2016 7th International Conference on Intelligent Systems, Modelling and Simulation

Chatbot Using A Knowledge in Database

Human-to-Machine Conversation Modeling

Bayu Setiaji
Department of Informatics Engineering
STMIK AMIKOM Yogyakarta
Yogyakarta, Indonesia
e-mail: bayusetiaji@amikom.ac.id

Ferry Wahyu Wibowo
Department of Informatics Engineering
STMIK AMIKOM Yogyakarta
Yogyakarta, Indonesia
e-mail: ferry.w.wibowo@ieee.org

***Abstract* - A chatterbot or chatbot aims to make a conversation between both human and machine. The machine has been embedded knowledge to identify the sentences and making a decision itself as response to answer a question. The response principle is matching the input sentence from user. From input sentence, it will be scored to get the similarity of sentences, the higher score obtained the more similar of reference sentences. The sentence similarity calculation in this paper using bigram which divides input sentence as two letters of input sentence. The knowledge of chatbot are stored in the database. The chatbot consists of core and interface that is accessing that core in relational database management systems (RDBMS). The database has been employed as knowledge storage and interpreter has been employed as stored programs of function and procedure sets for pattern-matching requirement. The interface is standalone which has been built using programing language of Pascal and Java.**

Keywords - bigram; chatbot; database; sentence; similarity

I. INTRODUCTION

The development of the information technology and communication has been complex in implementing of artificial intelligent systems. The systems are approaching of human activities such as decision support systems, robotics, natural language processing, expert systems, etc. Even in the artificial intelligent fields, there are some hybrid methods and adaptive methods those make more complex methods. Not only that, but nowadays there is also a hybrid of natural language and intelligent systems those could understand human natural language. These systems can learn themselves and renew their knowledge by reading all electronics articles those has been existed on the internet. Human as user can ask to the systems like usually did to other human. These systems are often known as internet answering-engines.

In addition the internet answering-engines, currently in the internet also begins many applications of chatter-boot or known as chatbot which is often aimed for such purposes or just entertainment [1]. This application work is very simpler because the knowledge already programmed in advance [2]. One of methods used in this application is to match the pattern (pattern-matching) [3]. The chatbot would match the input sentence from the speaker or user with pattern that has existed on the knowledge. Each pattern paired with the knowledge of chatbot which taken from various sources. The input sentence prepared as the materials of chat pattern [4]. The chat patterns modeled in the pattern-template stored in a relational database management system (RDBMS) tables. The process of pattern matching is using a sentence similarity measurement scores. The calculation method to achieve the scores of sentence-similarity measurement may apply bigram method as one way of measurement methods, although there are some other methods. The function programs for pattern matching and other support purposes written as program stored in the RDBMS. Other knowledge storage method of chatbot is artificial intelligence markup language (AIML) [5,6]. The AIML has modularly knowledge processes. This system is a web service-based which could be accessed by client. The chat patterns are language knowledge in the format of AIML stored in the database. This system could be added a specific knowledge modules [7,8].

In this paper shows the collected facts as prepared references for chat-pattern and this chat uses Indonesian language. The chat used in this project is commonly Indonesian conversational pattern and the RDBMS used in this project is MySQL. When connecting chat application to the database, it can miss in defining a sentence and how to response it. So knowledge representation in the database tables and implementation of structured query language (SQL) in the pattern-matching operation are very needed. A data those have been modeled on the pattern of the conversation would be tested using a series of scenarios. The results of conversation with the chatbot would be crosschecked back to the basic pattern. This is done to add some knowledge to the database because it hasn't been modeled before. So if the input sentences don't match in the database then it will be remodeled.

II. RELATED WORKS

A natural language processing (NLP) gives capability of computer allows communication to happen between user-to-computer or human-to-machine and computer-to-computer or machine-to-machine using human natural languages. There are three analyses to understand natural language i.e. parsing, semantic interpretation, and knowledge-based structures. The parsing is an analysis of sentence syntax structures. In this step, identification of main linguistic

2166-0670/16 $31.00 © 2016 IEEE
DOI 10.1109/ISMS.2016.53

relations is done to parse into subject, predicate, and object of the sentences. The semantic interpretation step yields meaning representation of the texts. The semantic interpretation uses knowledge of word meaning and linguistic structure such as noun or verb transitivity. Actually, processing focus in NLP is a sentence. A sentence could be meant as biggest syntaxes consist of two or more words. A structural relationship between inter-word and inter-sentence is a different. Between both sentence and word, there are two media syntax units i.e. clause and phrase. The clause is a syntax unit that consists of two or more predicate elements. The predicate elements are a subject, predicate, object, complement, and adverb. The phrase is a syntax that consists of two or more words which is not including predicate elements.

