CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation

This application will create linked list visualization with Javascript applicationming language. This application will store all of the data that user input into doubly linked list.

The main menus in this application are add, delete, search, and sort. In add menu have two submenus, there are add and add middle submenu. And in delete menu have three submenus, there are delete, delete head and delete tail. Besides all menus this application have one menu to count the number of nodes. This is code for add submenu :

```
1. add(data){
2.    const node = new Node(data);
3.    if(this._length != 0){
4.       this.tail.next = node;
5.       node.prev = this.tail;
6.       this.tail = node;
7.    }
8.    else{
9.       this.head = node;
10.      this.tail = node;
11.      }
12.         this._length++;
13.         console.log(data);
14.         return node;
15.      };
```

From the code above, in first line is to create function add with data as parameter. In second line is use to create a new variable constant with name node, those node have node type of data. Those node store new node contain with parameter data, it mean the parameter data have node type of data and stored in variable node. In third line this application give a condition, if linked list is empty, then run line 4-6 to create new node become head and tail from this linked list. While if linked list is not empty, then this application will run line 9-10 to

create a new node become tail from this linked list. Line 12-14 for show the result of linked list and loop until the last node.

The second submenu in add menu is add middle. Those submenu is use to add new node with desirable position. In code below will show the code for add middle.

```
1. addtengah(baru, posisi){
2.    var node = new Node(baru);
3.    var temp;
4.    var current = this.head;
5.       while(current != null){
6.          if(current.data == posisi){
7.             temp = current;
8.             node.next = temp.next;
9.             node.prev = temp;
10.                 current.next.prev = node;
11.                 current.next = node;
12.                 console.log(node);
13.              }
14.              current = current.next;
15.           }
16.           this._length++;
17.    };
```

From the code above, in first line is to create function addtengah with two parameter there are baru and position. In line 2-4 is use to create new variable. In fifth line is use to loop line 6-15 until this node is null. In line sixth is use to give condition if the data in linked list is same with position that user want to add the new node, then this application will run line 7-13. Those code is use to add the new node in the next of the position. But if the data in linked list is node same with the position that user want, then this application will check the next node until the last node use code in line 14. If the data in linked list is not found then the new node cannot add in linked list.

Beside the add menu this application is have delete menu. The delete menu have three submenu, there are delete, delete head and delete tail. In code below will show the code for delete submenu.

```
1. hapus(value){
2.    var tmp=this.head;
3.       while(tmp != null){
4.          if(tmp.data == value){
5.             if(tmp == this.head && tmp.next == null){
6.                this.head = null;
7.             }
8.             else if(tmp == this.head){
9.                this.head = this.head.next;
10.                     this.head.prev =null;
11.                  }
12.                     else if(tmp.next == null){
13.                        var temp = tmp;
14.                        temp.prev.next=null;
15.                        tmp = temp;
16.                     }
17.                     else{
18.                        var node1=tmp;
19.                        node1.prev.next=node1.next;
20.                        node1.next.prev=node1.prev;
21.                        tmp = node1;
22.                     }
23.            }
24.                  tmp = tmp.next;
25.       }
26.               return this.head;
27.    };
```

From the code above, in first line is to create function hapus with value as parameter. In fisrt line is use to create new variable. In third line is code to loop line 4-25 until the node in this linked list is empty. In line fourth this application give condition if the data in linked list are same with value that user want to delete, then this application will run code in line 5-22. But if the data in linked list is not same with value that user want to delete, then this application will check the next node until this application found the same value. But if until the last node the data in linked list are not same with value, then this application wouldn't delete any data, this command exist in line 24. In line 5-7 is use to delete data if the data position in the head of the linked list and the next node is empty. In line 8-11 is use to delete the data if the data position in the head of linked list. In line 12-16 is use to delete the data if the data position in the tail of linked list. In line 18-22 is use to delete the data if the data position is not in the head or tail of thelinked list.

