

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation

This program is visualizing the Merge Sort and Quick Sort sorting algorithm. Linked List is used as place for visualization for this program. The Javascript and HTML5 Canvas used for drawing the linked lists. For make the program looks good, CSS is also used in here. This program divided into some functions that used to drawing and visualize the algorithms.

At the first time, the code below is used to read the number that user inputted. It will separate the number based on comma.

```
1. var bacaInput = document.getElementById('angka').value;  
2. var values = bacaInput.split(",");
```

All parameters for drawing square is written in function gambarKotak. This function will recursively called in this program for making the nodes of the linked lists. When making each part of linked list, the coordinates must be resetted at certain points (see Illustration 5.1.1).

```
1. function  
gambarKotak(warna,x,y,sisi,tebal,ukuran_font,posisi,warnatext,angka)  
2.     {  
3.         ctx.font = ukuran_font;  
4.         ctx.fillStyle = warna;  
5.         ctx.fillRect(x,y,sisi,sisi);  
6.         ctx.lineWidth = tebal;  
7.         ctx.fillStyle = warnatext;  
8.         ctx.textBaseline=posisi;  
9.         ctx.fillText(angka,x+9,y+25);  
10.    }
```

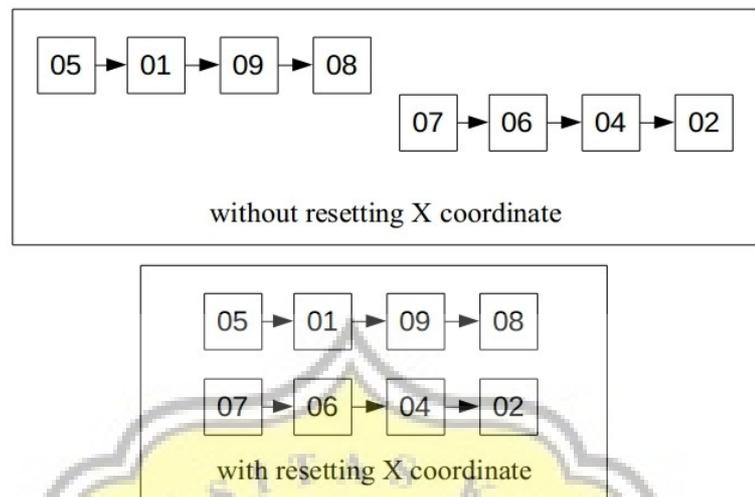


Illustration 5.1.1: X coordinate reset

Function below is for making the arrow line of the linked list. For drawing the arrow and line, it must takes calculations to make the shape of the lines and arrows looks proportional. This calculation can be seen on line 11 and 14.

```

1. function gambarGarisPannah(warna,a,b,c,tebal)
2. {
3.     ctx.beginPath();
4.     ctx.strokeStyle = warna;
5.     ctx.moveTo(a,b);
6.     ctx.lineTo(c,b);
7.     ctx.lineWidth = tebal;
8.     ctx.stroke();
9.     ctx.beginPath();
10.    ctx.fillStyle= warna;
11.    ctx.moveTo(a+50,b-10);
12.    ctx.lineTo(c,b);
13.    ctx.lineTo(c,b);
14.    ctx.lineTo(a+50,b+10);
15.    ctx.fill();
16. }

```

This function is for drawing the explanation text and the number inside of nodes. Same like the other, it must declare the coordinates, color, font size, and the content of the text itself.

```

1. function tulisan(warna,ukuran_font,isiteks,x,y)
2. {
3.     ctx.fillStyle = warna;
4.     ctx.font = ukuran_font;
5.     ctx.fillText(isiteks,x,y);
6. }

```

5.2 Testing



Picture 5.2.1: First interface of the program

Testing is ran by using Firefox Quantum web browser to see the visualization that is drawn in the canvas. The interface of the program is simple to make it easier for users to learning the sorting algorithms. There is a text field, merge and quick sort button and also a help button to show how to use this program.

