

Pencarian Tugas Akhir dengan Ontologi dan Boyer-Moore (Studi Kasus: Jurusan Teknik Informatika UNSRI)

Desty Rodiah*, Yunita, Novi Yusliani
Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya Indralaya, Indonesia
Email: destyrodiah@gmail.com

Abstract—Website sipeta.ilkom.unsri.ac.id adalah website yang menampung data tugas akhir mahasiswa Jurusan Teknik Informatika UNSRI. Namun website tersebut menggunakan penyimpanan dengan basis data biasa. Pada penelitian ini membuat pencarian data tugas akhir mahasiswa dengan memanfaatkan web semantik ontologi agar data yang dimiliki tidak hanya memiliki nilai, tetapi juga memiliki pengetahuan tentang relasi antar informasi yang saling berkaitan. Komponen yang digunakan dalam teknologi semantik adalah RDF yang dipergunakan sebagai representasi pengetahuan yang digunakan, kemudian SPARQL yang digunakan sebagai query untuk mengambil informasi yang terdapat dalam Ontologi RDF. Selain itu juga digunakan Algoritma Boyer Moore untuk mendapatkan nilai *similarity* antara data yang didapatkan dari hasil pencarian dengan *keyword* yang dimasukkan. Jenis pencarian yang dirancang ada 3 pencarian yaitu *keyword search*, *simple search* dan *advanced search*. Dan ketiga pencarian tersebut juga akan di kombinasikan dengan algoritma Boyer Moore. Hasil pencarian dengan ontologi dengan pencarian dengan ontologi dan Algoritma Boyer Moore dihasilkan bahwa pencarian dengan Boyer Moore membutuhkan waktu lebih lama secara rata-rata sekitar $\geq 0,0001$ perdetik dalam 5 kali percobaan dibandingkan pencarian dengan ontologi saja. Untuk Algoritma Boyer Moore dilakukan pengujian dengan ROC didapatkan hasil akurasi sebesar 99,84% untuk 16 kali percobaan.

Keywords—*boyer moore, ontologi, pencarian, tugas akhir, web semantik*

I. PENDAHULUAN

Jurusan Teknik Informatika UNSRI memiliki repositori yang menampung data tugas akhir mahasiswa dengan link website sipeta.ilkom.unsri.ac.id. Website ini juga dapat membantu mahasiswa dalam mencari tugas akhir terdahulu. Website tersebut memanfaatkan fitur *database* mysql untuk pencarian data tugas akhir. Fitur pencarian pada web tersebut melakukan proses pencarian terhadap data tugas akhir hanya didasarkan pada pencocokan kata kunci dengan data atau informasi yang dimiliki. Kemudian didapati masalah ketika ingin melakukan proses pencarian yang lebih spesifik seperti pencarian data berdasarkan suatu relasi atau keterkaitan tertentu, sehingga didapati semantik web menjadi solusi dalam proses pencarian. Dengan menggunakan semantik web, data yang dimiliki tidak hanya memiliki nilai, tetapi juga dapat memiliki pengetahuan tentang relasi antar informasi yang saling berkaitan. Dari adanya relasi ini dapat dimanfaatkan sebagai referensi dalam proses pencarian yang diinginkan, sehingga proses pencarian dapat dilakukan lebih mendalam [1]. Dengan semantik juga dapat melakukan penyimpanan

dan pencarian data yang begitu besar seperti halnya kasus tugas akhir mahasiswa yang banyak [2].

Penelitian yang dilakukan oleh Halim dan Gunawan [3], menerapkan web semantik untuk aplikasi pencarian pada repositori koleksi penelitian. Pada penelitian tersebut merancang ontologi penelitian, kemudian merancang RDF, lalu merancang Query dengan memanfaatkan SPARQL. Rancangan query yang dibuat terdiri dari 3 jenis, yaitu query untuk pencarian dengan kata kunci, query untuk pencarian dengan *simple*, query untuk pencarian *advanced*. Dan hasil yang di dapatkan bahwa aplikasi pencarian berhasil di kembangkan, namun masih butuh penambahan *vocabulary* kata kunci. Penelitian yang dilakukan oleh Wijayanto dkk[2], menerapkan web semantik dalam pencarian katlog buku di perpustakaan STMIK Sinar Nusantara Surakarta. Pada penelitian tersebut, representasi pengetahuan menggunakan RDF kemudian SPARQL yang digunakan sebagai query untuk mengambil informasi yang terdapat dalam ontologi RDF. Hasil pencarian diuji dengan 3 kategori pencarian yaitu *keyword*, *simple*, dan *advanced search*. Pemetaan dalam query SPARQL akan mempengaruhi seberapa cepat aplikasi akan menampilkan hasil. Hasilnya masih membutuhkan penambahan *vocabulary* kata kunci, dalam hal ini dapat menggunakan metode *Language Based Matching* atau *Vector Space Matching* sebagai metode pengukuran kemiripan.

Dalam Penelitian ini juga selain memanfaatkan web semantik akan menggunakan metode *Language Based Matching* untuk mengukur kemiripan. Algoritma Boyer Moore merupakan suatu solusi pencarian yang efisien dapat melakukan perbandingan *pattern* mulai dari kanan ke kiri. Karakteristik utama dari Algoritma Boyer-Moore adalah algoritma ini melakukan pencocokan string mulai dari kanan (belakang). Dengan karakteristik tersebut, ketidakcocokan saat terjadi perbandingan string akan membuat pergerakan *pattern* melompat lebih jauh untuk menghindari perbandingan karakter pada string yang diperkirakan gagal. Sehingga proses pencarian string akan lebih optimal. Algoritma Boyer Moore dapat digunakan untuk teknologi mesin pencari [4].

Pada penelitian yang dilakukn oleh Ahmad dkk [5] yang mengimplementasikan *string matching* dengan Algoritma Boyer Moore untuk menentukan tingkat kemiripan pada pengajuan judul skripsi/TA Mahasiswa. Penelitian tersebut menghasilkan sistem yang mampu menentukan kemiripan dari usulan judul Skripisi/TA berdasarkan repository judul yang telah ada. Dari hasil pengujian waktu proses dengan memasukan 100 dan kelipatannya

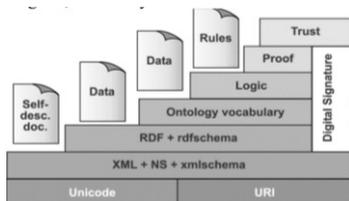
menunjukkan bahwa Algoritma Boyer-Moore mampu mencocokkan dengan cepat.