Beside those submenu the delete menu have delete head submenu, in code below will show those code submenu.

```
1. hapushead(){
2.    var node = new Node();
3.       node=this.head;
4.       this.head=node.next;
5.       node.next=null;
6.       this._length--;
7. };
```

From the code above, in first line is to create function hapushead. In second line is use to create a new variable. In line 3-6 is code to delete the head of the linked list. The last submenu in delete is delete tail. The code below will show the code for delete tail.

```
1. hapustail(){
2.    var node = new Node();
3.    node = this.head;
4.    while(node.next.next != null){
5.       node = node.next;
6.    }
7.       node.next = null;
8.       this.tail=node;
9.       this._length--;
10.      };
```

From the code above, in first line is to create function hapustail. In second line is use to create a new variable. In line fourth is use to loop until node in the last position. In line 7-8 is use to delete the tail of this linked list.

After the delete menu the next main menu in this application is search. This menu is use to searching data from the linked list that user want. If the data that user want to search is found in the linked list, then this application will highlight the node that searching for. The code below will show the code for search menu.

```
1. mencari(a){
2.    canvas.hapuscanvas();
3.    this.print();
4.    canvas.xawl=2;
5.    var node=new Node();
6.    var cari= new Node();
7.    this.aa = 2;
8.    node=this.head;
9.    cari=null;
10.           if(node==null){
```

```
11.              console.log("empty");
12.           }
13.           else{
14.              while(node !=null){
15.                 this.aa    =    this.aa    +    canvas.xawl    +
  canvas.panjangnode + canvas.jarakx;
16.                 if(canvas.xawl > 1000){
17.                    this.aa = 115;
18.                    canvas.yawl = canvas.yawl+50;
19.                 }
20.                    if(node.data == a){
21.                       cari=node;
22.                       canvas.drawcari(this.aa);
23.                       break;
24.                    }
25.                    else{
26.                       node=node.next;
27.                    }
28.                 }
29.           }
30.           return cari;
31.     };
```

From the code above, in first line is to create function search. In line 2-4 is use to call the other function in this application. In line 5-9 is use to create a new variable. In line 10-12 is use to give condition if the linked list is empty, then this application will show notification empty. If the linked list is not empty then this application will run code in line 14-29. In line 14 is use to loop line 15-27 until the node in linked list is empty. In line 16-19 is use to give limit to draw in canvas if the width is bigger than 1000. in line 20-24 is use to search the data. If the data that user want to search is same with the data in linked list, then this application will highlight the node that searching for.

The last menu in this application is sort menu. This menu is use to sort all of the data in linked list. The code below will show the code for sort menu.

```
1. sorting(){
2.    var node;
3.    var temp = new Node();
4.    node = this.head;
5.    var done = false;
6.      while (!done) {
7.         node = this.head;
8.         done = true;
9.        while(node != null){
10.          if(node.next != null && node.data > node.next.data ){
11.                  console.log(node);
```

```
12.                 temp.data = node.data;
13.                 node.data = node.next.data;
14.                 node.next.data = temp.data;
15.                 done = false;
16.             }
17.             node=node.next;
18.         }
19.     }
20.  };
```

From the code above, in first line is to create function sorting. In line 2-5 is use to create a new variable. The variable done is set false value for boolean. In line 6 is use to loop line 7-19 until the done value is true. In line 8 is use to set variable done be true value. In line 9 is use to loop line 10-18 until the node of the linked list is empty. In line 11-16 is use to sort the data in linked list. If the data is bigger than the next data, then the data will be swapped. But if the data is smaller than the next data, then the data will be check and compare the next data. In line 15 is use to set variable done be false value, then this application will loop to the next node.

