

## PEMANFAATAN BAHASA ALAMI DALAM PENELUSURAN INFORMASI SKRIPSI MELALUI DIGITAL LIBRARY

<sup>1,\*</sup>Dewi Soyusiawaty, <sup>2</sup>Anna Hendri Soleliza Jones

<sup>1</sup>Teknik Informatika, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

e-mail: dewi.soyusiawaty@tif.uad.ac.id, anna.jones@tif.uad.ac.id

\*correspondence email

### Abstrak

Daftar judul skripsi yang ada di web digilib.uad.ac.id belum digunakan secara optimal. Kesulitan menentukan topik serta pengelolaan data pada sistem yang belum memadai menjadi beberapa kendala mahasiswa. Penelitian ini bertujuan membangun pencarian informasi skripsi dengan antarmuka bahasa alami agar mudah dalam menulis kriteria pencarian tanpa harus terikat formulir pencarian. Penelitian ini menggunakan data skripsi untuk dikelola. Pengambilan data dilakukan menggunakan pendekatan Natural Language Processing. Masukan dalam bentuk kalimat bahasa alami digunakan untuk mencari data. Proses Parsing dilakukan untuk memecah kalimat input dan mendeteksi kata kunci yang relevan. Pengembangan aturan produksi dalam Context Free Grammar diperlukan untuk menerjemahkan bahasa alami ke dalam query. Kalimat yang melewati tahap parser diterjemahkan ke dalam bahasa SQL. Sistem ini berhasil menampilkan informasi skripsi berupa daftar judul berdasarkan topik, metode, dan objek penelitian sesuai kalimat pencarian dengan nilai *precision* sebesar 89,3% dan *recall* sebesar 100%. Keberadaan model pencarian informasi dengan antarmuka bahasa alami dapat menjadi alternatif dalam proses pencarian informasi skripsi guna menyediakan sistem yang lebih fleksibel.

*The use of the existing digital library has not been used optimally. Difficulties in determining topics and managing data in an inadequate system are among the obstacles for students. This study aims to build a thesis information search with a natural language interface so that it is easy to write search criteria without having to be tied to a search form. This research uses thesis data to be managed. Data were collected using the Natural Language Processing approach. Input in the form of natural language sentences is used to find data. The parsing process is carried out to break down the input sentence and detect relevant keywords. Development of production rules in Context Free Grammar is necessary to translate natural language into queries. Sentences that go through the parser stage are translated into SQL language. This system succeeds in displaying thesis information in the form of a list of titles based on topics, methods, and research objects according to the search sentence with a precision value of 89,3% and a recall of 100%. The existence of an information retrieval model with a natural language interface can be an alternative in the thesis information search process in order to provide a more flexible system.*

**Kata Kunci:** *digital library, natural language processing, query, skripsi, NLIDB*

---

### PENDAHULUAN

Seiring kemajuan ilmu kecerdasan buatan, pengolahan bahasa alami (*Natural Language Processing/NLP*) mulai banyak diaplikasikan pada berbagai bidang, salah satunya adalah pada pengolahan basis data. NLP merupakan bidang ilmu kecerdasan buatan yang secara khusus mempelajari bagaimana mesin/komputer mampu memahami bahasa alami manusia dengan memberlakukan berbagai macam pendekatan.

Dengan mengaplikasikan NLP pada pengolahan basis data, pencarian data dapat dilakukan hanya dengan menggunakan sebuah perintah tertulis dalam bahasa alami. Konsep ini dikenal dengan istilah *Natural Language Interface to Database (NLIDB)*.

NLIDB adalah suatu antarmuka yang mampu mengolah bahasa alami manusia menjadi SQL sehingga dapat dieksekusi pada suatu basis data tertentu. Hadirnya NLIDB memberikan opsi lain bagi *end user* untuk mengakses basis data selain pengisian form pada antarmuka berbasis form [1].