Pada penelitian ini akan memanfaatkan web semantik dimana representasi pengetahuan menggunakan RDF kemudian SPARQL yang digunakan sebagai query untuk mengambil informasi yang terdapat dalam ontologi RDF. Informasi yang didapatkan dari pencarian akan diukur kemiripan dengan *keyword* pencarian dengan memanfaatkan Algoritma Boyer Moore Pengujiannya dilakukan dengan 3 kategori pencarian yaitu *keyword*, *simple*, dan *advanced search*.

II. TINJAUAN PUSTAKA

A. Web Semantik

Web 3.0 yang juga dikenal sebagai Semantic Web dimunculkan pertama kali oleh Tim Berners-Lee penemu dan pencipta *World Wide Web* (WWW) [6]. Web Semantik merepresentasikan bentuk baru dari pengolahan informasi pada halaman website dengan menggunakan teknologi framework berbasis RDF (*Resource Description Framework*). Perbedaan Semantic Web dengan generasi web sebelumnya adalah kemampuan komputer atau mesin dalam mendefinisikan data dalam bentuk metadata, kemudian mempelajari data tersebut untuk menjadi sebuah pengetahuan, sehingga pada akhirnya komputer atau program komputer dapat mengerti makna dari sebuah data yang dihasilkan atau ditampilkan pada sebuah halaman website [7].



GAMBAR 1. ARSITEKTUR WEB SEMANTIK [6]

B. RDF

RDF merupakan sebuah bahasa yang digunakan untuk menghasilkan sebuah standar untuk metadata dari sumber daya yang ada pada halaman website. Dimana metadata yang dihasilkan tersebut dapat mendefinisikan informasi yang akan menjadi “pengetahuan” dalam *Semantic Web* [7]. RDF menggunakan graf untuk merepresentasikan kumpulan pernyataan. Simpul dalam graf mewakili suatu entitas, dan tanda panah mewakili relasi antar entitas [2]. RDF terdiri dari 3 obyek tipe, yaitu Sumber daya (*Resources*), Properti (*Property*), dan Pernyataan (*Statement*). Dalam RDF, sebuah deskripsi dari sumber direpresentasikan sebagai sejumlah triple. Tiga bagian dari setiap triple disebut *subject*, *predicate*, dan *object*. *Subject* dari triple adalah URI yang mendefinisikan sumber. *Object* dapat berupa nilai literal sederhana, seperti string, numerik, atau tanggal, atau URI dari sumber daya lainnya yang berkaitan dengan *subject*. *Predicate* mengindikasikan hubungan di antara *subject* dan *predicate*, misalnya ini adalah nama atau tanggal lahir. *Predicate* juga berupa URI. URI *predicate* didapatkan dari kamus data (*vocabularies*), yaitu sejumlah URI yang dapat digunakan untuk merepresentasikan informasi tentang domain tertentu [3].

C. Ontology Web Language (OWL)

OWL adalah bahasa ontologi untuk sebuah web semantik yang dikembangkan oleh W3C kelompok kerja Web Ontology. OWL yang diharapkan menjadi salah satu bahasa ontologi harus dapat merepresentasikan bagian-bagian yang berguna dalam sebuah ontologi. OWL menggunakan kemampuan RDF untuk penjabaran statis dan kemampuan struktur kelas serta properti dari skema RDF. Kelas OWL dapat dijelaskan sebagai kombinasi logikal (irisian, gabungan, komplemen) dari kelas lainnya, atau sebagai penjelasan satu-persatu dari objek yang dimaksud serta melebihi kemampuan skema RDF. OWL dapat juga mendeklarasikan properti, mengorganisasikan properti tersebut ke dalam hierarki “*subproperty*”, dan menyediakan domain dan *range* untuk properti tersebut, seperti pada skema RDF. Domain dari properti OWL adalah kelas dalam OWL, dan *range* dapat berupa kelas dalam OWL atau tipe data yang dideklarasikan dari luar seperti string atau integer. OWL dapat menetapkan bahwa properti tersebut adalah transitif, asimetrik, fungsional, atau bertolak belakang dengan properti lainnya [3].

OWL dapat mengekspresikan objek (dapat juga disebut ‘individu’) mana yang dimiliki oleh kelas yang mana, dan apa nilai properti untuk sebuah individu. Pernyataan yang sama dapat dibuat pada kelas dan properti, pernyataan *disjoint* dapat dibuat pada kelas, serta *equality* dan *inequality* dapat juga disisipkan di antara individu [3].

D. SPARQL

SPARQL merupakan bahasa *query* untuk RDF. Query SPARQL didasarkan pada pencocokan pola graf. Pola graf yang paling sederhana adalah *triple pattern* yang mirip dengan RDF triple, hanya saja pola pada *query* dimungkinkan pemberian nama diluar terminologi RDF pada posisi subyek, predikat dan obyek [2].

E. Algoritma Boyer Moore

Algoritma Boyer-Moore diperkenalkan oleh Bob Boyer dan J.S. Moore pada tahun 1977. Pada algoritma ini pencocokan kata dimulai dari karakter terakhir kata kunci menuju karakter awalnya. Boyer-Moore merupakan salah satu Algoritma *Pattern Matching* yang cukup terkenal. Algoritma ini menggunakan beberapa kasus pengecekan teks (*input* karakter yang akan dibaca) dengan *Pattern* (pola yang akan disaring) [8].

Algoritma *Pattern Matching* Boyer-Moore ini berbasis pada 2 metode yaitu:

1. The Looking-Glass Technique

The Looking-Glass Technique melakukan perbandingan suatu karakter akhir pada kata *w* dengan suatu karakter pada teks *s*. Jika karakter tersebut sama maka jendela karakter akan berjalan mundur pada kedua string dan mengecek kembali kedua karakter. Mencari Suatu kecocokan String pada Teks dengan pola yang akan dicari dengan cara memindahkan atau menggesernya sampai Teks string selesai.

2. The Character-Jump Technique

Character-jump Technique melakukan suatu aksi ketika perbandingan antara dua karakter yang berbeda. Ada dua aksi yang tergantung pada teks *s* dan kata *w*

yang dimiliki; jika p yaitu karakter pada s yang sedang diproses yang tidak cocok maka ada dua kemungkinan aksi. Mencari karakter yang sesuai dan cara penggeseran sebuah karakter perbandingan terakhir.

