CHAPTER 4

ANALYSIS AND DESIGN

4.1 Analysis

Heap Sort algorithm requires binary tree data structure. Student must input data into the program. The type of data must be row of numbers.

Table 4.1: Data Analysis.

Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9
17	15	23	10	16	20	29	6	12

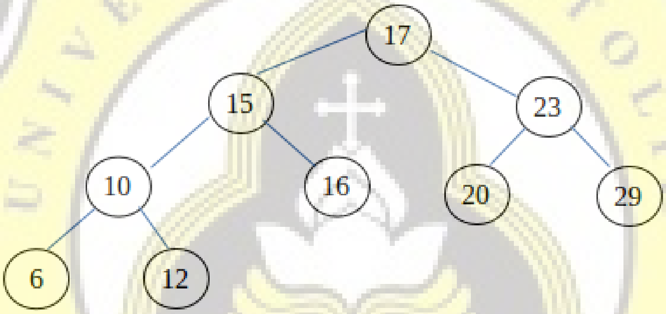

Illustration 4.1: Representation data into tree.

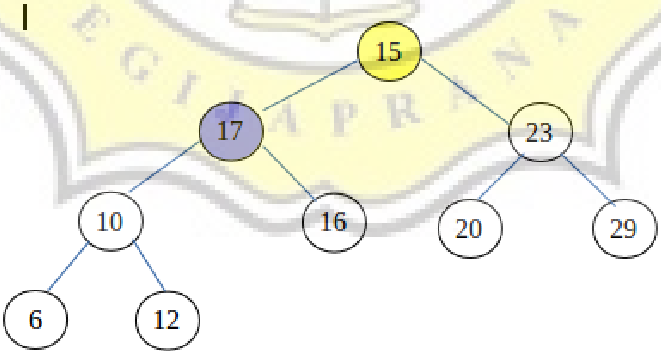

Illustration 4.2: Start comparing each nodes.

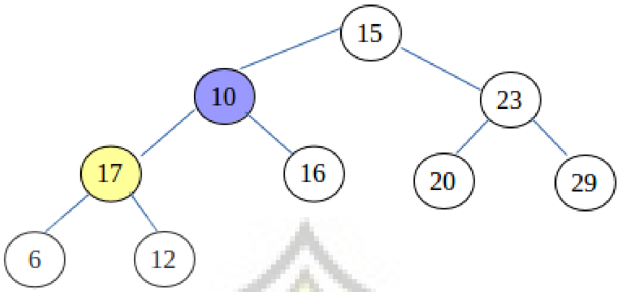

Illustration 4.3: Swapping each nodes.

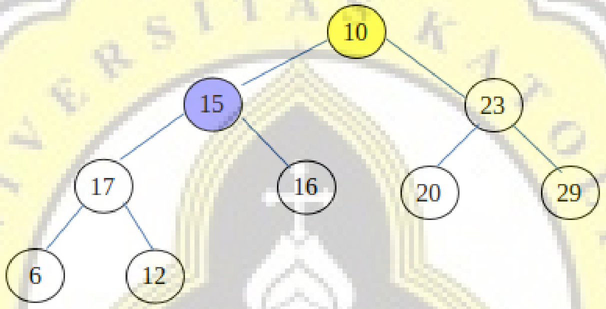

Illustration 4.4: Swapping more nodes.

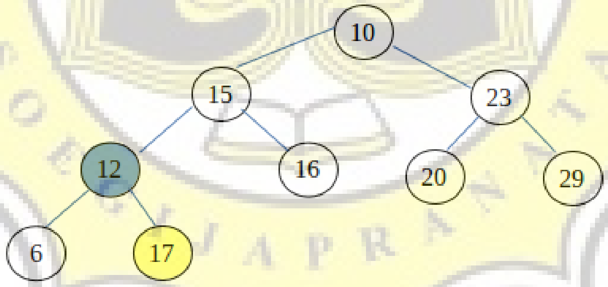

Illustration 4.5: Swapping the bottom nodes.

