

PENERJEMAHAN TEKS BAHASA INDONESIA MENJADI *DATA DEFINITION LANGUAGE* (DDL) DENGAN PENANGANAN KALIMAT MAJEMUK

Iqram Anwar¹, Ken Kinanti Purnamasari²

^{1,2} Teknik Informatika – Universitas Komputer Indonesia

Jl. Dipatiukur 112-114 Bandung 40132

Email : iqramanwar@email.unikom.ac.id¹, ken.kinanti@email.unikom.ac.id²

ABSTRAK

Perlu keahlian khusus untuk dapat berinteraksi dengan DBMS yaitu dengan memahami format baku bahasa SQL (*Structured Query Language*). Adapun cara lain untuk berinteraksi dengan DBMS yaitu menggunakan perantara bahasa alami seperti bahasa Indonesia. Bahasa alami tidak bisa langsung diproses oleh DBMS, sehingga diperlukannya sebuah sistem yang dapat menerjemahkan bahasa alami ke bahasa SQL. Salah satu bentuk bahasa basis data yaitu *Data Definition Language* (DDL) yang digunakan untuk membuat, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data. Metode yang digunakan untuk menerjemahkan kalimat perintah dalam bahasa Indonesia ke *query* DDL adalah metode *Rule-Based*. Proses pada sistem yang dibangun memiliki dua tahap utama yaitu *Preprocessing* dan Translasi. Tahap *Preprocessing* terdiri dari *case folding*, *filtering*, *tokenizing* kata, *stemming*, dan *removing stopword*. Tahap Translasi terdiri dari deteksi kata kunci, *tokenizing* perintah, identifikasi perintah DDL, identifikasi konten, dan penyusunan *query*. Berdasarkan hasil pengujian terhadap 25 jenis kalimat masukan dengan jumlah keseluruhan kalimat sebanyak 93, hasil rata-rata akurasi yang didapatkan yaitu 72,04%.

Kata kunci : Translasi, *Query*, Bahasa Alami, *Structured Query Language*, *Data Definition Language*

1. PENDAHULUAN

DBMS (*Database Management System*) merupakan media perantara bagi pengguna dengan suatu basis data yang tersimpan pada media penyimpanan. Cara berinteraksi antara pengguna dengan basis data tersebut dengan bahasa khusus yang sudah diatur oleh perusahaan DBMS. Sebuah Bahasa Basis Data biasanya dapat dipilah ke dalam 2 bentuk yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML)[1]. *Data Definition Language* (DDL) pada basis data digunakan untuk membuat, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data [1]. *Data Manipulation*

Language merupakan bentuk bahasa Basis Data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data [1].

Beberapa penelitian telah dilakukan dalam penggunaan basis data dengan perantara bahasa alami diantaranya pada bahasa Inggris [2][3] dan bahasa Indonesia [4][5][6]. Penelitian pada bahasa Inggris sudah dapat menerjemahkan bahasa alami ke *query* DDL dan DML namun nilai akurasi tidak diketahui. Penelitian pada bahasa Indonesia hanya menerjemahkan bahasa alami ke DML yang berfokus pada *query select* dalam kasus pengambilan data pada suatu basis data. Metode yang digunakan untuk mentranslasi bahasa alami adalah dengan melihat kata – kata yang ada pada kalimat bahasa alami tanpa melihat struktur kalimatnya kemudian kata – kata tersebut dibandingkan dengan daftar *keyword* untuk mendapatkan *field* yang dimaksudkan [5]. Hasil pada penelitian tersebut mendapatkan nilai akurasi sebesar 86% [5]. Adapun penelitian yang berfokus pada DDL diperoleh akurasi 91,57% [6]. Penelitian dalam bahasa Indonesia hanya dapat mentranslasikan kalimat masukan menjadi satu *query*, karena sistem yang telah dibangun tidak dapat mendeteksi kalimat masukan yang akan menghasilkan *query* lebih dari satu.

Berdasarkan uraian di atas, dibutuhkan suatu metode yang dapat mendeteksi kalimat masukan yang akan menghasilkan *query* lebih dari satu. Pada tahap proses translasi mengikuti pada tahap penelitian sebelumnya yang menggunakan metode *rule-based*.