Secara sistematis, langkah-langkah yang dilakukan Algoritma Boyer-Moore pada saat mencocokkan *pattern* adalah [9]:

1. Algoritma Boyer-Moore mulai mencocokkan *pattern* pada karakter paling akhir/kanan.
2. Dari kanan ke kiri, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
 - a. Karakter di *pattern* dan di teks yang dibandingkan tidak cocok (*mismatch*).
 - b. Semua karakter di *pattern* cocok, kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser *pattern* dengan mengambil nilai terbesar dari penggeseran *good-suffix* dan penggeseran *bad-character*, lalu mengulangi langkah 2 sampai *pattern* berada di ujung teks.

F. Receiver Operating Characteristic (ROC)

Receiver Operating Characteristic adalah hasil pengukuran klasifikasi dalam bentuk 2- dimensi. Berikut ada empat peluang yang dapat diformulasikan dalam tabel kontigensi 2x2 untuk menganalisis ROC [10]:

TABEL 1. PELUANG ANALISIS ROC

		Kelas Sebenarnya	
		True	False
Kelas Prediksi	True	True Positive (TP)	False Positive (FP)
	False	True Negative (TN)	False Negative (FN)

Untuk menghitung akurasi dapat menggunakan rumus (1):

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

III. METODOLOGI PENELITIAN

A. Perancangan Ontologi

Diperlukan sebuah ontologi yang merupakan teknologi web semantik untuk aplikasi pencarian tugas akhir. Ontologi digunakan untuk merepresentasikan pengetahuan pada saat pencarian. Dalam penelitian ini ontologi diberi nama "RepositoryTA".

Adapun struktur ontologi adalah sebagai berikut:

- a. Class TulisanIlmiah, terdiri dari subclass:
 - a. TugasAkhir, berisi *datatype* Judul (String) dan Abstrak (String)
- b. Class CivitasAkademika berisi *datatype* Nama (String), terdiri dari subclass
 - a. Dosen, berisi *datatype* NIP (Integer)
 - b. Mahasiswa, berisi *datatype* NIM (Integer)

Dari kelas dan *datatype* yang sudah dirancang, kemudian diperlukan juga suatu relasi yang menghubungkan antar kelas. Class bertindak sebagai subjek ontologi, dan *datatype* merupakan objek dalam ontologi. Predikat dalam ontologi

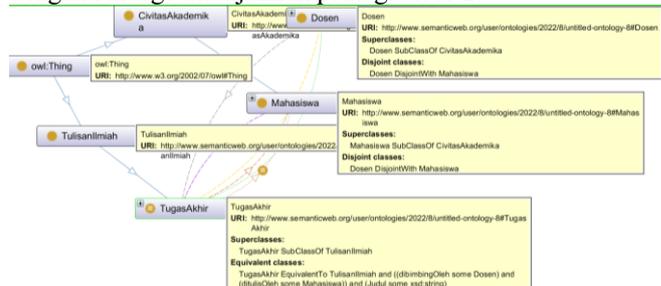
didefinisikan sebagai penghubung antara subjek dan objek agar memiliki arti kesatuan

Adapun predikat dari ontologi merupakan *object ontology*, bentuknya seperti tabel 2.

TABEL 2. DOMAIN RANGE "REPOSITORYTA" ONTOLOGI

Domain	Object Ontology	Range
TugasAkhir	ditulisOleh	Mahasiswa
Mahasiswa	menulis	TugasAkhir
TugasAkhir	dibimbingOleh	Dosen
Dosen	membimbing	TugasAkhir

Skema dari ontologi RepositoryTA yang dirancang dengan Protege ditunjukkan pada gambar 2:



GAMBAR 2. SKEMA ONTOLOGI REPOSITORYTA

B. Instance Ontology

Instance adalah memasukkan data-data kedalam sebuah ontologi. Dalam ontologi repositoryTA ini dilakukan instance dengan menggunakan editor Protégé. Sampel yang digunakan adalah sebanyak 10 tugas akhir, 10 mahasiswa dan 7 dosen yang selanjutnya dibagi kedalam subclass masing-masing tergantung dari jenis individu.

Judul	Abstrak	NamaMahasiswa	NIM	NamaDosen	NIP
1. Deteksi Wajah ...	Deteksi wajah merupakan tahap pertama yang harus dilakukan dalam ...	Winda Agustha	"9021381419070"	Desty Rodiah ...	"198912212020122011"
2. Deteksi Wajah ...	Deteksi wajah merupakan tahap pertama yang harus dilakukan dalam ...	Winda Agustha	"9021381419070"	M. Fachrudin ...	"198005222008121002"
3. Analisa Perband. ...	Logika fuzzy merupakan salah satu metode untuk melakukan analisis ...	Doni Milha	"9021381419107"	Dian Palagi Rin ...	"197802232006042002"
4. Analisa Perband. ...	Logika fuzzy merupakan salah satu metode untuk melakukan analisis ...	Doni Milha	"9021381419107"	Kanda Januar ...	"199001092019031012"
5. Pengaruh N-Gr. ...	Benita sudah menjadi kebutuhan utama bagi setiap orang, dengan ber ...	Fitria Khoirunnisa	"9021181520130"	Desty Rodiah ...	"198912212020122011"
6. Pengaruh N-Gr. ...	Benita sudah menjadi kebutuhan utama bagi setiap orang, dengan ber ...	Fitria Khoirunnisa	"9021181520130"	Novi Yuslani ...	"198211082012122001"
7. Ekstraksi Ciri Cit. ...	Penelitian tentang pengenalan ekspresi wajah manusia telah menjadi ...	Ayu Lestari	"9021181520021"	M. Fachrudin ...	"198005222008121002"
8. Ekstraksi Ciri Cit. ...	Penelitian tentang pengenalan ekspresi wajah manusia telah menjadi ...	Ayu Lestari	"9021181520021"	Oswari Arsalan ...	"198806282018031001"
9. Elefant Herdin. ...	Minyak mentah adalah salah satu energi utama yang dibutuhkan untuk ...	Muhammad Irfan T.	"9021281621046"	Dian Palagi Rin ...	"197802232006042002"
10. Elefant Herdin. ...	Minyak mentah adalah salah satu energi utama yang dibutuhkan untuk ...	Muhammad Irfan T.	"9021281621046"	M. Fachrudin ...	"198005222008121002"
11. Identifikasi Kal. ...	Dalam suatu penulisan kalimat, seringkali terdapat kesalahan yang tid ...	Ayu Hillana Putri	"9121402056"	M. Fachrudin ...	"198005222008121002"
12. Identifikasi Kal. ...	Dalam suatu penulisan kalimat, seringkali terdapat kesalahan yang tid ...	Ayu Hillana Putri	"9121402056"	Novi Yuslani ...	"198211082012122001"
13. Klasifikasi Swedi. ...	Pengklasifikasian daun dapat digunakan untuk mengeni identitas da ...	Rachman Hakim	"981320030"	M. Fachrudin ...	"198005222008121002"
14. Klasifikasi Swedi. ...	Pengklasifikasian daun dapat digunakan untuk mengeni identitas da ...	Rachman Hakim	"981320030"	Oswari Arsalan ...	"198806282018031001"
15. Klasifikasi peny. ...	Kucing adalah salah satu hewan peliharaan yang populer di dunia. Dal ...	Abdul Hafiz Muttaq.	"9021381621120"	Aki Syaikhri Ut. ...	"197812222006042002"
16. Klasifikasi peny. ...	Kucing adalah salah satu hewan peliharaan yang populer di dunia. Dal ...	Abdul Hafiz Muttaq.	"9021381621120"	Kanda Januar ...	"199001092019031012"
17. Penerapan Jatin. ...	Penelitian mengenai keamanan data telah banyak dilakukan dan dik ...	M. Yusuf	"9021381520043"	Oswari Arsalan ...	"198806282018031001"