Word categories in Indonesia language are verb (V), adjective (Adj), adverb (Adv), noun (N), preposition (Prep), and conjunctive (Conj). According to the phrase, the phrase in Indonesian language divided into main elements. The phrase categories in Indonesian language are Phrase of Noun (PhN), Phrase of Adjective (PhAdj), Phrase of Adverb (PhAdv), and Phrase of Prepositional (PhPrep). A kinds of word grouped in such category could have syntax function and difference semantic role in such sentence. A word can be functioned as subject, predicate, object, complement, and adverb. Relation among form, category, and function is more clearly with this Indonesian language sentence "induk burung sedang menangkap ikan segar untuk anaknya", the description of this sentence can be shown in Table I.

TABLE I. RELATION AMONG FORM, CATEGORY, AND FUNCTION OF SENTENCE ELEMENTS

Form	Word	Phrase	Function
Induk	N	PhN	Subject
Burung	Pron		
Sedang	Adv	PhV	Predicate
Menangkap	V		
Ikan	N	PhN	Object
Segar	Adj		
Untuk	Prep	PhPrep	Complement
Anaknya	N		

The application of chatbot core that has been made is applying database in the testing purposes. The application of database development tool for this project employed MySQL Workbench 6 and tools for generating application employed programming language of Pascal, Java, and PHP. In the chatbot, if the pattern is match, it will give suite template as a response to the user. The pattern-template has been employed as a main knowledge of the chatbot stored in database. The use of MySQL database in the chatbot is only limited to store the knowledge. All codes for requiring a pattern-matching written in programming language, so to make a service in other languages, the codes need to be rewritten.

The pattern-template storage as knowledge in the relational database management system (RDBMS) or called as database allows the use of structured query language (SQL) to handle the process of pattern-matching. The RDBMS already available in many built-in functions or procedures and can be made user-defined stored program which can be called using SQL. This allows many programming languages can be implemented easily as query of the database to send and receive an input response.

A. Bigram

The probability of calculating sentence could be represented mathematically as equation (1) [9].

$$\rho\left(\left|\psi_i\right|\right) \tag{1}$$

if ψ = "saya makan nasi", a sentence in Indonesian language which means "I eat rice", so it can be written as $\rho(|\psi|=3, \psi_1=$"saya", $\psi_2=$"makan", $\psi_3=$"nasi") = $\rho(\psi_1=$"saya"$|\psi_0=$"<s>") * $\rho(\psi_2=$"makan"$|\psi_0=$"<s>", $\psi_1=$"saya") * $\rho(\psi_3=$"nasi"$|\psi_0=$"<s>", $\psi_1=$"saya", $\psi_2=$"makan") * $\rho(\psi_4=$"</s>"$|\psi_0=$"<s>", $\psi_1=$"saya", $\psi_2=$"makan", $\psi_3=$"nasi"). The sentence is started with <s> and ended with </s>. The probability of ψ_0=<s> is 1 or written as $\rho(\psi_0=$<s>$)=1$. From this case, it can also be written as equation (2).

$$\rho(\psi) = \prod_{i=1}^{|\psi|+1} \rho\left(\psi_i \middle| \psi_0 \ldots \psi_{i-1}\right) \tag{2}$$

For bigram model, it adds one word of context that can be represented as equation (3).

$$\rho\left(\psi_i \middle| \psi_0 \ldots \psi_{i-1}\right) \approx \rho\left(\psi_i \middle| \psi_{i-1}\right) \tag{3}$$

So, the equation 3 can be enacted a linear interpolation using Witten-Bell smoothing algorithm. The Witten-Bell smoothing algorithm is applied to predict the probability of bigram model with zero count or $\rho(\psi_{i-1}|\psi_i)=0$ [10]. From Good-Turing estimation, the total mass of counts with a zero count in distribution is the number of things with one count. In this case the probability mass in the back-off distribution should be $\xi(\psi_{i-1})/\varsigma(\psi_{i-1})$. Where $\xi(\psi_{i-1})$ is a number of a unique words after ψ_{i-1} and $\varsigma(\psi_{i-1})$ is a count of ψ_{i-1}. So the probability for bigram is written as equation (4).

