The Implementation of *K-Nearest Neighbor* Algorithm in *Case-Based Reasoning* Model for Forming Automatic Answer Identity and Searching Answer Similarity of Algorithm Case

Yana Aditia Gerhana[1], Aldy Rialdy Atmadja[2], Wildan Budiawan Zulfikar[3], Nurida Ashanti[4]
[1,2,3,4]Department of Informatics UIN Sunan Gunung Djati Bandung
Jl. AH Nasution No. 105, Bandung, West Java, Indonesia
[1]yanagerhana@uinsgd.ac.id, [2]aldy@if.uinsgd.ac.id,
[3]wildan.b@uinsgd.ac.id, [4]nurida.ashanti@student.uinsgd.ac.id

***Abstract*—Case-Based Reasoning also known as CBR model has been widely used to solve the problem in various cases. This study aims to explain the implementation of K-Nearest Neighbor Algorithm in Case-Based Reasoning model. The research showed that KNN algorithm is suitable to be used in CBR model. The results of this study are to measure the accuracy level of automatic answer identity formation and search the similarity answer in algorithm case. From the test result showed that KNN accuracy score obtained is 0.9 when the value of k=5.**

Keywords— Case Base Reasoning, K-Nearest Neighbor, similarity, answer, algorithm

I. RESEARCH BACKGROUND

As matter of fact, that algorithm constitutes a necessary and inevitable element in programming. Student's ability to solve algorithm cases is essential competence for programmer or student majoring in informatics engineering. However, some students regarded that algorithm is challenging course. Based on the interview, 31.8% from 84 students of Informatics Engineering student suggest that algorithm is a complicated subject. The interview result was supported by test score, 28.52% of students got score below the score C. Case-Based Reasoning (CBR) is an approach to solving problem that uses a database or previous case problems handled when solving new problems, where database is a collection of data/cases stored in the computer [1]. CBR can be applied to look for proximity level of new case data with old cases data becoming a reference in the decision making of a new case [2]. CBR has been proved to be quite effectively used in learning and is able to improve student's problem-solving in computer troubleshooting [3].

II. LITERATURE REVIEW

A. Algorithm

Algorithm may be defined as a sequence of steps to solve a problem [4]. Algorithm constitutes a procedure that contains the steps to resolve a problem. Algorithm is used for calculating, data processing, and automated reasoning. Algorithm has some statements such as expression, selection, repetition, procedure, combination, and so on. Algorithm consist of steps to resolve problems that can form three constructions or basic structures such as sequences, selection, and repetition.

B. Case Based Reasoning (CBR)

The cycle of problem solving within *CBR* model by Aamodt and Plaza is described in Figure 1 [5]:

1) *Retrieve*
Finding/regaining the most relevant/similar case to the new case. Retrieval phases begins with describing/outlining some of the problems, and ends when a new case suits a previous problem with the highest level of compatibility. This section refers to the terms of identification, initial matching, searching, selection and execution.

2) *Reuse*
Modeling/reusing knowledge and information of an old case based on the extent of most relevant similarity to a new case, resulting in a proposed solution which may require adaptation to the new problem.

3*) Revise*
Reviewing the proposed solution and then testing it in a real case (simulation) and if it is necessary revising this solution may be taken to suit a new case.

4) *Retain*

Integrating/storing a new case that has already gained a solution in order that the new case can be used by subsequent cases similar to the case. But if a new solution fails, researcher can explain its failures, correct the solution used, and test it again.

