

Algoritma dan Struktur Data
SORTING CIRCULAR DOUBLY LINKED LIST



Oleh :

(M. Kevin A.) / (1203230096)

IF-03-01

Program Studi Informatika
Fakultas Informatika
Universitas Telkom Surabaya
Tahun 2024

1. Source code :

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct Node { int data;
5  struct Node* prev; struct Node* next;
6  } Node;
7
8  Node* createNode(int data) {
9  Node* newNode = (Node*)malloc(sizeof(Node)); newNode->data = data;
10 newNode->prev = newNode; newNode->next = newNode; return newNode;
11 }
12
13 void insertEnd(Node** head, int data) { Node* newNode = createNode(data); if (*head == NULL) {
14 *head = newNode;
15 } else {
16 Node* last = (*head)->prev; newNode->next = *head; (*head)->prev = newNode; newNode->prev = last;
17 last->next = newNode;
18 }
19 }
20
21 void sortList(Node** head) {
22 if (*head == NULL || (*head)->next == *head) return;
23 Node* current = *head; do {
24 Node* nextNode = current->next; while (nextNode != *head) {
25 if (current->data > nextNode->data) { int temp = current->data; current->data = nextNode->data; nextNode->data = temp;
26 }
27 nextNode = nextNode->next;
28 }
29 current = current->next;
30 } while (current->next != *head);
31 }
32
33 void displayList(Node* head) { if (head == NULL) {
34 printf("List kosong.\n"); return;
35 }
36 Node* temp = head; do {
37 printf("Alamat: %p, Data: %d\n", temp, temp->data); temp = temp->next;
38 } while (temp != head); printf("\n");
39 }
40
41 int main() {
42 Node* head = NULL; int N, data;
43
44 printf("Masukkan jumlah data: "); scanf("%d", &N);
45
46 for (int i = 0; i < N; i++) { printf("Masukkan data ke-%d: ", i+1); scanf("%d", &data);
47 insertEnd(&head, data);
48 }
49
50 printf("List sebelum pengurutan:\n"); displayList(head);
51
52 sortList(&head);
53
54 printf("List setelah pengurutan:\n"); displayList(head);
55
56 return 0;
57 }
58

```

Output :

```
Masukkan jumlah data: 5
Masukkan data ke-1: 5
Masukkan data ke-2: 3
Masukkan data ke-3: 8
Masukkan data ke-4: 1
Masukkan data ke-5: 6
List sebelum pengurutan:
Alamat: 00C615F0, Data: 5
Alamat: 00C61608, Data: 3
Alamat: 00C61620, Data: 8
Alamat: 00C61688, Data: 1
Alamat: 00C616A0, Data: 6

List setelah pengurutan:
Alamat: 00C615F0, Data: 1
Alamat: 00C61608, Data: 3
Alamat: 00C61620, Data: 5
Alamat: 00C61688, Data: 6
Alamat: 00C616A0, Data: 8
```

Penjelasan :

Bagian pertama dari kode ini mendefinisikan struktur yang disebut "Node" yang merepresentasikan satu node di dalam daftar. Setiap node punya tiga bagian: "data" (nilai yang disimpan dalam node), "prev" (pointer ke node sebelumnya), dan "next" (pointer ke node berikutnya).

Fungsi createNode membuat node baru dengan nilai tertentu dan mengembalikan pointer ke node tersebut.

Fungsi insertEnd menambahkan node baru ke akhir daftar. Kalau daftar kosong, fungsi ini bikin node baru dan mengatur node itu sebagai kepala daftar. Kalau nggak kosong, fungsi ini cari node terakhir di daftar dan menambahkan node baru setelahnya.

Fungsi sortList mengurutkan daftar dalam urutan naik. Fungsi ini ngelakuin dengan iterasi melalui daftar dan menukar node yang bersebelahan kalau mereka dalam urutan yang salah.

Fungsi displayList nge-print nilai-nilai dalam daftar, bersama dengan alamat memorinya.

Fungsi main adalah titik masuk program. Fungsi ini minta user untuk input beberapa nilai, nambahin setiap nilai ke daftar pakai insertEnd, ngurutkan daftar pakai sortList, dan kemudian menampilkan daftar yang sudah diurutkan pakai displayList.