2. ISI PENELITIAN

Pada bab ini akan membahas mengenai landasan teori, metode penelitian, hasil dan pembahasan yang merupakan masalah yang akan dianalisis, gambaran umum sistem, tahapan proses sistem dan hasil penelitian.

2.1 Pemrosesan Bahasa Alami

Pemrosesan Bahasa Alami sering diterapkan pada aplikasi tertulis yang sering menemui kesulitan karena tulisan setiap orang tidak selalu baku sesuai dengan tata bahasa sehingga tidak sesuai dengan aturan – aturan yang berlaku. Salah satu aplikasi dari pemrosesan bahasa alami adalah sistem penerjemah

bahasa alami (*natural language translator*) seperti sistem penerjemah dari bahasa Inggris ke bahasa Indonesia. *Translator* tidak hanya dapat menerjemahkan kata per kata (kamus) tetapi juga dapat mentranslasikan dari bahasa asal ke bahasa target dengan maksud yang dapat dimengerti [7].

2.2 Kalimat Majemuk

Kalimat majemuk adalah kalimat yang terdiri atas dua klausa atau lebih yang dimana antara klausa yang satu dengan klausa yang lain saling berhubungan [8]. Umumnya, ahli bahasa membagi hubungan antar klausa ke dalam dua jenis yaitu koordinasi dan subordinasi.

1. Hubungan koordinasi (Majemuk Setara)

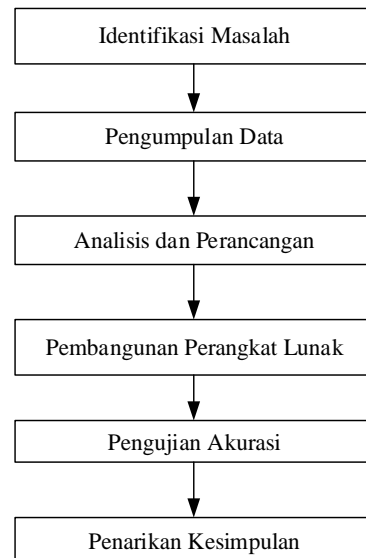
Hubungan koordinasi menggabungkan dua klausa atau lebih yang masing – masing mempunyai kedudukan setara dalam struktur kalimat [8]. Kalimat yang terbentuk dari dua klausa atau lebih dengan hubungan antar klausa terbentuk secara koordinatif dapat disebut dengan kalimat majemuk setara, sehingga kalimat tersebut terdiri dari klausa – klausa yang disusun secara koordinasi dengan kata penghubung yang dimana setiap klausa dapat berdiri sendiri menjadi kalimat.

2. Hubungan Subordinasi (Majemuk Bertingkat)

Hubungan subordinasi menunjukkan hubungan yang hierarkis, yakni menggabungkan dua klausa atau lebih secara bertingkat, ada yang berfungsi sebagai klausa utama dan ada yang berfungsi sebagai klausa bawahan[8]. Kalimat yang terbentuk dari beberapa klausa yang dihubungkan dengan kata penghubung yang bersifat subordinatif menyebabkan klausa yang satu menjadi bagian dari klausa yang lain, artinya salah satu klausa bergantung pada klausa lain dan tidak bisa berdiri sendiri. Kalimat tersebut disebut dengan kalimat majemuk bertingkat.

2.3 Metode Penelitian

Pada penelitian ini penulis akan menggunakan metode penelitian deskriptif. Metode ini digunakan karena penelitian ini menggambarkan secara sistematis fakta dan karakteristik objek dan subjek yang diteliti secara tepat[9]. Tahap penelitian yang dilakukan terdiri dari identifikasi masalah, pengumpulan data, analisis dan perancangan, pembangunan perangkat lunak, penarikan kesimpulan.



Gambar 1. 1 Alur Penelitian

Metode pembangunan perangkat lunak yang digunakan adalah *Sequential Linear Model* [10]. Adapun tahap – tahap yang ada pada model tersebut yaitu :

1. Analisis

Analisis data masukan yaitu kalimat perintah DDL dalam bahasa Indonesia untuk membangun tahap *preprocessing* yang sesuai dan penentuan kata kunci untuk tahap translasi.