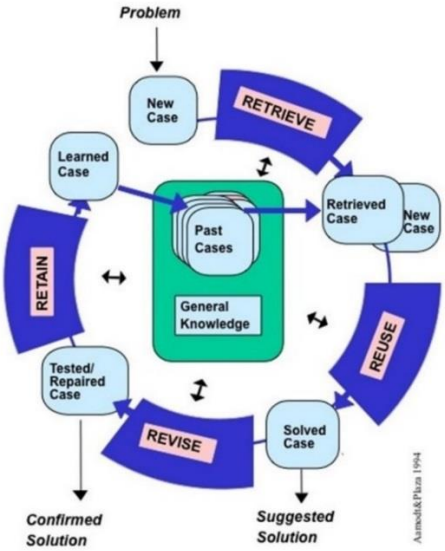

Figure 1. Cycle of CBR [5]

C. *K-Nearest Neighbor (KNN)*

K-Nearest Neighbor (KNN) is a method that uses supervised learning. In this approach, the value of k is used to selected category based on training examples. The purpose of the algorithm is classifying the new objects based on attribute and training data. Algorithm classification uses a majority voting of object k and uses the neighborhood as the predicted score of new query instance. KNN has several advantages when the training data that has a lot of noise and it must be effective when the training data is massive. Meanwhile, the weaknesses of KNN are a necessity for determining the exact score of the parameter k (the number of nearest neighbors). Besides that, the process on KNN depends only on the distance from one data to another, so there is no reference which attribute/feature has the effect to get the best results. Also, this algorithm has a high cost of computing because of the calculation of the distance from each query instance in the whole training [6].

The classification of *KNN* algorithm has the following steps:

a. Specify parameter *K*.

b. Calculate the distance between the data to be evaluated with whole trainings.

c. Sort formed distance (ascending).

d. Determine the shortest distance to the sequence *K*.

e. Pair the corresponding class.

f. Find the number of classes from the nearest neighbor and set the classes as data classes to be evaluated.

D. Classification and Similarity Stages

The classification and similar stages start from collecting algorithm questions. The amount of data required for training data is 68 documents. The data has been labeled as the category of algorithm questions. Later, the preprocessing step applied for the data such as case folding, tokenizing, filtering and stemming. After tokenizing levels in preprocessing stages, each term of the document is weighted with TF-IDF so that the results can be tested with KNN. Then, KNN algorithm is used to find similarities between new problems with the previous training data to obtain the results of the identity of the answer. Meanwhile, the CBR method (Decision Support System for problem-solving) is used as a consideration of new knowledge from prediction. In figure 2, the picture shows the process of Classification and Similarity Stages that implemented with KNN Algorithm and CBR model.

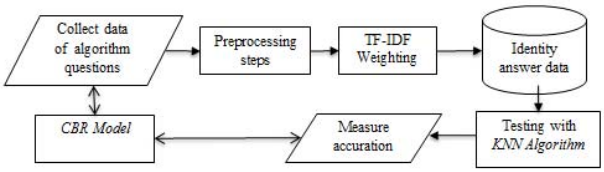

Figure 2. The Process of Classification and Similarity Stages

E. Text Mining and Preprocessing Technique

Text mining constitutes one of the specialized fields of data mining. Text mining may be defined as a process of finding information where a user interacts with a set of documents using the analysis tools as components of data mining one of which is categorization. Meanwhile, the special task of text mining is text categorization and text clustering [7].

There are some stages when implemented text mining. The most important steps to retrieve information from text is preprocessing stage. Some preprocessing level applied in this research as follows:

a. Case Folding

Case folding is a step usually done in the early process. This stage functions to convert text into small letters (lowercase) and eliminate characters except for a-z [8].

b. Tokenizing

Tokenizing process is a cutting step of input string based on every word that composes it. This process produces words that stand alone [9].

c. Filtering

Filtering process is a step to take important words from token. The process can use stop list algorithm (removing unnecessary words or word list. This process will produce only important words and remove unnecessary words) [8][9].

d. Stemming

Stemming process is a process to change word form to be base word. The process works through removing prefixe, affix, and suffix. The purpose of stemming process is improving the efficiency of system [10].

