COREFERENCE IMPLEMENTATION ON INDONESIAN LANGUAGE TO STRUCTURED QUERY LANGUAGE (SQL) CONVERTION ON CHANGING THE STRUCTURE OF THE TABLE

Fauzan Abdulwahid[1], Alif Finandhita[2]

[1,2] Teknik Informatika –Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung 40132
E-mail : dragonknightfarawell@email.unikom.ac.id[1], alif.finandhita@email.unikom.ac.id[2]

ABSTRAK

Natural Language Processing (NLP) is a computer science that processes natural language or language that is often used by humans to be understood by computers. There have been several studies of translating natural language into SQL (Structured Query Language) various methods, then research by Iqram Anwar which can handle plural sentences using the rule based method with a final accuracy of 72.04%. The acquisition of the final value is due to the process cannot translate changes in the structure of the table if there are reference words in the sentence. Coreference Implementation On Indonesian Language To Structured Query Language (Sql) Convertion On Changing The Structure Of The Table aims to build a process that can translate standard Indonesian sentences. Using a rule based method combined with coreference logic with stages of extracting the input sentence first and then taking the SQL template from the database based on the keywords that have been obtained. The accuracy of the final test is 89.47% which shows that this method is quite effective but not optimal because it cannot handle commands that refer to the previous command. In subsequent studies, refinement of coreference or referral order retrieval methods can overcome these problems.

Keywords : Natural Language Processing, Structured Query Language, Converter, Bahasa Indonesia, Coreference.

1. INTRODUCTION

NLP (Natural Language Processing) or natural language processing allows humans to interact with computers using human language. One of the applications of NLP is translation, which translates human natural language into language that can be processed by computers[1]. There have been several studies that discuss processing natural language into SQL (Structured Query Language), both using translation method that look at the structure of the sentences[2][3][4] or method that only translate based on keywords[5][6][7].

Indonesian to SQL translation research with a translation method that only looks at the keywords, including a research by Kuspriyanto that processes query select in academic databases with 86% accuracy [5]. Research by Defy M. A that process DDL (Data Definition Language) with an accuracy of 92.08% [6]. Research by Iqram A. that process DDL and can translate plural sentences with an accuracy of 72.04% [7]. Last result in Iqram's research is decreased because the process cannot translate sentences that have reference words in them.

Based on the explanation above, it can be concluded that there is a need for the development of a translatio method that can handle plural sentences that contain reference words in them. The method used is a rule based method and without seeing the structure of the input sentence. The translation method will then be tested with a simple web application built using HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) on the frontend [8] and PHP (Hypertext Preprocessor) on its backend [9].

2. RESEARCH CONTENT

2.1 *Natural Language Processing*

NLP is a branch of scientific AI (Artificial Intelligence) which discusses the relationship between humans and computers through the everyday language of humans. Natural language processing is not easy because of differences in how the human brain and computer work when processing language.

Humans can distinguish the meaning of a word by looking at the whole sentence, punctuation, location of words, and easily learn if they find a new word. While on computer, ambiguity often occurs because word processing is limited by rules that have been made [1].

2.2 Coreference

Coreference is a solution to solve the problem of determining a reference expression or reference to an entity, person, or thing in natural language [10].

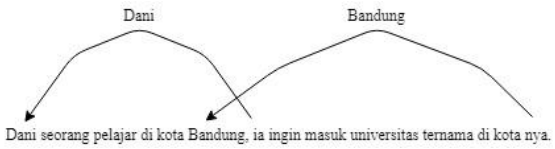

Figure 1. Coreference Example

In Figure 1, there are sentences in which there are words of people, 'Dani' and the city entity 'Bandung', then there are words that refer to the previous words, 'ia' and 'nya'. With the use of the word reference, coreference can have the meaning of people and city entities namely 'Dani' and 'Bandung'.

2.3 Research Methods

This research uses descriptive research method, which is a method that describes situations and events based on facts and appropriate interpretations [11].

The following are the steps carried out in this study.