MERGE SORT & QUICK SORT Visualization in Linked List

05,01,09,10,08 Merge Sort Quick Sort Help

- 1 Merge Sort: use the Divide and Conquer method
- 2 Divide: split the linked list into two parts, left and right. This is done recursively
- 3 Conquer: Sort each part of linked lists
- 4 Combine: Combine all the splitted and sorted parts into one big part of sorted linked list

-- Unsorted Linked Lists --

05 → 01 → 09 → 10 → 08

-- Left Partition --

05 → 01

-- Right Partition --

09 → 10 → 08

01 ← is a comparison result between [05,01]

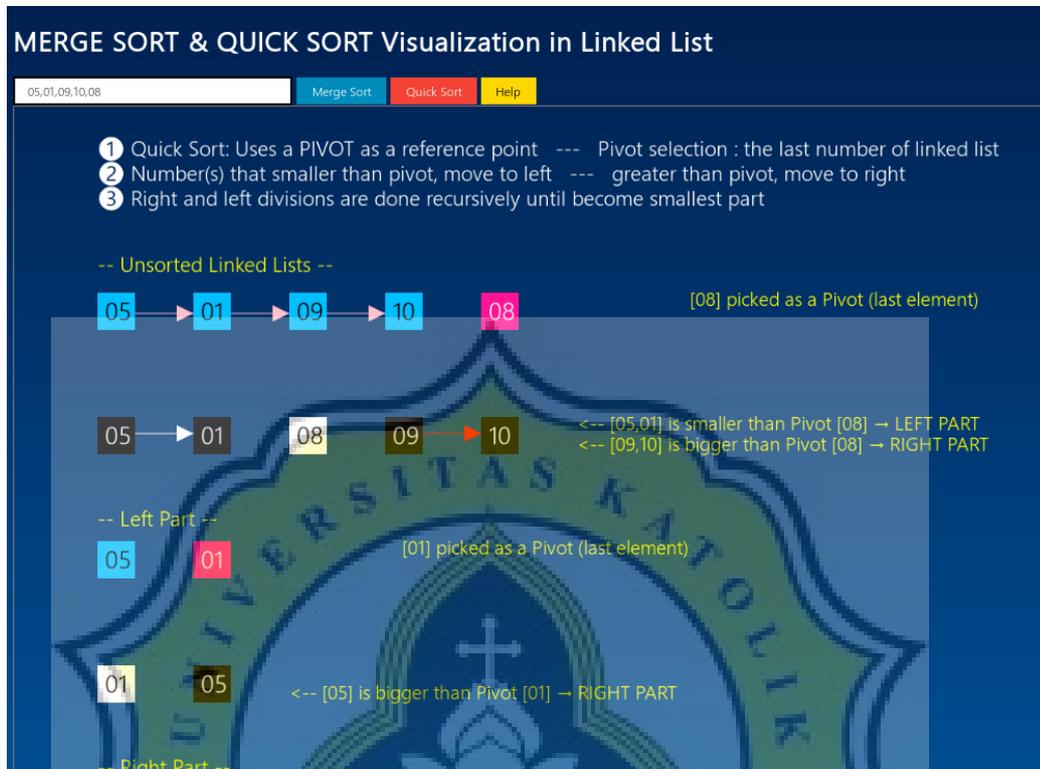
-- Right Partition --

10 → 08

08 ← is a comparison result between [10,08]

Picture 5.2.2: Visualization of Merge Sort

In the illustration above is a steps when partitioning the linked list into several parts. In this program there is also explanation about sorting algorithm that is used when visualization is run.



Picture 5.2.3: Visualization of Quick Sort

In the Quick Sort Visualization, after the quick sort button clicked, it will appear a picture of linked list along with some explanation of this sorting method. At each step of the pivot selection and separation to merge there is a explanation on the right side. Explanation purpose is to make the user of this program understand the running steps of sorting method.

Table 5.1: Result of the visualization testing

No	Data.length	Merge Sort	Quick Sort
1	5	All drawn with no problem	All drawn with no problem
2	10	All drawn with no problem	All drawn with no problem
3	12	All drawn except the sorting result	All drawn, explanation little bit trimmed
4	14	Some sorting step trimmed, sorting result not shown	Sort result not shown, some of explanation trimmed
5	16	Some sorting step trimmed, sorting result not shown	Sort result not shown, some of explanation trimmed
6	18	Unsorted linked list little bit trimmed, some sorting step and the sorting result not shown	Unsorted linked list trimmed, sort result not shown, explanation trimmed
7	20	Unsorted linked list little bit trimmed, some sorting step and the sorting result not shown	Unsorted linked list trimmed, explanation trimmed, sort result not shown.

The table above talk about test input data from 5 to 20 data length. The inputting data from 5 to 10 will show the result without problem. The visualization completely appears. But after inputting more than 12 data, some parts of the visualization does not appear. So this program is suggested to be used for 5 to 10 data only, due the limitation of canvas size.