F. TF-IDF

The initial step of Text Mining before applying *TF-IDF* process are token and stopword steps. *TF-IDF* is one method that can function to make the weighting of term. *TF (*Term Frequency) is word weighting (term) that is based on calculating the number of words appearing in a document. *IDF* (Inverse Document Frequency) is word weighting (term) that is based on calculating the number of words appearing in all documents [11]. The more words are available in a document, the greater is the weight of the words, and vice versa. *TF-IDF* (Term Frequency - Inverse Document Frequency) is word weighting in a document to be processed further by other algorithms [12].

$$W_{dt} = tf_{dt} * IDF_t = tf_{dt} \log \frac{D}{df_t} \quad (1)$$

Where:

d = document to-d
t = word to-t keyword
W = document weight to-d toward word to-t
tf = number of words searched in a document
D = total document

G. Cosine Similarity

Cosine similarity is used to calculate query relevance approach toward a document. Determining a query relevance toward a document is viewed as similarity measurement between the query vector and document vector. The greater the similarity score of query vector to document vector is, the query is seen to be more relevant to the document. When an engine receives query, the engine will build a vector Q (Wq1, Wq2, ... Wqt) based on the terms in the query and a vector D (di1, di2, ... dit) sized t for each document. In general, cosine similarity (CS) is calculated by cosine measure formula [13]

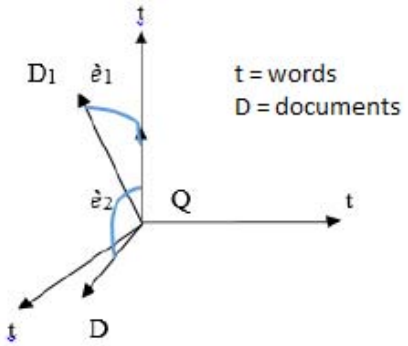

Figure 3. Scalar Vector

Cosine Similarity calculation can be performed with the formulation:

$$CS(b_1, b_2) = \frac{\sum_{t=1}^{n} W_{t,b1} W_{t,b2}}{\sqrt{\sum_{t=1}^{n} W^2_{t,b1} \sum_{j=1}^{n} W^2_{t,b2}}} \quad (2)$$

Where:

T = term in a sentence
Wt, b1 = weight of *term* t in *block* b1
Wt, b2 = weight of *term* t in block b2

III. RESULT AND DISCUSSION

A. Case Folding and Tokenizing

In figure 4 showed the implementation of case folding and tokenizing. As described earlier, case folding used for converting text to lowercase. Besides, tokenizing is the method for split word into a letter. To prove, in this below picture tell that two approaches which changing words into lowercase and splitting into letters.

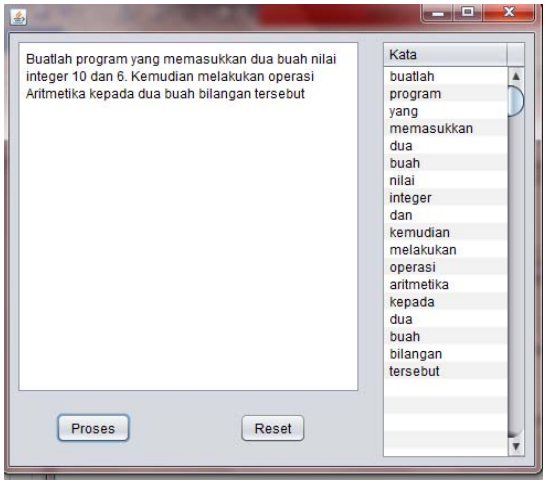

Figure 4. Implementation of Case Folding and Tokenizing

B. Algorithm Stopword Removal and Porter Stemmer algorithm

The implementation of algorithm Stopword Removal and Porter Stemmer is not directly demonstrated in the application, but the result is directly stored into database. Here is the result of algorithms Stopword Removal and Porter Stemmer processing in the database taken from appindex table last line or the result of question pre-processing to find the answers.

