Jurnal Ilmiah Teknik Informatika dan Komunikasi
Halaman Jurnal: http://journal.sinov.id/index.php/juitik/index
Halaman UTAMA Jurnal : https://journal.sinov.id/index.php

Autocomplete Indonesian Dictionary with Trie and Depth-First Search Algorithm

Aaqila Dhiyaanisafa Goenawan [a], Abdullah Ammar [b],
Mutiara Persada Pulungan [c], Desy Komalasari [d]
[a] Program Studi Ilmu Komputer, a.dhiyaanisafa.g@students.esqbs.ac.id, STIMIK ESQ
b Program Studi Ilmu Komputer, a.ammar@students.esqbs.ac.id, STIMIK ESQ
[c] Program Studi Ilmu Komputer, m.persada.p@students.esqbs.ac.id, STIMIK ESQ
[d] Program Studi Ilmu Komputer, desy.komalasari@esqbs.ac.id, STIMIK ESQ

ABSTRAK

In the current era of technology, the internet provides many conveniences for our daily lives. There are no limitations in accessing the internet, we can even search for information that we do not know through the internet. One of them is by using an electronic dictionary. By using the electronic dictionary the user only needs to enter a keyword and the machine will search for the appropriate data. But when the user finishes entering the word in the search engine, the word is not necessarily found, this makes the electronic dictionary less than optimal in terms of time usage. Therefore, we use the autocomplete feature with the trie data structure and the dfs algorithm in order to shorten the time the user is typing a word, where this feature will display a list of words that the user might mean without having to type the word in full.

Keywords: Aplikasi Indonesian Dictionary, Autocomplete, Trie, Depth-First Search (DFS)

1. INTRODUCTION

Dictionary is a reference book that contains words of a language with information on their meanings, forms, pronunciations, functions, spellings, and idiomatic uses. It was first introduced by Samuel Johnson, who wrote the first comprehensive English dictionary in 1750 in a book. Ever since then, people have written dictionaries in their own language and distributed them in the form of books.

As technology grows, dictionaries are made available in the form of websites and mobile applications, where we have to input the word that we want to find manually. However, we as a human being will make mistakes such as typos, so that the word we want to find out cannot be found in the dictionary. Thus, this study is made to implement the Autocomplete feature in an electronic dictionary.

Autocomplete is a feature where it can give recommendations for users without having to type the query completely. Trie is one of data structures that we can use to implement the autocomplete feature in our electronic dictionary because it is a sorted tree-based structure whose nodes store the letters of an alphabet. In order to complete the autocomplete feature, we will be using Depth First Search as the algorithm for traversing and searching the Trie data structure.

2. TINJAUAN PUSTAKA

2.1. Algorithm

Algorithm is a collection of steps that are arranged in writing and sequentially to solve certain problems, such as math and logic problems. The development of science and technology makes humans able to produce works that are increasingly sophisticated and complex. Although computers can perform calculations faster than humans in general, computers cannot simply solve problems without being taught by humans through a sequence of steps (algorithms) defined in advance. Besides being used for problem solving using computers,

Received JANUARI 10, 2022; Revised FEBRUARI 22, 2022; Accepted MARET 27, 2022

algorithms can also be applied in solving everyday problems that require a series of processes or procedural steps.

2.2. Autocomplete

Autocomplete is a feature provided by many web browsers such as search engine interfaces, word processors, and command line interpreters. Moreover, it can predict which word the user is typing and complete it automatically and give suggestions to the user without writing the whole word completely.

2.3. Trie

A trie is a tree data structure used to efficiently store and retrieve keys in a string data set. The experiment consists of nodes and edges. Each node has a maximum of 26 children and an edge connects each parent node with its children. There are various applications of this data structure, such as autocomplete and spell checker. Here's how it's implemented:

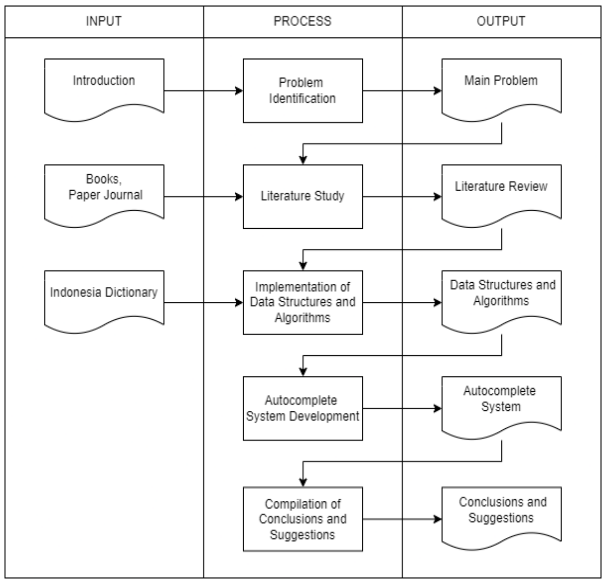
- Trie() Initializes the Trie object.
- void insert(String word) Inserts a string word into the trie.
- boolean search(Word string) Returns true if the word string is in the trie (that is, entered previously), and false otherwise.
- boolean startWith(String prefix) Returns true if any of the previously entered string words has a prefix, and false otherwise.