GAMBAR 3. INSTANCE ONTOLOGI REPOSITORYTA

C. Perancangan RDF

Setelah ontologi di bentuk dan *instance* ditambahkan ke dalam protégé maka ontologi tersebut disimpan dan bisa didapatkan file dalam bentuk RDF yang berisikan semua meta data dari ontologi dan *instance* yang telah didefinisikan. Setelah ontologi buat dalam bentuk RDF database, *prefix*, *namespace* maka dapat dilakukan simulasi query SPARQL didalamnya.

D. Perancangan Query

Perancangan query dibagi dalam 3 macam, yaitu perancangan query untuk pencarian berdasarkan *keyword*,

query untuk pencarian berdasarkan pencarian *simple* dan query untuk *advanced search*.

a. Perancangan query *keyword search*

```

PREFIX
skripsi:<http://www.semanticweb.org/user/ontologies/2022/8/untitled-ontology-8#>
SELECT distinct ?Judul ?Abstrak ?NamaMahasiswa ?NIM
?NamaDosen ?NIP
WHERE
{
    ?tugasakhir skripsi:Judul ?Judul.
    ?tugasakhir skripsi:Abstrak ?Abstrak.
    ?tugasakhir skripsi:ditulisOleh ?Mahasiswa.
    ?Mahasiswa skripsi:Nama ?NamaMahasiswa.
    ?Mahasiswa skripsi:NIM ?NIM.
    ?tugasakhir skripsi:dibimbingOleh ?Dosen.
    ?Dosen skripsi:Nama ?NamaDosen.
    ?Dosen skripsi:NIP ?NIP.
    FILTER (regex(STR(?Judul) , '$keyword',
'i')||regex(STR(?NamaMahasiswa) , '$keyword',
'i')||regex(STR(?NamaDosen) , '$keyword', 'i'))
}
    
```

Pada query diatas, “\$keyword” merupakan kata kunci yang dimasukkan kedalam form pencarian.

b. Perancangan query *simple search*

```

PREFIX
skripsi:<http://www.semanticweb.org/user/ontologies/2022/8/untitled-ontology-8#>
SELECT distinct ?Judul ?Abstrak ?NamaMahasiswa ?NIM
?NamaDosen ?NIP
WHERE
{
    ?tugasakhir skripsi:Judul ?Judul.
    ?tugasakhir skripsi:Abstrak ?Abstrak.
    ?tugasakhir skripsi:ditulisOleh ?Mahasiswa.
    ?Mahasiswa skripsi:Nama ?NamaMahasiswa.
    ?Mahasiswa skripsi:NIM ?NIM.
    ?tugasakhir skripsi:dibimbingOleh ?Dosen.
    ?Dosen skripsi:Nama ?NamaDosen.
    ?Dosen skripsi:NIP ?NIP.
    FILTER (regex(STR(?$kategori) , '$keyword', 'i'))
}
    
```

Pada query diatas menampilkan seluruh data berdasarkan kategori. Sintak “\$kategori” merupakan pilihan kategori yang dideklarasikan dalam bentuk *combobox*, dan “\$keyword” merupakan kata kunci yang dimasukkan dalam *form* pencarian.

c. Perancangan query *advanced search*

```

PREFIX
skripsi:<http://www.semanticweb.org/user/ontologies/2022/8/untitled-ontology-8#>
SELECT distinct ?Judul ?Abstrak ?NamaMahasiswa ?NIM
?NamaDosen ?NIP
WHERE
{
    ?tugasakhir skripsi:Judul ?Judul.
    ?tugasakhir skripsi:Abstrak ?Abstrak.
    ?tugasakhir skripsi:ditulisOleh ?Mahasiswa.
    ?Mahasiswa skripsi:Nama ?NamaMahasiswa.
    ?Mahasiswa skripsi:NIM ?NIM.
    ?tugasakhir skripsi:dibimbingOleh ?Dosen.
    
```

```

?Dosen skripsi:Nama ?NamaDosen.
?Dosen skripsi:NIP ?NIP.
FILTER ($regex)
}
    
```

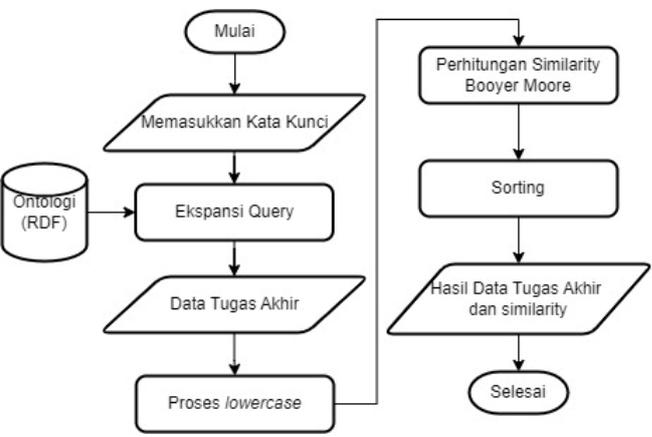
Pada query “\$regex” adalah pemecahan kunci pencarian yang bersifat opsional, dan menggunakan fungsi AND, OR, dan NOT, yang telah dideklarasikan dalam sintak PHP sebelumnya.