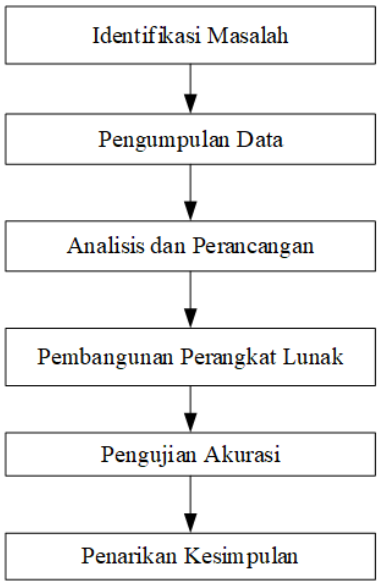

Figure 2. Research Stages

1. Problem Identification

   Based from previous research, what was done at this stage was finding and identifying problems - problems in previous research in order to focus on solving those problems.
2. Gathering Data

   The method used for gathering data is the study of literature, which is collecting data from literature, papers, journals, and books that are related and helpful to research.
3. Analysis and Design

   At this stage an analysis of the problems that have been found, for designing solutions that will solve the problem.
4. Software Development

   At this stage software development is carried out to test the design of the solution that has been obtained.
5. Accuracy Testing

   Accuracy testing using the case test method is carried out to see how much accuracy the value obtained from the solution has been determined.
6. Conclusion

   At this stage what is done is to draw conclusions from research that has been done.

2.4 Problem Analysis

Research by Iqram Anwar [7] can handle plural sentences that cannot be handled by other studies [2] [3][4][5][6], but there are deficiencies in the translation method so that it cannot properly process changes in structure table. This happens because the system cannot process words that refer to other words so adding coreference can solve the problem.

Table 1. Incorrect Query Example

Input	**Expectation**	**Result**
hapus index idx_nim pada tabel mahasiswa lalu tambah kolom umur dengan tipe data integer	DROP INDEX idx_nim ON mahasiswa;	DROP INDEX idx_nim ON mahasiswa;
	ALTER TABLE mahasiswa ADD umur integer(25);	ALTER TABLE umur ADD integer varchar(255);

In Table 1, it appears that the second command did not match expectations because the table name was not mentioned. If it is assumed that the table name is an entity, then the problem can be overcome if coreference is added so the process will take the table name from the first command [10].

2.5 System Overview

Basically, this research is a development of two previous studies [6] [7],

So there are two main stages, namely the Preprocessing and Translation stages.

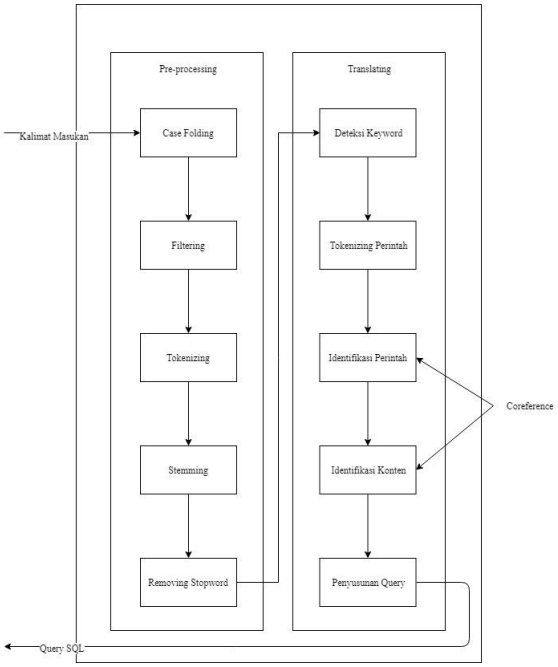

Figure 3. System Overview

The processes carried out at the Preprocessing stage including:

1. Case Folding

Uniforming of characters is done at this stage, therefore the input sentences are lowercase.

Table 2. Tabel Case Folding

Before	After
Hapuslah index idx_nim pada tabel mahasiswa lalu tambah kolom umur dengan tipe data integer!	**h**apuslah index idx_nim pada tabel mahasiswa lalu tambah kolom umur dengan tipe data integer!

2. Filtering

At this stage, filtering of unneeded characters, leaving only certain characters.

Table 3. Tabel Filtering

Before	After
hapuslah index idx_nim pada tabel mahasiswa lalu tambah kolom umur dengan tipe data integer**!**	hapuslah index idx_nim pada tabel mahasiswa lalu tambah kolom umur dengan tipe data integer

3. Tokenizing

At this stage the input sentences that have been processed will be separated word by word.

Table 4. Tabel *Tokenizing*

Before	After	
	Index	**Token**
hapuslah index idx_nim pada tabel mahasiswa lalu tambah kolom umur dengan tipe data integer	1	hapuslah
	2	index
	3	idx_nim
	4	pada
	5	tabel
	6	mahasiswa
	7	lalu
	8	tambah
	9	kolom
	10	umur
	11	dengan
	12	tipe
	13	data
	14	integer

4. Stemming

At this stage each word token is returned to its basic form by removing the prefixes and postfixes.

