Implementation of the Insertion Sort Algorithm to Sort Positive Integers in Ascending Order Using Flowgorithm

Zulham Sitorus[1], Dhimas Prayogi[2], M. Azhari Rizko[3], Ade Guna Suteja[4], Muhammad Raihan Harahap[5]
[1,2,3,4,5]Master of Information Technology, Universitas Pembangunan Panca Budi

ABSTRACT

The advancement of information technology has had a significant impact on various fields, including education. One area that has been greatly influenced is the teaching of programming algorithms, where technology helps simplify the understanding of fundamental concepts, such as data sorting algorithms. This study focuses on the implementation of the Insertion Sort algorithm using the Flowgorithm application to facilitate the understanding of the data sorting process. Flowgorithm is a software tool that enables the creation of flowcharts to visually represent the steps of an algorithm, which can then be translated into programming languages. The Insertion Sort algorithm was chosen due to its simplicity and its effectiveness in sorting small datasets. This study covers the design and implementation of the algorithm using Flowgorithm, testing with positive integer data, and verifying the correctness of the sorting results through manual calculations. The test results show that the algorithm works correctly, producing the proper sequence from unsorted data. Additionally, the pseudocode generated from the Flowgorithm design can be translated into a programming language like Python. This research contributes to enhancing the understanding of algorithmic concepts, particularly sorting algorithms, by using Flowgorithm as an effective learning tool.

Keyword : Algorithm; Insertion Sort; Flowgorithm.

Corresponding Author:
Zulham Sitorus,
Master of Information Technology
Universitas Pembangunan Panca Budi
Jl. Jend. Gatot Subroto Km. 4,5 Sei Sikambing 20122, Medan, Indonesia.
Email : zulhamsitorus@dosen.pancabudi.ac.id

Article history:
Received Oct 29, 2024
Revised Oct 30, 2024
Accepted Oct 31, 2024

1. INTRODUCTION

The rapid advancements in technology and science are inevitable, as technology continues to evolve across various fields of knowledge. One area particularly impacted is education, where technology facilitates learning, including in understanding programming algorithm concepts. Although algorithms are typically studied in computer science programs, their understanding is also crucial in other disciplines (Mutaqin et al., 2022) (Sihombing, 2023) (Cindy Kawilda Hasibuan & Yahfizham Yahfizham, 2023). An algorithm is a systematic set of steps used to solve a problem, executed by either a computer or a human. Therefore, well-designed algorithms are essential, as errors in algorithm creation can lead to incorrect outputs (Amalia et al., 2023).

One effective way to represent algorithms is through the use of flowcharts. A flowchart is a diagram that visually and systematically illustrates the sequence of steps involved in problem-solving (Damasta et al., 2023). Learning algorithms with flowcharts can aid students' understanding by clearly demonstrating how algorithms function (Melinda Azizah & Yahfizham Yahfizham, 2023). For drawing flowcharts, Flowgorithm is a popular application. It uses standard symbols, making it easier for users to depict algorithms that can later be translated into programming (Hisamuddin & Siregar, 2024).

In the context of computer applications, one of the most common processes is data sorting. Sorting is the process of arranging initially unordered data into a more structured sequence based on specific rules. One popular algorithm for sorting data is Insertion Sort, which is known for its simplicity in implementation (Nasution et al., 2023) (Yusuf Asyhari et al., 2023) (Rahman Hakim et al., 2024) .This algorithm is suitable for sorting small datasets or for understanding the basics of sorting algorithms.

However, when creating flowcharts and implementing algorithms, questions often arise about the correctness and accuracy of the results produced. Is the flowchart correct, and does the output match the expected result? To address these issues, software like Flowgorithm is very useful, as it aids in visualizing algorithms and ensuring that the results are accurate (Smrti et al., 2023). This study aims to implement the Insertion Sort algorithm using Flowgorithm to sort positive integers in ascending order, with the goal of providing a better understanding of the sorting process in algorithm education.

