

Konversi Bahasa Indonesia ke Perintah *Data Manipulation Language* pada *Structured Query Language* menggunakan *Natural Language Processing*

Sirajuddin^{a,1}, Yulita Salim^{b,2}, Mardiyah Hasnawi^{c,6}

^{a b c} Universitas Muslim Indonesia, Jln. Urip Sumoharjo km. 05, Makassar 90231, Indonesia

¹ sirajuddin.cloudstring@gmail.com; ² yulita.salim@umi.ac.id; ³ mardiyah.hasnawi@umi.ac.id

INFORMASI ARTIKEL	ABSTRAK
<p>Diterima : xx – xx – 20xx Direvisi : xx – xx – 20xx Diterbitkan : xx – xx – 20xx</p> <p>Kata Kunci: Bahasa <i>Natural Language Processing</i> <i>Structured Query Language</i> <i>Data Manipulation Language</i> <i>Query</i></p>	<p>Bahasa merupakan alat yang digunakan manusia untuk berkomunikasi. Tidak hanya untuk melakukan komunikasi antara manusia dengan manusia yang lainnya, bahasa juga dapat menjadi perantara komunikasi manusia dengan komputer. Agar bahasa alami dapat dimengerti oleh komputer, maka harus diolah menggunakan <i>Natural Language Processing</i> (NLP). <i>Structured Query Language</i> (SQL) merupakan salah satu bahasa pemrograman yang digunakan dalam <i>database</i>. Bahasa SQL terdapat perintah <i>Data Manipulation Language</i> (DML) yang memiliki perintah dasar <i>select</i>, <i>insert</i>, <i>update</i> dan <i>delete</i>. Perintah tersebut memungkinkan pengguna memanipulasi <i>database</i> misalnya menampilkan, menambah, mengubah atau menghapus data. Namun, tidak semua pemula dalam dunia pemrograman yang baru mempelajari tentang DML dapat memahami penggunaan perintah-perintah tersebut dengan cepat karena adanya kendala bahasa. Untuk itu, penelitian ini bertujuan untuk membuat aplikasi yang dapat mengonversi bahasa Indonesia menjadi perintah DML menggunakan NLP.</p> <p>Penelitian ini telah berhasil membuat aplikasi yang dapat mengonversi bahasa Indonesia menjadi perintah dasar DML menggunakan NLP. Dari proses uji coba didapatkan hasil dari keseluruhan 40 kali uji coba, ada 29 perintah yang berhasil dikonversi dengan benar sehingga didapatkan akurasi sebesar 72.5%.</p>

I. Pendahuluan

Bahasa merupakan alat yang digunakan manusia untuk berkomunikasi. Fungsi bahasa sebagai alat komunikasi yaitu sebagai sarana penyampaian informasi kepada orang lain baik secara lisan maupun tulisan. Sedangkan bahasa alami atau *natural language* adalah bahasa yang dapat dimengerti dan dipahami oleh seseorang pada lokasi tertentu, sebagai contoh bahasa alami dari orang Indonesia adalah bahasa Indonesia. Tidak hanya untuk melakukan komunikasi antara manusia dengan manusia yang lainnya, bahasa juga dapat menjadi perantara komunikasi manusia dengan komputer.

Bahasa alami tidak akan langsung dimengerti oleh komputer. Agar bahasa alami dapat dimengerti oleh komputer, maka harus diolah pada setiap tahapan *Natural Language Processing* (NLP). NLP atau pengolahan bahasa alami merupakan salah satu bidang ilmu *Artificial Intelligence* (Kecerdasan Buatan) yang mempelajari komunikasi antara manusia dengan komputer melalui bahasa alami [1].

Bahasa yang digunakan manusia untuk berkomunikasi dengan komputer dikenal dengan bahasa pemrograman. Bahasa *Structured Query Language* (SQL) merupakan salah satu bahasa pemrograman yang digunakan dalam *database*. *Database* dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah [2].

