Tugas double linked-list circular

Nama:danendra urdha bhanu cetta harenndy

Nim:1203230110

Code:

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

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

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

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

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

void swapNodes(Node **head, Node *node1, Node *node2) {
```

```
    if (node1 == node2) return;

    Node *prev1 = node1->prev, *next1 = node1->next;
    Node *prev2 = node2->prev, *next2 = node2->next;

    if (next1 == node2) {
        node1->next = next2;
        node2->prev = prev1;
        node1->prev = node2;
        node2->next = node1;
    } else if (next2 == node1) {
        node2->next = next1;
        node1->prev = prev2;
        node2->prev = node1;
        node1->next = node2;
    } else {
        node1->next = next2;
        node1->prev = prev2;
        node2->next = next1;
        node2->prev = prev1;
    }

    if (prev1 != node2) prev1->next = node2;
    if (next1 != node2) next1->prev = node2;
    if (prev2 != node1) prev2->next = node1;
    if (next2 != node1) next2->prev = node1;

    if (*head == node1) *head = node2;
    else if (*head == node2) *head = node1;
}

void sortList(Node **head) {
    if (*head == NULL) return;
    int swapped;
    Node *ptr1;
    Node *lptr = NULL;

    do {
        swapped = 0;
        ptr1 = *head;

        while (ptr1->next != lptr && ptr1->next != *head) {
            if (ptr1->data > ptr1->next->data) {
                swapNodes(head, ptr1, ptr1->next);
                swapped = 1;
            }
            ptr1 = ptr1->next;
        }
```

```
        lptr = ptr1;
    } while (swapped);
}

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

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

    printf("Masukkan %d data:\n", N);
    for (int i = 0; i < N; i++) {
        scanf("%d", &A);
        insert(&head, A);
    }

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

    sortList(&head);

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

    return 0;
}
```

Output:

```
Masukkan jumlah data (N): 5
Masukkan 5 data:
5
3
8
1
6
List sebelum pengurutan:
Address: 00BB2980, Data: 5
Address: 00BB2998, Data: 3
Address: 00BB29B0, Data: 8
Address: 00BB29C8, Data: 1
Address: 00BB29E0, Data: 6

List setelah pengurutan:
Address: 00BB29C8, Data: 1
Address: 00BB2998, Data: 3
Address: 00BB2980, Data: 5
Address: 00BB29B0, Data: 8
Address: 00BB29E0, Data: 6

PS C:\Users\danen\OneDrive\Desktop\Khusus buat blajar>
```

```
Masukkan jumlah data (N): 3
Masukkan 3 data:
31
2
123
List sebelum pengurutan:
Address: 00C72980, Data: 31
Address: 00C72998, Data: 2
Address: 00C729B0, Data: 123

List setelah pengurutan:
Address: 00C72998, Data: 2
Address: 00C72980, Data: 31
Address: 00C729B0, Data: 123
```

Penjelasan program:

Struk: menyimpan variabel data berupa integer dan pointer berupa next da prev

Fungsi createnode: berfungsi untuk membuat dan menyimpan data dari node baru nengan cara menepatkan newnode->data = data, newnode->prev = newnode->data=next = NULL dan diembalikan ke nilai newnode

Fungsi insert: berfungsi untum menambahkan node baru yang sudah berisi data kedalam list,kemudian penempatan pointer prev atau next pada setiap node

Fungsi printlist:untuk mencetak data dalam node jika node tersebut tidak kosong dan mencetak alamat memori pointer pada masin masing node:

Fungsi swapnode: untuk mengubah posisi data node tampa harus mengubah isi dari datanya cara kerjanya adalah dengan membuat berberapa algoritma seperti pengecekan,mendapatkan pointer,mengatur ulang pointer,memperbarui pointer di node lain dan yang terakhir memperbarui posisi head

Fungsi sortlist: mengurutkan posisi pointer dengan mengunakan sorting buble sort

Fungsi utama: mengelola input dan utput dan memanggil fungsi lain