2. RESEARCH METHOD

1. Design and Implementation, The author will design the Insertion Sort algorithm and implement it using the Flowgorithm software. This process includes creating a flowchart with standard symbols to detail the steps involved in the Insertion Sort algorithm.
2. Testing, The author will test and verify the Insertion Sort algorithm implemented in Flowgorithm using a set of positive integer data: 42, 17, 9, 58, 31, 5, 63, 22, 47, and 11, to ensure that the algorithm can correctly sort the numbers in ascending order. The author will also perform a correctness test by comparing the sorting results produced by Flowgorithm with manually calculated sorting results to verify the algorithm's accuracy.
3. Pseudocode, Pseudocode is an algorithm presentation that resembles the syntax of a programming language (Yahfizham, 2023). Python is one of the most widely used programming languages by large companies and developers for creating various types of desktop, web, and mobile applications (Khairudin, 2024). The author will generate Python pseudocode based on the design and implementation of the Insertion Sort algorithm using the Flowgorithm application. The author will also analyze the flow of the Insertion Sort algorithm derived from the generated pseudocode to ensure that each step is correct and that it produces the expected output.

3. RESULTS AND DISCUSSION

A. Design and Implementation of the Insertion Sort Algorithm

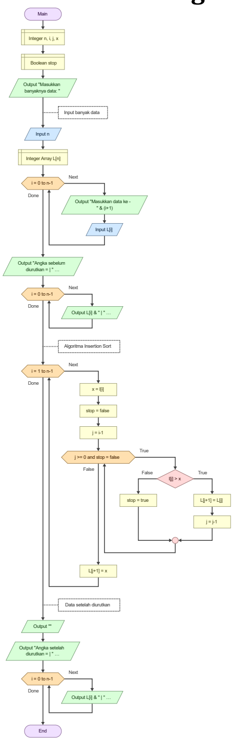
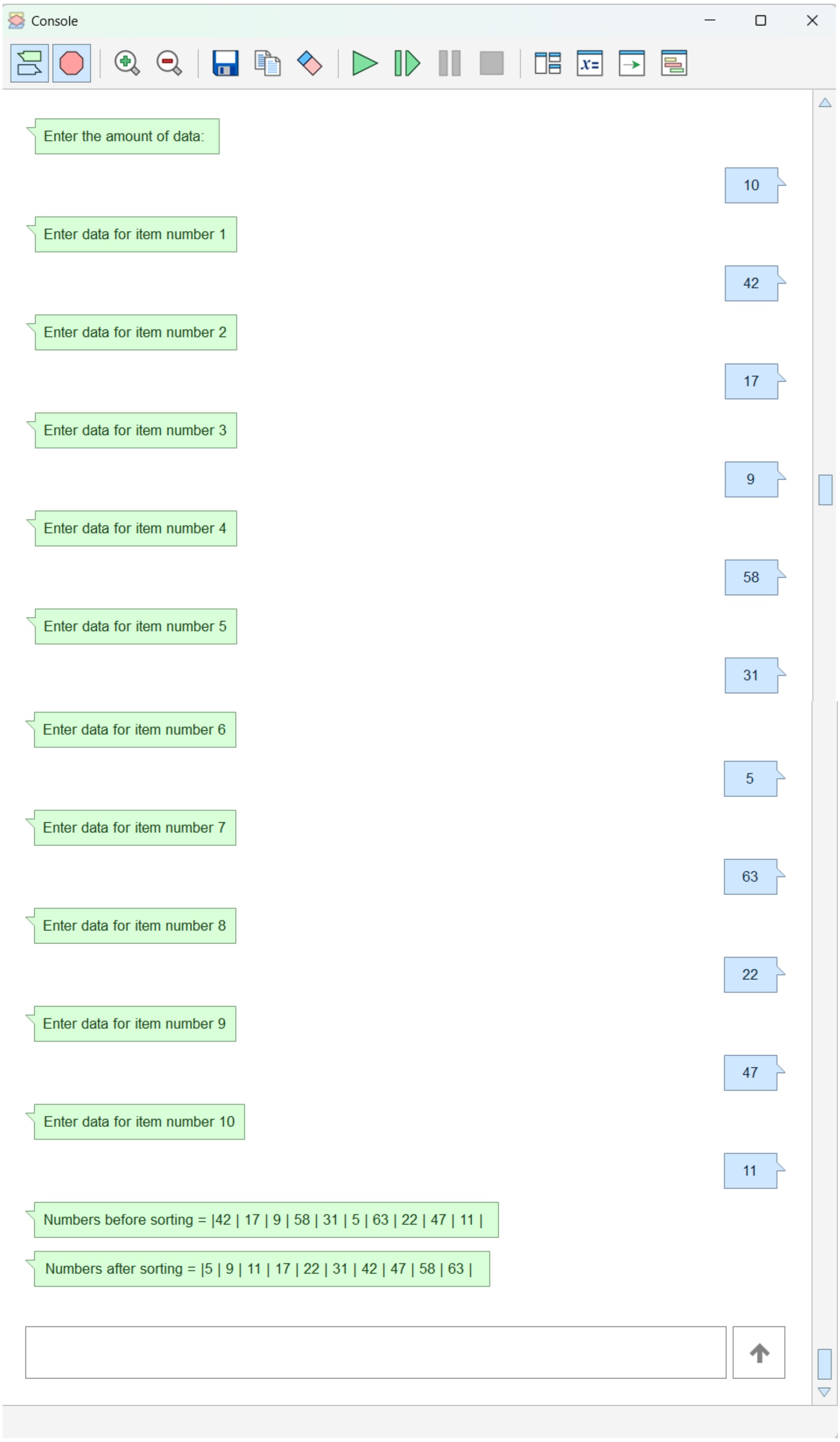