Bahasa SQL terbagi dalam dua bagian yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). DDL adalah perintah yang digunakan untuk mendefinisikan skema ke dalam *database* sedangkan DML adalah perintah yang digunakan untuk memanipulasi *database* misalnya menghapus, mengubah dan menambah [3].

Pada DML terdapat beberapa perintah seperti *Select*, *Insert*, *Update* dan *Delete*. Untuk memanipulasi *database* pengguna dapat menggunakan perintah-perintah tersebut. Namun, tidak semua pemula dalam dunia pemrograman yang baru mempelajari tentang DML dapat memahami penggunaan perintah-perintah tersebut dengan cepat karena adanya kendala bahasa. Oleh karena itu, dengan menggunakan bahasa alami khususnya bahasa Indonesia untuk mengeksekusi *query*, pengguna akan lebih mudah memahami perintah-perintah tersebut dengan cepat. Untuk itu, penulis membangun sebuah aplikasi yang dapat mengonversi bahasa Indonesia menjadi perintah DML menggunakan NLP.

II. Natural Language Processing

Natural Language Processing (NLP) dapat didefinisikan sebagai kemampuan suatu komputer untuk memproses bahasa, baik lisan maupun tulisan yang digunakan oleh manusia dalam percakapan sehari-hari. Untuk proses komputasi, bahasa harus direpresentasikan sebagai rangkaian simbol yang memenuhi aturan tertentu. Secara sederhana, NLP mencoba untuk membuat komputer dapat mengerti perintah-perintah yang ditulis dalam standar bahasa manusia [4]. Dalam NLP terdapat beberapa tahapan yaitu Analisis Leksikal (*Scanner*), Analisis Sintaksis (*Parser*), Analisis Semantik (*Translator*) dan Analisis Pragmatik (*Evaluator*) [5].

A. Analisis Leksikal (*Scanner*)

Pada tahap analisis leksikal atau *pre-processing* terdapat 4 (empat) aktifitas yang akan dilakukan yaitu:

- 1) *Case Folding*
Proses mengubah semua huruf dalam dokumen menjadi huruf kecil, agar sama untuk setiap kata yang ada [6].
- 2) *Tokenizing*
Proses untuk mengubah kalimat, paragraf, dokumen menjadi teks atau token-token tertentu dimana spasi dan tanda baca digunakan sebagai acuan pemisah antar token [7].
- 3) *Stopword*
Tahap pemilihan kata-kata penting dari hasil token yang akan mewakili dokumen. *Stop-words removal* ini membuang kata-kata yang tidak penting [7].
- 4) *Stemming*
Proses mencari kata dasar dari sebuah kata imbuhan, dengan menghilangkan imbuhan-imbuhan pada kata [7].

B. Analisis Sintaksis (*Parser*)

Proses menganalisis daftar token yang terbentuk dari proses leksikal yang akan dianalisis oleh *parser* dan *lexicon*. Analisis sintaksis melakukan pelacakan terhadap token-token yang dihasilkan oleh *scanner* kemudian dibandingkan dengan daftar token yang tersedia [8].

C. Analisis Semantik (*Translator*)

Untuk melakukan analisis semantik, terdapat istilah *understander* dan *knowledge base*. *Understander* bekerja sama dengan *knowledge base*. *Understander* akan menentukan makna sebuah kalimat dengan melihat kata-kata atau frasa-frasa yang terdapat dalam *knowledge base* bermakna [4]. *Knowledge base* merupakan basis pengetahuan yang dijadikan acuan oleh *understander* dalam menentukan makna suatu kalimat. *Knowledge base* berisi sekumpulan informasi yang terorganisasi dengan baik, yang secara unik diformat ke dalam bentuk struktur data [10].