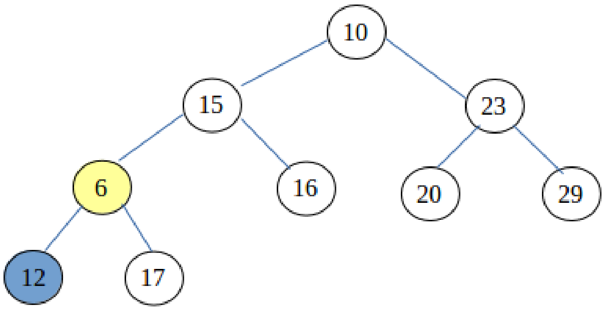

Illustration 4.6: Swapping nodes.

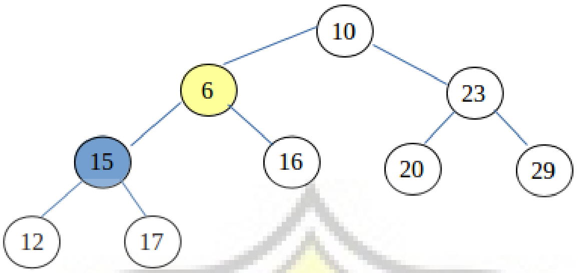

Illustration 4.7: Swapping the rest of the nodes.

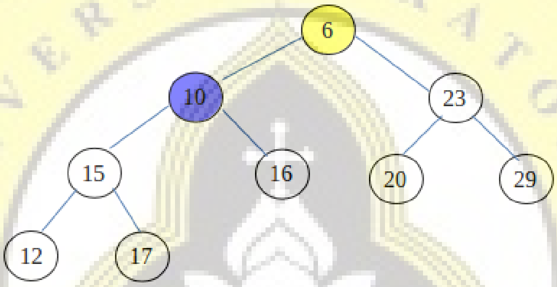

Illustration 4.8: The root has ben set to minimum value.

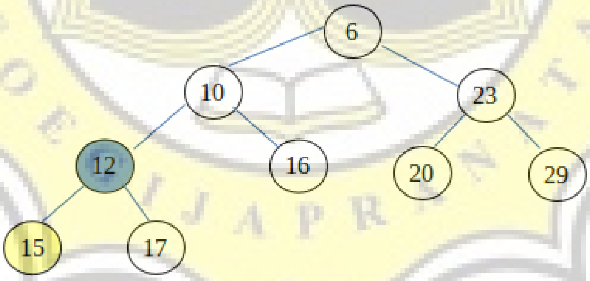

Illustration 4.9: Swapping last nodes.

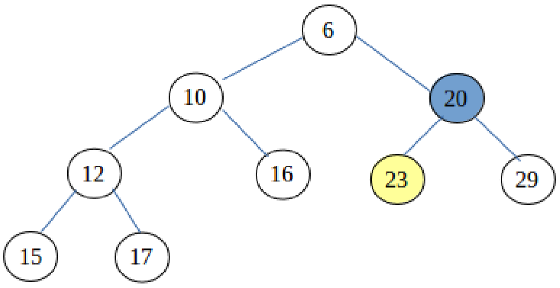

Illustration 4.10: Sorting is done after all nodes been swapped.

The multiheaps algorithm begins when the single heap procedure is done. The heap separated into two heaps which its node has just been removed. Then both heaps start sorting the data at the same time.

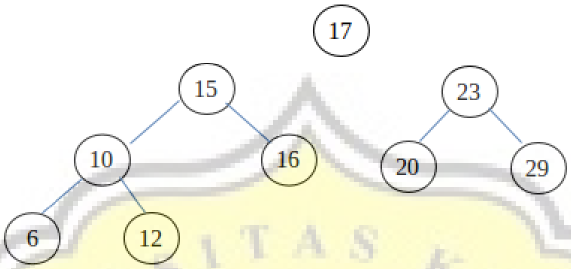

Illustration 4.11: Multi heaps process start.

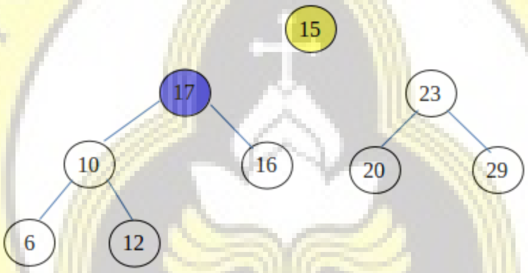

Illustration 4.12: Tree split into two sides.