2.4. Depth-First Search (DFS)

Depth First Search (DFS) algorithm is a search method in a tree with one branch trip of a tree until it finds a solution. The search is carried out on one node in each level from the far left and continues on the right node. If a solution is found, there is no need for a backtracking process, namely backtracking to get the desired path. In the DFS method, the usage of the memory is not much because only nodes on the active path are stored. In addition, if the searched query is at the leftmost level, then DFS will find it quickly.

3. METODOLOGI PENELITIAN

The research steps are designed according to which is presented in Figure 1.


INPUT
PROCESS
OUTPUT
Introduction
Problem Identification
Main Problem
Books, Paper Journal
Literature Study
Literature Review
Indonesia Dictionary
Implementation of Data Structures and Algorithms
Data Structures and Algorithms
Autocomplete System Development
Autocomplete System
Compilation of Conclusions and Suggestions
Conclusions and Suggestions

Gambar 1. Research Steps

Figure 1 illustrates the research design flow. The following is an explanation of the stages of the research carried out:

1. Problem Identification. Purpose to determine background research problem main research problem.

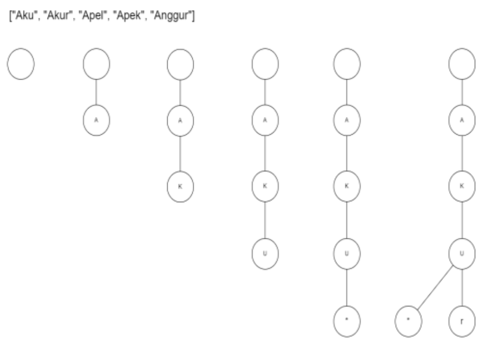
2. Literature Study. The implementation of a library review is carried out to find theories, research, and methodologies relevant to the research questions that have been determined in the previous stage sourced from textbooks, papers, and international journals so as to produce outputs equal to the literature review that underlies the study.
3. Implementation of Data Structure and Algorithm. Implementation of Trie data structure and Depth-First Search algorithm using JavaScript.
4. Autocomplete System Development. Developing website and autocomplete features using HTML, CSS, and JavaScript.
5. Preparation of Conclusions and Suggestions. The last step is to make conclusions and suggestions based on the results and analysis of the previous stage.

4. HASIL DAN PEMBAHASAN

In this section, we will explain the implementation of Data Structures and Algorithms and Autocomplete Systems.

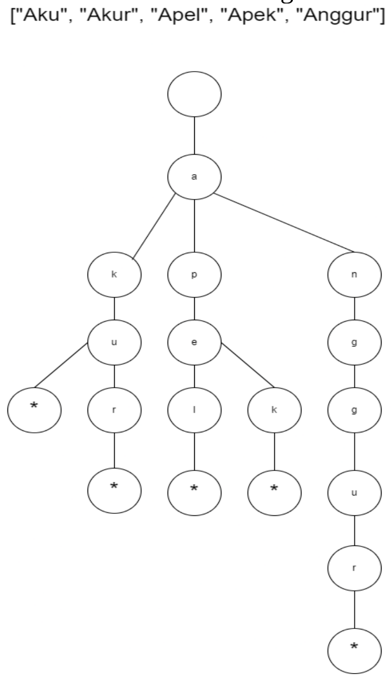
4.1 Implementation of Data Structures and Algorithm

Before building the autocomplete feature, a Trie data structure is needed that will be used to store words from the Kamus Besar Bahasa Indonesia. The Trie data structure breaks words into letters and stores each letter into a node which is a HashMap data structure. The process of forming the Trie data structure can be seen in Figure 2.



Gambar 2. Trie formation process

The Trie data structure that has been formed can be seen in Figure 3.

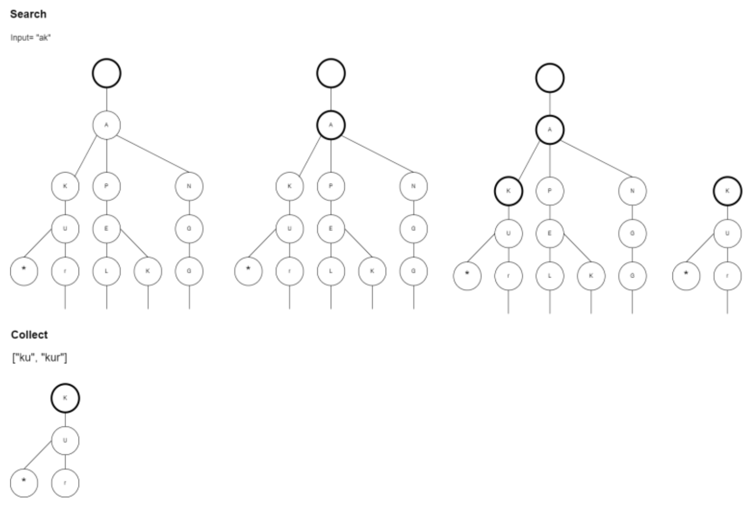


Gambar 3. Trie Data Structure

After the Trie data structure is formed, it is possible to search for words using the Depth-First Search algorithm.