$$\rho\left(\psi_i \middle| \psi_{i-1}\right) = \omega \rho_{MLE}\left(\psi_i \middle| \psi_{i-1}\right) + \left(1-\omega\right)\rho\left(\psi_i\right) \tag{4}$$

which $\rho_{MLE}(\psi_i|\psi_{i-1})$ is the probability of maximum likelihood estimation which can be written as equation (5).

$$\rho_{MLE}\left(\psi_i \middle| \psi_{i-1}\right) = \frac{\varsigma\left(\psi_{i-1}\psi_i\right)}{\varsigma\left(\psi_{i-1}\right)} \tag{5}$$

while $\rho(\psi_i)$ is the estimated probability which can be written as equation (6).

$$\rho\left(\psi_i\right) = \omega \rho_{MLE}\left(\psi_i\right) + \frac{\left(1-\omega\right)}{\nu} \tag{6}$$

In this case the symbol of v is the vocabulary size. Applying the value of ω is taken to make easier calculation and ω is a value set which can be written as equation (7).

$$\omega = 1 - \frac{\xi(\psi_{i-1})}{\xi(\psi_{i-1}) + \varsigma(\psi_{i-1})} \tag{7}$$

B. *Sentence Similarity Measurement*

Semantic similarity is giving score for semantic relation between two sentences or strings. So, if there are two sentences or strings, from measuring it can be determined the similar of two sentences or strings. The higher score of the sentence semantic similarity, the more similar meaning of two sentences. The score of the sentence semantic similarity is from 0 until 1. The equation of sentence similarity represented by equation (8).

$$\frac{\varsigma(s_1 \in s_2) \cup \varsigma(s_2 \in s_1)}{\varsigma(s_1) \cup \varsigma(s_2)} \tag{8}$$

The symbol of ς is count of the sentence or string that is symbolized as s. The used method to compute the semantic similarity between two sentences could be written as bellow,

a. Each sentence is divided into a list of tokens
b. The divided sentence can implement bigram set [11]
c. After getting the divided sentence, it is counted and applied the equation (8)

For example this method can be written as bellow,

s_1 = "burung"
s_2 = "burrungg"

so s1 and s2 can be divided into bigram set as

s_1 = {"bu", "ur", "ru", "un", "ng"} ≈ 5
s_2 = {"bu", "ur", "rr", "ru", "un", "ng", "gg"} ≈ 7
$s_1 \in s_2$ = {"bu", "ur", "ru", "un", "ng"} ≈ 5
$s_2 \in s_1$ = {"bu", "ur", "ru", "un", "ng"} ≈ 5

Thus the similarity score for these sentences is obtained as follow,

$$\frac{5 \cup 5}{5 \cup 7} \approx \frac{10}{12} \approx 0{,}83333$$

C. *Database*

A relational database management system (RDBMS) is a program set that is employed to define, manage, and process a database. The database is a structure that is built to be functioned as data storage. MySQL is a server database or RDBMS software that can manage the database and can store data in many numbers. It can be accessed by multi-user and can do multi-threaded.

III. METHODOLOGY

There are some methods applied to the pattern similarity process. A sentence-similarity measurement scores which employed is to obtain similarity level between both input and pattern. This process is done in the RDBMS. Before entering design process, it needs to know global architecture of the chatbot. The scheme of the chatbot design shown in Figure 1.

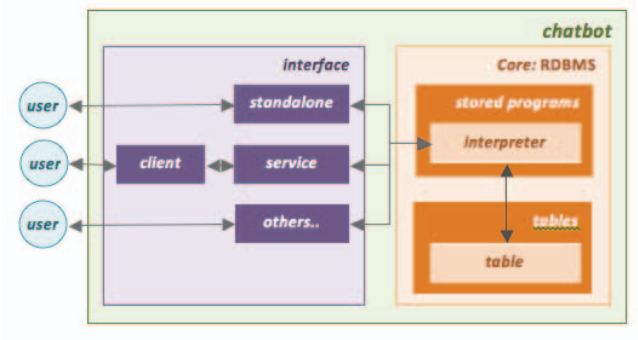

Figure 1. Global Design of Chatbot

The chatbot consists of core and interface accessing that core. The core is in RDBMS being database. The database consists of tables to store knowledge, while the interpreter is a stored program of function and procedure sets for requiring of pattern matching. The interface could be a standalone application that can be employed by user for chatting or conversation. It can also be employed by service that needs additional client application to converse with the user. This application in the interface side can be expanded more over as user needs it and can also be written using other programming languages.

