

## TEMPLATE TUGAS

Nama : Muhammad Afzal  
Nim : 1203230039  
Jurusan/Fakultas : Informatika / FTIB  
Mata Kuliah : Asd

---

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

// Definisi struktur node
typedef struct Node {
    int data;           // Menyimpan data integer di node
    struct Node* prev;  // Pointer ke node sebelumnya
    struct Node* next;  // Pointer ke node berikutnya
} Node;

// Fungsi untuk membuat node baru
Node* createNode(int data) {
    Node* newNode = (Node*)malloc(sizeof(Node)); // Alokasi memori untuk node baru
    newNode->data = data; // Mengisi node dengan data
    newNode->prev = NULL; // Inisialisasi prev dengan NULL
    newNode->next = NULL; // Inisialisasi next dengan NULL
    return newNode; // Mengembalikan pointer ke node baru
}

// Fungsi untuk menambahkan node ke dalam list
void append(Node** head, int data) {
    Node* newNode = createNode(data); // Membuat node baru
    if (*head == NULL) { // Jika list kosong
        *head = newNode; // Node baru menjadi head
        newNode->next = newNode; // List sirkular: next menunjuk ke diri sendiri
        newNode->prev = newNode; // List sirkular: prev menunjuk ke diri sendiri
    } else {
        Node* last = (*head)->prev; // Node terakhir dalam list
        last->next = newNode; // Menghubungkan node terakhir ke node baru
        newNode->prev = last; // Menghubungkan node baru ke node terakhir
        newNode->next = *head; // Menghubungkan node baru ke head
    }
}
```

```

        (*head)->prev = newNode;    // Menghubungkan head ke node baru
    }
}

// Fungsi untuk menampilkan list
void printList(Node* head) {
    if (head == NULL) return; // Jika list kosong, keluar dari fungsi
    Node* temp = head;
    do {
        printf("Address: %p, Data: %d\n", (void*)temp, temp->data); //
Mencetak alamat dan data node
        temp = temp->next; // Bergerak ke node berikutnya
    } while (temp != head); // Berhenti jika kembali ke head
}

// Fungsi untuk menukar dua node
void swap(Node* node1, Node* node2) {
    // Menukar data di dalam node
    int tempData = node1->data;
    node1->data = node2->data;
    node2->data = tempData;
}

// Fungsi untuk mengurutkan list
void sortList(Node* head) {
    if (head == NULL) return; // Jika list kosong, keluar dari fungsi
    Node* current, *index;
    int swapped;

    do {
        swapped = 0; // Inisialisasi tidak ada pertukaran
        current = head;

        do {
            index = current->next; // Node berikutnya untuk perbandingan
            if (current->data > index->data) { // Jika data node sekarang
lebih besar dari node berikutnya
                swap(current, index); // Tukar data di antara node
                swapped = 1; // Tanda bahwa terjadi pertukaran
            }
            current = current->next; // Bergerak ke node berikutnya
        } while (current->next != head); // Berhenti jika kembali ke head

    } while (swapped); // Ulangi selama ada pertukaran
}

int main() {
    int N;

```

```

Node* head = NULL;

// Input jumlah data
scanf("%d", &N);

// Input data
for (int i = 0; i < N; i++) {
    int data;
    scanf("%d", &data);
    append(&head, data); // Menambahkan data ke list
}

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

// Urutkan list
sortList(head);

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

return 0;
}

```

```

C double linked list circular.c x
Prj
C:\windows\system32\cmd.exe x + v
5
5
3
8
1
6
List sebelum pengurutan:
Address: 00871878, Data: 5
Address: 00871520, Data: 3
Address: 00871538, Data: 8
Address: 00871550, Data: 1
Address: 00871568, Data: 6
List setelah pengurutan:
Address: 00871878, Data: 1
Address: 00871520, Data: 3
Address: 00871538, Data: 5
Address: 00871550, Data: 6
Address: 00871568, Data: 8
Press any key to continue . . . |

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS [2] C/C++ Compiler Run: Compile - Task

Executing task: C:\MinGW\bin\gcc.exe -Wall -Wextra -g3 d:\KULIAH GWEICH\WATKUL GWEJH-s2\ALPRO\algorithma dan struktur data\Praktikum\pertemuan 9 (circular)\double linke