5.2 Testing

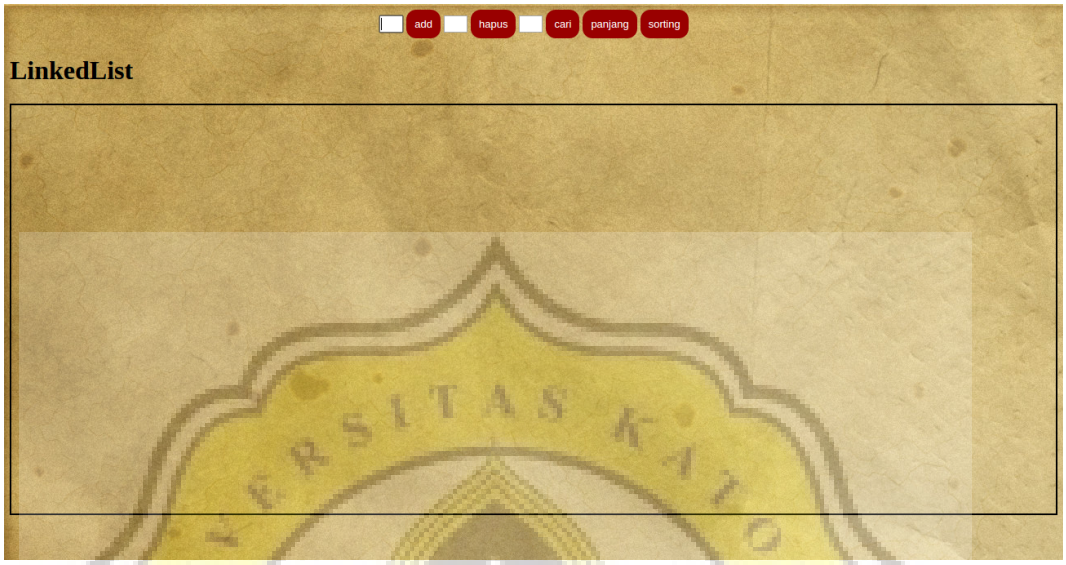

Illustration 5.2.1: User Interface

From the illustration 5.2.1 is show the user interface of this application. This application have five main menu there are add, delete, search, sort, and count the number of node. In add menu have two submenu there are add and add middle. In illustration below will show the proccess for add menu.

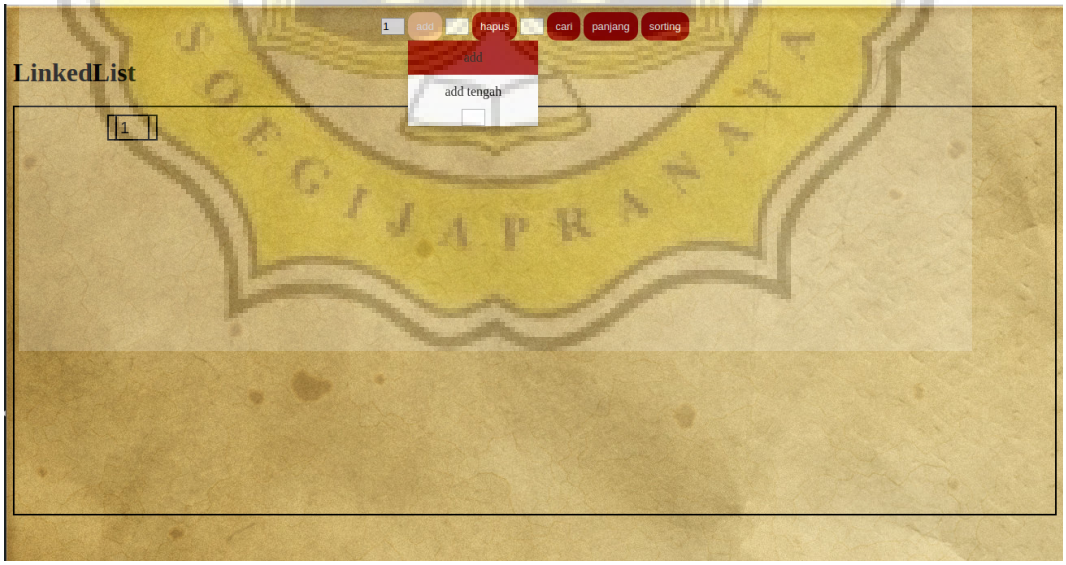

Illustration 5.2.2: Add Submenu

From the illustration 5.2.2 is show the procces of add submenu. If user choose the add submenu, then this application will add new node in the linked list and show the result. If the linked list not have a node, then this application will

add new node in head and tail position. But if the linked list have nodes, then this application will add new node in tail position. The second submenu in add main menu is add middle, in illustration below will show the proccess of add middle.