E. Perancangan Alur Kerja Sistem

Aplikasi yang akan dirancang menggunakan bahasa pemrograman PHP. Data RDF disimpan pada apache jena fuseki. apache jena Fuseki juga bertindak sebagai server untuk mengeksekusi SPARQL dalam mengolah data rdf. Untuk membuat RDF pada php digunakan library EasyRDF. EasyRDF ditulis dalam bahasa pemrograman PHP berbasis obyek. EasyRDF membangun sebuah grafik objek PHP yang dapat digunakan untuk mendapatkan data untuk ditampilkan pada halaman web [11].

Pada penelitian ini pencarian tugas akhir bukan hanya dengan semantik, tetapi juga menggunakan Algoritma Boyer Moore untuk mengukur kemiripan hasil pencarian. Alur kerja sistem pencarian dengan semantik dan algoritma Boyer Moore dapat dilihat pada gambar 4. Berikut ini adalah proses pencarian tugas akhir dengan semantik dan Algoritma Boyer Moore.

1. Memasukkan kata kunci pencarian, tahap selanjutnya adalah proses pencarian semantik.
2. Ontologi digunakan untuk memperluas makna dari tugas akhir yang digunakan.
3. Ekspansi Query merupakan hasil perluasan tugas akhir
4. Data tugas akhir merupakan database tugas akhir
5. Mengubah hasil pencarian menjadi *lowercase*
6. Algoritma Boyer Moore digunakan untuk mengukur kemiripan hasil pencarian dengan keyword yang dimasukkan pada form pencarian.
7. Mengurutkan data hasil pencarian berdasarkan hasil kemiripan yang paling tinggi.
8. Hasil pencarian tugas akhir yang relevan ditampilkan.

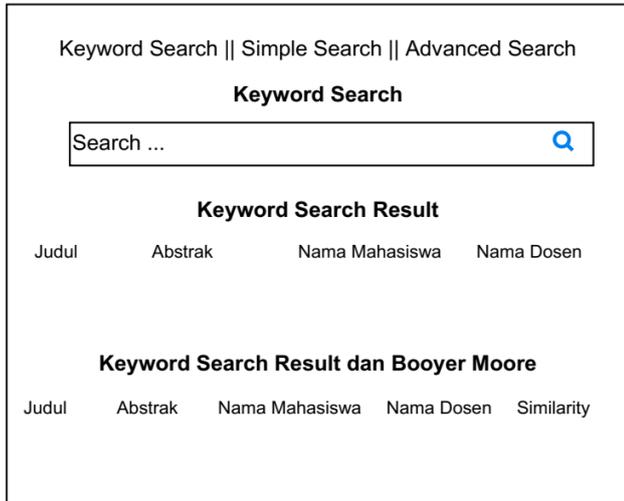


GAMBAR 4. ALUR KERJA SISTEM

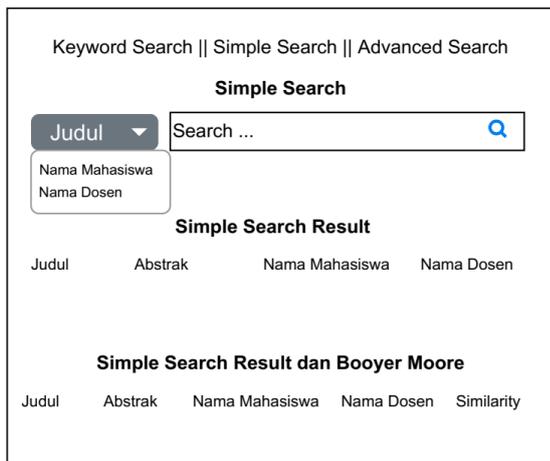
F. Perancangan Aplikasi Pencarian

Bentuk rancangan aplikasi akan dibagi menjadi 3 bagian, yaitu *Keyword Search*, *Simple Search* dan *Advanced Search*.

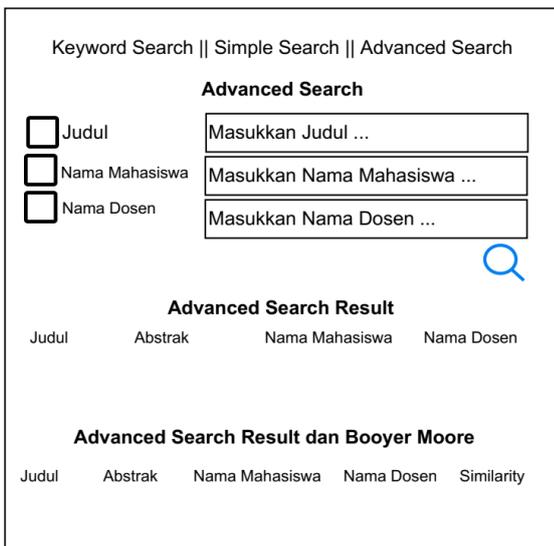
Gambaran umum dari rancangan aplikasi pencarian ditunjukkan pada gambar 5,6, dan 7.



