITS dimodelkan dalam bentuk pasangan triplets pada triple store Apache Jena Fuseki. Metadata tersebut dapat digunakan untuk menjawab query SPARQL tentang *software*, metode, dan *keyword* yang terkandung di dalam *file* POMITS.

DAFTAR PUSTAKA

[1] S. Mesbah, K. Fragkeskos, C. Lofi, A. Bozzon, dan G. J. Houben, "Semantic annotation of data processing pipelines in scientific publications," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10249 LNCS, hlm. 321–336, 2017, doi: 10.1007/978-3-319-58068-5_20/TABLES/6.

[2] N. F. Ariyani, A. Munif, dan P. Q. Ayunin, "An automatic annotation method on MOOC's learning content," dalam *Proceedings of 2019 International Conference on Information and Communication Technology and Systems, ICTS 2019*, 2019. doi: 10.1109/ICTS.2019.8850965.

[3] P. Lopez, "GROBID: Combining automatic bibliographic data recognition and term extraction for scholarship publications," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5714 LNCS, hlm. 473–474, 2009, doi: 10.1007/978-3-642-04346-8_62.

[4] B. Ghavimi, P. Mayr, S. Vahdati, dan C. Lange, "Identifying and improving dataset references in social sciences full texts," *Positioning and Power in Academic Publishing: Players, Agents and Agendas - Proceedings of the 20th International Conference on Electronic Publishing, ELPUB 2016*, hlm. 105–114, 2016, doi: 10.3233/978-1-61499-649-1-105.

[5] F. Osborne, H. de Ribaupierre, dan E. Motta, "TechMiner: Extracting technologies from academic publications," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10024 LNAI, hlm. 463–479, 2016, doi: 10.1007/978-3-319-49004-5_30.

[6] "Publikasi Online ITS (POMITS)." https://ejurnal.its.ac.id/ (diakses 15 April 2022).

[7] "Apache Jena - Apache Jena Fuseki." https://jena.apache.org/documentation/fuseki2/ (diakses 12 April 2022).

[8] N. H. Pribadi, "Sistem Rekomendasi Karya Ilmiah Berdasarkan Semantic Similarity Menggunakan FastText dan Word Mover's Distance," 2020.

[9] "Natural Language Processing." https://socs.binus.ac.id/2013/06/22/natural-language-processing/ (diakses 12 April 2022).

[10] "spaCy 101: Everything you need to know · spaCy Usage Documentation." https://spacy.io/usage/spacy-101 (diakses 12 April 2022).

[11] "Prodigy 101 – everything you need to know · Prodigy · An annotation tool for AI, Machine Learning & NLP." https://prodi.gy/docs (diakses 12 April 2022).

[12] "Flask · PyPI." https://pypi.org/project/Flask/ (diakses 12 April 2022).

[13] "SPARQL Query Language for RDF." https://www.w3.org/TR/rdf-sparql-query/ (diakses 12 April 2022).