Illustration 5.2.3: Add Middle Submenu

In the illustrasion 5.2.3 is submenu add middle. In those submenu, user can add the data with desirable position. If the desirable data position is found, then this application will add new node in those desirable position and stored in linked list. After that this application will show the result.

In second main menu in this application is delete, in those menu have three submenu there are delete, delete head and delete tail. In illustration below will show the proccess of delete submenu.

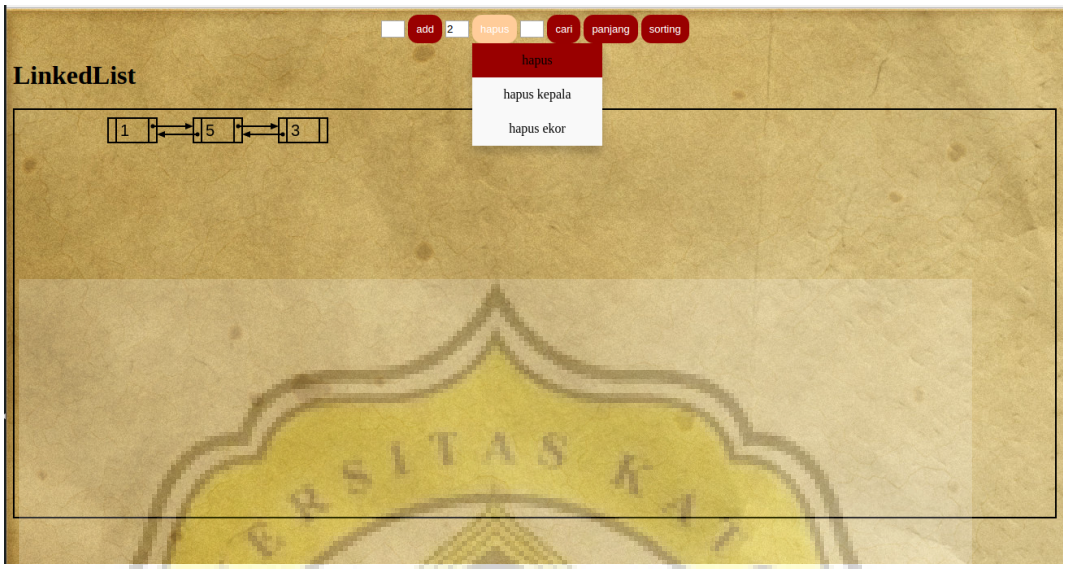

Illustration 5.2.4: Delete Submenu

From this illustration above user can delete the data that user want to delete. The data will be check in linked list, if the data that user want to delete is found, then the data will be delete. But if the data that user want to delete is not found, then the data wouldn't be delete. Beside those submenu this delete menu have delete head and tail submenu. If the user choose those menus, then this application will delete the head or tail of the linked list In accordance with the user choosen before. In illustration 5.2.5 willl show the delete head submenu and in the illustration 5.2.6 will show the delete tail.