GAMBAR 5. KEYWORD SEARCH



GAMBAR 6. SIMPLE SEARCH



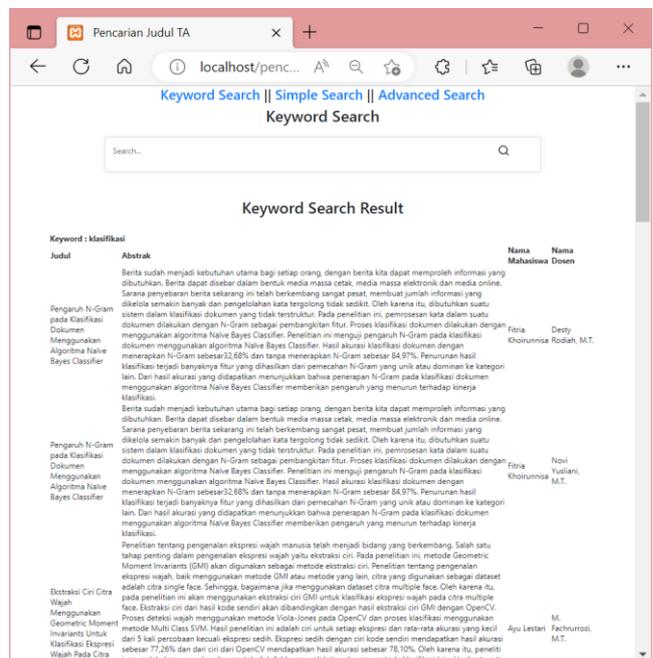
GAMBAR 7. ADVANCED SEARCH

IV. HASIL DAN DISKUSI

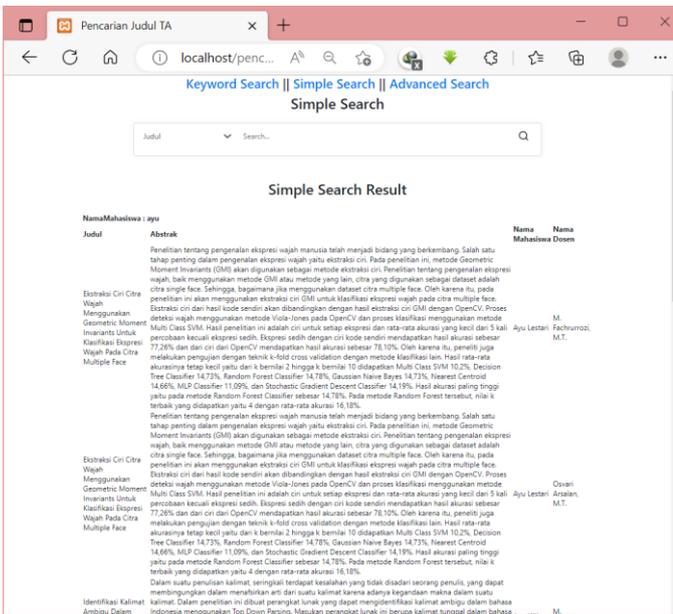
Untuk melakukan implementasi yang dipersiapkan adalah Jena Fuseki server, Apache XAMPP sebagai HTTP Request dan web browser.

Cara mengakses halaman aplikasi simulasi pencarian tugas akhir, alamat yang dituju dalam web browser adalah <http://localhost/PencarianTA>. Kemudian akan tampil halaman pencarian. Untuk melakukan pencarian, dapat dilakukan dengan memilih salah satu menu yang terdapat dibagian atas halaman yaitu *Keyword Search*, *Simple Search* dan *Advanced Search*. *Keyword Search* adalah pencarian yang memasukkan kata kunci yang diinginkan oleh pengguna yang berhubungan dengan tugas akhir. *Simple Search* adalah pencarian yang berdasarkan kategori, pengguna dapat memilih 3 kategori yaitu pencarian berdasarkan judul, pencarian berdasarkan nama mahasiswa dan pencarian berdasarkan nama dosen. *Advanced search* adalah pencarian yang dapat menggabungkan beberapa kategori pencarian.

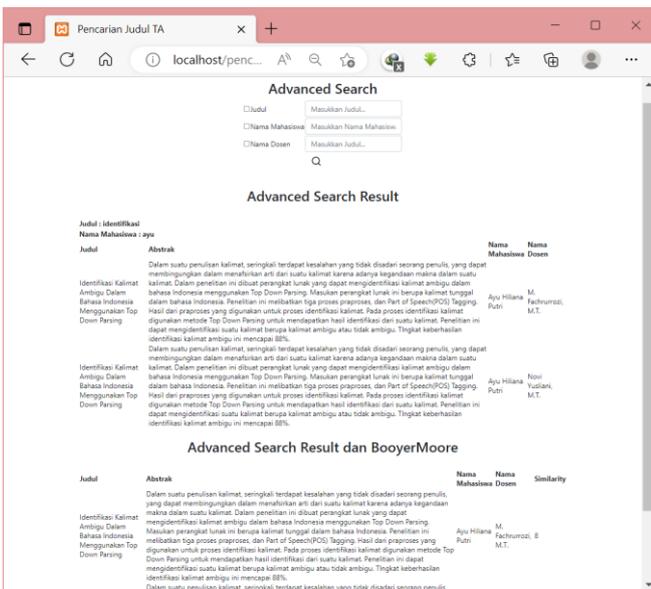
Berikut ini adalah contoh untuk *keyword search* menggunakan kata kunci “klasifikasi” untuk pencarian *Simple search* berdasarkan kategori “Nama Mahasiswa” dengan kata kunci “Ayu”. *Advanced search* mengkombinasikan kategori “Judul” (Kata kunci “identifikasi”) dan “Nama Dosen” (Kata kunci “fachrurrozi”).



GAMBAR 8. TAMPILAN HASIL KEYWORD SEARCH



GAMBAR 9. TAMPILAN HASIL SIMPLE SEARCH



GAMBAR 10. TAMPILAN HASIL ADVANCED SEARCH

A. Hasil Pengujian Pencarian

Dari ketiga model pencarian ontologi dan pencarian dengan ontologi dan Algoritma Boyer Moore maka dapat dilakukan pengujian untuk membandingkan seberapa cepat pencarian dengan ontologi dibandingkan dengan pencarian dengan ontologi dan Boyer Moore. Sehingga nantinya dapat ditarik sebuah kesimpulan apakah algoritma Boyer Moore berpengaruh besar terhadap lamanya eksekusi pencarian atau tidak.

Proses pengujian dilakukan dengan ketentuan sebagai berikut:

1. Kondisi *hardware* dan *software* dalam keadaan yang sama saat dilakukan pengujian.
2. Sampel pengujian diberikan sebanyak 10 sampel tugas akhir.
3. Ditetapkan 6 klausa kunci (Q) yang akan menjadi kata kunci sama dalam pencarian.

4. Setiap pencarian diulang sebanyak 5 kali pengulangan untuk satu buah klausa kunci (Q). Sehingga dapat ditarik rata-rata untuk dijadikan pedoman penentuan waktu dalam setiap hasil pencarian dari klausa kunci (Q)

Untuk hasil pengujian yang telah dilakukan dengan perhitungan rata-rata, maka diperoleh hasil yang dapat ditampilkan pada tabel 3.