Penentuan topik skripsi merupakan masalah awal yang dialami mahasiswa tingkat akhir. Mahasiswa biasanya kesulitan, ragu bahkan tidak tahu untuk menetapkan topik sehingga memerlukan waktu lebih untuk memulai skripsi. Salah satu fasilitas yang bisa dimanfaatkan mahasiswa untuk mencari skripsi yang sudah ada yaitu melalui web perpustakaan kampus. Informasi yang ditampilkan berupa judul skripsi, penulis, program studi, dan lokasi skripsi tersebut disimpan. Sistem belum dapat memfilter judul skripsi berdasarkan topik dan tahun tertentu.

Pengambilan informasi adalah proses mendapatkan informasi yang diperlukan dalam waktu yang diinginkan. Proses ini tergantung pada berbagai faktor untuk mengambil data yang akurat dalam waktu yang efisien. Faktor dapat berupa sistem yang digunakan untuk mengambil data, bagaimana data disimpan, desain skema dan lain-lain. Dalam era ledakan informasi dibutuhkan teknik yang ditingkatkan dan efisien untuk mengumpulkan data cerdas selain data terkait. Teknologi *Natural Language Processing* (NLP) mempertimbangkan cara interaksi antara manusia dan juga mesin yang memproses untuk mempelajari berbagai aturan dan kemudian menerapkan aturan itu untuk tugas yang sama dan untuk membuat data yang cerdas [2].

Pengambilan kembali informasi yang tersimpan dalam basis data bisa dilaksanakan dengan menggunakan pemrosesan bahasa alami. Beberapa teknik dan pendekatan digunakan serta dikembangkan dengan tujuan agar komputer lebih mampu memahami instruksi/permintaan manusia melalui penggunaan bahasa alami. Bahasa alami banyak digunakan oleh sebagian orang karena kemudahan dan keluwesan dalam menentukan kata kunci penelusuran dari informasi yang dicari. Dengan kemudahan dan spesifikasi yang tinggi dalam merumuskan sebuah kata kunci maka bahasa alami dapat melakukan penelusuran informasi lebih efektif [3].

Untuk itu diperlukan aturan produksi yang secara khusus akan mengenali model pola kalimat/pertanyaan yang diberikan dari masukan sistem. Aturan produksi ditetapkan berdasarkan pola keteraturan kalimat/pertanyaan yang digunakan untuk mengakses basis data. Keteraturan pola pertanyaan dikenali dengan mendaftar seluruh kalimat-kalimat pertanyaan yang mungkin dapat diberikan dalam pengaksesan basis data [4].

Penelitian menggunakan data skripsi untuk dikelola agar dapat dimanfaatkan banyak pihak baik bagi mahasiswa atau pengelola skripsi atau pengelola program studi untuk menganalisis data tersimpan sehingga dapat memberikan informasi yang bermanfaat untuk menyelesaikan permasalahan kesulitan menentukan topik dan juga lama penyelesaian skripsi. Pengambilan data dari sebuah sistem basis data relasional dapat dilakukan dengan menggunakan pendekatan Pemrosesan Bahasa Alami untuk keluwesan interaksi [5].

## **METODE PENELITIAN**

### **1. Analisis Data**

Sistem yang dirancang hanya dapat menerima pertanyaan sesuai dengan pola kalimat yang ditetapkan. Pertanyaan yang tidak sesuai dengan pola aturan akan diabaikan. Data masukan atau pola kalimat yang disiapkan seperti contoh di bawah beserta varian kalimat lainnya:

- a. Menampilkan semua data

Membuat perintah untuk menampilkan semua data skripsi (tanpa ada pengecualian). Contoh: Tampilkan semua data skripsi

b. Menampilkan beberapa kolom

Membuat perintah untuk menampilkan data skripsi dengan hanya menampilkan beberapa kolom yang dibutuhkan saja dari basis data. Contoh: Tampilkan semua topik dari skripsi, Tampilkan semua objek dari skripsi, Tampilkan metode dari skripsi.

c. Menampilkan semua data dengan kondisi

Membuat perintah untuk menampilkan skripsi sesuai dengan kondisi. Contoh: Tampilkan topik *Natural Language Processing*, Tampilkan skripsi *Data Mining*, Tampilkan metode *Apriori*, Daftar skripsi *data mining*, Tampilkan skripsi topik *data mining* dengan *fuzzy*.

