Speed Analysis of Binary Search and Interpolation Search for Searching Identification Numbers on National Identity Cards

ABSTRACT

Aims: The research purpose is to make comparisons and find the best search algorithm between binary search and interpolation search algorithms for searching identity number on Indonesian identity cards.
Study design: This study was designed by implementing binary and interpolation algorithms into C language, with a dataset of 5000.
Place and Duration of Study: Department of InformaticUniversitas Multimedia Nusantara, between November 2022untilMarch 2023.
Methodology: Binary algorithms and interpolation search will be tested by searching for identity number on identity card data, totaling 5000 data. To measure the level of speed and effectiveness of the two algorithms, this will be done by trying to search for data in various amounts, namely 1000, 2000 and 5000 data, with the data sought between the two algorithms having the same value. Testing is carried out by implementing the two algorithms into C language to get the number of loops and execution time, and testing of the algorithms on each sample dataset is carried out five times to get the average value of the test results.
Results: The results of the tests that have been carried out between the binary search and interpolation search algorithms from the existing dataset, found that the number of loops in the interpolation loop is 36.57% better than the binary search algorithm, while for execution time, binary search has a faster execution speed of 12.43% than interpolation search.
Conclusion: It can be concluded that if the dataset is large enough, such as identity card data on Indonesian citizens, and if the computer specifications are adequate, binary search, which has a faster execution time, should be used; however, if the computer specifications are inadequate, interpolation search, which has a smaller number of loop processes, should be used.

Keywords:Binary Search, Execution time, Identity number on identity card, Interpolation Search, Loop time,Searching Algorithm

1. INTRODUCTION

The development of science and technology today greatly affects human life. With the development of technology, many computer applications can be used to simplify human life. These technological developments greatly affect the amount of current data usage[1]. One example of the use of data that is very necessary is data on citizen identification numbers where this data is very important in a government.

Indonesia, as the fourth most populous country in the world, has a population of more than 273 million people and has 16,771 islands, 98 cities and 38 provinces[2]. Data is too much and very complex. As well as further handling is needed in managing the large amount of

data, especially when searching for a parent number or a certain word so that processing time does not take a long time[3].

To perform fast, precise and accurate searches on large and complex data, it is necessary to apply an appropriate search algorithm. A search algorithm is an algorithm that accepts a key argument and with certain steps will search for records with that key[4]. while searching is the process of finding certain values in a set of data of the same type[5], [6]. Searching aims to find out whether the value sought is in a data set or not. There are several algorithms that can be used to perform searches[7], [8], one example of a search algorithm is the binary search algorithm and also the interpolation search algorithm. The speed of processing time and the complexity of the search algorithm were tested in this study to search for identification numbers on Indonesian national identity cards, with the goal of providing a comparison of the processing time and complexity between the two algorithms[9].

The choice of binary search and interpolation search algorithms in this study and the problem of searching for identification numbers on identity cards in Indonesia, is because the two algorithms start searching starting from the midpoint and the search does not start from the frontmost index or randomly[10], [11]. This is because on the Indonesian national identity card, which has many provinces, the first digit number to the middle is the code for the province, district, sub-district up to the date of birth, therefore the search can be focused on the middle index of the population number to the last index of the population number, so it is expected can perform searches more effectively and quickly. The purpose of this research is to compare and find the best search algorithm between binary search and interpolation search algorithms in searching for identification number data on identity cards.

2. LITERATURE REVIEW

2.1 Binary Search

Binary search is a search algorithm for sorted data. Searching is done by guessing whether the data being sought is in the middle of the data, then comparing the data being sought with the data in the middle. If the data in the middle is the same as the data you are looking for, it means that the data is found[12]. However, if the data in the middle is greater than the data being searched for, then it is certain that the data being searched for is likely to be on the left of the middle data and the data on the right of the middle data can be ignored[13], [14]. The upper bound of the new left data is the index of the middle data itself. Conversely, if the data in the middle is smaller than the data being searched for, it can be ascertained that the data being sought is most likely to be to the right of the middle data. The lower bound of the data to the right of the middle data is the index of the middle data itself plus one, and so on[6].

