

Nama : Muhammad Arif Irfan

Nim : 202210370311030

Kelas : Struktur Data J

TUGAS 1 STRUKTUR

DATA SINGLE LINKED LIST

Algoritma :

1. Buat node baru (newNode) dengan nilai data (int data)
2. Periksa apakah linked list masih kosong (first == null).
 - a. Jika ya, atur first dan last ke newnode karena ini akan menjadi node pertama dan terakhir.
 - b. Jika tidak, lanjut ke langkah berikutnya.
3. Periksa apakah data baru kurang dari atau sama dengan data pada node pertama (data <= first.data).
 - a. Jika ya, atur newNode.next ke first dan kemudian atur first ke newNode karena data baru akan menjadi node pertama.
 - b. Jika tidak, lanjut ke langkah berikutnya.
4. Inisialisasi variabel current dengan first dan before dengan null.
5. Traverse linked list dengan iterasi menggunakan loop while sampai menemukan posisi yang tepat atau mencapai akhir linked list.
 - a. Di setiap iterasi, perbarui before ke current dan current ke current.next.
 - b. Ulangi langkah ini selama data baru lebih besar dari data pada current.
6. Sisipkan node baru pada posisi yang sesuai dengan mengatur before.next ke newNode dan newNode.next ke current.
7. Perbarui node terakhir (last) jika node baru disisipkan di akhir linked list (ketika current == null).

Source Code :

- InsertSort

```
1 public void insertSort(int data) {
2     Node newNode = new Node(data);
3     if (first == null) {
4         first = newNode;
5         last = newNode;
6     } else if (data <= first.data) {
7         newNode.next = first;
8         first = newNode;
9     } else {
10        Node current = first;
11        Node before = null;
12        while (current != null && data > current.data) {
13            before = current;
14            current = current.next;
15        }
16        before.next = newNode;
17        newNode.next = current;
18
19        if (current == null) {
20            last = newNode;
21        }
22    }
23 }
```

- Main + Output

```
1 public class Linki {
2     public static void main(String[] args) {
3         LinkedList my = new LinkedList();
4         my.insertSort(10);
5         my.insertSort(20);
6         my.insertSort(30);
7         my.insertSort(50);
8         my.insertSort(60);
9         my.insertSort(80);
10
11        // my.delete(60);
12        // my.insertFisrt(100);
13        my.display();
14    }
15 }
```

```
PS D:\College Folder\Semester 4\StrukturData\KelasTeori\Tugas1\LinkedList> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ADMIN\AppData\Roaming\Code\User\workspaceStorage\4997c8c28b757006a5e5851e0372326a\redhat.java\jdt_ws\LinkedList_13c5f546\bin' 'Linki'
10 20 30 50 80
PS D:\College Folder\Semester 4\StrukturData\KelasTeori\Tugas1\LinkedList> |
```