The development of chatbot core that is in the RDBMS covers knowledge storage and pattern-matching process. The boundaries of chatbot in this method have some requirements:

1. Chatbot should be able to differ each conversation session that is running, so it has to store data due to the conversation session such as session of identity (sessionid), user name on the session. The sessionid must always be sent together with user input by application in the interface side along conversation process
2. Chatbot must store knowledge in the pattern-template form
3. All user inputs must be free of misspellings, punctuation, and must be in lower case, so to anticipate these cases the chatbot should be able to do normalization of the input that doesn't fit
4. For the purposes of misspellings correction, the chatbot should have a list of misspelled words and the correction stored in the tables of database
5. The chatbot should be able to pick up the keywords from the user input, so the chatbot should have a list of keywords which is stored in the tables of database
6. The chatbot should be able to do a search template using a sentence-similarity measurement scores between both pattern and input. The searching of pattern is narrowed based on the result identification as described at point 4

7. The chatbot should keep a conversation log containing sessionid, time, input, and response.

Some points those have been identified above provide a global description scheme of the chatbot core related with conversation processing that is shown in Figure 2.

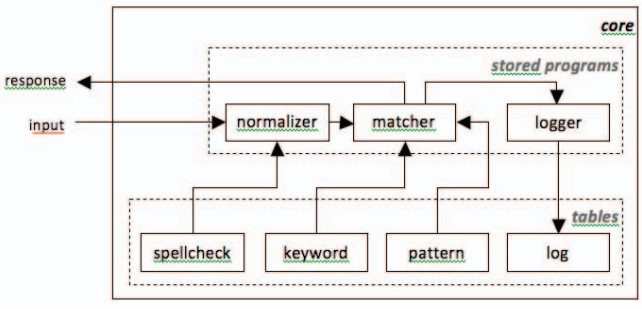

Figure 2. Chatbot Core Scheme

Based on Figure 2, the chatbot core consists of tables and stored programs.

A. Tables

The table blocks consist of main tables as scheme and supporting tables i.e. spellcheck, keyword, pattern, and log. The spellcheck stores list of misspelling words from user input and correction. The keyword stores list of keywords that is probably found in the user input. The keywords are used to narrow the range of pattern searching. The pattern stores pattern-template pairing and roles as main knowledge. Each pattern could be paired with one or more templates and each template could be paired with one or more patterns, so it would be divided into three tables i.e. pattern, template, and pattern-template. The pattern is functioned to store pattern containing patternid and pattern, template is functioned to store template containing templateid and template, and the pattern-template roles as table connecting pattern and template tables. The convlog stores conversation history containing sessionid, time, user input, and response given by user. The session stores attributes of such conversation session including sessionid, user name, and other attributes. The array is a temporary table outside the main tables. It is used as array data structure representation. This table is only supporting for internal operation requirements of array operation function. And tid is used as supporting table for id generator process that is applied in other requiring tables. These tables contain id as id name, counter as id counter, and rtable as table name reference.

B. Stored Programs

The stored program is containing stored procedures and functions for pattern-matching requirements. The normalizer is a function to norm user input that corrects spellings, eliminates punctuations, and changes into lower case. The matcher is a main function in pattern matching to find appropriate template based-on the sentence-similarity measurement scores between both input and pattern. Before doing pattern-matching process, it needs to take keywords on the input, so it is just pattern with same keywords that will be matched. The logger is a procedure functioned as conversation history storing for user input and response of question. The array is a set of functions and procedures for array tables used for the internal operations of 3 major processes that have been described previously. The array of functions and procedures includes array_intersect() which is a procedure for operating the intersection of two arrays, array_push() which is a procedure for operating the push element to the array, array_pop() which is a function for pop operation element of the array, array_count() which is a function to count the number of elements in the array, array_clear() which is a procedure to clear the contents of the array, and bigram which is a procedure for preparing bigram of a string.

C. Design

Based on the problem identification, the database that is built consist of some tables, process input of normalization and pattern matching with other supporting tables like spellcheck, keywords, pattern, template, pattern_template, convlog, session, array, and tid tables. The inter-table relationship shown in Figure 3.