2.2Interpolation Search

Interpolation search is a search method that searches for key values in a set of data or arrays that have been ordered. InterpolationSearch is a technique of developing binary search. The binary search technique will always check the middle value of each array, while the interpolation search can go to different locations based on the obtained key. If the key value is closer to the last array, then the interpolation search technique will start searching from the last array[15], [16]. This method isbased on searching the phone book where the search is based on the key values in the phone book.

Binary search, on the other hand, always chooses the middle of the remaining search space, discarding one half or the other based on a comparison between the key found at the estimated position and the key sought; it does not require numerical values for the keys, only a total order on them. The remaining search space is constrained to the region preceding or

following the estimated position. Because it compares elements one by one from the start, the linear search only uses equality[17].

If the elements are uniformly distributed, the interpolation search makes approximately log(log(n)) comparisons, where n is the number of elements to be searched. In the worst-case scenario (for example, if the numerical values of the keys increase exponentially), it can perform up to O(n) comparisons.Interpolation-sequential search employs interpolation to locate an item that is similar to the one being sought, followed by linear search to locate the exact item[18], [19].

For a Sorted and Uniformly Distributed array, Interpolation Search outperforms Binary Search. Regardless of search-key, Binary Search checks the middle element. Interpolation Search, on the other hand, may go to different locations based on the search-key. If the search-key value is close to the last element, Interpolation Search will most likely begin searching on the end side.Both interpolation and binary search are algorithms for finding a specific element in a sorted list or array. The average-case time complexity of both algorithms is O(log n), which means that the time required to perform the search grows logarithmically with the size of the list[19], [20].

3. RESEARCH METHODOLOGY

The stages in this research are divided into several steps, the first step is problem identification, where it has been explained that searching for identification numbers on a large number of resident cards creates many problems, namely the search process is long and very complex. After identifying the problem, then the next research step is a literature review and literature study related to the search algorithm that will be used in this study,

The next step is to do a needs analysis to compare the two search algorithms and the method to be used to find out the results of the comparison, the identity card dataset sample that will be tested in this study totals 5000 data, where the search test will be carried out in three phases, the first quests in the amount of 1000, then the next phase, amounting to 2000, the last phase amounting to 5000.

The method used to search for identification number data on identity cards using a binary algorithm and interpolation of the dataset that has been collected is by implementing the search algorithm into a programming language, namely the C programming language to obtain a comparison of the search time between the two algorithms. then the final step is to carry out a test analysis of the results of the comparison of the binary search and interpolation algorithms and then record and calculate as a result of the research that has been carried out which is the conclusion from the results of this study. The application or implementation of the binary search algorithm in the C programming language used to search for identification numbers on identity cards can be seen in Figure 1.