## 2. Perancangan Aturan Produksi

Bahasa Indonesia telah memiliki grammar dan aturan produksi namun dalam penggunaannya untuk melakukan *query* basis data perlu diberikan aturan produksi secara khusus. Pola pertanyaan yang diberikan dalam bahasa alami untuk mendapatkan hasil dari *query* sistem basis data perlu disusun ke dalam suatu aturan produksi.

Aturan produksi dibangun dengan melakukan pemetaan keteraturan pola dari suatu pertanyaan [6]. Pertanyaan kalimat input memiliki pola keteraturan yang dapat disusun dengan notasi dari aturan *Context Free Grammars* (CFG) atau tata bahasa bebas konteks. CFG memiliki tujuan untuk menunjukkan bagaimana menghasilkan suatu untaian dalam sebuah bahasa. Notasi aturan CFG dapat disusun sebagai berikut: [7]

$$G = (T, N, P, S)$$

- a. T didefinisikan sebagai token yang tidak bisa diuraikan lagi. Dalam hal ini token merupakan daftar kata yang disimpan dalam tabel di basis data.

$T = \{\text{Tampilkan, Daftar, Berikan, Tunjukkan, ....., Skripsi, topik, metode, judul skripsi, ....., Natural Language Processing, Data Mining, Multimedia, Jaringan Komputer, ....., Porter Stemming, Levenstein Distance, K-Means, .....,}\}$

- b. N didefinisikan sebagai variabel untuk menyimpan string yang masih bisa diuraikan. Dalam hal ini adalah variabel untuk menyimpan komponen dari kalimat input bahasa alami.

$N = \{\text{Kalimat, Subjek, Atribut, Objek}\}$

1. Kalimat untuk menyimpan pola kalimat
2. Subjek untuk menyimpan daftar kata awal yang digunakan dalam kalimat
3. Atribut untuk menyimpan daftar kata tambahan yang mengikuti setelah kata awal dalam kalimat
4. Objek untuk menyimpan daftar kata terkait keyword dari data yang akan dicari, yaitu berupa topik dan metode.

- c. P merupakan pola aturan produksi dari kalimat yang dikenali, yaitu :

1. Kalimat  $\rightarrow$  <Subjek> <atribut> <objek>  
Merupakan pola umum dari kalimat input
2. Subjek  $\rightarrow$  {Tampilkan, Daftar, Berikan, Tunjukkan, ....}  
Subjek berisi daftar kata awal dari kalimat input.
3. Atribut  $\rightarrow$  {Skripsi, topik, metode, judul skripsi}

Atribut berisi kata lanjutan yang mungkin mengikuti setelah subjek.

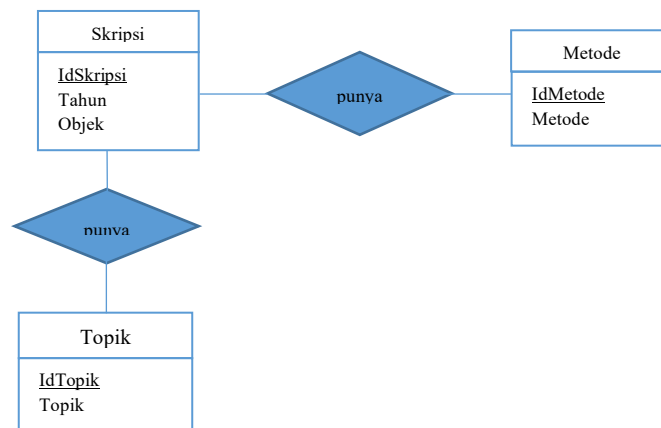
4. Objek berisi keyword dari pencarian. Biasanya berisi topik dan metode yang dicari <objek> → <topik> , <objek> → <metode>, <objek> → <topik> <metode>
5. Topik berisi daftar topik skripsi yang didefinisikan dalam basis data.  
<topik> → {Natural Language Processing, Data Mining, Multimedia, Jaringan Komputer, ...}
6. Metode berisi daftar metode penelitian yang didefinisikan dalam basis data.  
<metode> → {Porter Stemming, Levenstein Distance, K-Means, ...}