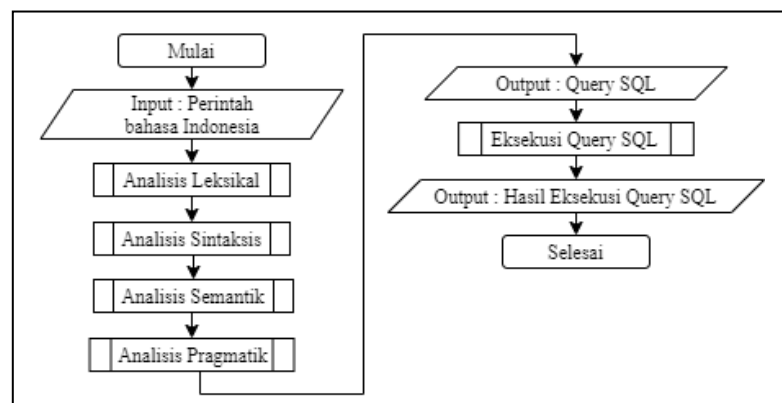
D. Analisis Pragmatik (*Evaluator*)

Analisis pragmatik atau *evaluator* berfungsi untuk menentukan jawaban akhir *query* berdasarkan hasil keluaran *translator* (analisis semantik) yang berupa perintah *query* menurut tipe dan atribut sesuai dengan notasi masing-masing tipe *query* [9].

III. Hasil dan Pembahasan

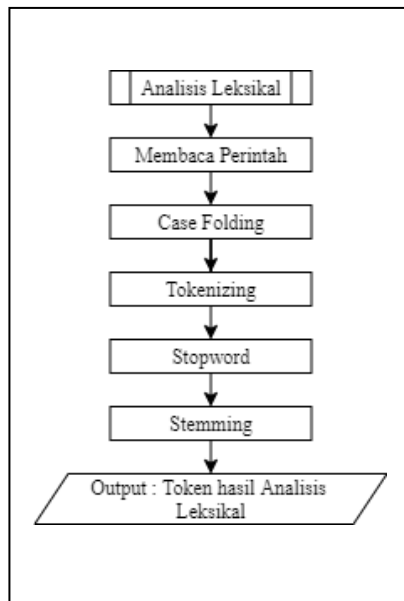
A. Perancangan Proses Natural Language Processing

Flowchart merupakan cara yang digunakan untuk menggambarkan proses atau tahapan *natural language processing* dalam penelitian ini. *Flowchart* tahapan NLP secara umum dapat dilihat pada gambar berikut:

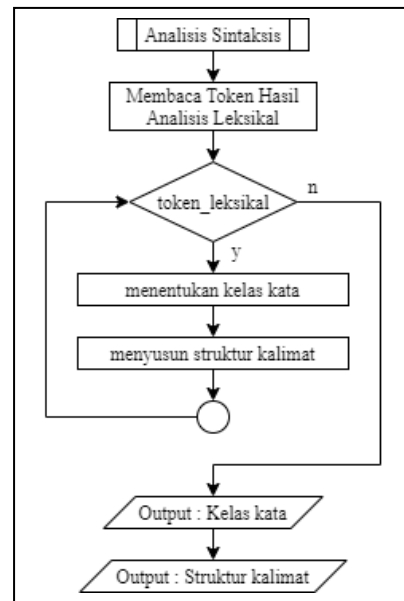


Gambar 1. *Flowchart* tahapan NLP

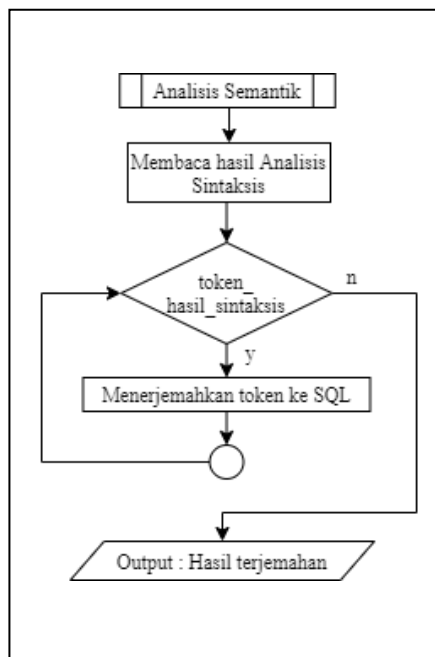
Flowchart analisis leksikal, analisis sintaksis, analisis semantik dan analisis pragmatik secara berurutan dapat dilihat pada gambar berikut ini:



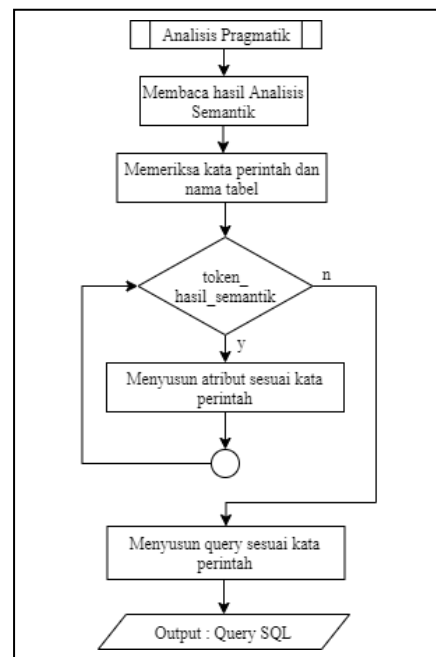
Gambar 2. *Flowchart* proses analisis leksikal



Gambar 3. *Flowchart* proses analisis sintaksis



Gambar 4. *Flowchart* proses analisis semantik



Gambar 5. *Flowchart* proses analisis pragmatik

B. Perancangan Knowledge Base

Knowledge base pada penelitian ini akan disimpan dalam format *file* XML. Berikut ini daftar kata *stopword* yang digunakan pada proses analisis leksikal:

Tabel 1. Daftar *stopword*

<i>Stopword</i>
adalah
adapun
agak
agaknya

akan
akankah
aku
akulah
amatlah
anda
andalah
antar
antaranya
antara
dan seterusnya

Proses *parsing* atau penentuan kelas kata dan struktur kalimat pada proses analisis sintaksis memerlukan daftar kelas kata yang tidak termasuk kata *stopword* seperti pada tabel berikut:

Tabel 2. Daftar kelas kata

Kata Perintah	Kata Kuantitas	Kata Sambung	Operator
tampil	semua	jika	atau
tunjuk	jumlah	kalau	dan
cari	banyak	dengan	tidak
berapa	total	yang	bukan
hapus		mana	kecuali
ubah			selain
edit			sama
masuk			kurang
tambah			lebih
			!=
			<>
			=
			<
			<=
			>
			>=

Untuk melakukan analisis semantik, *translator* memerlukan *understander* yang berisi daftar token dan *sintak SQL* yang disimpan dalam *knowledge base*. Isi dari *understander* dapat dilihat pada tabel berikut:

Tabel 3. Daftar *sintak SQL*

Token	Sintak SQL
tampil	<i>select</i>
tunjuk	<i>select</i>
cari	<i>select</i>
berapa	<i>select</i>
tambah	<i>insert</i>
masuk	<i>insert</i>
ubah	<i>update</i>
edit	<i>update</i>
hapus	<i>delete</i>

jika	<i>where</i>
mana	<i>where</i>
dan	<i>and</i>
atau	<i>or</i>
tidak	<i>!=</i>
bukan	<i>!=</i>
kecuali	<i>!=</i>
selain	<i>!=</i>
sama	<i>=</i>
tidak sama dengan	<i>!=</i>
kurang dari	<i><</i>
kurang atau sama dengan	<i><=</i>
lebih dari	<i>></i>
lebih atau sama dengan	<i>>=</i>
kurang lebih	<i><></i>
<i>!=</i>	<i>!=</i>
<i><=</i>	<i><=</i>
<i><</i>	<i><</i>
<i>>=</i>	<i><=</i>
<i>></i>	<i>></i>
<i><></i>	<i><></i>
jumlah	<i>count(*)</i>
banyak	<i>count(*)</i>
total	<i>count(*)</i>
semua	<i>count(*)</i>

C. Implementasi

Aplikasi yang dibangun adalah aplikasi yang dapat mengonversi perintah bahasa alami menjadi perintah DML. Bahasa alami yang dimaksud adalah bahasa Indonesia dengan kosakata yang baku. Beberapa perintah yang dapat diproses adalah perintah menampilkan, menambah, mengubah dan menambah data. Uji coba konversi perintah *select* atau menampilkan data dapat dilihat pada Gambar 6.