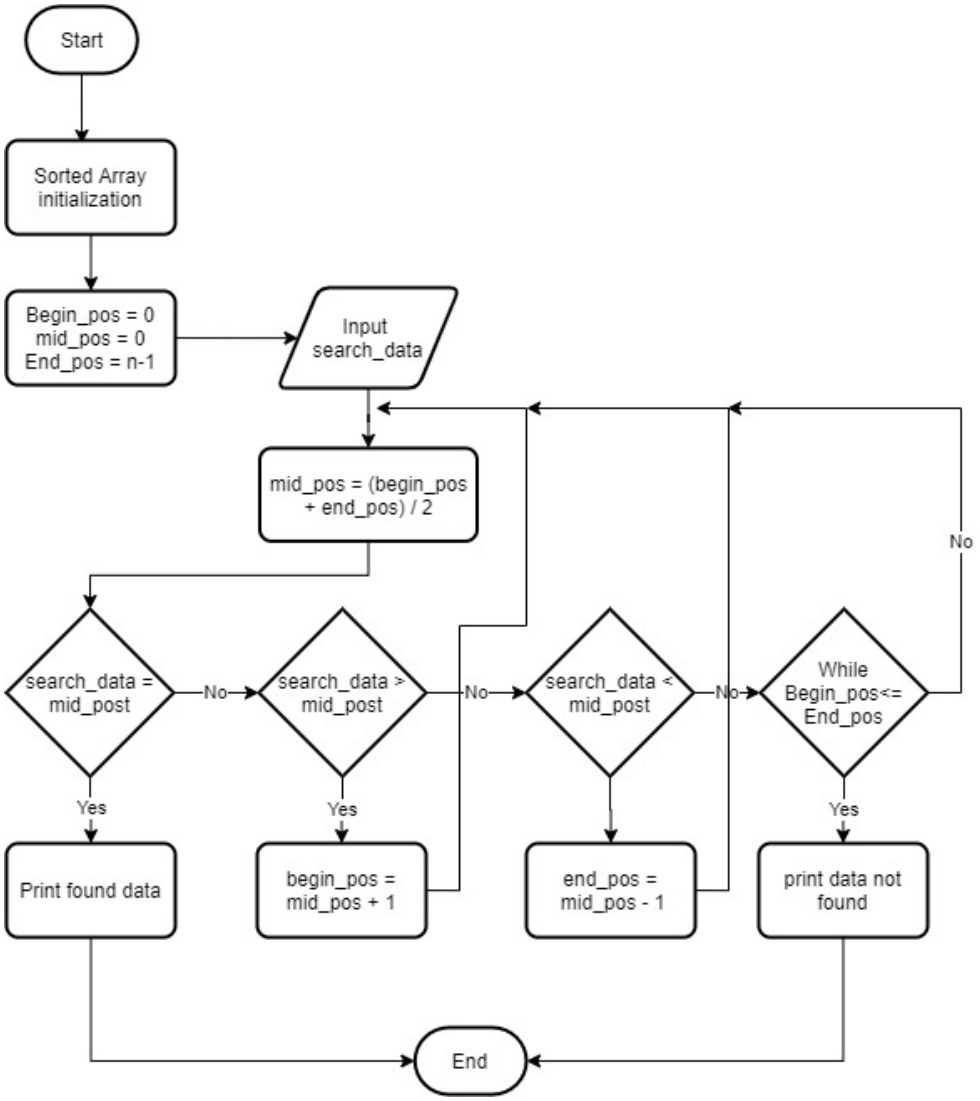

Fig. 1.Binary Search implemantation for search Identity number on Identity Card

In addition to the implementation of the binary search algorithm in the C programming language and the C++ programming language, an interpolation search algorithm was also created in this study using the same method to search for identification numbers on identity cards, the application of the interpolation search method can be seen in Figure 2.