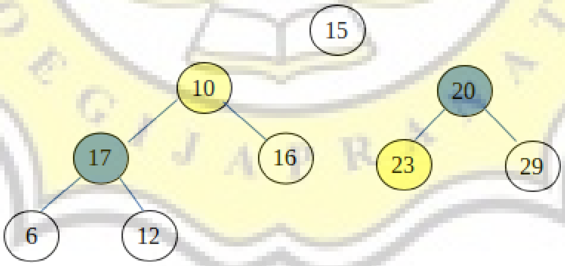

Illustration 4.13: Start the sorting at the same time.

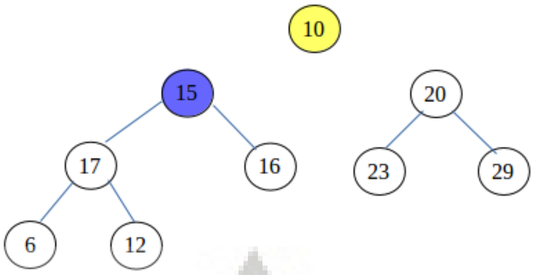

Illustration 4.14: Swapping each nodes.

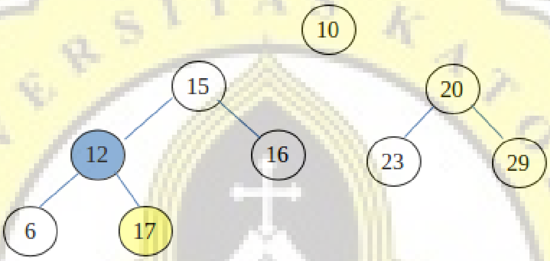

Illustration 4.15: Swapping bottom nodes.

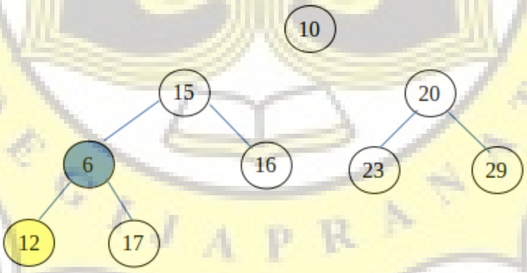

Illustration 4.16: Swapping values 6 and 12.

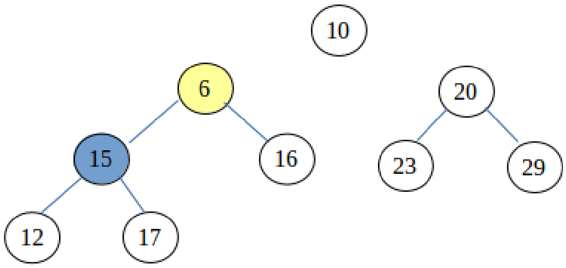

Illustration 4.17: Swapping values 6 and 15.

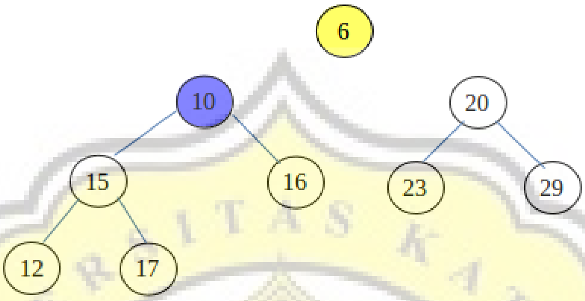

Illustration 4.18: The root is the minimum value.

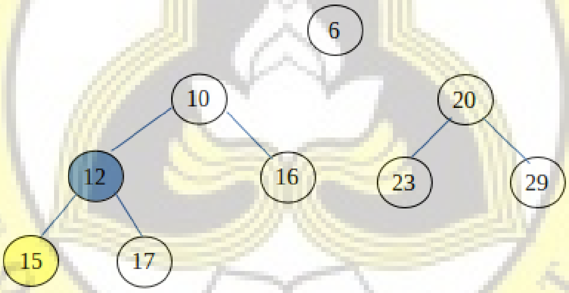

Illustration 4.19: Sorting process is done.