2. Desain

Pada tahap ini, langkah awal yang dilakukan adalah membuat rancangan sistem secara keseluruhan, meliputi skema *Preprocessing* dan Translasi.

3. Implementasi

Tahap ini merupakan tahap pembuatan aplikasi dengan desain yang telah dibuat pada tahap perancangan serta implementasi proses translasi perintah bahasa Indonesia ke dalam sintak SQL.

4. Pengujian

Pada tahap ini, pengujian penerjemah perintah DDL menggunakan bahasa Indonesia apakah sudah benar dan dapat menjalankan perintah sesuai yang diharapkan.

2.4 Analisis Masalah

Penelitian yang sudah dilakukan terkait proses translasi bahasa alami ke *query* SQL dengan perantara bahasa Indonesia yang berfokus pada DML *query SELECT* didapatkan akurasi sebesar 86%[5], sedangkan untuk kasus translasi pada DDL didapatkan rata – rata akurasi sebesar 91,57%[6]. Namun pada penelitian sebelumnya yang sudah disebutkan, penelitian tersebut hanya mentranslasikan kalimat masukan ke dalam satu *query* SQL, sehingga sistem yang sudah ada tidak dapat mentranslasi kalimat yang mengandung perintah *query* lebih dari satu.

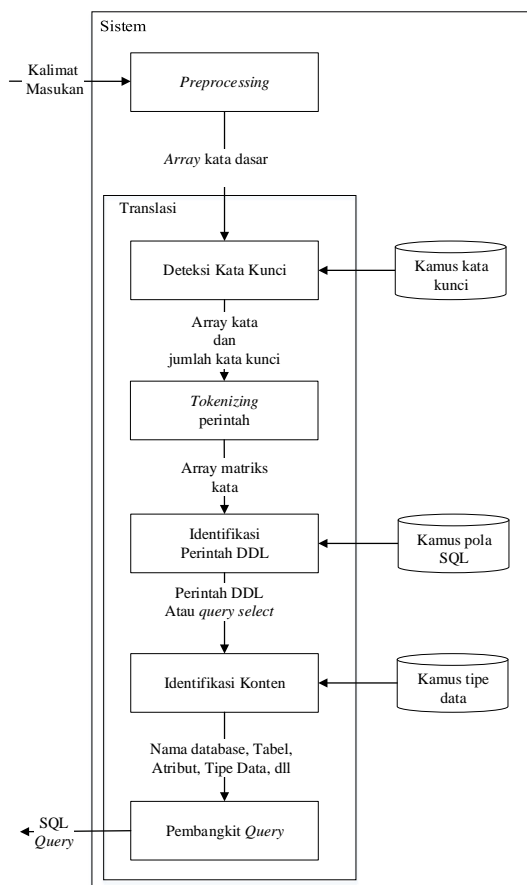
Tabel 1. 1 Contoh Perintah Dalam Bahasa Indonesia

No	Perintah DDL dalam Bahasa Indonesia	Hasil Translasi
1.	buatkan database dengan nama unikom kemudian hapus database dengan nama db_kampus	CREATE DATABASE unikom;
		DROP DATABASE db_kampus;

Penelitian ini bertujuan untuk menambah fitur agar kalimat masukan yang mempunyai perintah *query* lebih dari satu dapat diproses translasi dan menghasilkan *query* DDL lebih dari satu.

2.5 Gambaran Umum Sistem

Penelitian ini memiliki dua proses utama yaitu proses *preprocessing* dan translasi. Proses *preprocessing* memiliki tahap – tahap antara lain *case folding*, *filtering*, *tokenizing* kata, *stemming*, *removing stopword* dan proses translasi memiliki tahap – tahap antara lain deteksi kata kunci, *tokenizing* perintah, identifikasi perintah DDL, identifikasi konten, penyusunan *query*.