### 3. Perancangan Basis Data

Perancangan data yang diperlukan meliputi:

- a. Data Skripsi, yaitu berupa id skripsi, subjek/judul skripsi, tahun penyelesaian skripsi
- b. Data Topik, merupakan daftar topik skripsi, meliputi idtopik dan nama topik
- c. Data Metode, yaitu idmetode dan daftar metode yang digunakan dalam penelitian

Hubungan antar data skripsi, metode dan topik dapat dilihat pada Gambar 2. Tiap skripsi dapat dibedakan metode penyelesaian dan topik. Contoh : Skripsi dengan judul “Penerapan Algoritma Levenstein Distance untuk Pendeteksi Kemiripan Dokumen Tugas Mahasiswa” adalah skripsi dengan topik dari Natural Language Processing dan metode Levenstein Distance.



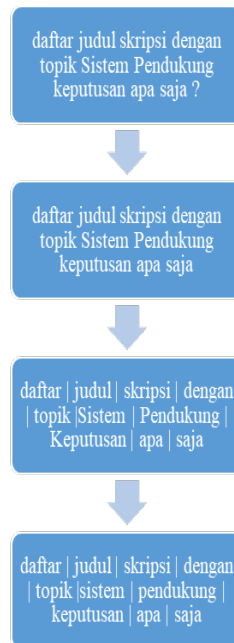
Gambar 2. Perancangan Basis Data

### 4. Perancangan NLIDB

Dengan adanya NLIDB, pengguna tidak perlu mempelajari bahasa *query* basis data melainkan cukup melakukan perintah dengan bahasa alami. Penyederhanaan dalam penyusunan query SQL merupakan hal yang sangat dibutuhkan. NLIDB merupakan alat bagi semua orang untuk mengakses data secara mudah dari basis data relasional [8]. Pengguna akan merasakan kemudahan dengan antar muka yang digunakan, tanpa ada kekhawatiran akan kesalahan interpretasi dari NLIDB, bahkan saat pengguna tidak ingat nama dari kolom pada tabel menyimpan data [9]. Berikut tahapan perancangan *NLIDB* yang diterapkan pada penelusuran informasi skripsi dengan pemanfaatan bahasa alami:

### 1. Natural Language Parsing

Merupakan proses *parsing* atau pemecahan kalimat bahasa alami dari input yang diberikan oleh pengguna. Pemrosesan awal kalimat input dilakukan oleh *scanner* yang akan membuat kalimat input menjadi sebuah daftar token. Pada penelitian ini *scanner* melakukan 3 aktivitas yakni: menghilangkan tanda baca, memecah kalimat dan mengubah menjadi huruf kecil. Pada Gambar 3 menampilkan proses pemecahan kalimat atau parsing kalimat dengan contoh satu input. Pada tahap pertama ini dihasilkan daftar token atau kata dari kalimat input bahasa alami oleh pengguna.



Gambar 3. Tahap *Natural Language Parsing*

### 2. Node Mapper

Pada tahap ini dilakukan pemetaan hasil dari proses *parsing* sebelumnya ke komponen *SQL*. Contoh pada kalimat input yang didefinisikan sebelumnya terdapat kata “Tampilkan”, maka secara logika perintah *SQL*, kata tersebut relevan dengan perintah *SQL* yaitu “Select”. Tabel 1 merupakan daftar token yang kemudian dapat dikonversi menjadi komponen *SQL*.