Illustration 5.2.5: Delete Head Submenu

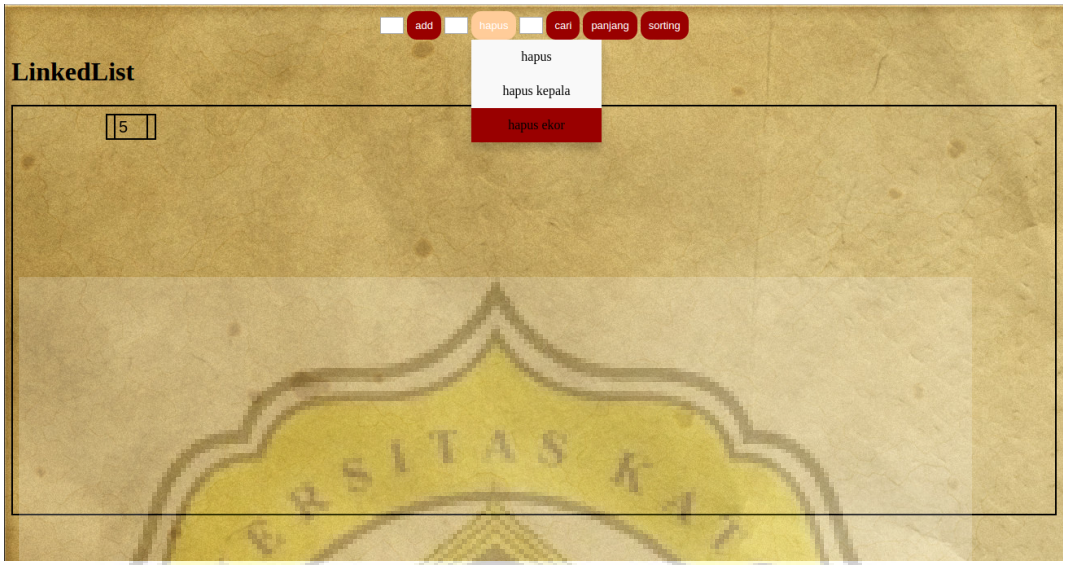

Illustration 5.2.6: Delete Tail Submenu

In third main menu in this application is search, in those menu user can searching the data that exist in the linked list. In illustration below will show the procces of search menu.

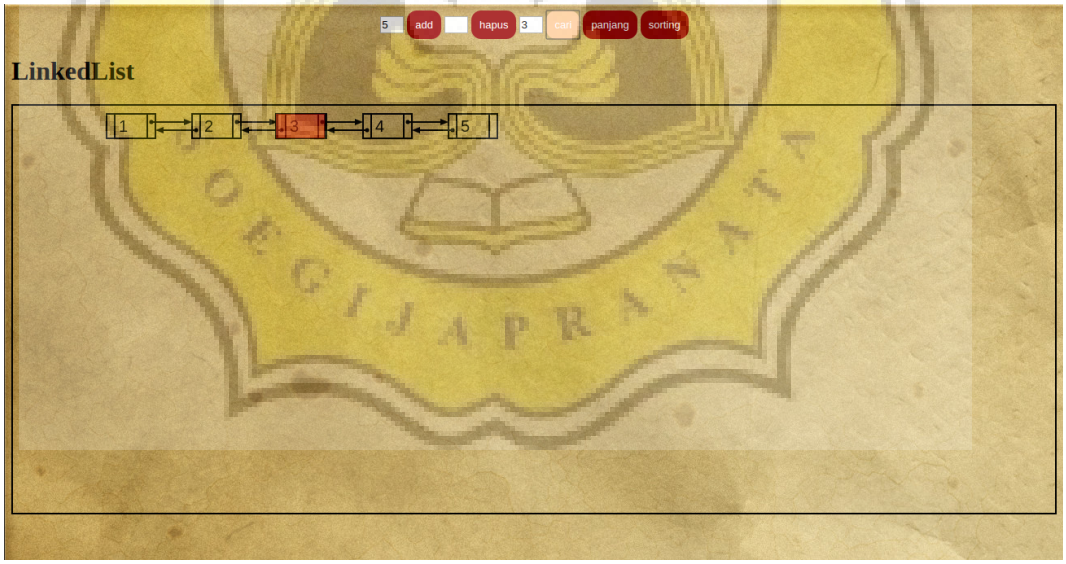

Illustration 5.2.7: Menu Search

In illustration above is show that the user want to searh data with value 3 in the linked list. This application will find the Node according to the data that user input before. If the data is found then this application will highlight the node that searching for. But if the data is not found this application will show

notification that data is not exist. Beside those menus this application also have count menu, the illustration will show in below.

