M Iqbal Ilham Prabowo

IF 03-01

1203230088

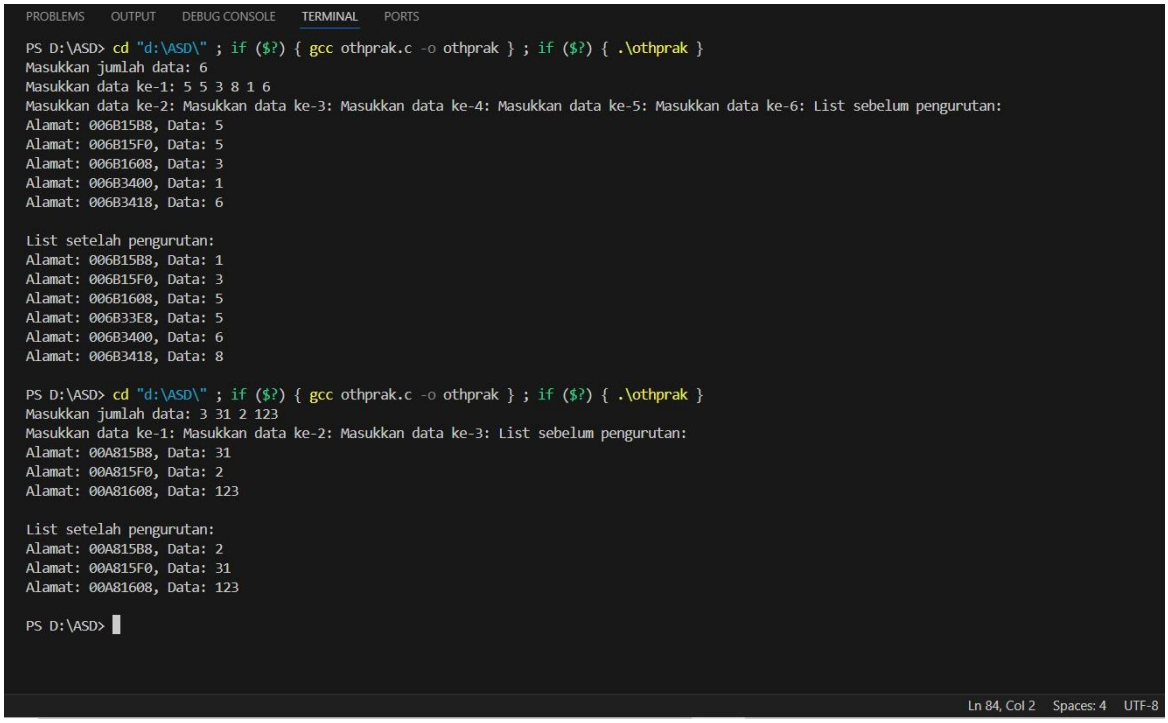
OTH

Source code

```c
#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;
}
```

Output



Penjelasan

Fungsi Utama

1. **createNode(int data)**: Membuat node baru dengan data yang diberikan dan mengatur **prev** serta **next** ke dirinya sendiri.
2. **insertEnd(Node** *head, int data)**: Menambahkan node baru ke akhir linked list.
3. **sortList(Node** *head)**: Mengurutkan linked list menggunakan metode bubble sort dengan menukar nilai data antar node.
4. **displayList(Node** *head)**: Mencetak alamat dan data dari setiap node dalam linked list.

Alur Program (main function)

1. Meminta pengguna untuk memasukkan jumlah data (**N**).
2. Menginput data dari pengguna dan menambahkannya ke linked list.
3. Menampilkan linked list sebelum diurutkan.
4. Mengurutkan linked list.
5. Menampilkan linked list setelah diurutkan.