Gambar 1. 2 Blok Diagram Alur Sistem

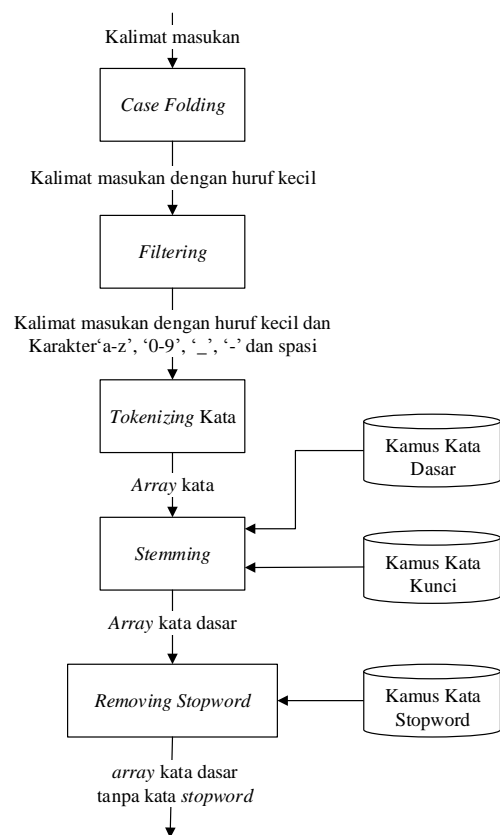
Sebelum dilakukan proses *preprocessing*, terdapat kalimat masukan yang dimasukan oleh pengguna.

Tabel 1. 2 Contoh Kalimat Masukan

Kalimat Masukan	
<i>Query Tunggal</i>	<i>Query Jamak</i>
Buatlah basisdata Unikom!	Tambahkan field pada Tabel user dengan tglahir tipe data date kemudian ubah Field 'alamat' menjadi 'kota' pada tabel mahasiswa..!

2.6 Preprocessing

Preprocessing merupakan proses awal untuk mempersiapkan data masukan yang berupa kalimat teks bahasa Indonesia sebelum memasuki proses translasi. Proses ini terdiri dari beberapa tahap yaitu *case folding*, *filtering*, *tokenizing* kata, *stemming*, dan *removing stopword*.



Gambar 1. 3 Blok Diagram Preprocessing

1. Case Folding

Case folding merupakan proses menyeragamkan setiap huruf pada kalimat masukan dengan mengubah semua huruf menjadi kapital (*uppercase*) atau kecil (*lowercase*). Pada penelitian ini proses *case folding* menyeragamkan semua huruf menjadi kecil (*lowercase*) yang berguna pada proses *stemming*, *removing stopword*, dan translasi karena pada proses tersebut ukuran huruf harus seragam.

2. Filtering

Filtering merupakan tahap untuk memilah setiap karakter pada kalimat masukan hanya karakter tertentu. Tujuan proses *filtering* yaitu

untuk mengurangi karakter yang tidak diperlukan sehingga mengurangi *noise* saat proses pengambilan informasi. Pada penelitian ini, karakter yang menjadi masukan hanya ‘a’ sampai ‘z’, ‘0’ sampai ‘9’, ‘_’, ‘,’ dan spasi.

3. Tokenizing Kata

Tokenizing kata merupakan tahap untuk memotong kalimat menjadi kata - kata dengan pemisah berupa spasi (‘ ’). *Tokenizing* kata dilakukan karena pada tahap selanjutnya akan dilakukan pemrosesan setiap kata. Hasil dari *tokenizing* kata yaitu berupa token – token kata.

4. Stemming

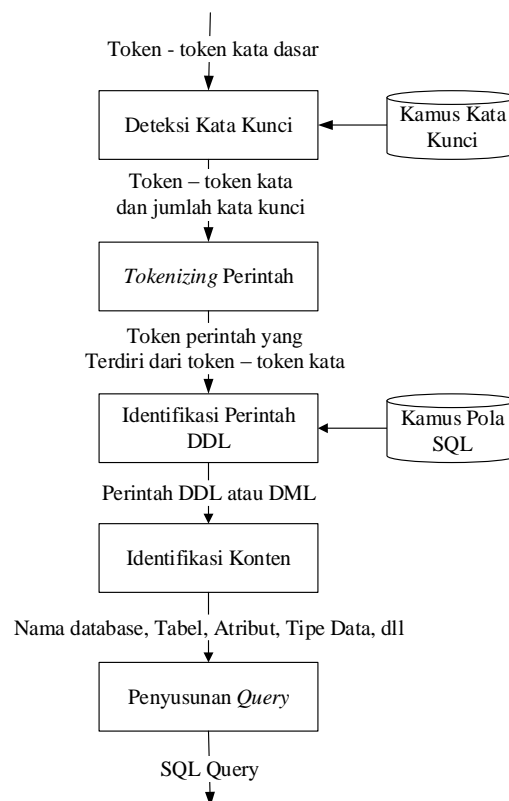
Stemming merupakan tahap untuk mengubah suatu kata yang memiliki partikel imbuhan menjadi kata dasar. Proses *stemming* dapat dilakukan menggunakan teknik *crawling* dengan membandingkan setiap token kata dengan kata dasar yang terdapat pada kamus KBBI [11]. Pada penelitian ini, proses *stemming* menggunakan *library stemmer* untuk bahasa Indonesia dari sastrawi yang telah mengadopsi Algoritma Nazief & Andriani [12]. Contoh penggunaan *stemmer* yaitu menghilangkan imbuhan ‘mem-’ pada kata ‘membuat’ sehingga menjadi kata dasar yaitu ‘buat’. Kata yang diubah menjadi kata dasar hanya kata yang terdapat pada kamus kata kunci.