Table 5. Tabel Stemming

Index	Before	After
1	hapus**lah**	hapus
2	index	index
3	idx_nim	idx_nim
4	pada	pada
5	tabel	tabel
6	mahasiswa	mahasiswa
7	lalu	lalu
8	tambah	tambah
9	kolom	kolom
10	umur	umur
11	dengan	dengan
12	tipe	tipe
13	data	data
14	integer	integer

5. Removing Stopword

At this stage, filtering of words that are not needed is done, leaving only certain characters.

Table 6. Tabel Removing Stopword

Index	Before	After	Result	
			Index	**Token**
1	hapus	hapus	1	hapus
2	index	index	2	index
3	idx_nim	idx_nim	3	idx_nim
4	pada		4	tabel
5	tabel	tabel	5	mahasiswa
6	mahasiswa	mahasiswa	6	tambah
7	lalu		7	kolom
8	tambah	tambah	8	umur
9	kolom	kolom	9	tipe
10	umur	umur	10	data

11	dengan		11	integer
12	tipe	tipe		
13	data	data		
14	integer	integer		

After the Preprocessing stage is carried out, the next stage is Translations which include:

1. Keyword Detection

At this stage each token will be compared with a keyword database. This stage is to determine a single or plural command.

Table 7. Tabel Keyword Detection

Index	Token	Keyword?
1	**hapus**	Ya
2	index	Bukan
3	idx_nim	Bukan
4	tabel	Bukan
5	mahasiswa	Bukan
6	**tambah**	Ya
7	kolom	Bukan
8	umur	Bukan
9	tipe	Bukan
10	data	Bukan
11	integer	Bukan

2. Commands Tokenizing

At this stage tokens that have been identified as keywords will be separated so that the command sentence can be separated.

Table 8. Tabel Commands Tokenizing

Before		After		
Index	**Token**	**Commands**	**Index**	**Token**
1	**hapus**	1	1	hapus
2	index		2	index
3	idx_nim		3	idx_nim
4	tabel		4	tabel
5	mahasiswa		5	mahasiswa
6	**tambah**	2	1	tambah
7	kolom		2	kolom
8	umur		3	umur
9	tipe		4	tipe
10	data		5	data
11	integer		6	integer

3. Commands Identification

At this stage, each token is identified by comparing it with the initial keyword dictionary and the derived keyword so that it gets the structure of the SQL sentence, if the initial keyword and the derived keyword do not produce an SQL template, it will be compared with the second derived keyword to get its SQL sentence structure.

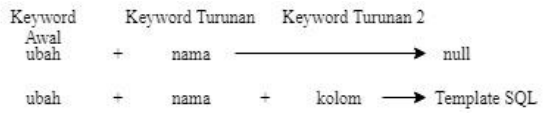

Figure 4. Obtaining SQL Template

Table 9. Tabel Commands Identification

Commands	Index	Token	Keyword	Template
1	1	hapus	hapus + index	ALTER TABLE {{nama_tabel}} DROP {{nama_index}};
	2	index		
	3	idx_nim		
	4	tabel		
	5	mahasiswa		
2	1	tambah	tambah + kolom	ALTER TABLE {{nama_tabel}} ADD {{kolom_tipedata}};
	2	kolom		
	3	umur		
	4	tipe		
	5	data		
	6	integer		

After getting the SQL template, the keyword token is deleted, leaving unused tokens.

Table 10. Tabel Unused Tokens

Commands	Index	Token
1	1	idx_nim
	2	tabel
	3	mahasiswa
2	1	umur
	2	tipe
	3	data
	4	integer

4. Contents Identification

At this stage the remaining tokens are checked whether the contents of the table to be processed are like the table name and column names of the remaining tokens, this check uses the handlebar template of the template that has been obtained. At this stage that the reference word or coreference is processed.

In the case of SQL queries, most of the reference words are used when selecting the table to be used, so that the coreference processing can be done at the content identification stage, where the word 'tabel' token that is considered as entity [10] will be searched in the command token array, and if there isn't 'table' word then the word search will be carried out to the previous command.

Figure 5. Obtaining Template

In the example above, there is no ‘tabel’ word in the second command sentence so the process will take the name of the table from the first sentence by breaking down the first command sentence that has been arranged then looking for the word 'tabel' in it.

There are three types of handle bar templates that are identified namely {{nama_tabel}}, {{kolom_tipedata}} and {{nama_index}}.

