

Nama : Juan Axl Ronaldo Zaka Putra

NIM : 220411100066

Kelas : Struktur Data IF 2D

1. Jika terdapat antrian Input pada Linked List 1, 3, 2, 4, 5, 8, 7, 6, bagaimana cara agar data yang terkoleksi pada linkedlist dapat terurut 1,2,3,4,5,6,7,8. Jelaskan Model, Script, serta penjelasan komputasinya.

```
5 struct Node
6 {
7     int data;
8     Node *next;
9 } *head, *tail, *current, *newNode, *del;
10
11 void createNode(int data)
12 {
13     head = new Node();
14     head->data = data;
15     head->next = NULL;
16     tail = head;
17 }
```

Pertama, saya membuat mendeklarasikan sebuah struct dengan nama Node yang berisi data dengan tipe data integer dan variabel pointer \*next dengan tipe data Node. Lalu, mendeklarasikan juga beberapa variabel pointer seperti \*head, \*tail, \*current, \*newNode, dan \*del.

Berikutnya, membuat fungsi tanpa pengembalian data yaitu void createNode dengan parameter integer data untuk membuat anggota node pertama yang didalamnya terdapat head = new Node() yang berarti mengalokasikan memori untuk node baru. Lalu, data pada head berisi data yang dimasukkan pada parameter fungsi, next pada head berisi NULL karena merupakan data yang dimasukkan pertama kali, dan nilai dari variabel tail sama dengan variabel head.

```
27 // tambahAkhir single linked list
28 void insertTail(int data)
29 {
30     newNode = new Node();
31     newNode->data = data;
32     newNode->next = NULL;
33     tail->next = newNode;
34     tail = newNode;
35 }
```

Berikutnya, membuat fungsi void insertTail dengan parameter integer data untuk menambahkan data pada urutan linked list terakhir yang didalamnya terdapat newNode = new Node() yang berarti mengalokasikan memori untuk node baru. Lalu, data pada newNode berisi data yang dimasukkan pada parameter fungsi, next pada newNode berisi NULL karena merupakan data ke terakhir, next pada tail berisi newNode, dan nilai variabel tail diisi newNode.

```

57 // print single linked list
58 void printSingleLinkedList ()
59 {
60     cout << "Anggota single linked list : " << endl;
61     current = head;
62     while (current != NULL)
63     {
64         cout << current->data << " ";
65         current = current->next;
66     }
67     cout << "\n" << endl;
68 }
69

```

Berikutnya, saya membuat fungsi void printSingleLinkedList untuk menampilkan semua data anggota linked list yang berisi print teks “Anggota single linked list : “, variabel current dengan nilai head atau node pertama, perulangan while selama current tidak sama dengan NULL lalu menjalankan print data dari current dan current diganti oleh next dari current. Akan dilakukan perulangan sampai node terakhir.

```

71 int main()
72 {
73     createNode(1);
74     insertTail(2);
75     insertTail(3);
76     insertTail(4);
77     insertTail(5);
78     insertTail(6);
79     insertTail(7);
80     insertTail(8);
81     printSingleLinkedList();
82     return 0;
83 }
84

```

Selanjutnya, untuk menginputkan node dan menampilkan anggota, saya melakukannya pada fungsi utama / main. Pertama saya memanggil fungsi createNode dengan parameter 1 untuk membuat node pertama, lalu seterusnya menggunakan fungsi insertTail dengan parameter 2 dan melakukannya secara berulang sampai 8. Lalu terakhir memanggil fungsi printSingleLinkedList untuk menampilkan semua data anggota linked list.

Output:

```

Anggota single linked list :
1 2 3 4 5 6 7 8

```

2. Bagaimana cara menghapus data melalui Tail dengan Urutan hapus 8,7,6 dari koleksi yang terbentuk, jelaskan Model, Script, serta penjelasan komputasinya.

```

44 void deleteTail ()
45 {
46     del = tail;
47     current = head;
48     while (current->next != tail)
49     {
50         current = current->next;
51     }
52     tail = current;
53     tail->next = NULL;
54     delete del;
55 }

```

Untuk menghapus data melalui tail / data terakhir saya membuat fungsi deleteTail yang berisi variabel del dengan nilai tail, variabel current dengan nilai head, perulangan while dengan kondisi selama current->next tidak sama dengan tail dan didalamnya current akan menjadi current->next selama perulangan dilakukan. Setelah perulangan dilakukan variabel tail diganti current, tail->next diganti NULL dan delete / menghapus variabel del.

```

71 int main()
72 {
73     createNode(1);
74     insertTail(2);
75     insertTail(3);
76     insertTail(4);
77     insertTail(5);
78     insertTail(6);
79     insertTail(7);
80     insertTail(8);
81     printSingleLinkedList();
82
83     deleteTail();
84     printSingleLinkedList();
85
86     deleteTail();
87     printSingleLinkedList();
88
89     deleteTail();
90     printSingleLinkedList();
91
92     return 0;
93 }
94

