Implementation of Sorting Algorithm in Marketplace to Sort Price Using Insertion Sort

Bilal Abdul Qowiy 1*, Amanda Muchsin Chalik 2, Faiz Hanafi 3, Andika Sundawijaya 4

Program Studi Ilmu Komputer STIMIK ESQ Jakarta, Kampus Menara 165, Lantai 18 & 19, Jl. Tb. Simatupang Kav. 1, Jakarta, Indonesia

Abstract. Price is the first thing that customers look at before they buy something. So, sorting is the most important feature that every single person needs when we want to shop on the online marketplace or we can call it e-commerce. In this case we would focus on the price, because price is the most important thing that customers consider before they buy something. However, customers see each other too closely but price is the most important thing. Sort Algorithm is not just about Insertion Sort, there are many of them, like bubble sort, Quicksort, mergesort, etc. but we use insertion sort based on the runtime complexities and find the fastest time to reach the final results. And our implementation in the website base, it is common to use and implement on that scope.

Keywords: Algorithm; E-Commerce; Insertion Sort; Sorting, Website.

1. Introduction

Sorting is a common thing that humans do today, if you ever imagine how machines do that. Sorting some stuff based on the values, shape, or maybe color of an object that we want to sort. Machines are exactly the same as humans do, but faster and with more ways to do it. When someone or machine wants to do something, there must be some step that someone has to go through, the steps that we call an algorithm. An algorithm is a finite sequence of well-defined instructions, typically used to solve a class of specific problems or to perform a computation (Saptadi & Sari, 2012).

Algorithm is so many option to choose, what is the best algorithm that we gonna implement in our system, it's about sorting algorithm. In computer science, a sorting algorithm is an algorithm that puts elements of a list into an order. The most frequently used orders are numerical order and lexicographical order, and either ascending or descending. Efficient sorting is important for optimizing the efficiency of other algorithms (such as search and merge algorithms) that require input data to be in sorted lists. Sorting is also often useful for canonicalizing data and for producing human-readable output.

A sorting algorithm is amede used in classifying a data, one of the most widely used sorting methods is insertion sorting. The insertion sorting method is also the simplest sorting method of various models and sorts of sorting that exist. Algorithm sorting insertion sorting has similarities with bubble sorting algorithms, but has a significant difference if faced with many data models. If fulfilled then the left will be locked as the right array, this insertion sort algorithm has a complexity as large as $O(n^2)$.

2. Methods

In this research method, to sort product price data on a website from the highest price to the highest price or vice versa using a data sorting algorithm with the insertion sort
method. This website is divided into two stages, the first filtering data, which is grouping product data according to the type of product. The Second stage of the program can also sort products from the highest price to the lowest price and vice versa.

For data that wants to be filtered and sorted directly entered into the program manually, this is because the author does not use the database in testing filters and sorting product data. Based on how to obtain data, using primary data because the data is directly collected by the researcher from the product that wants to be filtered and sorted. Based on the nature of the data, using quantitative data because the exact number data is using the price that has been determined on each produk. Based on the time of collection of data, use cross section / incidental because the data is collected at a certain time.

The next stage includes developing and testing with the same data on the localhost website using html asit looks and javascript as a programming language against filters and sorting

2.1. Filtering

Process filtering is the sorting of data in the form of a category or group or uniform data set, in filtering there is usually data yang is the same, shaped and oriented. In this case filtering is grouping of items available on the marketplace website according to the available categories.

2.2. Sorting

In the sorting stage, the data that has been collected can be sorted, thus making it easier for website users to choose products at prices that have been neatly arranged.

In general, there are 2 types of data sorting:

2.2.1. Ascending

It is a process done in sorting to calculate the value of data in the form of Char, String, and Numeric arranged from the smallest value to the largest value in a given model. For example, if there is random data [3, 4, 2, 7, 1, 8] then in Ascending will be sorted into [1, 2, 3, 4, 7, 8] by sorting the smallest value towards the largest value. Conversely, if there is data in the form of strings that are not numerical data then it is sorted in alphabetical order. For example, researchers try to sort shoe type products at varying prices, the data is sorted with price details from cheapest to most expensive, using the following program code that can be seen at Figure 1.

![Figure 1 Program code to sorting ascending](image)

2.2.2. Descending

Descending is the opposite of the model used in the ascending process, which is by sorting data from the largest value into the smallest value according to the type and model
of the data inputted, for example if there is data [3, 4, 2, 7, 1, 8] then the result obtained if using the function of the model descending is [8, 7, 4, 3, 2, 1]. Program code in Figure 2.

As an example of research in this case, researchers will conduct a trial of products of shoe type to be sorted by price, from the highest price to the lowest price. Safter being sorted using source code above hence the results as in Figure 3.

![Figure 2 Program code to sorting descending](image)

![Figure 3 Product item after sorting](image)

Shoe-type product data has been sorted from the highest price to the lowest price.

3. Results and Discussion

3.1. Table and Picture

In the time complexity notation on the sort sorting insertion algorithm, it can be said that the best condition if the value of a data is as large as $O(n)$, at this time it means that the sequence in an array is appropriate and there is no insertion or transfer of data on the array. This is when a data is not fully organized, there will be a data check on each node that produces the complexity of time $O(n^2)$. In the Table 1 can be clearly seen the complexity of the time required insertion sort algorithm in certain cases.
### Table 1 Complexity of time

<table>
<thead>
<tr>
<th>Insertion Sort</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Worst time complexity</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Average time complexity</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Best time complexity</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Space complexity</td>
<td>$O(1)$</td>
</tr>
</tbody>
</table>

#### 3.2. Mathematical Notation

Complexity insertion sort, the best condition (best case) is achieved if the data has been sorted. Only one perbandingan is done for each position $i$, so there is $n - 1$ ratio, or $O(n)$. The worst case is achieved if the data has been sequenced but in reverse order. In this case, for each $i$, the data element[i] is smaller than the elemen data[0], ..., data[i-1], each of the elements is moved one position. For each iteration $i$ in the outermost kalang, there is always a comparison of $i$, so the total number of comparisons for all iterations in this kalang is:

$$T(n) = 1 + 2 + ... + n - 1 = \sum_{i=1}^{n} i = \frac{n(n - 1)}{2} = o(n^2)$$

Insertion Sort Analysis For the best case this algorithm runs 1 time, that is, if the elements in the table have been sorted. Kalang (loop) while never run. For the worst case this algorithm runs $N$ max times. times faster and more efficiently compared to bubble sequencing. However, this algorithm remains less efficient for large tables (storing many values). With data intervals between 100 to 1,000 elements. Execution time is measured by the second(s) unit.

#### 4. Conclusions

From the implementation results on category filtering trial programs and price sorting on ecommerce websites it can be concluded that the program can run well, the program can recognize objects according to the category filter, and also in sorting the appropriate results menu price. but this program has obstacles if faced with two or more variables of the same value to insert the corresponding position in order.

In the creation of programs built for sorting trials using insertion sorts need to be further developed on the focus of possible areas that occur in other programs, with this then the insertion sort needs to be tested again on the number with real case studies.

#### References