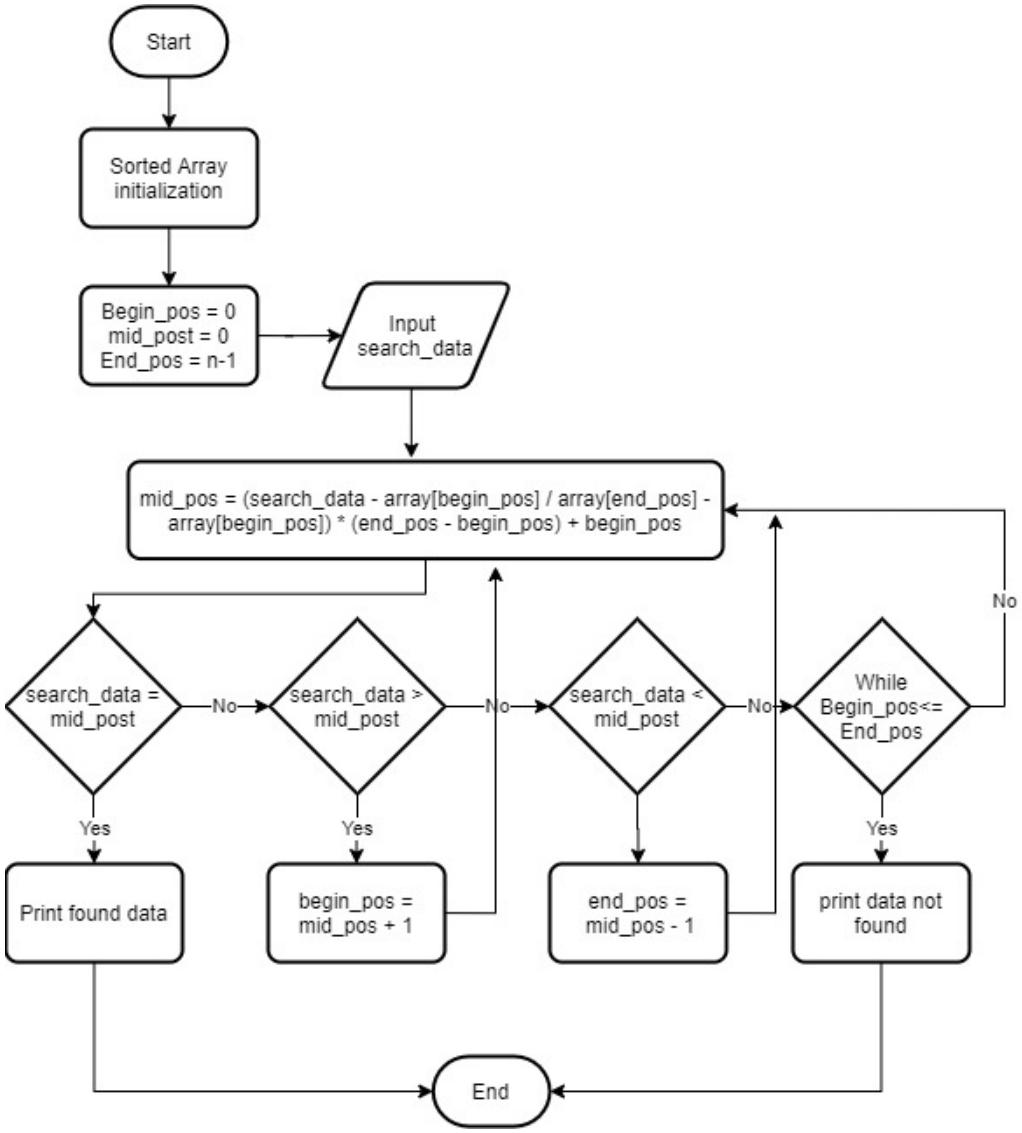

Fig. 2.Interpolation Search implemantation for search Identity number on Identity Card

4. RESULT AND DISCUSSION

In algorithmsinterpolation andbinary search data must be sorted first, with data sorting making it easier to search. There are two possibilities that will occur when searching for data usingbinary search orinterpolation search, this possibility can be described with big O notation. Big O notation is a mathematical notation that describes the limits of the behavior of a function when the argument goes to a certain unlimited value or value. the first possibility is the best possibility (best case) where the value sought is right in the middle of a set of data so it doesn't require removing or selecting data, in big O notation it can be written O(1).

The second possibility is the worst possibility (worst case) where the value sought is not found, this algorithm will continue to search for that value until the final position is reversed from the initial position. This condition can be written in big O notation, namely O(log n). In

the algorithm there is also a data type, the data type is an attribute of the data where the attribute describes how the programmer should use the data. In other words, the data type algorithm plays an important role in determining an action. Interpolation search and binary search are search algorithms that string data types, but in the use of algorithmsinterpolation search the use of the string data type is small, because the interpolation search uses a search with a mathematical formula.

In the testing phase of the binary search algorithm using different data types and the same amount of data, and testing is carried out by searching for data that is not contained in the array list in order to obtain the worst case condition of the two binary search algorithms and the interpolation. Then the data search was also carried out five times to get the average execution time of each of these algorithms. The dataset to be searched has also been sorted according to the data search criteria using binary and interpolation. After the two algorithms are implemented into the C programming language with the search string data type in the form, the search time results are obtained which can be seen in Table 1.