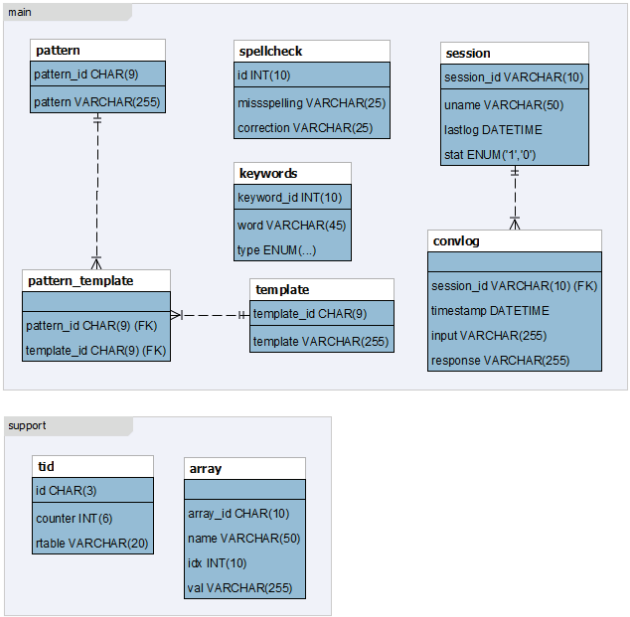

Figure 3. Entity-Relationship Diagram of Chatbot

Figure 3 shows that the tables of pattern, template, pattern_template, spellcheck, keywords, session, and convlog are grouped as main tables. Meanwhile other tables are grouped as supporting tables.

IV. RESULTS

In making a table of database for chatbot, it had implemented a forward-engineering technique. This technique is generating Entity Relationship (ER) into DDL scripts those could be executed as table generating. All designs of tables and stored programs had been implemented

in the RDBMS MySQL. Before testing process was done, it ought to be entered some knowledge which input sentence patterns stored in the pattern table and response sentences stored in the template table. In additional it had to be entered mapping as representation of relationship between both pattern and template stored in the pattern_template table.

Some tests had been applied to know the functionality of the application. A modular test was done to check and analyze the functionality of the stored program. This test was done with executing each stored program using various parameters as an inputs and analyzing an outputs. A normalizer consists of some stored programs those have been implemented to process input normalization. A function of remmarks() is employed to omit punctuation of dots, commas, semicolons, colons, exclamation points, and question marks. Testing of function of remmarks() is shown in Table II.

TABLE II. RESULT OF FUNCTION TESTING OF REMMARKS()

No	Input	Result
1	Halo, apa kabar	Halo apa kabar
2	Sekarang hari apa?	Sekarang hari apa
3	Sekarang hari Minggu.	Sekarang hari Minggu
4	Senin; Selasa; Rabu	Senin Selasa Rabu
5	Tanda Seru!!	Tanda Seru
6	Next: continue	Next continue
7	Under_score	Under_score
8	Stripped-line	Stripped-line

According to the Table II shows that testing results of number 7 and 8 yield same strings as input strings because the function of underscore (_) and stripped line (-) hasn't been included in the function of remmarks(). Thus the function of remxspaces() has been employed to eliminate more spaces. Testing of function of remxspaces() is shown in Table III.

TABLE III. RESULT OF FUNCTION TESTING OF REMXSPACES()

No	Input	Result
1	Halo, apa kabar	Halo,apa kabar
2	Halo	Halo

According to the Table III, the function of remxspaces() has been successful to eliminate spaces which are more than one including tab as it is shown in number 1. While the function of spellcorrection() is employed to fix spelling in writing. The correction of spelling depends on the number of data that has been stored in the table of missspelling. Testing of the spellcorrection() function is shown in Table IV.

TABLE IV. RESULT OF FUNCTION TESTING OF SPELLCORRECTION()

No	Input	Result
1	Halo pa kabar	Halo apa kabar
2	Kamu sdh tau blm	Kamu sudah tahu belum
3	Brp jml bintang di langit	Berapa jml bintang di langit
4	Aq ga tau	Aku ga tahu

According to the Table IV, the function of spellcorrection() could be employed to fix spelling except for "jml" as stated in the number 3 because it hasn't been stored in the table of missspelling. And the function of normalize() has been employed to call function of remmarks(), remxspaces(), and spellcorrection(). These functions are used in the matcher. Testing of function of normalize() is shown in Table V.