Gambar 6. Gambar hasil konversi perintah *select*

Uji coba konversi perintah *insert* atau menambah data dapat dilihat pada Gambar 7.

Tambahkan data "1304BB802", "Tugas Akhir", "4", "8" ke dalam tabel mata_kuliah

Hasil Konversi

insert into mata_kuliah values ('1304BB802', 'Tugas Akhir', '4', '8')

Jalankan SQL Bersihkan

Gambar 7. Gambar hasil konversi perintah *insert*

Uji coba konversi perintah *update* atau mengubah data dapat dilihat pada Gambar 8.

h data pada tabel mata_kuliah pada kolom nama_mata_kuliah menjadi "Skripsi" jika kode_mata_kuliah sama dengan "1304BB802"

Hasil Konversi

update mata_kuliah set nama_mata_kuliah = 'Skripsi' where kode_mata_kuliah = '1304BB802'

Jalankan SQL Bersihkan

Gambar 8. Gambar hasil konversi perintah *update*

Uji coba konversi perintah *delete* atau menghapus data dapat dilihat pada Gambar 9.

Hapus data pada tabel mata_kuliah jika kolom sks lebih dari 3

Hasil Konversi

delete from mata_kuliah where sks > '3'

Jalankan SQL Bersihkan

Gambar 9. Gambar hasil konversi perintah *delete*

D. Pembahasan Proses Natural Language Processing

Sebelum sistem menampilkan hasil konversi dari perintah bahasa Indonesia yang dimasukkan *user*, terdapat beberapa proses yang terjadi. Perintah tersebut akan melalui beberapa tahap *natural language processing* yang dimulai dari proses analisis leksikal sampai proses analisis pragmatik. Berikut ini pembahasan bagaimana perintah diproses dengan *natural language processing* sampai menghasilkan *query* DML.

1) Penginputan Perintah

Misalkan pengguna *menginputkan* perintah berikut ini:

Perintah *Select*: Tampilkan data pada Tabel Mata_Kuliah

Perintah *Insert*: Tambahkan data "1304BB802", "Tugas Akhir", "4", "8" ke dalam tabel mata_kuliah

Perintah *Update*: Ubah data pada tabel mata_kuliah pada kolom nama_mata_kuliah menjadi "Skripsi" jika kode_mata_kuliah sama dengan "1304BB802"

Perintah *Delete*: Hapus data pada tabel mata_kuliah jika kolom sks lebih dari 3

2) Analisis Leksikal

- Perintah *Select*

Perintah: Tampilkan data pada Tabel Mata_Kuliah

Case Folding: tampilkan data pada tabel mata_kuliah

Tokenizing: tampilkan | data | pada | tabel mata_kuliah

Stopword: tampilkan | mata_kuliah

Stemming: tampil | mata_kuliah

- *Perintah Insert*

Perintah: Tambahkan data “1304BB802”, “Tugas Akhir”, “4”, “8” ke dalam tabel mata_kuliah

Case Folding: tambahkan data “1304BB802”, “Tugas Akhir”, “4”, “8” ke dalam tabel mata_kuliah

Tokenizing: tambahkan | data | “1304BB802” | “Tugas Akhir” | “4” / “8” | ke | dalam | tabel | mata_kuliah

Stopword: tambahkan | “1304BB802” | “Tugas Akhir” | “4” | “8” | mata_kuliah

Stemming: tambah | “1304BB802” | “Tugas Akhir” | “4” | “8” | mata_kuliah

- *Perintah Update*

Perintah: Ubah data pada tabel mata_kuliah pada kolom nama_mata_kuliah menjadi “Skripsi” jika kode_mata_kuliah sama dengan “1304BB802”