Tabel 1. Node Mapper

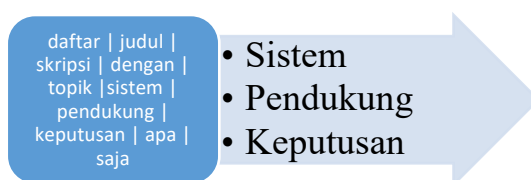
Sintaks Indonesia	Sintaks Query SQL
<b>Tampilkan</b>	Select
<b>Daftar</b>	Select
<b>Berikan</b>	Select
<b>Tunjukkan</b>	Select
<b>Semua</b>	*
<b>Dan</b>	and
<b>atau</b>	or
<b>Seperti</b>	Like

### 3. Parsing Reformulation

Daftar token yang terbentuk akan dianalisa oleh *parser* apakah sesuai dengan pembentukan pola kalimat yang telah ditetapkan. Penentuan struktur kalimat bisa jadi merupakan pekerjaan yang sulit bergantung dari bahasa yang digunakan. Pada tahap ini secara umum *parser* akan melakukan:

- a. Membaca daftar token
- b. Membaca cacah token
- c. Membaca isi token
- d. Membandingkan dengan aturan produksi

Analisis sintaks melakukan pelacakan terhadap token-token yang dihasilkan oleh *scanner* kemudian dibandingkan dengan daftar token yang tersedia. Jika sesuai dengan daftar token yang ada maka dilihat apakah terdapat kecocokan dengan aturan produksi yang ada. Sebagai contoh Gambar 4 merupakan tahap *Parsing Reformulation* dengan membaca isi token dalam termin topik apakah sesuai dengan daftar token yang diijinkan.



Gambar 4. Tahap Parsing Reformulation

### 4. SQL Translation

Merupakan tahap menterjemahkan hasil *parsing* ke dalam format *SQL*, contoh kalimat input: "Tampilkan topik data mining" maka akan menghasilkan query: "SELECT\*FROM skripsi WHERE topik="data mining"

### 5. Pengujian Recall dan Precision

*Recall* adalah rasio antara dokumen yang relevan yang berhasil ditemukan kembali dari seluruh dokumen relevan yang ada didalam sistem nilai tertinggi. *Recall* adalah 1 yang berarti seluruh dokumen dalam koleksi berhasil ditemukan kembali. Recall dapat dihitung dengan rumus (1).

$$\text{Recall} = \frac{\text{Jumlah dokumen relevan ditemukan}}{\text{Jumlah dokumen relevan}} \quad (1)$$

*Precision* adalah rasio dokumen relevan yang berhasil ditemukan kembali dari seluruh dokumen yang ditemukan kembali nilai tertinggi. *Precision* adalah 1 yang berarti seluruh dokumen yang ditemukan adalah relevan. Precision dapat dihitung dengan rumus (2).

$$\text{Precision} = \frac{\text{Jumlah dokumen relevan ditemukan}}{\text{Jumlah dokumen}} \quad (2)$$

Tujuan uji *Recall* dan *Precision* adalah untuk mendapatkan informasi hasil pencarian yang didapatkan oleh sistem. Hasil pencarian sistem bisa dinilai tingkat *recall* dan *precision*nya. *Precision* dapat dianggap sebagai ukuran ketepatan atau ketelitian,

sedangkan *recall* adalah kesempurnaan. Nilai *precision* adalah proporsi dokumen yang terambil oleh sistem adalah relevan.

## HASIL DAN PEMBAHASAN

Proses implementasi sistem dimulai dengan memberikan beberapa kriteria input seperti yang ditunjukkan pada kalimat-1 hingga kalimat-4.

Pada kalimat-1: “Tampilkan skripsi data mining”. Selanjutnya akan dilakukan tahap NLIDB dengan melakukan pemisahan antar kata. Rincian tahapan proses NLIDB dapat dilihat pada Tabel 2 dan hasil pencarian kalimat-1 dapat dilihat pada Gambar 5.

Pada tahap *SQL Transformation* generate *query SQL* menjadi: “SELECT \* FROM penelitian, topik WHERE penelitian.id\_topik = topik.id\_topik AND Topik.nama\_topik LIKE '%datamining%'”.