Figure 5. Implementation of Question Input Stopword and Porter Stemmer

C. Tf-Idf

Figure 5 below describes the example of *TF-IDF* implementation result. From this picture shows each of term that have weight that calculate from the documents.

Term	TF	IDF
abc	2	3.806662489770...
absolute	1	4.499809670330...
ada	2	3.806662489770...
agustus	2	3.806662489770...
air	2	4.499809670330...
akhir	6	2.890371757896...
algoritma	53	0.567984037605...
aman	1	4.499809670330...
ambah	3	3.806662489770...
ambil	1	4.499809670330...
anak	2	4.499809670330...
analisis	1	4.499809670330...
andai	3	3.806662489770...
anggap	1	4.499809670330...
aritmetika	2	3.806662489770...
arsip	1	4.499809670330...
arti	1	4.499809670330...
asosiasi	1	4.499809670330...
asumsi	3	3.401197381662...
atas	1	4.499809670330...
atm	1	4.499809670330...
awal	4	3.401197381662...
awar	1	4.499809670330...
ayam	2	4.499809670330...
baca	66	0.607989372219...
bagi	4	3.113515309210...

Figure 6. Implementation of TF-IDF in Processing Page

D. Cosine Similarity

Figure 6 below describes the example of *Cosine Similarity* Implementation result

ID	Similarity
1	0.00371352696910...
2	0.00242229676492...
3	0.01217695536889...
4	0.26122992100253...
5	0.03252670551930...
6	0.00347939371287...
7	0.00147728285161...
8	0.10802995104060...
9	0.01581323642007...
10	0
11	0.18326702668037...
12	0.147169735781559
13	0.66640671797737...
14	0.03367999580219...
15	0.00981320318439...
16	0.11641292191514...
17	0.01853607976816...
18	0.02950643014222...
19	0.01281083121679...
20	0.00744687191809...
21	0.00258814121923...
22	0.00579979668350...
23	0.00636636225792...
24	0.00298495125493...
25	0.12440671876179...
26	0.00737436595425...

Figure 7. Implementation of Cosine Similarity in Processing Page

E. KNN Algorithm

The implementation of KNN is obtained from the sequencing score of cosine similarity with the score k=5 and classification based on vote of majority category; however if the number of vote category is equal 2:2:1 the category that has the highest cosine similarity is taken from the equal number of documents above. For example, the implementation of the running program below will produce a sequence classification due to a vote of the five categories as shown below:

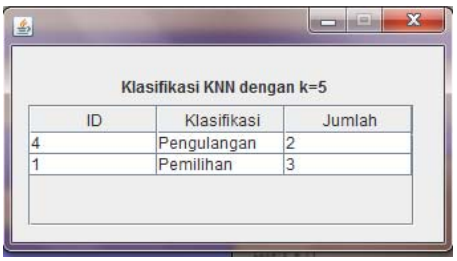

Figure 8. Implementation of KNN in Processing Page

F. Testing Document

The applied testing is category classification with a score of k = 5, k = 7 and k = 9. To find out the processing result of answer similarity search is using 10 questions compared with 90 knowledge data of old cases that are viewed to have valid status in the database and consists of three categories: sequencing, selection and repetition, which have 30 documents in each category. The testing results can be seen in Figure 8 below:

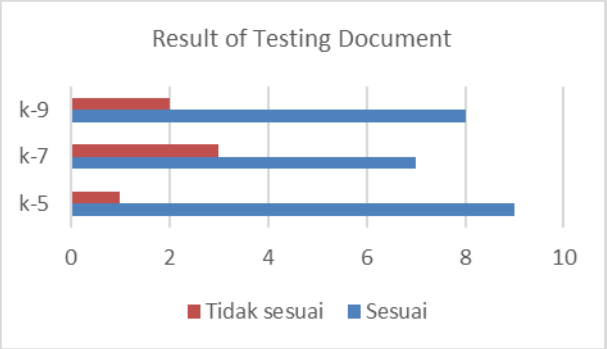

Figure 9. Result of Testing Document

Testing is conducted to find out the precision and accuracy of document classification, using the testing of accuracy score. Based on comparison with the score k above, the implementation of algorithm basic structure classification can work well when using *KNN* method with a score of k = 5, which has the following accuracy score:

$$Accuracy = \frac{\sum relevant\ document}{\sum total\ document} = \frac{9}{10} = 0{,}9$$