TABLE V. RESULT OF FUNCTION TESTING OF SPELLCORRECTION()

No	Input	Result
1	Hai…. pa kbr?	Hai apa kabar
2	Hitung! Brapa: 2+3	Hitung Berapa 2+3
3	Jarak Jauh sekali	Jarak jauh sekali

The matcher consists of stored programs for requiring a pattern-matching process. The function of getkeyword() is employed to get the keywords in the input. The keywords identification in the input depends on the keywords list stored in the table of keywords. The output of this function is a string which is applied as pattern in regular process of the function of gettemplate(). Testing of the function of getkeyword() is shown in Table VI.

TABLE VI. RESULT OF FUNCTION TESTING OF GETKEYWORD()

No	Input	Result
1	Apa kabar kamu	[[:<:]]apa[[:>:]]
2	Sekarang hari apa	[[:<:]]apa[[:>:]]
3	Benarkah 1+1=2	.
4	Di mana ibukota Indonesia	.

According to the Table VI, the testing results of number 1 and 2 yield a regular expression patterns. While testing results of number 3 and 4 yield dot (.) because the keywords weren't found in the pattern table. The function of similar() is employed to count similarity scores of two strings. This function has 2 parameters of string to be compared with the similarity scores. Testing of function of similar() is shown in Table VII.

TABLE VII. RESULT OF FUNCTION TESTING OF SIMILAR()

No	String Input(s)		Score
	1	2	
1	siapa namamu	siapa namamu	1
2	siapa namamu	namamu siapa	0,81818
3	siapa namamu	siapa nama kamu	0,92000
4	siapa namamu	Test	0

According to the Table VII, testing number 1 yields score of 1 because the string inputs of 1 and 2 have same sentences. The string input of 1 is the reference and the string input of 2 is the variation. The function of gettemplate() is employed to get the response from the input. The template that becomes a response can differ from one to another when it results responses because the response has been set randomly. Testing of the function of gettemplate() is shown in Table VIII.

TABLE VIII. RESULT OF FUNCTION TESTING OF GETTEMPLATE()

No	Input	Result
1	Apa kabar?	Kabar baik
2	Apa kabar?	Baik
3	Apa kabar?	Alhamdulillah kabar baik
4	Apa kabar?	Baik

Application has been built to make a conversation with chatbot. The application is standalone in the console and it

also needs a libraries to call the database. For application using Pascal language is shown in Figure 4.

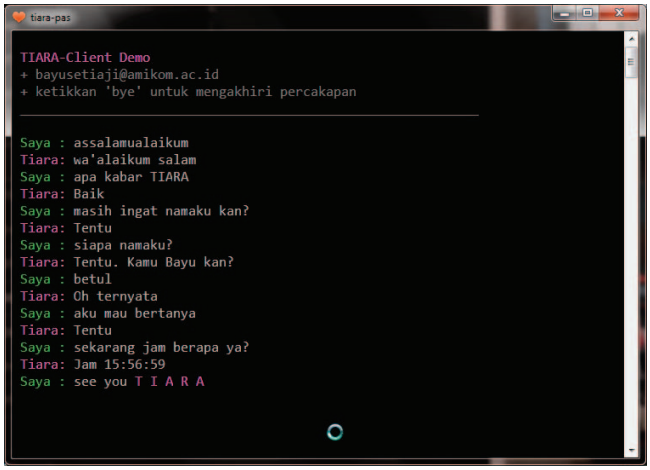

Figure 4. Chatbot Application using Pascal Language

It has also been built using Java language as shown in Figure 5.