Case Folding: ubah data pada tabel mata_kuliah pada kolom nama_mata_kuliah menjadi “Skripsi” jika kode_mata_kuliah sama dengan “1304BB802”

Tokenizing: ubah | data | pada | tabel | mata_kuliah | pada | kolom | nama_mata_kuliah | menjadi | “Skripsi” | jika | kode_mata_kuliah | sama | dengan | “1304BB802”

Stopword: ubah | mata_kuliah | nama_mata_kuliah | “Skripsi” | jika | kode_mata_kuliah | sama | dengan | “1304BB802”

Stemming: ubah | mata_kuliah | nama_mata_kuliah | “Skripsi” | jika | kode_mata_kuliah | sama | dengan | “1304BB802”

- *Perintah Delete*

Perintah: Hapus data pada tabel mata_kuliah jika kolom sks lebih dari 3

Case Folding: hapus data pada tabel mata_kuliah jika kolom sks lebih dari 3

Tokenizing: hapus | data | pada | tabel | mata_kuliah | jika | kolom | sks | lebih | dari | 3

Stopword: hapus | mata_kuliah | jika | sks | lebih | 3

Stemming: hapus | mata_kuliah | jika | sks | lebih | 3

3) Analisis Sintaksis

- *Perintah Select*

Perintah: Tampilkan data pada Tabel Mata_Kuliah

Kelas Kata: kata perintah -> tampil

nama tabel -> mata_kuliah

Struktur Kalimat: kata_perintah, nama_tabel

- *Perintah Insert*

Perintah: Tambahkan data “1304BB802”, “Tugas Akhir”, “4”, “8” ke dalam tabel mata_kuliah

Kelas Kata: kata perintah -> tambah

data -> “1304BB802” “Tugas Akhir” “4” “8”

nama tabel -> mata_kuliah

Struktur Kalimat: kata_perintah, data, nama_tabel

- *Perintah Update*

Perintah: Ubah data pada tabel mata_kuliah pada kolom nama_mata_kuliah menjadi “Skripsi” jika kode_mata_kuliah sama dengan “1304BB802”

Kelas Kata: kata perintah -> ubah

nama tabel -> mata_kuliah

nama kolom -> nama_mata_kuliah

data -> “Skripsi”

kata sambung -> jika

nama kolom -> kode_mata_kuliah

operator -> sama

data -> “1304BB802”

Struktur Kalimat: kata_perintah, nama_tabel, nama_kolom, data, kata_sambung, nama_kolom, operator, data

- *Perintah Delete*

Perintah: Hapus data pada tabel mata_kuliah jika kolom sks lebih dari 3

Kelas Kata: kata perintah -> hapus

```

nama tabel -> mata_kuliah
kata sambung -> jika
nama kolom -> sks
operator -> lebih
data -> 3

```

Struktur Kalimat: kata_perintah, nama_tabel, kata_sambung, nama_kolom, operator, data

4) Analisis Semantik

- *Perintah Select*

Perintah: Tampilkan data pada Tabel Mata_Kuliah

Terjemahan : tampil -> select

```
nama_tabel -> mata_kuliah
```

- *Perintah Insert*

Perintah: Tambahkan data “1304BB802”, “Tugas Akhir”, “4”, “8” ke dalam tabel mata_kuliah

Terjemahan: tambah -> insert

```
data -> “1304BB802” “Tugas Akhir” “4” “8”
```

```
nama_tabel -> mata_kuliah
```

- *Perintah Update*

Perintah: Ubah data pada tabel mata_kuliah pada kolom nama_mata_kuliah menjadi “Skripsi” jika kode_mata_kuliah sama dengan “1304BB802”

Terjemahan: ubah -> update

```
nama_tabel -> mata_kuliah
```

```
nama_kolom -> nama_mata_kuliah
```

```
data -> “Skripsi”
```

```
jika -> where
```

```
nama_kolom -> kode_mata_kuliah
```

```
sama -> =
```

```
data -> “1304BB802”
```