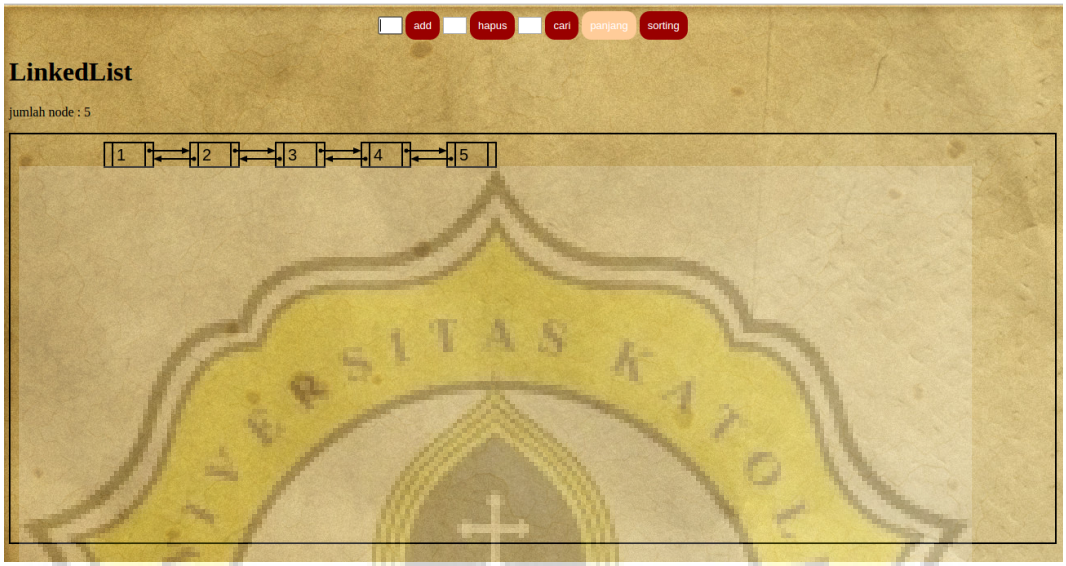

Illustration 5.2.8: Count Menu

From ilustration above is show the count main menu. This menu is use to count the number of node in the linked list. If user choose this menu, than this application will count the number of node and show the result.

The last menu in this application is sort menu. In illustration below will show the linked list that has not been sorting.

Illustration 5.2.9: Menu Sort Before

From this illustration above all the data in linked list will be comparison with the next data. If the data is bigger than the next data, then the data will be swapped. But if the data is smaller than the next data, then the data will be check and compare the next data. After the data have already sort, then this application will show the result. In illustration below will show the result after all the data is done be sort.

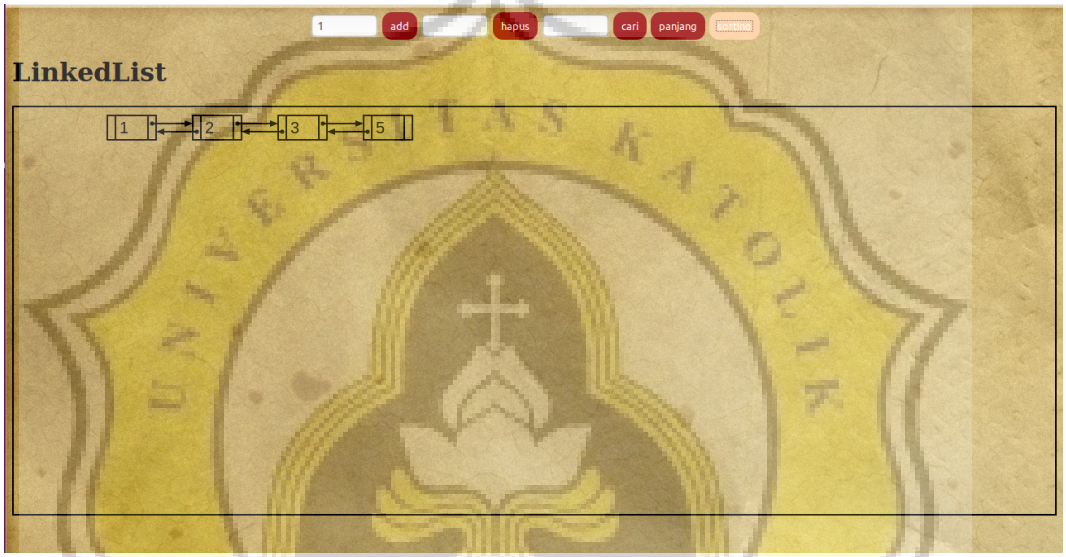

Illustration 5.2.10: Menu Sort after