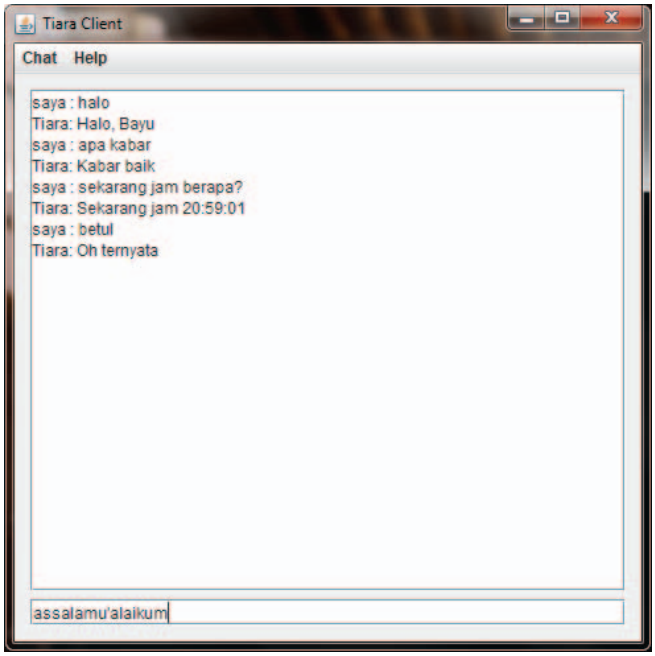

Figure 5. Chatbot Application using Java Language

V. CONCLUSIONS

The development of chatbot application in various programming language had been done with making a user interface to send input and receive response. Designing and building tables as representation of knowledge in the database had been started from entity-relationship diagram resulting 11 entities and its cardinalities. Making use of structured query language (SQL) for pattern matching had been done within stored program. The stored program consists of 4 stored procedures and 21 stored functions employed as pattern matching and supporting processes. Bigram method can be used not only for Indonesian language words, but also other languages with some boundaries.

ACKNOWLEDGMENT

We thank STMIK AMIKOM Yogyakarta which have given us chance for presenting our research paper

REFERENCES

[1] A. Augello, G. Pilato, A. Machi, and S. Gaglio, "An Approach to Enhance Chatbot Semantic Power and Maintainability: Experinces Within The FRASI Project," Proc. of 2012 IEEE Sixth International Conference on Semantic Computing, 2012, pp. 186-193, doi:10.1109/ICSC.2012.26.

[2] H. Al-Zubaide and A. A. Issa, "OntBot: Ontology Based Chatbot," Proc. IEEE of 2011 Fourth International Symposium on Innovation in Information & Communication Technology (ISIICT), 2011, pp. 7-12, doi:10.1109/ISIICT.2011.6149594.

[3] C. Erdogan, H. Nusret Bulus, and B. Diri, "Analyzing The Performance Differences Between Pattern Matching and Compressed Pattern Matching on Texts," Proc. IEEE of 2013 International Conference on Electronics, Computer and Computation (ICECCO), 2013, pp. 135-138, doi:10.1109/ICECCO.2013.6718247.

[4] J. P. McIntire, L. K. McIntire, and P. R. Havig, "Methods for Chatbot Detection in Distributed Text-Based Communications," Proc. IEEE of 2010 International Symposium on Collaborative Technologies and Systems (CTS), 2010, pp. 463-472, doi:10.1109/CTS.2010.5478478.

[5] Y. Wu, G. Wang, W. Li, and Z. Li, "Automatic Chatbot Knowledge Acquisition from Online Forum via Rough Set and Ensemble Learning," Proc. IEEE of 2008 IFIP International Conference on Network and Parallel Computing, 2008, pp. 242-246, doi:10.1109/NPC.2008.24.

[6] S. Ghose and J. J. Barua, "Toward The Implementation of A Topic Specific Dialogue Based Natural Language Chatbot As An Undergraduate Advisor," Proc. IEEE of 2013 International Conference on Informatics, Electronics & Vision (ICIEV), 2013, pp. 1-5, doi:10.1109/ICIEV.2013.6572650.

[7] A. Augello, M. Scriminaci, S. Gaglio, and G. Pilato, "A Modular Framework for Versatile Conversational Agent Building," Proc. IEEE of 2011 International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), 2011, pp. 577-582, doi:10.1109/CISIS.2011.95.

[8] G. Pilato, A. Augello, and S. Gaglio, "A Modular Architecture for Adaptive Chatbots," Proc. IEEE of 2011 Fifth IEEE International Conference on Semantic Computing (ICSC), 2011, pp. 177-180, doi:10.1109/ICSC.2011.68.

[9] G. Neubig, "NLP Programming Tutorial 2 – Bigram Language Models," Presentation Module of Nara Institute of Science and Technology (NAIST).

[10] M. Dickinson, "Smoothing," Presentation Module of Dept. of Linguistics, Indiana University, Fall 2009.

[11] Y. Bin, P. Cunlin, and L. Dan, "Chinese Text Feature Extraction Method Based on Bigram," Proc. IEEE of 2013 International Communications, Circuits and Systems (ICCCAS), 2013, pp. 342-346, doi: 10.1109/ICCCAS.2013.6765352