- Template Handle Bar {{nama_tabel}}

In this handle bar template process, what to do is look for the word 'tabel', if the word 'tabel' is found then the name of the table is in the index of the word 'table' + 1. If the word 'tabel' is not found then the name of the table is taken from the word table in the previous sentence or template. If the word 'tabel' is not found at all then the name of the table is at index 1. After the table name is specified, the ‘tabel’ token and the table name token are deleted.

In the example above, the word 'tabel' is found in the 1st index in the first command but not found in the 2nd command, so the table name is in the 2nd index for the first command and the table name for the second command will follow the table name in first command. After that, delete the tokens.

Table 11. Tabel *Handle Bar* {{nama_tabel}}

Commands	Token	Template Handle Bar	Content
1	idx_nim	{{nama_tabel}}	mahasiswa
		{{nama_index}}	
2	umur	{{nama_tabel}}	mahasiswa
	tipe		
	data	{{kolom_tipedata}}	
	integer		

- Template Handle Bar {{nama_index}}

In this handle bar template process all you have to do is look for the word 'indeks', if the word is found then the index name is in the index word 'indeks' + 1. If the word is not found then the index name is in the 1st index.

In the example above, the handle bar is only in the first command and the 'indeks' token is not found in the command, so the index name is in the 1st index for the first command. After that, delete the tokens.

Table 12. Tabel *Handle Bar* {{nama_index}}

Commands	Token	Template Handle Bar	Content
1	kosong	{{nama_tabel}}	mahasiswa
		{{nama_index}}	idx_nim
2	umur	{{nama_tabel}}	mahasiswa
	tipe		
	data		
	integer	{{kolom_tipedata}}	

- TemplateHandle Bar {{kolom_tipedata}}

In this handle bar template process, what you have to do is look for the word 'kolom' or 'field', if the word is found then the column name is in the index of the word 'kolom’ + 1. If the word is not found then the column name is in the index- 1 The token after the column name is a typedata token, if there is a ‘type’ or ‘data’ token then the token is deleted. After the column name and data type are determined, the next token is the size token, after the size token there are several other tokens such as null status, and column position. Specifically for column name token, data type, and size, the tokens must have valid values, otherwise the output will produce an error input sentence detail. After the column name and constraint has been determined the word ‘kolom’ and the tokens afterwards are deleted.

In the example above, the handle bar is only in the second command and the word 'kolom' token is not found in the command, so the column name is in the 1st index for the first command and the other constraints in the index afterwards. After that, delete the word tokens.

Table 13. Tabel *Handle Bar* {{kolom_tipedata}}

Commands	Token	Template Handle Bar	Content
1	kosong	{{nama_tabel}}	mahasiswa
		{{nama_index}}	idx_nim
2	kosong	{{nama_tabel}}	mahasiswa
		{{kolom_tipedata}}	umur, integer

5. Query Building

At this stage, building the query is based on command tokens and content tokens that have been obtained from previous processes. This process is done by replacing the Handle Bar Template with the content that has been obtained.

Table 14. Tabel Query Building

Commands	Before	After
1	ALTER TABLE {{nama_tabel}} DROP {{nama_index}};	ALTER TABLE mahasiswa DROP idx_nim;

2	ALTER TABLE {{nama_tabel}} ADD {{kolom_tipedata}};	ALTER TABLE mahasiswa ADD umur integer (Size invalid!);

At the second command output there is the word 'Size invalid!' Because the rules in the {{kolom_tipedata}} handlebar that require a minimum of column names, data types, and sizes.

2.6 Test Result

In this chapter, testing the accuracy of the translation method that has been developed. Input sentences are plural sentences with or without reference words. The command sentence will be considered correct if the query generated is same with expectations. Accuracy testing is performed on 45 types of non-repetitive ALTER sentences, and each type of sentence is given at least 2 types of input sentences namely true and false sentences.