Fig 1. Flowchart Design of the Insertion Sort Algorithm

B. Testing the Insertion Sort Algorithm



```
Enter the amount of data:
10
Enter data for item number 1
42
Enter data for item number 2
17
Enter data for item number 3
9
Enter data for item number 4
58
Enter data for item number 5
31
Enter data for item number 6
5
Enter data for item number 7
63
Enter data for item number 8
22
Enter data for item number 9
47
Enter data for item number 10
11
Numbers before sorting = |42 | 17 | 9 | 58 | 31 | 5 | 63 | 22 | 47 | 11 |
Numbers after sorting = |5 | 9 | 11 | 17 | 22 | 31 | 42 | 47 | 58 | 63 |
```

Fig 2. Testing the Insertion Sort Algorithm Using Flowgorithm

In Figure 2, it can be seen that the testing of the Insertion Sort algorithm produces a sorted sequence, where the numbers before sorting were 42, 17, 9, 58, 31, 5, 63, 22, 47, 11, and after sorting, the numbers become 5, 9, 11, 17, 22, 31, 42, 47, 58, and 63.

Table 1. Testing the Insertion Sort Algorithm Using Manual Method

Step	Array After Process	Brief Explanation
0	[42, 17, 9, 58, 31, 5, 63, 22, 47, 11]	The numbers before sorting.
1	[17, 42, 9, 58, 31, 5, 63, 22, 47, 11]	17 is compared with 42, and 17 is moved to the front.
2	[9, 17, 42, 58, 31, 5, 63, 22, 47, 11]	9 is compared with 17 and 42, and 9 is moved to the front.
3	[9, 17, 42, 58, 31, 5, 63, 22, 47, 11]	58 is already greater than 42, so no changes are made.
4	[9, 17, 31, 42, 58, 5, 63, 22, 47, 11]	31 is compared with 58 and 42, and it is moved after 17.
5	[5, 9, 17, 31, 42, 58, 63, 22, 47, 11]	5 is compared with 9, 17, 31, and 42, and is moved to the front.
6	[5, 9, 17, 31, 42, 58, 63, 22, 47, 11]	63 is greater than 58, so no changes are made.
7	[5, 9, 17, 22, 31, 42, 58, 63, 47, 11]	22 is compared with 63, 58, 42, and 31, and is moved after 17.
8	[5, 9, 17, 22, 31, 42, 47, 58, 63, 11]	47 is compared with 63, 58, and 42, and is moved after 42.
9	[5, 9, 11, 17, 22, 31, 42, 47, 58, 63]	11 is compared with 63, 58, 47, and 42, and is moved after 9.

In Table 1, it can be seen that the data before sorting was 42, 17, 9, 58, 31, 5, 63, 22, 47, 11, and after sorting, the numbers became 5, 9, 11, 17, 22, 31, 42, 47, 58, 63. The results of the manual testing are consistent with the results obtained through testing using Flowgorithm.