Tabel 2. Proses NLIDB Kalimat-1

Natural Language Parsing	CFG	Node Mapper
<i>Tampilkan</i>	Subjek	Select
<i>Skripsi</i>	Atribut	-
<i>Data Mining</i>	Objek	Like

Pencarian

Password

**QUERY :** SELECT \* FROM penelitian.topik WHERE topik.id\_topik = penelitian.topic and topik.nama\_topik LIKE '%data mining%'

12 ditemukan

Judul	Tahun
<a href="#">Penerapan Data Mining Untuk Pengelompokan Pelayanan Kesehatan Ibu Hamil Menggunakan Metode K-Medoids Di Provinsi Maluku Utara</a>	2017-2018
<a href="#">Tahun 2015</a>	
<a href="#">Implementasi Data Mining Untuk Melakukan Prediksi Jenis Tindakan Abortus Menggunakan K-Nearest Neighbor (K-NN)</a>	2017-2018
<a href="#">Implementasi Data Mining Algoritma K-Menas Untuk Clustering Data Psien Berdasarkan Usia dan Status Gizi Pada BPRB Dharma Husada</a>	2013-2014
<a href="#">Rancang Bangun Aplikasi Data Mining Menggunakan Metode Clustering AHC dan Time Series Linier pada Data Transaksi Penjualan</a>	2013-2014
<a href="#">Penerapan Data Mining Untuk Mengetahui factor-Faktor yang Mempengaruhi Kelahiran Bayi Menggunakan Association Rule</a>	2013-2014
<a href="#">Penerapan Data Mining Untuk Memprediksi Jumlah Penumpang Bus Trnas Jogja Menggunakan Time Series</a>	2013-2014
<a href="#">Penerapan Text Mining pada Klasifikasi Email Spam dengan Metode Naive Bayes</a>	2013-2014
<a href="#">Implementasi Data Mining Menggunakan Algoritma Cart Untuk Memprediksi Kelulusan Berdasarkan Data Akademis Mahasiswa Studi Kasus Teknik Informatika UAD</a>	2013-2014
<a href="#">Penerapan Data Mining dengan Algoritma Hash Based untuk Mengetahui Faktor-Faktor yang mempengaruhi prestasi Siswa Sekolah Dasar</a>	2013-2014
<a href="#">Implementasi Data Mining Menggunakan Fuzzy C-Clustering pada Data Transaksi Penjualan di Toko Safira</a>	2014-2015
<a href="#">Penerapan Data Mining Untuk Meramalkan Permintaan Produk Air minum Kemasan pada Perusahaan Air Minum "XYZ" dengan Menggunakan Metode Time Series dengan Teknik Exponential Smoothing</a>	2014-2015
<a href="#">Penerapan Algoritma FB-Growth dalam Data mining pada data Transaksi Penjualan Obat di Apotek</a>	2014-2015

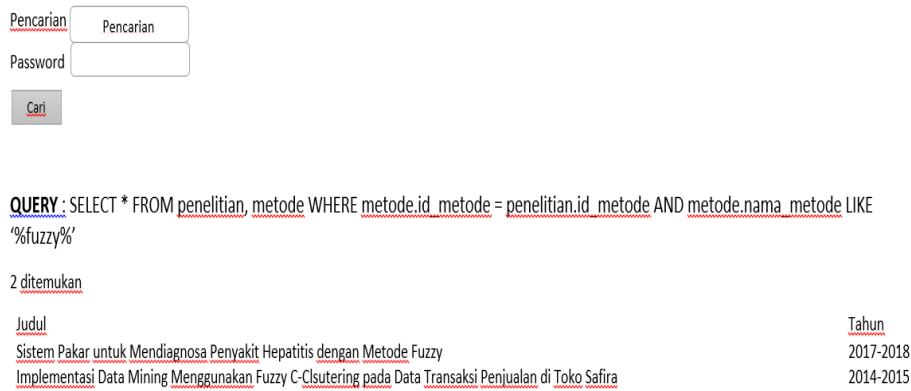
Gambar 5. Tampilan hasil kalimat “Tampilkan skripsi data mining”

Pada kalimat-2 akan dilakukan pencarian dengan kata kunci “Tampilkan skripsi metode fuzzy”. Tahapan proses NLIDB dapat dilihat pada Tabel 3 dan hasil pencarian dapat dilihat pada Gambar 6.