TABEL 3. HASIL UJI COBA WAKTU PENCARIAN

Ke y w r d	T o t a l	Rata Rata dalam 5 kali								
		Keyword Search			Simple Search			Advanced Search		
		H1	H2	Selisih	H1	H2	Selisih	H1	H2	Selisih
Q1	8	0,0	0,0		0,0359	0,036	0,00009	0,0314	0,031	0,000
		60	60		41	63	0,0002	84	281	0,096
Q2	2	0,0	0,0		0,0320	0,032	0,00009	0,0243	0,024	0,000
		40	40		53	61	0,0000	39	816	0,063
Q3	2	0,0	0,0		0,0326	0,032	0,00009	0,0325	0,032	0,000
		34	34		54	63	0,0000	32	66	0,034
Q4	1	0,0	0,0		0,0292	0,029	0,00009	0,0364	0,036	0,000
		43	44		69	76	0,0000	04	69	0,0874
Q5	1	0,0	0,0		0,0362	0,036	0,00010	0,0311	0,031	0,000
		96	04	0,0000	804	941	3997	56	870	2634
Q6	2	0,0	0,0		0,0318	0,031	0,00007	0,0413	0,041	0,000
		36	36		25	34	0,0000	80	78	0,0896
Rata - rata					0,0001083		0,0000923			0,0000836

Keterangan:

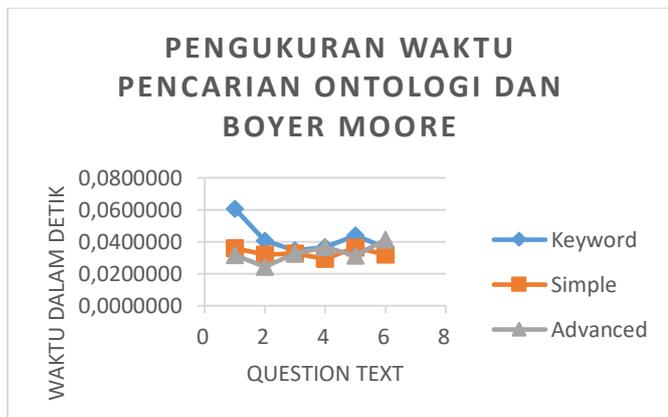
- Q1 = "klasifikasi"
- Q2 = "identifikasi"
- Q3 = "deteksi wajah"
- Q4 = "jaringan saraf"
- Q5 = "jaringan saraf propagasi"
- Q6 = "identifikasi kalimat ambigu"

H1 = Rata rata waktu pencarian dengan Ontologi

H2 = Rata rata waktu pencarian dengan Ontologi dan Boyer Moore

Berdasarkan tabel 3, dapat dilihat bahwa perbandingan waktu eksekusi antara pencarian ontologi dan pencarian ontologi dengan Boyer Moore adalah pencarian ontologi dan Boyer Moore memiliki waktu eksekusi lebih lama dibandingkan pencarian dengan query ontologi saja. Namun waktu yang dibutuhkan Boyer Moore dalam melakukan pengecekan *similarity* tidak membutuhkan waktu yang terlalu lama. Rata-rata selisih dari waktu eksekusi $\leq 0,0001$ perdetik. Sehingga tidak terlalu berpengaruh besar untuk lamanya pencarian.

Hasil rata-rata setiap pencarian ontologi dengan Boyer Moore pada tabel 3 dapat ditampilkan dalam bentuk grafik seperti terlihat pada gambar 11.



GAMBAR 11. PENGUKURAN WAKTU PENCARIAN ONTOLOGI DENGAN BOYER MOORE

Grafik gambar 11 merupakan hasil pengukuran rata-rata untuk pencarian ontologi dan Boyer Moore yang dilakukan sebanyak 5 kali untuk setiap *Question Test* (Q) terhadap model pencarian. Pada *keyword* search rata-rata waktu eksekusi lebih lama dibandingkan metode pencarian yang lain, dikarenakan pada pencarian *keyword query* yang digunakan lebih panjang dan *query filter* yang dilakukan lebih banyak. Sedangkan untuk *simple search* dan *advanced search* rata-rata waktu eksekusi lebih singkat dikarenakan *query* yang digunakan lebih pendek langsung ke kategori yang di tuju saja. Untuk *simple search* menghasilkan waktu eksekusi lebih stabil di bandingkan *advanced search*, karena *advanced search query filter* yang digunakan tergantung dari jumlah kategori yang dipilih oleh *user* sedangkan *simple search query filter* hanya menggunakan 1 (satu) *filter* saja terhadap 1 (satu) kategori yang dipilih.

B. Hasil Pengujian Algoritma Boyer Moore

Pengujian dalam pencarian data menggunakan Algoritma Boyer Moore ini dilakukan dengan melihat keakuratan sistem dalam mendapatkan data hasil pencarian. Uji coba dilakukan sebanyak 16 kali percobaan dengan 6 *keyword* yang berbeda. Perhitungan akurasi menggunakan rumus akurasi ROC sehingga diperoleh hasil pengujian seperti pada tabel 4.

TABEL 4. HASIL UJI COBA SIMILARITY PENCARIAN

No	Keyword	Judul	Total kata	TP	Data Benar	Hasil Salah	TN	Akurasi
1	Klasifikasi	Pengaruh N-Gram pada Klasifikasi Dokumen Menggunakan Algoritma Naïve Bayes Classifier	167	8	8	0	159	100
2		Pengaruh N-Gram pada Klasifikasi Dokumen Menggunakan Algoritma Naïve Bayes Classifier	167	8	8	0	159	100
3		Ekstraksi Ciri Citra Wajah Menggunakan Geometric Moment Invariants Untuk Klasifikasi Ekspresi Wajah Pada Citra Multiple Face	260	4	4	0	256	100
4		Ekstraksi Ciri Citra Wajah Menggunakan Geometric Moment Invariants Untuk Klasifikasi Ekspresi	260	4	4	0	256	100