5. Removing Stopword

Removing stopword merupakan tahap untuk menghilangkan kata pada kalimat masukan yang terdapat pada kamus kata *stopword*. Pada penelitian ini, penentuan *stopword* disesuaikan pada kasus yang sedang diteliti, biasanya pemilihan *stopword* tergantung pada kata yang sering muncul dalam suatu teks, contohnya kata ‘yang’, ‘dan’, ‘dengan’, ‘atau’ dan lain – lain.

2.7 Translasi

Translasi merupakan proses untuk menerjemahkan token – token kata kalimat masukan dari hasil *preprocessing* ke *query* DDL.



Gambar 1. 4 Blok Diagram Translasi

Pada penelitian ini, proses translasi memiliki 5 tahap yaitu deteksi kata kunci, *tokenizing* perintah, identifikasi perintah ddl, identifikasi konten, penyusunan *query*.

1. Deteksi Kata Kunci

Pada proses ini, token – token kata dasar yang didapatkan dari hasil *preprocessing* akan dilakukan deteksi kata kunci awal dengan pengecekan setiap token pada kamus kata kunci. Tujuan pengecekan kata kunci untuk menentukan banyaknya kata perintah yang akan menghasilkan *query*.

Tabel 1. 3 Hasil Deteksi Kata Kunci

Kalimat Masukan	Kata Kunci Terdeteksi	Posisi Token	Jumlah Kata Kunci
<i>Query Tunggal</i>	buat	1	1
<i>Query Jamak</i>	tambah	1	2
	ubah	9	

2. Tokenizing Perintah

Pada proses ini akan dilakukan proses pemisahan token – token berdasarkan data deteksi kata kunci dan posisi *array* hasil dari proses deteksi kata kunci. Tujuan proses *tokenizing* perintah yaitu untuk memisahkan setiap kalimat perintah yang utuh, sehingga setiap kalimat dapat diproses translasi.

3. Identifikasi Perintah DDL

Proses ini merupakan tahap untuk mengidentifikasi setiap token – token kata pada setiap kalimat perintah agar mendapatkan pola SQL DDL. Setiap token pada setiap kalimat perintah dibandingkan dengan kamus kata kunci untuk mendapatkan perintah DDL. Tujuan proses identifikasi DDL yaitu untuk mendapatkan *template* pola SQL pada setiap kalimat perintah. Token kata pada setiap kalimat perintah yang sudah teridentifikasi sebagai kata kunci akan dibandingkan dengan kamus pola SQL, sehingga *template* pola SQL akan teridentifikasi.

4. Identifikasi Konten

Pengecekan *template handlebar* dilakukan dengan mencari karakter pada pola SQL yang diawali dengan “{{” dan diakhiri dengan “}}”. Pada pola SQL, hasil identifikasi perintah DDL ditemukan *template handlebar* {{nama_db}}, {{nama_tabel}}, {{nama_kolom}}, {{nama_kolom_baru}}, dan {{kolom_tipe_opsi}}, maka proses ini untuk menentukan nama tabel, nama kolom, nama kolom baru, dan nama kolom yang akan dibuat dengan melakukan pencarian pada setiap token kata. Tujuan proses ini yaitu untuk menentukan isi dari semua konten *template handlebar* dari sisa token hasil identifikasi perintah DDL.