The sorting process using multiple heaps is done. The result is same with single heap. Multiple heaps allow program to sort the data faster and more efficient.

4.2 Design

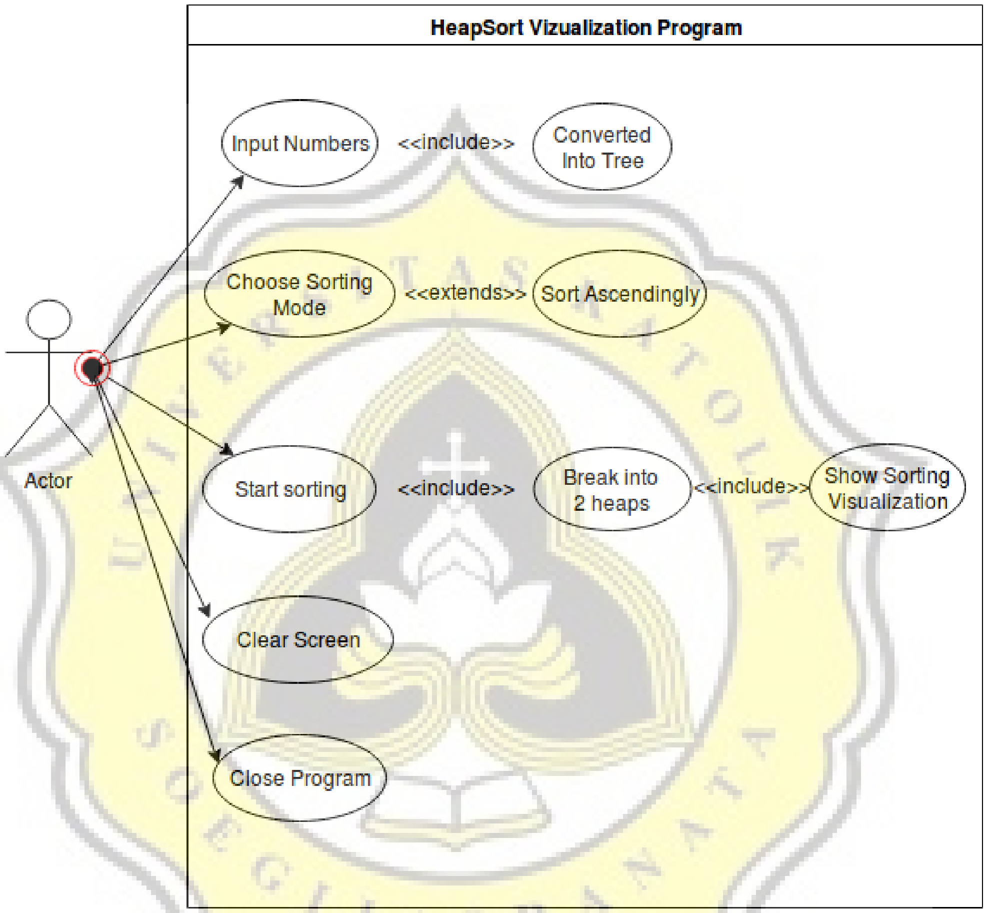

Illustration 4.20: Use case diagram.

Based on the picture above, there are 5 interactions between the user and the program. First, the user must enter rows of numbers in which the numbers will be automatically converted to binary tree by the program. Secondly user chooses the sorting mode then selects the start sorting menu. Rows of numbers that have been inputted will be split into 2 heaps and sorted together. There is also an interaction to clean the program panel and close the program

