

# OTH Linked List Circular

M Fadli Zamzami

1203230054

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

#include <stdlib.h>

// Struktur untuk node dalam linked list
typedef struct Node {
    int data; // Data yang disimpan dalam node
    struct Node* next; // Pointer ke node berikutnya
    struct Node* prev; // Pointer ke node sebelumnya (untuk linked list doubly)
} Node;

// Fungsi untuk membuat node baru
Node* createNode(int data) {
    Node* newNode = (Node*)malloc(sizeof(Node)); // Alokasikan memori untuk node baru
    if (!newNode) { // Jika memori tidak bisa dialokasikan
        printf("Memory alokasi gagal\n"); // Cetak pesan error
        return NULL; // Kembalikan NULL
    }
    newNode->data = data; // Inisialisasi data pada node baru
    newNode->next = NULL; // Inisialisasi pointer next ke NULL
    newNode->prev = NULL; // Inisialisasi pointer prev ke NULL
    return newNode; // Kembalikan pointer ke node baru
}

// Fungsi untuk menambahkan node di akhir linked list
void append(Node** head, int data) {
    Node* newNode = createNode(data); // Buat node baru dengan data
    if (*head == NULL) { // Jika linked list kosong
        *head = newNode; // Node baru menjadi head
    } else {
        Node* lastNode = *head; // Inisialisasi pointer ke head
        while (lastNode->next) { // Iterasi sampai node terakhir
            lastNode = lastNode->next; // Pindah ke node berikutnya
        }
        lastNode->next = newNode; // Set next dari node terakhir ke node baru
        newNode->prev = lastNode; // Set prev dari node baru ke node terakhir
    }
}

// Fungsi untuk mencetak linked list
```

```

void printList(Node* head) {
    Node* current = head; // Mulai dari head
    while (current) { // Iterasi selama current tidak NULL
        printf("Address: %p, Data: %d\n", (void*)current, current->data); //
        // Cetak alamat dan data node
        current = current->next; // Pindah ke node berikutnya
    }
}

// Fungsi untuk mengurutkan linked list secara ascending
void sortList(Node** head) {
    if (*head == NULL) return; // Jika linked list kosong, keluar dari fungsi

    Node* current = *head; // Mulai dari head
    Node* nextNode; // Pointer untuk node berikutnya
    int swapped; // Flag untuk mengetahui apakah ada pertukaran

    do {
        swapped = 0; // Reset flag swapped
        current = *head; // Mulai lagi dari head

        while (current->next != NULL) { // Iterasi sampai node terakhir
            if (current->data > current->next->data) { // Jika data node saat
                // ini lebih besar dari data node berikutnya
                // Tukar data
                int temp = current->data;
                current->data = current->next->data;
                current->next->data = temp;
                swapped = 1; // Set flag swapped ke 1
            }
            current = current->next; // Pindah ke node berikutnya
        }
    } while (swapped); // Lanjutkan iterasi jika ada pertukaran
}

int main() {
    int N; // Variabel untuk menyimpan jumlah node
    scanf("%d", &N); // Input jumlah node

    Node* head = NULL; // Inisialisasi head ke NULL
    for (int i = 0; i < N; i++) {
        int data; // Variabel untuk data node
        scanf("%d", &data); // Input data node
        append(&head, data); // Tambahkan node ke linked list
    }

    printf("Sebelum pengurutan:\n"); // Cetak pesan sebelum pengurutan
    printList(head); // Cetak linked list
}

```

```
sortList(&head); // Urutkan linked list

printf("Setelah pengurutan:\n"); // Cetak pesan setelah pengurutan
printList(head); // Cetak linked list

return 0; // Kembalikan 0 untuk mengakhiri program
}
```

```
f ($?) { .\Pweek8 }
```

5

5

3

8

1

6

Sebelum pengurutan:

Address: 00D42FC0, Data: 5

Address: 00D416D8, Data: 3

Address: 00D416F0, Data: 8

Address: 00D41708, Data: 1

Address: 00D41720, Data: 6

Setelah pengurutan:

Address: 00D42FC0, Data: 1

Address: 00D416D8, Data: 3

Address: 00D416F0, Data: 5

Address: 00D41708, Data: 6

Address: 00D41720, Data: 8

PS D:\SEMESTER 2\ALGORITMA DAN STRUKTUR DATA\SEM2.C> cd "d:\s

```
f ($?) { .\Pweek8 }
```

3

31

2

123

Sebelum pengurutan:

Address: 00EB2FC0, Data: 31

Address: 00EB16D8, Data: 2

Address: 00EB16F0, Data: 123

Setelah pengurutan:

Address: 00EB2FC0, Data: 2

Address: 00EB16D8, Data: 31

Address: 00EB16F0, Data: 123

PS D:\SEMESTER 2\ALGORITMA DAN STRUKTUR DATA\SEM2.C>