5. Penyusunan Query

Pada proses ini, penyusunan *query* dilakukan sesuai dari hasil yang didapatkan dari proses identifikasi perintah DDL serta proses identifikasi konten. Tujuan proses penyusunan *query* yaitu untuk memetakan setiap token hasil identifikasi konten ke dalam *template handlebar*.

Tabel 1. 4 Proses Penyusunan Query

Kalimat Masukan	Sebelum	Sesudah
	Pola SQL	Pola SQL
Query Tunggal	CREATE DATABASE {{nama_db}};	CREATE DATABASE E unikom;
Query Jamak	ALTER TABLE {{nama_tabel}} ADD {{kolom_tipe_opsi}};	ALTER TABLE user ADD tgl_lahir date;
	ALTER TABLE {{nama_tabel}} CHANGE {{nama_kolom}} {{nama_kolom_baru}};	ALTER TABLE mahasiswa CHANGE alamat kota;

2.8 Hasil Pengujian

Tahap pengujian akurasi pada penelitian ini dilakukan untuk mengukur tingkat keberhasilan pada proses penerjemah kalimat masukan ke query DDL

SQL. *Query* DDL yang diuji mencakup fungsi *CREATE*, *ALTER*, *RENAME*, *TRUNCATE*, dan *DROP*. Data kalimat masukan yang merupakan satu kalimat perintah dalam bahasa Indonesia yang mengandung dua *query* DDL. Pengujian dilakukan dengan menerjemahkan kalimat masukan lalu hasil *query* dari proses penerjemahan oleh sistem akan dibandingkan dengan *query* harapan yang sudah ditentukan. Salah satu contoh kalimat masukan yang dipakai dalam pengujian ada pada tabel berikut.

Tabel 1. 5 Contoh Kalimat Uji

Data Uji
Buatlah database dengan nama db_kampus dan buat tabel mahasiswa dengan field nim tipe data char (7), nama tipe data varchar (50), alamat tipe data varchar (100), tahun_msk tipe data integer

hasil bernilai benar jika *query* yang dihasilkan sistem sama dengan *query* harapan yang sudah ditentukan. Pengujian akurasi yang dilakukan mencakup 25 jenis kalimat masukan dengan 93 kalimat perintah yang merupakan kombinasi dari setiap fungsi DDL.

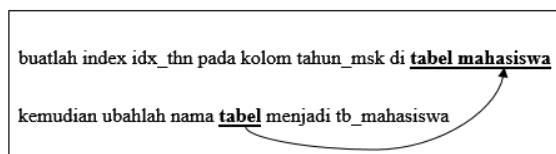
Tabel 1. 6 Hasil Pengujian Akurasi

No	Jenis Perintah	Kalimat	Hasil Klarifikasi	
			Benar	Salah
1	CREATE-CREATE	9	8	1
2	CREATE-ALTER	4	2	2
3	CREATE-RENAME	4	3	1
4	CREATE-TRUNCATE	4	2	2
5	CREATE-DROP	4	4	0
6	ALTER-CREATE	4	3	1
7	ALTER-ALTER	4	3	1
8	ALTER-RENAME	3	2	1
9	ALTER-TRUNCATE	3	2	1
10	ALTER-DROP	3	3	0
11	RENAME-CREATE	3	2	1
12	RENAME-ALTER	4	3	1
13	RENAME-RENAME	3	2	1
14	RENAME-TRUNCATE	3	1	2
15	RENAME-DROP	3	2	1
16	TRUNCATE-CREATE	3	2	1
17	TRUNCATE-ALTER	3	1	2

18	TRUNCATE- RENAME	3	2	1
19	TRUNCATE- TRUNCATE	3	3	0
20	TRUNCATE- DROP	3	2	1
21	DROP- CREATE	3	2	1
22	DROP- ALTER	5	4	1
23	DROP- RENAME	3	2	1
24	DROP- TRUNCATE	5	3	2
25	DROP- DROP	4	4	0
Total		93	67	26
Rata – Rata Akurasi = 72,04%				

Berdasarkan hasil pengujian *black box* dan akurasi pada sistem yang telah dibangun, maka kesimpulan yang diperoleh adalah sebagai berikut.