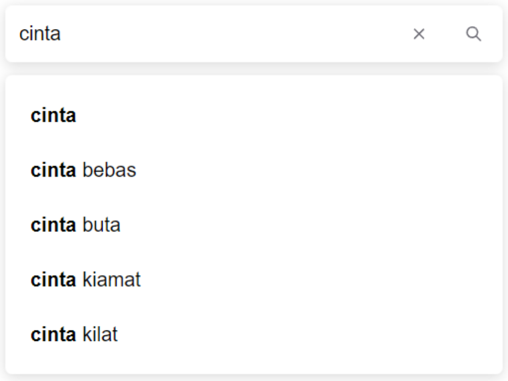
4.2 Autocomplete System Development.

The autocomplete feature is executed in two stages. The first stage is the search for the last node using the Depth-First Search algorithm from the input to be broken down into a list of letters. After the node is found, the remaining words are collected in the form of an array. The process of searching for nodes and collecting words can be seen in Figure 4.



Gambar 4. Search for nodes and collect words

After the collection of remaining words is obtained, the remaining words will be combined with user input and displayed in the form of an HTML list. The word list can be seen in Figure 5.



cinta

cinta

cinta bebas

cinta buta

cinta kiamat

cinta kilat

Gambar 5. Word list

4.3 Analysis

Depth First Search Algorithm

The autocomplete feature in this research is made using the Depth First Search algorithm because we think it is the right algorithm for the data structure that is implemented here. The Depth First Search algorithm here is used to form all possible letters in the node from the user input. After the collection of remaining letters is obtained, the results will be displayed in the form of a recommendation list for the user's input. Based on the results obtained, the Depth First Search algorithm is the best algorithm to be implemented in this Autocomplete feature since the definition of the Depth First Search itself is an algorithm for traversing or searching tree or graph data structures, such as Trie data structure.

Time Complexity

Time complexity for search and input is O(k), where k is the length of the input string. For example, if input is “ak” then time complexity is O(2), where 2 is the length of “ak”. For collecting words, the time complexity is O(m), where m is the sum of node children from the current node.

Space Complexity
Space complexity for Trie data structure is O(n), where n is length of input string since n new nodes are added which takes up space O(n).

5. CLOSING STATEMENT

5.1 Conclusion

1. Autocomplete is implemented in the KBBI system by adding this feature to the word search input box. Then the data trie is used to store string data in the autocomplete feature and the dfs algorithm is implemented during the word input process.
2. The added features of autocomplete successfully display several search suggestions based on the location of the nodes in the trie. This makes it easier for users without having to type in the words they want to search for in their entirety.

5.2 Suggestion

The authors of this research recommend improving the results of the Autocomplete feature by showing the most searched or the most popular words to be shown first on the recommendation.

REFERENCES

[1] A. Jeklin, “DEPTH FIRST SEARCH (DFS) UNTUK MENENTUKAN DIAMETER GRAF HIRARKI,” vol. 11, no. July, pp. 1–23, 2016.
[2] D. Sianturi, “PERANCANGAN FITUR AUTOCOMPLETE PADA APLIKASI KAMUS ISTILAH TEKNOLOGI INFORMASI MENGGUNAKAN ALGORITMA BOYER-MOORE,” J. Pembang. Wil. Kota, vol. 1, no. 3, pp. 82–91, 2021.
[3] E. Sarigul, “The importance of using dictionary in language learning and teaching,” 2016.
[4] M. M. Yulianto, R. Arifudin, and A. Alamsyah, “Autocomplete and Spell Checking Levenshtein Distance Algorithm To Getting Text Suggest Error Data Searching In Library,” Sci. J. Informatics, vol. 5, no. 1, p. 75, 2018, doi: 10.15294/sji.v5i1.14148.
[5] R. Rohmatillah, “Dictionary Usage In English Language Learning,” English Educ. J. Tadris Bhs. Ingg., vol. 9, no. 1, pp. 186–197, 2016.https://docs.moodle.org/29/en/About_Moodle. [Accessed 23 10 2015].
[6] W. Dakun, “SHOULD THEY LOOK IT UP ? THE ROLE OF DICTIONARIES IN LANGUAGE Review by Wang Dakun,” vol. 2001, no. 1, pp. 27–33, 2001.