Table 1. Binary Search with string data type and with the execution time that was tried 5 times the trial

No	Dataset	Binary Search Execution Time (in second)						
		Loop count	First try	Second try	Third try	Forth try	Fifth try	Average
1	1000	10	1.672	1.663	1.771	1.665	1.656	1.685
2	2000	11	1.703	1.672	1.668	1.863	1.610	1.703
3	5000	12	1.738	1.702	1.703	1.713	1.785	1.728

In the testing phase of the interpolation search algorithm using the string data type and the same amount of data using the C programming language and the dataset to be searched for has been sorted, and with the conditions the dataset and value sought are the same as the dataset and value in the binary search algorithm experiment, the results from the interpolation search can be seen in Table 2.

Table 2. Interpolation Search with string data typeand with the execution time that was tried 5 times the trial

No	Dataset	Interpolation Search Execution Time (in second)						
		Loop count	First try	Second try	Third try	Forth try	Fifth try	Average
1	1000	4	1.601	1.664	1.694	1.741	1.577	1.655
2	2000	4	1.726	1.805	1.575	1.643	2.073	1.764
3	5000	4	1.949	1.736	1.696	2.638	2.329	2.070

From the results of tests that have been carried out on the binary and interpolation search algorithms from five trials by implementing them into the C programming language, the results of the execution time and loop count for each algorithm can be seen in Table 1 and Table 2, then from the results of the two tables , a comparison can be made, which can be seen in Table 3

Table 3. Comparing result table between Binary Search algorithm and Interpolation Search

Dataset	Binary Search		Interpolation Search		Comparing result		Percentage Gap	
	Loop count	Average Exe Time (in Second)	Loop count	Average Exe Time (in Second)	Loop Count	Average Exe Time (in Second)	Loop Count	Average Exe Time (in Second)
1000	10	1.685	4	1.655	6	0.03	40.00%	3.00%
2000	11	1.703	4	1.764	7	0.06	36.36%	6.10%
5000	12	1.728	4	2.070	8	0.34	33.33%	34.20%
Average							36.57%	12.43%

5. CONCLUSION

From the test results between the binary search and interpolation search algorithms in finding identity numbers on identity cards in Indonesian format, with test sample data, namely 1000, 2000 and 5000 data, using the C programming language, and with the conditions the dataset being tested has been sorted previously and data has been taken from one area. So when in this study, an experiment was carried out to search for values in the dataset, the results obtained from the comparison of the number of loops carried out between the binary search and interpolation search algorithms had a difference of 36.57%, with less interpolation search in carrying out repetition comparisons in the dataset used. But at the execution time between binary search and interpolation search, binary search is 12.43% faster than interpolation search. From these results, it can be concluded that if the dataset is large enough, such as identity card data on Indonesian citizensand if the computer specifications are good enough, then binary search should be used which has a faster in execution time, but if the computer specifications are not too good, it's better using interpolation search which has a smaller number of loop processes.

ACKNOWLEDGEMENTS

Thank you to the Universitas Multimedia Nusantara, Indonesia which has become a place for researchers to develop this journal research. Hopefully, this research can make a major contribution to the advancement of technology in Indonesia.

REFERENCES

1. X. Song, Y. Cong, Y. Song, Y. Chen, and P. Liang, "A bearing fault diagnosis model based on CNN with wide convolution kernels," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 8, pp. 4041–4056, 2022, doi: 10.1007/s12652-021-03177-x.
2. Bps.go.id, "Jumlah Penduduk Pertengahan Tahun (Ribu Jiwa), 2020-2022," *https://www.bps.go.id/*, 2022. https://www.bps.go.id/indicator/12/1975/1/jumlah-penduduk-pertengahan-tahun.html (accessed Mar. 03, 2023).
3. L. Cameron, D. C. Suarez, and K. Cornwell, "Understanding the determinants of maternal mortality: An observational study using the Indonesian Population Census," *PLoS ONE*, vol. 14, no. 6, pp. 1–18, 2019, doi: 10.1371/journal.pone.0217386.
4. Vanessa Ardelia Layustira and W. Istiono, "Comparative Analysis of Brute Force and