1. Fungsionalitas sistem berjalan dengan baik, karena ditandai dengan suksesnya setiap fungsi yang ada pada sistem.
2. Rata – rata akurasi yang didapat sebesar 72.04%. Nilai rata – rata akurasi yang rendah disebabkan karena sistem belum dapat menangani kalimat yang memiliki kata yang mengacu pada kata di klausa sebelumnya. Contoh kalimat dapat dilihat pada gambar.



Gambar 1. 5 Contoh Kalimat Mengandung Kata Acuan

Pada contoh di gambar 1.5, nama tabel ‘mahasiswa’ hanya disebutkan sekali, sehingga nama tabel pada klausa kedua tidak disebutkan karena kata tabel pada klausa kedua mengacu pada nama tabel di klausa pertama.

3. PENUTUP

3.1 Kesimpulan

Berdasarkan hasil implementasi dan pengujian pada penelitian yang telah dilakukan, maka dapat diperoleh informasi bahwa sistem yang dibangun dapat menerjemahkan kalimat perintah DDL dalam bahasa Indonesia yang mengandung *query* jamak dan menghasilkan *query* DDL lebih dari satu. Penelitian ini berhasil menerjemahkan kalimat perintah yang mengandung *query* DDL jamak dalam bahasa Indonesia ke *query* DDL dengan akurasi 72,04%.

3.2 Saran

Berdasarkan analisis dan pengujian yang telah dilakukan, maka terdapat saran agar penelitian ini mendapatkan hasil lebih baik diantaranya sebagai berikut.

1. Menambahkan *co-reference* dalam mengetahui kata ganti untuk kata yang mengacu pada klausa sebelumnya.
2. Menambahkan fitur koreksi kata yang salah eja (*typo-checking*).
3. Menambahkan *rule* yang dapat membaca kalimat yang menghasilkan *query* dengan format atribut *null* atau *not null*, *foreign key*, dan *primary key*.

DAFTAR PUSTAKA

- [1] Fathansyah, *Basis Data*. Bandung: Informatika, 2004.
- [2] S. Phadtare, “SQL Query Generation Using Human Language Processing,” *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 4, no. 4, hal. 696–698, 2016.
- [3] S. Kaur dan R. S. Bali, “SQL generation and execution from natural language processing,” *Int. J. Comput. Bus. Res. ISSN*, no. 1972, hal. 2229–6166, 2012.
- [4] S. Wibisono, “Aplikasi Pengolah Bahasa Alami untuk Query Basisdata Akademik dengan Format Data Xml,” *J. Teknol. Inf. Din.*, vol. 18, no. 1, hal. 65–79, 2013.
- [5] Kuspriyanto, H. Sujani, H. Tjahjono, dan S. Kusuma, “Perancangan Translator Bahasa Alami ke dalam Format SQL (Structured Query Language),” *Teknoin*, vol. 10, no. 3, hal. 225–236, 2005.
- [6] D. M. Aminuddin, “Data Definition Language (DDL) Pada Structured Query Language (SQL) Menggunakan Bahasa Indonesia,” Universitas Komputer Indonesia, 2018.
- [7] D. Anita dan M. Arhami, *Konsep Kecerdasan Buatan*. Yogyakarta: Andi Offset.
- [8] K. Miftahul dan S. Ridwan, *SINTAKSIS Memahami Satuan Kalimat Perspektif Fungsional*. Jakarta: Bumi Aksara, 2014.
- [9] Sugiyono, *Metode Penelitian Kuantitatif*. Bandung: Alfabeta, 2014.
- [10] R. S. Pressman, *Software Engineering A Practitioner’s Approach 7th Ed*. New York: McGraw-Hill, 2010.
- [11] R. H. Fauzi, “Pembangunan Kamus Bahasa Indonesia sebagai Sumber Daya Natural Language Processing,” *Pros. Semin. Nas. Komput. dan Inform.*, vol. 2017, hal. 193–200, 2017.
- [12] M. Adriani, J. Asian, B. Nazief, dan E. Williams, “Stemming Indonesian : A Confix-Stripping Approach,” *ACM Trans. Asian Lang. Inf. Process*, vol. 6, no. 4, hal. 1–33, 2007.