B. Pseudocode

Table 2. Pseudocode Algoritma Insertion Sort

Line	Pseudocode	Description
0	print("Enter the amount of data: ")	Displays a prompt asking the user to input the amount of data they want to enter.
1		Blank line for clarity in the structure.
2	# Input a large amount of data	Comment: This section is used to take input data from the user.
3	n = int(input())	Takes the number of data items (n) from the user and converts it to an integer.
4	l = [0] * (n)	Creates a list l of size n with all elements initialized to 0.
5		Blank line for clarity in the structure.
6	for i in range(0, n - 1 + 1, 1):	A loop to iterate n times, asking the user to input data for each item.
7	print("Enter data for item number " + str(i + 1))	Displays a prompt asking the user to enter the data for the i+1-th item.
8	l[i] = int(input())	Takes input from the user and stores it in the i-th index of list l.
9	print("Numbers before sorting = |", end='', flush=True)	", end='', flush=True)`
10	for i in range(0, n - 1 + 1, 1):	Loop to print all elements of the list l before sorting.
11	print(str(l[i]) + " | ", end='', flush=True)	", end='', flush=True)`
12		Blank line for clarity in the structure.
13	# Insertion Sort Algorithm	Comment: This section implements the Insertion Sort algorithm to sort the data.
14	for i in range(1, n - 1 + 1, 1):	A loop starting from i = 1 to i = n-1, which performs sorting using the Insertion Sort method.
15	x = l[i]	Stores the value at l[i] into variable x, which will be placed in the correct position in the list.
16	stop = False	A flag indicating that the comparison process has not finished yet.
17	j = i - 1	Initializes index j to one position before i, for comparing with previous elements.
18	while j >= 0 and stop == False:	A condition to loop while j is within bounds of the list and the process is not stopped.
19	if l[j] > x:	If the element at index j is greater than x, shift the element to the next position.
20	l[j + 1] = l[j]	Shifts the element l[j] to the next position, l[j + 1].
21	j = j - 1	Decrements j to compare with the previous element.
22	else:	If l[j] is not greater than x, it means the list is in order, and the process can stop.
23	stop = True	Stops the while loop as the element is now in the correct position.
24	l[j + 1] = x	Places x in the correct position in the list after the larger elements.
25		Blank line for clarity in the structure.
26	# Data after sorting	Comment: This section displays the data after sorting.

27	print("")	Prints an empty line for cleaner output formatting.
28	print("Numbers after sorting = |", end='', flush=True)	", end='', flush=True)`
29	for i in range(0, n - 1 + 1, 1):	Loop to print all elements of the list l after sorting.

In Table 2, the pseudocode implements the Insertion Sort algorithm to sort data entered by the user. The process begins by asking the user to input the number of data items to be processed (lines 0 to 3). Afterward, the program creates a list l with a length of n and initializes all its elements to 0 (line 4). The program then prompts the user to input n data values (lines 6 to 8), storing them in the list l at the appropriate indices. Once the data has been entered, the program displays all the input data before sorting (lines 9 to 11), showing the data on a single line with spaces between elements. Next, the Insertion Sort algorithm is applied to sort the data (lines 13 to 23), starting by selecting the second element (index 1), comparing it with previous elements, and moving it to the correct position by shifting larger elements to the right. This process repeats until all the data is sorted. Finally, after sorting is complete, the program displays the sorted data in the correct order (lines 24 to 29).

4. CONCLUSION

This study demonstrates that the use of the Flowgorithm application to illustrate and implement the Insertion Sort algorithm helps simplify the understanding of programming algorithm concepts, particularly in data sorting. The clearly designed flowchart visualizes the steps of the algorithm, making it easier to grasp the concept of Insertion Sort. Testing with positive integer data yields results consistent with manual calculations, confirming that the algorithm works correctly. Moreover, the pseudocode generated by Flowgorithm can be easily adapted into a programming language such as Python. Overall, this study contributes to enhancing the understanding of sorting algorithms by using Flowgorithm as an effective learning tool.