```

Untuk menghapus urutan 8, 7, 6 saya menggunakan fungsi deleteTail tiga kali seperti pada script di atas. Lalu menghasilkan output:

```

Anggota single linked list :
1 2 3 4 5 6 7 8

Anggota single linked list :
1 2 3 4 5 6 7

Anggota single linked list :
1 2 3 4 5 6

Anggota single linked list :
1 2 3 4 5

```

3. Bagaimana cara membuat struktur Circular-LinkedList, Model, Script, serta penjelasan komputasinya.

```

5 struct Node
6 {
7     int data;
8     Node *next;
9 } *last, *current, *newNode, *del;
10
11 void createNode (int data)
12 {
13     last = new Node();
14     last->data = data;
15     last->next = last;
16 }

```

Pertama, saya membuat mendeklarasikan sebuah struct dengan nama Node yang berisi data dengan tipe data integer dan variabel pointer \*next dengan tipe data Node. Lalu, mendeklarasikan juga beberapa variabel pointer seperti \*last, \*current, \*newNode, dan \*del.

Berikutnya, membuat fungsi tanpa pengembalian data yaitu void createNode dengan parameter integer data untuk membuat anggota node pertama yang didalamnya terdapat last = new Node() yang berarti mengalokasikan memori untuk node baru. Lalu, data pada last berisi data yang dimasukkan pada parameter fungsi, dan next pada last berisi last karena merupakan data yang dimasukkan pertama kali.

```

18 void insertStart (int data)
19 {
20     newNode = new Node();
21     newNode->data = data;
22     newNode->next = last->next;
23     last->next = newNode;
24 }

```

Berikutnya saya membuat fungsi insertStart untuk menambahkan data pada urutan pertama yang didalamnya terdapat newNode = new Node() yang berarti mengalokasikan memori untuk node baru. Lalu, data pada newNode berisi data yang dimasukkan pada parameter fungsi, next pada newNode berisi last->next agar menghubungkan ke data terakhir, dan next pada last berisi newNode agar menghubungkan ke newNode.

```

26 void insertEnd (int data)
27 {
28     newNode = new Node();
29     newNode->data = data;
30     newNode->next = last->next;
31     last->next = newNode;
32     last = newNode;
33 }

```

Berikutnya saya membuat fungsi insertEnd untuk menambahkan data pada urutan terakhir yang didalamnya terdapat newNode = new Node() yang berarti mengalokasikan memori untuk node baru. Lalu, data pada newNode berisi data yang dimasukkan pada parameter fungsi, next pada newNode berisi last->next agar menghubungkan ke data terakhir, next pada last berisi newNode agar menghubungkan ke newNode, dan nilai variabel last diganti ke variabel newNode.

```

35 void printSingleLinkedList ()
36 {
37     cout << "Anggota circular single linked list : " << endl;
38     current = last->next;
39     do {
40         cout << current->data << " ";
41
42         current = current->next;
43     } while (current != last->next);
44     if (current == last->next) {
45         cout << current->data;
46     }
47
48     cout << "\n" << endl;
49 }

```

Berikutnya, saya membuat fungsi void printSingleLinkedList untuk menampilkan semua data anggota linked list yang berisi print teks "Anggota circular single linked list : ", variabel current dengan nilai last->next atau node pertama, perulangan do while selama current tidak sama dengan last->next lalu menjalankan print data dari current dan current diganti oleh next dari current. Akan dilakukan perulangan sampai node pertama terulang kembali.

```

51 int main()
52 {
53
54     createNode(1);
55     insertEnd(2);
56     insertEnd(3);
57     insertEnd(4);
58     insertEnd(5);
59     insertEnd(6);
60     insertEnd(7);
61     insertEnd(8);
62
63     printSingleLinkedList();
64     return 0;
65 }

```

Selanjutnya, untuk menginputkan node dan menampilkan anggota, saya melakukannya pada fungsi utama / main. Pertama saya memanggil fungsi createNode

dengan parameter 1 untuk membuat node pertama, lalu seterusnya menggunakan fungsi insertEnd dengan parameter 2 dan melakukannya secara berulang sampai 8. Lalu terakhir memanggil fungsi printSingleLinkedList untuk menampilkan semua data anggota linked list.

Output:

```
Anggota circular single linked list :  
1 2 3 4 5 6 7 8 1
```