The score 0.9 is defined to be nearly accurate score

IV. CONCLUSION AND FUTURE WORKS

The application of *KNN* in *CBR* model is suitable in finding algorithm case similarity. The testing with 10 algorithm questions by means of training data of 90 documents with 3 categories: sequencing, selection and repetition, each of which has 30 data in every category, meanwhile the testing result showed that *KNN* made nearly accurate score when the score of k = *5*, where the accuracy result obtained is 0.9.

This research can still be further develop. The development can be improve with another learning algorithm such as Support Vector Machine, Decision Tree and Naïve Bayess.

V. REFERENCES

[1] Maher, L, M., Balachandran B, M. & Zhang, M, D. *Case-Based Reasoning in Design. Australia. Psycology Press.*1995.

[2] Luthfi ET. Penerapan Case Based Reasoning dalam Mendukung Penyelesaian Kasus. JURNA DASI, p. 10. .2010

[3] Gerhana YA, Djohar A. Case-based Reasoning Learning Model to Develop Skill in Problem Solving of Student of Vocational Education. International Journal of Basic and Applied Science. 04(11). 2016 April.

[4] Munir R. Algoritma dan Pemrograman dalam Bahasa Pascal dan C, Edisi ke-3, Buku 1 Bandung: Informatika Bandung; 2011.

[5] Aamodt, A. & Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, Journal of Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications. IOS Press, Vol VII No I, 1994, hlm. 39-59.

[6] Yovianto E. Buku TA : K-Nearest Neighbor (KNN). [Online].; 2010 [cited 2016 Januari 9. Available from: https://kuliahinformatika.wordpress.com/2010/02/13/buku-ta-k-nearest-neighbor-knn/.

[7] Diaz R. Pengertian Data Mining,Teks Mining,dan Web Mining. [Online].; 2013 [cited 2016 Januari 10. Available from: http://yosephoriolryandiaz.blogspot.co.id/2013/03/pengertian-data-miningteks-miningdan.html.

[8] Atmadja, A. R., and Purwarianti, A. (2015). Comparison on the rule based method and statistical based method on emotion classification for Indonesian Twitter text. In *Information Technology Systems and Innovation (ICITSI), 2015 International Conference on* (pp. 1-6). IEEE.

[9] Hasanah U. Label: Analyzing, Dokumen, Filtering, Kata, Proses Filtering, Stemming, Tagging, [Online].; 2012 [cited 2016 Januari 27. Available from: http://sistemtemukembaliinformasi.blogspot.co.id/2012/07/tokenisasi.html.

[10] Rizki AS, Indriati , Muflikhah L. Text Mining Klasifikasi Soal Biologi Sekolah Menengah Atas Dengan Metode Improved KNN. Repositori Jurnal Mahasiswa PTIIK UB. 2014;: p. 8.

[11] Ridok A. Pembuatan Judul Otomatis Dokumen Berita Berbahasa Indonesia Menggunakan Metode KNN. ISSN: 1907-5022. p. 5. 2012.

[12] Samuel Y, Delima R, Rachmat A. Implementasi Metode K-Nearest Neighbor dengan Decision Rule untuk Klasifikasi Subtopik Berita. Jurnal Informatika, Vol. 10 No. 1. 4; p. 15. 201

[13] Salsabilla SM. Sistem Peringkasan Jurnal Ilmiah Menggunakan Metode Maximal Marginal Relevance Bandung. Universitas Islam Negeri Sunan Gunung Djati Bandung; 2016