Boyer Moore Algorithms in Word Suggestion Search," *International Journal of Emerging Trends in Engineering Research*, vol. 9, no. 8, pp. 1064–1068, 2021, doi: 10.30534/ijeter/2021/05982021.
5. F. S. Gharehchopogh, M. Namazi, L. Ebrahimi, and B. Abdollahzadeh, "Advances in Sparrow Search Algorithm: A Comprehensive Survey," *Archives of Computational Methods in Engineering*, vol. 30, no. 1, pp. 427–455, 2023, doi: 10.1007/s11831-022-09804-w.
6. G. Xiong, X. Yuan, A. W. Mohamed, J. Chen, and J. Zhang, "Improved binary gaining-sharing knowledge-based algorithm with mutation for fault section location in distribution networks," *Journal of Computational Design and Engineering*, vol. 9, no. 2, pp. 393–405, 2022, doi: 10.1093/jcde/qwac007.
7. M. Tubishat, N. Idris, L. Shuib, M. A. M. Abushariah, and S. Mirjalili, "Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection," *Expert Systems with Applications*, vol. 145, p. 113122, 2020, doi: 10.1016/j.eswa.2019.113122.
8. C. KOMALASARI and W. ISTIONO, "A Comparative Study of Cocktail Sort and Insertion Sort," *Journal of Applied Computer Science & Mathematics*, vol. 15, no. 1, pp. 21–25, 2021, doi: 10.4316/jacsm.202101003.
9. L. Zhou, X. Zhou, and C. Yi, "A Hybrid STA Based on Nelder–Mead Simplex Search and Quadratic Interpolation," *Electronics (Switzerland)*, vol. 12, no. 4, 2023, doi: 10.3390/electronics12040994.
10. S. Orhani, "Binary search algorithm for natural number targets," *Journal Basic Science and Technology*, vol. 11, no. 3, pp. 88–92, 2022.
11. Z. Liu and H. Han, "A fast interpolation-based multi-objective evolutionary algorithm for large-scale multi- objective optimization problems," *Springer Nature*, 2023.
12. P. Pappula, "A Novel Binary Search Tree Method to Find an Item Using Scaling," *International Arab Journal of Information Technology*, vol. 19, no. 5, pp. 713–720, 2022, doi: 10.34028/iajit/19/5/2.
13. Y. Feng and G. G. Wang, "A binary moth search algorithm based on self-learning for multidimensional knapsack problems," *Future Generation Computer Systems*, vol. 126, pp. 48–64, 2022, doi: 10.1016/j.future.2021.07.033.
14. M. H. O. Rashid and A. Imtiaz, "Paw Search-A Searching Approach for Unsorted Data Combining with Binary Search and Merge Sort Algorithm Paw Search Algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 2, pp. 227–236, 2023, doi: 10.14569/IJACSA.2023.0140228.
15. P. Van Sandt, Y. Chronis, and J. M. Patel, "Efficiently searching in-memory sorted arrays: Revenge of the interpolation search?," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 36–53, 2019, doi: 10.1145/3299869.3300075.
16. A. Baviskar, A. D. Hansen, K. Das, and F. U. Nazir, "Minmax-optimal list searching with 0(log2 log2 n) average cost," 2022.
17. R. Rahim, S. Nurarif, M. Ramadhan, S. Aisyah, and W. Purba, "Comparison Searching Process of Linear, Binary and Interpolation Algorithm," *Journal of Physics: Conference Series*, vol. 930, no. 1, 2017, doi: 10.1088/1742-6596/930/1/012007.
18. M. Karane and A. Panteleev, "Hybrid multi-agent optimization method of interpolation search," *AIP Conference Proceedings*, vol. 2181, no. November, 2019, doi: 10.1063/1.5135688.
19. T. Brown, A. Prokopec, and D. Alistarh, "Non-blocking interpolation search trees with doubly-logarithmic running time," *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPOPP*, pp. 276–291, 2020, doi: 10.1145/3332466.3374542.
20. A. Kaporis, C. Makris, S. Sioutas, A. Tsakalidis, K. Tsichlas, and C. Zaroliagis, "Dynamic Interpolation Search revisited," *Information and Computation*, vol. 270, p.

268 104465, 2020, doi: 10.1016/j.ic.2019.104465.