

## Tugas Oth Double link list circular

Nama:M.Agung balya

NIM:1203230095

Kelas:IF 03-01

Code

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct Node {
5      int data;
6      struct Node* next;
7      struct Node* prev;
8  } Node;
```

StructNode mendefinisikan data untuk menyimpan nilai integer, next untuk menunjuk node berikutnya, dan prev untuk menunjuk node sebelumnya.

```
10 void insertNode(Node** head, Node** tail, int data) {
11     Node* newNode = (Node*) malloc(sizeof(Node));
12     newNode->data = data;
13     newNode->next = NULL;
14     newNode->prev = NULL;
15
16     if (*head == NULL) {
17         *head = newNode;
18         *tail = newNode;
19     } else {
20         (*tail)->next = newNode;
21         newNode->prev = *tail;
22         *tail = newNode;
23     }
24 }
```

insertNode untuk memasukkan node baru ke dalam link list dengan data yang diberikan, kemudian ditambahkan ke akhri link list.

```
26 void printList(Node* head) {
27     Node* current = head;
28     while (current != NULL) {
29         printf("Address: %p, Data: %d\n", (void*) current, current->data);
30         current = current->next;
31     }
32     printf("\n");
33 }
```

printList untuk mencetak seluruh is linked list dengan parameter head yang setiap iterasinya dilakukan melalui linked list, dan setiap node dicetak dengan Alamat memori dan datanya.

```

35 void sortList(Node** head) {
36     int swapped;
37     Node *ptr1;
38     Node *lptr = NULL;
39     Node *temp = NULL;
40
41     if (*head == NULL)
42         return;
43
44     do {
45         swapped = 0;
46         ptr1 = *head;
47
48         while (ptr1->next != lptr) {
49             if (ptr1->data > ptr1->next->data) {
50                 if (ptr1 == *head) {
51                     *head = ptr1->next;
52                 }
53                 if (ptr1->next == lptr) {
54                     lptr = ptr1;
55                 }
56
57                 temp = ptr1->next->next;
58                 ptr1->next->prev = ptr1->prev;
59                 if (ptr1->prev != NULL) {
60                     ptr1->prev->next = ptr1->next;
61                 }
62                 ptr1->prev = ptr1->next;
63                 ptr1->next = temp;
64                 if (temp != NULL) {
65                     temp->prev = ptr1;
66                 }
67                 ptr1->prev->next = ptr1;
68
69                 swapped = 1;
70             }
71             else {
72                 ptr1 = ptr1->next;
73             }
74         } while (swapped);
75     }
76 }

```

Sortlist berfungsi untuk mengurutkan linked list menggunakan bubble sort dengan fungsi pointer ke pointer kepala linked list sebagai parameter dengan algoritma bubble sort digunakan untuk menurunkan posisi node berdasarkan nilai datanya.

```

78  int main() {
79      int N;
80      Node* head = NULL;
81      Node* tail = NULL;
82
83      scanf("%d", &N);
84
85      for (int i = 0; i < N; i++) {
86          int data;
87          scanf("%d", &data);
88          insertNode(&head, &tail, data);
89      }
90
91      printf("sebelum diurutkan:\n");
92      printList(head);
93
94      sortList(&head);
95
96      printf("sesudah diurutkan:\n");
97      printList(head);
98
99      return 0;
100 }
101

```

Fungsi main sebagai tempat program dimulai dengan jumlah elemen linked list dari masukan pengguna, setiap elemen dimasukkan ke linked list menggunakan insertNode, linked list sebelum dan setelah pengurutan dicetak menggunakan printList, dan linked list diurutkan menggunakan sortList.

Output

```
PS C:\agung\Februari\Tugas Sing
ugas-Oth-Double-Link-List-Circ
3 31 2 123
```

sebelum diurutkan:

```
Address: 006A2A20, Data: 31
Address: 006A2A38, Data: 2
Address: 006A2A50, Data: 123
```

sesudah diurutkan:

```
Address: 006A2A38, Data: 2
Address: 006A2A20, Data: 31
Address: 006A2A50, Data: 123
```

```
PS C:\agung\Februari\Tugas Sing
le-Link-List-Circular.c -o T
5 5 3 8 6 1
```

sebelum diurutkan:

```
Address: 008E2A20, Data: 5
Address: 008E2A38, Data: 3
Address: 008E2A50, Data: 8
Address: 008E2A68, Data: 6
Address: 008E2A80, Data: 1
```

sesudah diurutkan:

```
Address: 008E2A80, Data: 1
Address: 008E2A38, Data: 3
Address: 008E2A20, Data: 5
Address: 008E2A68, Data: 6
Address: 008E2A50, Data: 8
```