- *Perintah Delete*

Perintah: Hapus data pada tabel mata_kuliah jika kolom sks lebih dari 3

Terjemahan: hapus -> delete

```
nama_tabel -> mata_kuliah
```

```
jika -> where
```

```
nama_kolom -> sks
```

```
lebih -> >
```

```
data -> 3
```

5) Analisis Pragmatik

- *Perintah Select*

Perintah: Tampilkan data pada Tabel Mata_Kuliah

Query SQL: select * from mata_kuliah

- *Perintah Insert*

Perintah: Tambahkan data “1304BB802”, “Tugas Akhir”, “4”, “8” ke dalam tabel mata_kuliah

Query SQL: insert into mata_kuliah values (‘1304BB802’, ‘Tugas Akhir’, ‘4’, ‘8’)

- *Perintah Update*

Perintah: Ubah data pada tabel mata_kuliah pada kolom nama_mata_kuliah menjadi “Skripsi” jika kode_mata_kuliah sama dengan “1304BB802”

Query SQL: update mata_kuliah set nama_mata_kuliah = ‘Skripsi’ where kode_mata_kuliah = ‘1304BB802’

- *Perintah Delete*

Perintah: Hapus data pada tabel mata_kuliah jika kolom sks lebih dari 3

Query SQL: delete from mata_kuliah where sks > ‘3’

IV. Kesimpulan

Dari hasil penelitian ini dengan melalui proses perancangan, implementasi dan pengujian, maka diperoleh kesimpulan sebagai berikut:

1. Aplikasi yang dibuat mampu memproses perintah menampilkan (*select*), menambah (*insert*), mengubah (*update*) dan menghapus data (*delete*).
2. Aplikasi yang dibuat mampu mengonversi perintah bahasa Indonesia menjadi perintah DML. Berdasarkan hasil uji coba konversi perintah, didapatkan akurasi sebesar 72.5%.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada pihak-pihak yang telah membantu dalam proses penelitian ini sampai penerbitan paper ini.

Daftar Pustaka

- [1] J. Karlos, H. Sujaini, and H. Anra, "Konversi Bahasa Indonesia ke SQL (Structured Query Language) dengan Pendekatan Mesin Penerjemah Statistik," vol. 3, no. 1, 2016.
- [2] Y. Priadi, *Kolaborasi SQL dan ERD dalam Implementasi Database*. Yogyakarta: Andi Yogyakarta, 2014.
- [3] Rachmawati, "Analisis Kesalahan Menerapkan Bahasa SQL (Structure Query Language) Mata Kuliah Basis Data," vol. 1, no. 2, pp. 27–34, 2019.
- [4] E. A. Lisangan, "Natural Language Processing dalam Memperoleh Informasi Akademik Mahasiswa Universitas Atma Jaya Makassar," *J. Temat.*, vol. 1, no. March 2013, pp. 1–9, 2015.
- [5] D. Soyusiawaty and A. H. S. Jones, "Penerapan Antar Muka Bahasa Alami dalam Pencarian Informasi Skripsi pada Suatu Program Studi," pp. 13–18, 2018.
- [6] R. Rosyadi, S. Al-faraby, and Adiwijaya, "Penerapan Question Answering System Pada Pembahasan Agama Islam Dengan Pendekatan Metode Pattern Based," vol. 2, no. 4, pp. 140–148, 2018.
- [7] E. Supriyati and M. Iqbal, "Pengukuran Similarity Tema pada Juz 30 Al Qur'an Menggunakan Teks Klasifikasi," vol. 9, no. 1, pp. 361–370, 2018.
- [8] S. Raharjo and S. Hartati, "Antarmuka Bahasa Alami Untuk Melakukan Query Terhadap Terjemahan Al-Quran," *J. Teknol.*, vol. 7, no. 1, pp. 12–19, 2014.
- [9] S. Wibisono, "Aplikasi Pengolah Bahasa Alami untuk Query Basisdata Akademik dengan Format Data XML," vol. 18, no. 1, pp. 65–79, 2013.
- [10] F. Fahmi, "Implementasi Perintah Menampilkan Data Menggunakan Bahasa Indonesia Dengan Natural Language Processing," no. 021.