Tabel 3. Proses NLIDB Kalimat-2

Natural Language Parsing	CFG	Node Mapper
<i>Tampilkan</i>	Subjek	Select
<i>Skripsi</i>	Atribut	-
<i>Metode</i>	Atribut	-
<i>Fuzzy</i>	Objek	Like

Pada tahap *SQL Transformation* mengenerate *query SQL* menjadi: "SELECT \* FROM penelitian, metode WHERE metode.id\_metode=penelitian.id\_metode AND metode.nama\_metode LIKE '%fuzzy%'. Gambar 6 menampilkan hasil kalimat contoh.



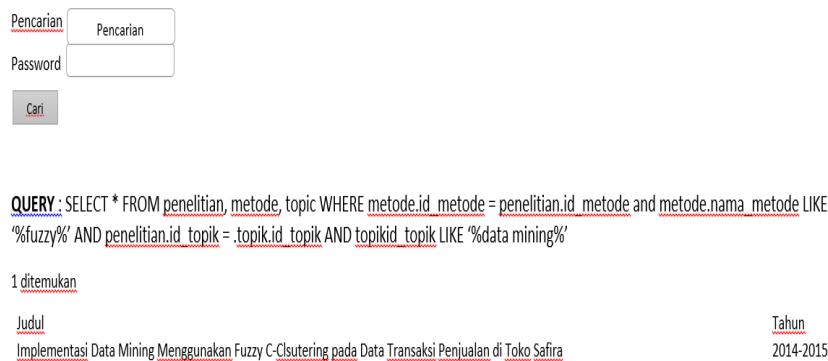
Gambar 6. Tampilan hasil kalimat "Tampilkan skripsi metode fuzzy"

Pada kalimat-3 akan dilakukan pencarian dengan kata kunci "Tampilkan skripsi topik data mining metode fuzzy". Tabel 4 dan Gambar 7 menunjukkan proses NLIDB untuk kalimat-3.

Tabel 4. Proses NLIDB Kalimat-3

Natural Language Parsing	CFG	Node Mapper
<b>Tampilkan</b>	Subjek	Select
<b>Skripsi</b>	Atribut	-
<b>Topik</b>	Atribut	-
<b>Data Mining</b>	Objek	Like
<b>Metode</b>	Atribut	-
<b>Fuzzy</b>	Objek	Like

Pada tahap *SQL Transformation* mengenerate *query SQL* menjadi: SELECT \* FROM penelitian, metode, topik WHERE metode.id\_metode = penelitian.id\_metode AND metode.nama\_metode LIKE '%fuzzy%' AND penelitian.id\_topik = topik.id\_topik AND topik.id\_topik LIKE '%data mining%'. Gambar 7 menampilkan pencarian berdasar jumlah.



Gambar 7. Pencarian berdasarkan jumlah



Pada Kalimat-4 dilakukan pencarian dengan kata kunci “Tampilkan topik skripsi data mining pada jenis tindakan abortus”. Tabel 5 dan Gambar 8 menunjukkan proses NLIDB untuk kalimat-4.

Tabel 5. Proses NLIDB Kalimat-4

Natural Language Parsing	CFG	Node Mapper
<b>Tampilkan</b>	Subjek	Select
<b>topik</b>	Atribut	-
<b>skripsi</b>	Atribut	-
<b>Data Mining</b>	Objek	Like
<b>pada</b>	Atribut	-
<b>Jenis tindakan abortus</b>	Objek	Like

Pada *SQL Transformation* mengenerate *query SQL* menjadi: `SELECT * FROM penelitian, topik WHERE penelitian.id_topik = topik.id_topik AND Topik.nama_topik LIKE '%data mining%' AND skripsi.objek LIKE '%jenis tindakan abortus%'`. Gambar 8 menampilkan hasil kalimat “Tampilkan topik skripsi data mining pada jenis tindakan abortus”

Pencarian

Password

**QUERY :** `SELECT * FROM penelitian, topik WHERE penelitian.id_topik = topik.id_topik AND Topik.nama_topik LIKE '%data mining%' AND skripsi.objek LIKE '%jenis tindakan abortus%'`

1 ditemukan

Judul

Implementasi Data Mining untuk Melakukan Prediksi Jenis Tindakan Abortus Menggunakan K-Nearest Neighbor (K-NN)

Tahun

2017-2018

Gambar 8. Tampilkan topik skripsi data mining pada jenis tindakan abosrtus.