REFERENCES

Amalia, R., Assani', S., & Effindi, M. A. (2023). Rancang Bangun Media Pembelajaran Algoritma Perograman Berbasis Android. *Jurnal Ilmiah Edutic : Pendidikan Dan Informatika*, *9*(2), 188–200. https://doi.org/10.21107/edutic.v9i2.20215

Cindy Kawilda Hasibuan, & Yahfizham Yahfizham. (2023). Analisis Pembelajaran Algoritma Pemrograman. *Jurnal Arjuna : Publikasi Ilmu Pendidikan, Bahasa Dan Matematika*, *1*(5), 274–285. https://doi.org/10.61132/arjuna.v1i5.337

Damasta, I. R., Rosid, A., Eviyanti, A., & Indahyanti, U. (2023). Implementasi Online Marketplace pada Industri Rumahan di Desa Bluru Permai Sidoarjo. *Innovative Technologica: Methodical Research Journal*, *2*(4), 1–13. https://doi.org/10.47134/innovative.v2i4

Hisamuddin, M. Z., & Siregar, M. U. (2024). Evaluasi Penggunaan Flowgorithm dalam Pembelajaran Algoritma Pemrograman menggunakan Technology Acceptance Model (TAM). *Edumatic: Jurnal Pendidikan Informatika*, *8*(1), 84–92. https://doi.org/10.29408/edumatic.v8i1.25413

Khairudin. (2024). PENGENALAN BAHASA PEMROGRAMAN PYHTON UNTUK MENINGKATKAN KOMPETENSI YATIM PIATU RW 01 KELURAHAN PANUNGGAN KEC PINANG. *Praxis: Jurnal Pengabdian Kepada Masyarakat*, *4*(3), 36–43.

Melinda Azizah, & Yahfizham Yahfizham. (2023). Pengembangan Sistem Kuis Alogoritma Pemprograman Berbasis Web. *Jurnal Arjuna : Publikasi Ilmu Pendidikan, Bahasa Dan Matematika*, *1*(5), 286–300. https://doi.org/10.61132/arjuna.v1i5.338

Mutaqin, R., Shinta Dharmopadni, D., Mutaqin, G., Gadjah Mada, U., Besar TNI, M. A., & Maulana Malik Ibrahim, U. (2022). Dampak Perkembangan Teknologi Informasi Dan Komunikasi Terhadap Dinas Militer. *Jurnal Ilmiah Multidisiplin*, *2*(3), 199–204. https://jurnal.alungcipta.com/index.php/JIM

Nasution, R., Syahputra, A., Widiyanto, A., Subuhanto, D., & Abdillah, A. Y. (2023). Persepsi Mahasiswa Informatika Terhadap Keefektifan Algoritma Bubble Sort dalam Mengurutkan Data. *Blend Sains Jurnal Teknik*, *1*(3), 220–225. https://doi.org/10.56211/blendsains.v1i3.186

Rahman Hakim, F., Syadi Ismallah, H., & Luthfi Firdaus, M. (2024). PENERAPAN ALGORITMA INSERTION SORTING TERHADAP DATA TRANSAKSI SAHAM PER KOTA DI INDONESIA. *Prosiding Seminar Nasional Sains Dan Teknologi Seri 02*, *1*(2), 3047–6569.

Sihombing, S. O. (2023). Transformasi Kepemimpinan Dosen: Menghadapi Tantangan dan Mengoptimalkan Peluang Teknologi dalam Pendidikan. In S. O. Sihombing (Ed.), *Transformasi Kepemimpinan Dosen: Menghadapi Tantangan dan Mengoptimalkan Peluang Teknologi dalam Pendidikan* (pp. 11–21). Institut Informatika dan Bisnis Darmajaya.

Smrti, N. N. E., Andisana, I. P. G. S., Rahayu, N. K. D. T., Adnan, & KW, P. P. O. J. (2023). Flowgorithm Sebagai Penunjang Pembelajaran Algoritma dan Pemrograman. *Jurnal Bangkit Indonesia*, *12*(1).

Yahfizham, H. F. (2023). STUDI KASUS PENERAPAN ALGORITMA DALAM KEHIDUPAN SEHARI-HARI. *AL ITIHADU JURNAL PENDIDIKAN*, *2*(2), 214–226. https://jurnal.asrypersadaquality.com/index.php/alittihadu

Yusuf Asyhari, M., Dewi Hapsari, E., Yulianto, S., & Teknik, F. (2023). Pendekatan Metode Kolokasi untuk Text Processing Ulasan Aplikasi Android Surveilans Penyebaran Covid-19 di Indonesia. *NFORMATION SYSTEM DEVELOPMENT*, *8*(3), 33–2. https://doi.org/10.19166/isd.v8i1