No	Jenis Perintah	Jumlah Kalimat	Hasil Klasifikasi	
			Benar	Salah
1	ADD – ADD	3	2	1
2	ADD – DROP	2	2	0
3	ADD – CHANGE	2	1	1
4	ADD – ADD PRIMARY	3	1	2
5	ADD – DROP PRIMARY	3	2	1
6	ADD – ADD FOREIGN	3	2	1
7	ADD – DROP FOREIGN	2	2	0
8	ADD – MODIFY	2	2	0
9	ADD – RENAME	4	2	2
10	DROP – DROP	2	2	0
11	DROP – CHANGE	2	2	0
12	DROP – ADD PRIMARY	2	2	0
13	DROP – DROP PRIMARY	2	2	0
14	DROP – ADD FOREIGN	2	2	0
15	DROP – DROP FOREIGN	2	2	0
16	DROP – MODIFY	2	2	0
17	DROP – RENAME	2	2	0
18	CHANGE – CHANGE	2	2	0
19	CHANGE – ADD PRIMARY	2	1	1
20	CHANGE – DROP PRIMARY	2	2	0
21	CHANGE – ADD FOREIGN	2	1	1
22	CHANGE – DROP FOREIGN	2	2	0
23	CHANGE – MODIFY	2	2	0
24	CHANGE – RENAME	2	2	0
25	ADD PRIMARY – ADD PRIMARY	2	1	1
26	ADD PRIMARY – DROP PRIMARY	2	2	0
27	ADD PRIMARY – ADD FOREIGN	2	2	0
28	ADD PRIMARY – DROP FOREIGN	2	2	0
29	ADD PRIMARY – MODIFY	2	2	0
30	ADD PRIMARY – RENAME	2	2	0
31	DROP PRIMARY – DROP PRIMARY	2	2	0
32	DROP PRIMARY – ADD FOREIGN	2	2	0
33	DROP PRIMARY – DROP FOREIGN	2	2	0
34	DROP PRIMARY – MODIFY	2	2	0
35	DROP PRIMARY – RENAME	2	2	0
36	ADD FOREIGN – ADD FOREIGN	2	2	0
37	ADD FOREIGN – DROP FOREIGN	2	2	0
38	ADD FOREIGN – MODIFY	2	2	0
39	ADD FOREIGN – RENAME	2	2	0

40	DROP FOREIGN – DROP FOREIGN	2	2	0
41	DROP FOREIGN – MODIFY	2	2	0
42	DROP FOREIGN – RENAME	2	2	0
43	MODIFY – MODIFY	2	2	0
44	MODIFY – RENAME	2	2	0
45	RENAME – RENAME	2	2	0
Total		96	85	11

The following is a calculation for the overall accuracy of the input data.

$$\frac{\sum Score}{n} x\ 100\% = \frac{85}{96} x\ 100\% = 88.54\% \qquad (1)$$

3. CLOSING

3.1 Conclusion

Based on the results of tests that have been done, the system can translate plural sentences that contain words that refer to other sentence. The final value of testing is 88.54%. These results are not optimal because the system has not been able to process word that refer to other command words.

3.2 Suggestion

Based on the analysis of accuracy testing, the cause of translation results is not optimal because the coreference logic is not perfect. Handling referral command words can solve this problem.

REFERENCES

[1] D. Anita dan M. Arhami, Konsep Kecerdasan Buatan. Yogyakarta: Andi, 2006.

[2] Saravjeet Kaur, SQL *Generation And Execution From Natural Language Processing*. MMU. Mullana.

[3] Setyawan Wibisono, Aplikasi Pengolahan Bahasa Alami untuk Query Basisdata Akademik dengan Format Data XML, Skripsi S1. Teknik Informatika. Universitas Stikubank, 2013.

[4] Andri, Penerapan Bahasa Alami Sederhana Pada *Online Public Access Catalog*(OPAC) Berbasis Web Semantik. Sistem Informasi. Universitas Binadarma Palembang, 2012.

[5] Kuspriyanto, Perancangan Translator Bahasa Alami Ke Dalam Format SQL (*Structured Query Language*). Departemen Teknik Elektro. Institut Teknologi Bandung.

[6] Defy M. Aminuddin, *Data Definition Language* (DDL) Pada *Structured Query Language* (SQL) Menggunakan Bahasa Indonesia, Skripsi S1. Teknik Informatika. Universitas Komputer Indonesia.

[7] Iqram Anwar, Penanganan Teks Bahasa Indonesia Menjadi *Data Definition Language* (DDL) Dengan Penanganan Kalimat Majemuk, Skripsi S1. Teknik Informatika. Universitas Komputer Indonesia.

[8] J. Enterprice, Pengenalan HTML dan CSS, Jakarta: PT Elex Media Komputindo, 2016.

[9] Ramus Lerdorf, Kevin Tatroe, *Programing PHP*, United States of America: O'Reilly Media, 2002.

[10] Maciej Ogrodniczuk, "*Coreference*". Walter de Gruyter GmbH & Co KG.

[11] I. Afrianto, A. Heryandi, A. Finandhita, Sufa'atin, *E-Document Autentification With Digital Signature Fro Smart City : Reference Model. Informatic Engineering Department, Faculty of Engineering and Computer Science.* Universitas Komputer Indonesia.