Pengujian ini dilakukan sebanyak 5 kali sesuai tampilan sebelumnya, dengan mencoba menginputkan beberapa *query*. Hasil pencarian selanjutnya dilakukan perhitungan untuk pengujian *recall* dan *precision* pada sistem dengan menggunakan sebanyak data temuan. Tabel 6 merupakan rekapitulasi pengujian. Ra adalah jumlah dokumen relevan ditemukan, Rs merupakan jumlah dokumen relevan dalam koleksi dan Rt adalah jumlah dokumen ditemukan.

Tabel 6. Hasil Pengujian Recall dan Precison

Pengujian	Ra	Rs	Rt	Recall	Precision
<b>1</b>	11	11	12	1	0,916
<b>2</b>	2	2	2	1	1
<b>3</b>	1	1	1	1	1
<b>4</b>	1	1	1	1	1
<b>5</b>	1	1	2	1	0,5
<b>6</b>	17	17	18	1	0,94
<b>Nilai Rata-rata</b>				1	0.893

Nilai rata-rata *Recall* yang dihasilkan sebesar 1 menunjukkan semua dokumen relevan dapat ditemukan oleh sistem dan *Precision* sebesar 0,893 menunjukkan terdapat dokumen lain ditemukan sistem selain dokumen relevan.

## KESIMPULAN

Penelitian menghasilkan sebuah aplikasi yang mampu melakukan *query* terhadap data yang disimpan dalam basis data relasional dengan menggunakan bahasa Indonesia. Kalimat pencarian yang diajukan oleh user hanya dapat diproses sesuai dengan aturan produksi yang dibangun. Pertanyaan yang tidak sesuai dengan aturan produksi yang ditetapkan akan diberikan pesan kesalahan atau diabaikan. Membebaskan pertanyaan menjadi lebih luas adalah saran yang bisa dilakukan untuk penelitian selanjutnya sehingga pengguna dapat memiliki variasi pertanyaan yang lebih banyak. Pengujian *recall* dan *precision* menunjukkan semua dokumen relevan dapat ditemukan dengan hasil masing-masing sebesar 100% untuk *recall* dan 89,3% untuk *precision*, walaupun masih terdapat dokumen lain yang ditemukan selain dokumen relevan.

## DAFTAR PUSTAKA

- [1] D. Soyusiawaty and A. H. S. Jones, "Penerapan Antar Muka Bahasa Alami dalam Pencarian Informasi Skripsi pada suatu Program Studi," *Semnasteknomedia*, February 2018.
- [2] P. Devale and A. Deshpande, "Probabilistic Context Free Grammar : An Approach to Generic Interactive Natural Language Interface to Database," *Journal of Information, Knowledge and Research in Computer Engineering*, vol. 01, no. 02, 2011.
- [3] K. W. C. X. Yilin Gao, "Natural Language Interface for Relational Database," *Final Report*, 2 12 2016.
- [4] N. Nihalani, S. Silakari and M. Motwani, "Natural Language Interface for Database : A Brief Review," *International Journal of Computer Science Issues*, vol. 8, March 2011.
- [5] M. Tyagi, "Natural Language Interface to Database," *International Journal of Science and Research*, vol. 3, no. 5, May 2014.
- [6] R. John and S. Govilkar, "Information Retrieval Technique for Web using NLP," *International Journal of Natural Language Computing*, vol. 6, no. 5, October 2017.
- [7] Y. Gao, K. Wang and C. Xu, "Natural Language Interface for Relational Database," *Final Report*, 2 Dec 2016.
- [8] S. Raharjo and S. Hartati, "Antar Muka Bahasa Alami untuk Melakukan Query terhadap Terjemahan Al Quran," *Teknologi*, vol. 7, no. 1, 2014.
- [9] J. Patel and J. Dave, "A Survey : Natural Language Interface to Database," *International Journal of Advanced Engineering and Research Development*, 2015.