

Nama : Ahmad Farhan QF

NIM : 1203230008

Kelas : IF-03-03

TUGAS CIRCULAR DOUBLE LINKED LIST

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

// Struktur Node
struct Node {
    int data;                // variabel untuk menyimpan data
    struct Node* memori_asli; // Menyimpan alamat asli dari node sebelum
    sorting
    struct Node* next;       // Pointer ke node berikutnya
    struct Node* prev;       // Pointer ke node sebelumnya
};

// Fungsi memasukkan angka diakhir
void insertEnd(struct Node** start, int value)
{
    // Jika kosong, buat node
    if (*start == NULL) {
        struct Node* new_node = (struct Node*)malloc(sizeof(struct Node)); //
        Mengalokasikan memori untuk node baru
        new_node->data = value; // Mengatur nilai input pada node baru
        new_node->memori_asli = new_node; // Menyimpan alamat asli dari node
        baru
        new_node->next = new_node->prev = new_node; // Menjadikan node baru
        sebagai next dan prev dari diri sendiri
        *start = new_node; // Menjadikan node baru sebagai awal
        return;
    }

    // Jika tidak kosong

    // Mencari node terakhir
    struct Node* last = (*start)->prev;

    // Membuat node baru, tetapi jika sudah ada node sebelumnya
    struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
    new_node->data = value; // Mengatur nilai input pada node baru
    new_node->memori_asli = new_node; // Menyimpan alamat asli dari node baru

    // Menyisipkan node baru di akhir
```

```

    new_node->next = *start;
    (*start)->prev = new_node;
    new_node->prev = last;
    last->next = new_node;
}

// Fungsi untuk menampilkan output
void display(struct Node* start)
{
    struct Node* temp = start;

    // Menampilkan output
    printf("Output\n");
    printf("Address: %p, Data: %d\n", temp->memori_asli, temp->data); //
    Menampilkan alamat asli dan data dari node pertama
    temp = temp->next;
    while (temp != start) {
        printf("Address: %p, Data: %d\n", temp->memori_asli, temp->data); //
        Menampilkan alamat asli dan data dari node selanjutnya
        temp = temp->next;
    }
}

void sort(struct Node** start) {
    struct Node *temp, *node, *min;
    if (*start == NULL) { // Jika kosong, tampilkan pesan kosong
        printf("List is empty\n");
        return;
    }
    for (temp = *start; temp->next != *start; temp = temp->next) {
        min = temp;
        for (node = temp->next; node != *start; node = node->next) {
            if (node->data < min->data) { // Jika data node berikutnya lebih
            kecil dari data node minimum
                min = node; // Update node minimum
            }
        }
        if (min != temp) { // Jika node minimum tidak sama dengan node saat
        ini
            struct Node* temp = temp->memori_asli; // Simpan alamat asli dari
            node saat ini
            temp->memori_asli = min->memori_asli; // Update alamat asli dari
            node saat ini dengan alamat asli dari node minimum
            min->memori_asli = temp; // Update alamat asli dari node minimum
            dengan alamat asli dari node saat ini

            int tempData = temp->data; // Simpan data dari node saat ini

```

```

        temp->data = min->data; // Update data dari node saat ini dengan
data dari node minimum
        min->data = tempData; // Update data dari node minimum dengan
data yang disimpan sebelumnya dari node saat ini
    }
}

int main()
{
    // Memulai start dengan NULL
    struct Node* start = NULL;

    int banyak_data, data;
    printf("Input\n");
    scanf("%d", &banyak_data);

    for(int i = 0; i < banyak_data; i++) {
        scanf("%d", &data);
        insertEnd(&start, data); // Memasukkan data
    }

    display(start); // Menampilkan sebelum sorting

    // Melakukan sorting
    sort(&start);

    printf("\nAfter sorting:\n");
    display(start); // Menampilkan setelah sorting

    return 0;
}

```

```
PS D:\Visual Studio Code data\Learn\C\Semester 2> cd "d:\Visual Studio C  
ode data\Learn\C\Semester 2\" ; if ($?) { gcc praktikum_ASD_11.c -o prak  
tikum_ASD_11 } ; if ($?) { .\praktikum_ASD_11 }
```

Input

5
5
3
8
1
6

Output

Address: 00BD19A0, Data: 5
Address: 00BD19B8, Data: 3
Address: 00BD1908, Data: 8
Address: 00BD1920, Data: 1
Address: 00BD1938, Data: 6

After sorting:

Output

Address: 00BD1920, Data: 1
Address: 00BD19B8, Data: 3
Address: 00BD19A0, Data: 5
Address: 00BD1938, Data: 6
Address: 00BD1908, Data: 8