		Wajah Pada Citra Multiple Face						
5		Klasifikasi Swedish Leaf Berdasarkan Bentuk Daun	170	4	4	0	166	100
6		Klasifikasi Swedish Leaf Berdasarkan Bentuk Daun	170	4	4	0	166	100
7		Klasifikasi penyakit kucing menggunakan metode random forest	161	2	4	2	157	98,75776
8		Klasifikasi penyakit kucing menggunakan metode random forest	161	2	4	2	157	98,75776
9	identifikasi	Identifikasi Kalimat Ambigu Dalam Bahasa Indonesia Menggunakan Top Down Parsing	122	7	7	0	115	100
10		Identifikasi Kalimat Ambigu Dalam Bahasa Indonesia Menggunakan Top Down Parsing	122	7	7	0	115	100
11	deteksi wajah	Deteksi Wajah Menggunakan Variasi Segmentasi dan Randomized Hough Transform (RHT)	150	3	3	0	147	100
12		Deteksi Wajah Menggunakan Variasi Segmentasi dan Randomized Hough Transform (RHT)	150	3	3	0	147	100
13	jaringan saraf	Penerapan Jaringan Saraf Propagasi Balik pada Kriptografi Simetri untuk Data Teks	173	2	2	0	171	100
14	jaringan saraf propagasi	Penerapan Jaringan Saraf Propagasi Balik pada Kriptografi Simetri untuk Data Teks	173	2	2	0	171	100
15	identifikasi kalimat	Identifikasi Kalimat Ambigu Dalam Bahasa Indonesia Menggunakan Top Down Parsing	122	3	3	0	119	100
16	identifikasi kalimat	Identifikasi Kalimat Ambigu Dalam Bahasa Indonesia Menggunakan Top Down Parsing	122	3	3	0	119	100
Rata-rata								99,84472

Ket:

Total Kata = Total Kata Judul + Total kata abstrak

TP = Hasil Pencarian yang benar (TP)

TN = Data yang tidak terseleksi dengan benar (TN)

Data Benar = Data yang seharusnya benar

Hasil Salah = Hasil Pencarian Salah

Berdasarkan hasil uji coba didapatkan hasil akurasi sebesar 99.84% pada 16 kali percobaan pencarian Rata-rata hasil pencarian memberikan *output* yang sesuai karena algoritma ini bersifat *match case* yang memudahkan dalam pencocokan kata kunci.

V. KESIMPULAN

Dari hasil penelitian ini didapatkan bahwa pencarian dengan memanfaatkan semantik web ontologi dapat diaplikasikan dan dapat juga dikombinasikan dengan Algoritma Boyer Moore untuk mengukur *similarity* antara data yang didapatkan dari hasil pencarian dengan keyword yang dimasukkan. Perbandingan waktu eksekusi antara pencarian dengan ontologi dengan pencarian dengan ontologi

dan Algoritma Boyer Moore dihasilkan bahwa pencarian dengan Boyer Moore membutuhkan waktu lebih lama secara rata-rata sekitar $\geq 0,0001$ perdetik dalam 5 kali percobaan dibandingkan pencarian dengan ontologi saja.

Untuk pengujian untuk Algoritma Boyer Moore dilakukan pengujian dengan ROC didapatkan hasil akurasi sebesar 99,84% untuk 16 kali percobaan. Dan dapat disimpulkan bahwa Algoritma Boyer Moore dapat mengukur *similarity* dengan cukup tepat.

DAFTAR PUSTAKA

- [1] N. S. Haq and K. Muslim, "Penerapan Sparql Dan Ontology Pada Pencarian Data Buku Dan Perbandingannya Dengan Pendekatan Relasional," *e-Proceeding of Engineering*, vol. 5, no. 2, pp. 3812–3819, 2018.
- [2] H. Wijayanto, W. L. YS, and T. Susyanto, "PENERAPAN WEB SEMANTIK DALAM Pencarian KATALOG BUKU DI PERPUSTAKAAN STMIK SINAR NUSANTARA SURAKARTA," *Jurnal Teknologi Informasi dan Komunikasi (TIKoSIN - STMIK Sinar Nusantara)*, 2013.
- [3] Gunawan and F. Halim, "PENERAPAN WEB SEMANTIK UNTUK APLIKASI Pencarian PADA REPOSITORI KOLEKSI PENELITIAN, STUDI KASUS: PROGRAM STUDI SISTEM INFORMASI STMIK MIKROSKIL MEDAN," *Jurnal SIFO Mikroskil*, vol. 15, no. 1, 2014.
- [4] E. Rahmanita, "PENCARIAN STRING MENGGUNAKAN ALGORITMA BOYER MOORE PADA DOKUMEN," *Jurnal Ilmiah NERO*, vol. 1, no. 1, 2014.
- [5] I. Ahmad, R. Indra Borman, G. G. Caksana, and J. Fakhrurozi, "IMPLEMENTASI STRING MATCHING DENGAN ALGORITMA BOYER-MOORE UNTUK MENENTUKAN TINGKAT KEMIRIPAN PADA PENGAJUAN JUDUL SKRIPSI/TA MAHASISWA (STUDI KASUS: UNIVERSITAS XYZ)," *SINTECH JOURNAL*, vol. 4, no. 1, 2021, [Online]. Available: <https://doi.org/10.31598>
- [6] N. Choudhury, "World Wide Web and Its Journey from Web 1.0 to Web 4.0," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 5, no. 6, pp. 8096–8100, 2014, [Online]. Available: www.ijcsit.com
- [7] Himawan and Supriati, "Evolusi Penggunaan Teknologi Web 3.0: Semantic Web," *JOURNAL OF INFORMATION SYSTEM, GRAPHICS, HOSPITALITY AND TECHNOLOGY*.
- [8] E. Rahmanita, "PENCARIAN STRING MENGGUNAKAN ALGORITMA BOYER MOORE PADA DOKUMEN," *Jurnal Ilmiah NERO*, vol. 1, no. 1, 2014.
- [9] C. Irawan, M. Riyan Pratama, and V. Wahanggara, "PERBANDINGAN ALGORITMA BOYER MOORE DAN BRUTE FORCE PADA Pencarian KAMUS BESAR BAHASA INDONESIA BERBASIS ANDROID."
- [10] asifatul Mu'awanah, "IMPLEMENTASI ALGORITMA BOYER MOORE PADA Pencarian DATA DI SISTEM INFORMASI MANAJEMEN LABORATORIUM FAKULTAS SAINS DAN TEKNOLOGI," 2018.
- [11] N. Fajrin Ariyani, A. Yoga Priyanto, and R. Sarno, "PEMODELAN GRANULARITAS TEMPORAL UNTUK Mencari RELASI ANTAR OBJEK WARISAN BUDAYA INDONESIA DENGAN MENGGUNAKAN ONTOLOGI," *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 15, no. 1, pp. 72–83, 2017.