

Circular Double Link List

Nama: Jehova Putra Yan Nehru

Kelas: IF0301

Nim: 1203230107

Source Code

```
#include <stdio.h>
#include <stdlib.h>

typedef struct Node {
    int data;
    struct Node* prev;
    struct Node* next;
} Node;

Node* createNode(int data) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->prev = newNode;
    newNode->next = newNode;
    return newNode;
}

void insertEnd(Node** head, int data) {
    Node* newNode = createNode(data);
    if (*head == NULL) {
        *head = newNode;
    } else {
        Node* last = (*head)->prev;
        newNode->next = *head;
        (*head)->prev = newNode;
        newNode->prev = last;
        last->next = newNode;
    }
}

void sortList(Node** head) {
    if (*head == NULL || (*head)->next == *head) return;

    Node* current = *head;
    do {
        Node* nextNode = current->next;
        while (nextNode != *head) {
            if (current->data > nextNode->data) {
```

```

        int temp = current->data;
        current->data = nextNode->data;
        nextNode->data = temp;
    }
    nextNode = nextNode->next;
}
current = current->next;
} while (current->next != *head);
}

void displayList(Node* head) {
    if (head == NULL) {
        printf("List kosong.\n");
        return;
    }
    Node* temp = head;
    do {
        printf("Alamat: %p, Data: %d\n", temp, temp->data);
        temp = temp->next;
    } while (temp != head);
    printf("\n");
}

int main() {
    Node* head = NULL;
    int N, data;

    printf("Masukkan jumlah data: ");
    scanf("%d", &N);

    for (int i = 0; i < N; i++) {
        printf("Masukkan data ke-%d: ", i+1);
        scanf("%d", &data);
        insertEnd(&head, data);
    }

    printf("List sebelum pengurutan:\n");
    displayList(head);

    sortList(&head);

    printf("List setelah pengurutan:\n");
    displayList(head);

    return 0;
}

```

Input 1

5

3

8

1

6

Output 1:

```
● PS C:\Users\JEHOVA\Documents\VSC\Bahasa C or C++> cd
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: 00CF1598, Data: 5
Alamat: 00CF15B0, Data: 3
Alamat: 00CF15C8, Data: 8
Alamat: 00CF2370, Data: 1
Alamat: 00CF2388, Data: 6

List setelah pengurutan:
Alamat: 00CF1598, Data: 1
Alamat: 00CF15B0, Data: 3
Alamat: 00CF15C8, Data: 5
Alamat: 00CF2370, Data: 6
Alamat: 00CF2388, Data: 8
○ PS C:\Users\JEHOVA\Documents\VSC\Bahasa C or C++> █
```

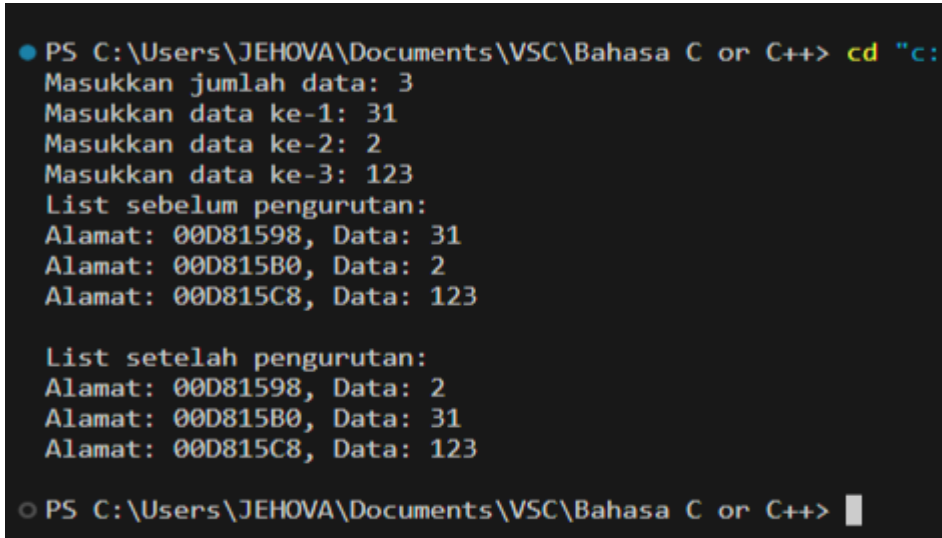
Input 2

31

2

123

Output 2:



```
PS C:\Users\JEHOVA\Documents\VSC\Bahasa C or C++> cd "c:\
Masukkan jumlah data: 3
Masukkan data ke-1: 31
Masukkan data ke-2: 2
Masukkan data ke-3: 123
List sebelum pengurutan:
Alamat: 00D81598, Data: 31
Alamat: 00D815B0, Data: 2
Alamat: 00D815C8, Data: 123

List setelah pengurutan:
Alamat: 00D81598, Data: 2
Alamat: 00D815B0, Data: 31
Alamat: 00D815C8, Data: 123

PS C:\Users\JEHOVA\Documents\VSC\Bahasa C or C++>
```

Penjelasan:

1. Fungsi CreateNode
Membuat dan menginisialisasi node baru dengan data yang diberikan dan mengalokasikan memori untuk node menggunakan malloc.
2. Fungsi InsertEnd
Menyisipkan node baru di akhir daftar, jika daftar kosong maka node baru di akhir akan menjadi elemen pertama namun jika tidak maka node baru disisipkan setelah terakhir dan sebelum elemen pertama
3. Fungsi SortList
Mengurutkan daftar menggunakan algoritma bubble sort dan membandingkan setiap pasangan node dan menukar data jika dibutuhkan
4. Fungsi DisplayList
Menampilkan semua elemen dalam daftar, dan melooping untuk mengunjungi setiap node
5. Fungsi Main:

Membaca jumlah data yang dimasukan dan menampilkan daftar setelah dan sebelum diurutkan