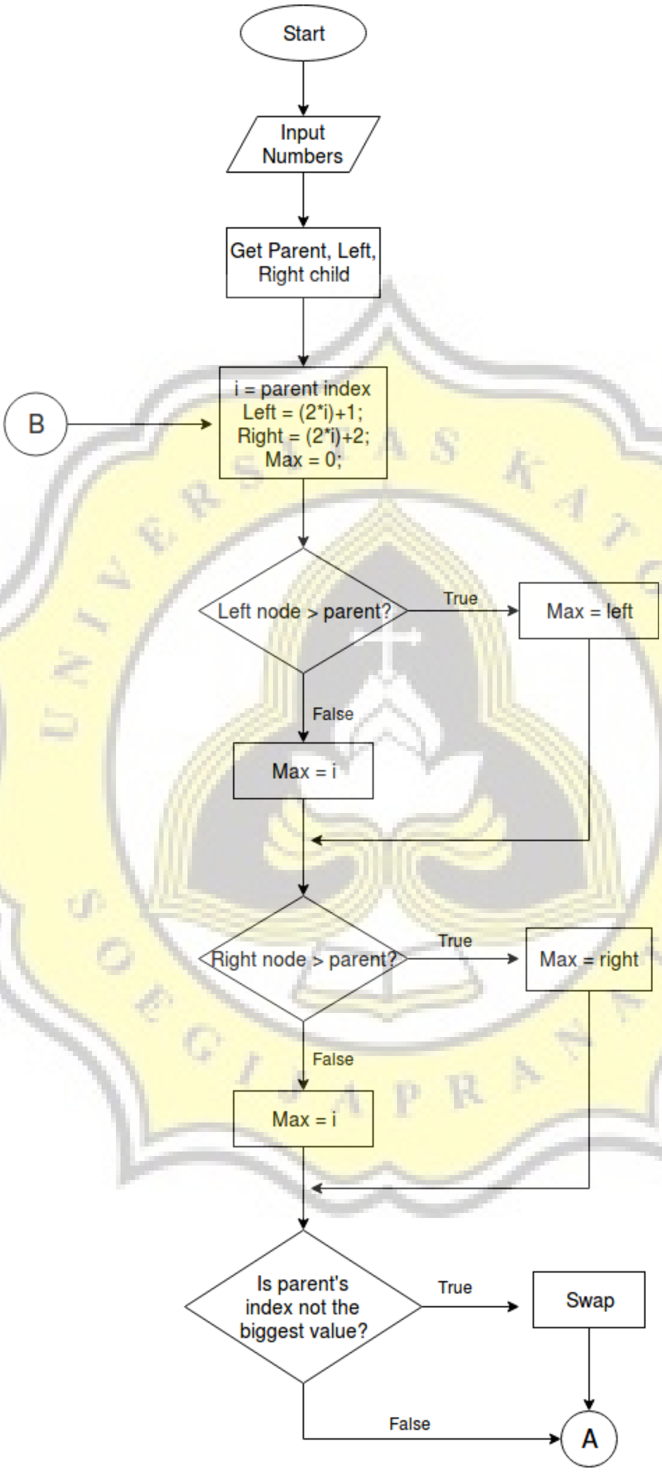

Illustration 4.21: Flowchart 1.

The flowchart starts with user inputting various of numbers, then program represent the numbers into binary tree. From there, the program starts split the tree and comparing each nodes to find minimum value. Root must have the smallest value, if root don’t have minimum value, program will swap its child value which have the smallest value with root’s. Swapping nodes using recursive looping until all parents have the smallest value and the numbers sorted ascendingly.

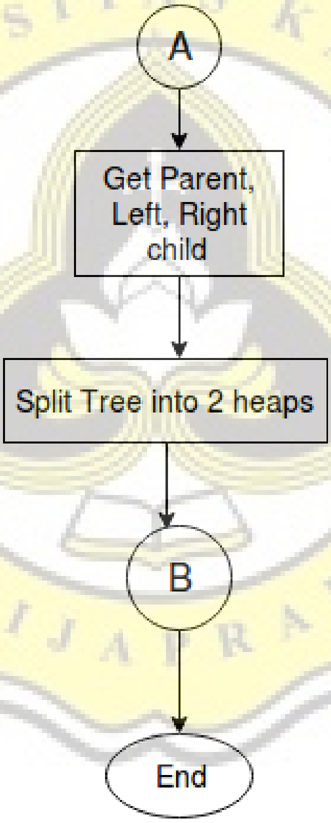

Illustration 4.22: Flowchart 2.