

Nama : Henri Kurniawan Candra

Kelas : IF-03-02

NIM : 1203230086

Matkul: ASD

Tugas OTH

Source Code

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

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

typedef struct Node node;

node *pHead = NULL;
node *pTail = NULL;

node *alokasiNodeBaru()
{
    node *pNew = NULL;
    pNew = (node *)malloc(sizeof(node));
    return (pNew);
}

void insert(int data)
{
    node *pNew = alokasiNodeBaru();

    if (pNew == NULL)
    {
        printf("\n[ALOKASI GAGAL]");
    }
    else
    {
        pNew->data = data;
        pNew->prev = NULL;
        pNew->next = NULL;
    }
}
```

```

        if (pHead == NULL)
        {
            pHead = pNew;
            pTail = pNew;
            pHead->next = pHead;
            pHead->prev = pHead;
        }
        else
        {
            pNew->prev = pTail;
            pNew->next = pHead;
            pTail->next = pNew;
            pHead->prev = pNew;
            pTail = pNew;
        }
    }
}

void view()
{
    node *pWalker = pHead;
    int i = 1;

    if (pWalker == NULL)
    {
        printf("\n[DATA KOSONG]");
    }
    else
    {
        printf("\n");
        while (pWalker != pTail)
        {
            printf("%d ", pWalker->data);
            i++;
            pWalker = pWalker->next;
        }
        printf("%d ", pWalker->data);
    }
    printf("\n");
}

void sortNode(node *pWalker, node *pWalkerNext)
{
    node *temp = NULL;

    if (pWalker->data > pWalkerNext->data)
    {

```

```

        if (pWalker == pHead)
        {
            pHead = pWalkerNext;
        }
        if (pWalkerNext == pTail)
        {
            pTail = pWalker;
        }

        if (pWalker->prev != NULL)
        {
            pWalker->prev->next = pWalkerNext;
        }
        if (pWalkerNext->next != NULL)
        {
            pWalkerNext->next->prev = pWalker;
        }

        temp = pWalkerNext->next;
        pWalkerNext->next = pWalker;
        pWalkerNext->prev = pWalker->prev;
        pWalker->next = temp;
        pWalker->prev = pWalkerNext;
    }
}

void viewWithAddress()
{
    node *pWalker = pHead;
    int i = 1;

    if (pWalker == NULL)
    {
        printf("\n[DATA KOSONG]");
    }
    else
    {
        printf("\n");
        while (pWalker != pTail)
        {
            printf("Address: %p | Data: %d\n ", pWalker, pWalker->data);
            i++;
            pWalker = pWalker->next;
        }
        printf("Address: %p | Data: %d\n ", pWalker, pWalker->data);
    }
    printf("\n");
}

```

```

int main()
{
    node *pNew = NULL;
    int numOfData, data;

    printf("Masukkan jumlah data: ");
    scanf("%d", &numOfData);
    for (int i = 0; i < numOfData; i++)
    {
        printf("Masukkan data ke-%d: ", i + 1);
        scanf("%d", &data);
        insert(data);
    }

    printf("\nData awal: ");
    viewWithAddress();
    printf("\nData setelah diurutkan: ");
    sortNode(pHead, pHead->next);
    viewWithAddress();

    return 0;
}

```

Ss Output

```

PS C:\HENRI\Campus\Project semester 2\Alpro\TugasASD> .\DoubleLinkedListCircular.exe
Masukkan jumlah data: 2
Masukkan data ke-1: 22
Masukkan data ke-2: 11

Data awal:
Address: 009818B0 | Data: 22
Address: 009818C8 | Data: 11

Data setelah diurutkan:
Address: 009818C8 | Data: 11
Address: 009818B0 | Data: 22

```

Penjelasan

Struktur dan Tipe Data

struct Node: Mendefinisikan struktur untuk simpul dari linked list dengan tiga anggota: prev (penunjuk ke simpul sebelumnya), data (nilai data yang disimpan dalam simpul), dan next (penunjuk ke simpul berikutnya).

typedef struct Node node: Memberikan alias node untuk struct Node, sehingga node bisa digunakan sebagai tipe data yang lebih ringkas.

****node pHead = NULL, node pTail = NULL:** Mendeklarasikan dua penunjuk global pHead dan pTail yang masing-masing menunjuk ke kepala dan ekor dari linked list.

Fungsi alokasiNodeBaru

alokasiNodeBaru(): Fungsi ini mengalokasikan memori untuk sebuah simpul baru dan mengembalikan penunjuk ke simpul tersebut. Jika alokasi gagal, pNew akan menjadi NULL.

Fungsi insert

insert(int data): Fungsi ini menyisipkan simpul baru dengan nilai data ke dalam linked list. Jika list kosong (pHead == NULL), simpul baru menjadi kepala dan ekor list. Jika list sudah berisi elemen, simpul baru ditambahkan di akhir list dan dihubungkan dengan kepala untuk menjaga circular linked list.

Fungsi view

view(): Fungsi ini mencetak nilai data dari setiap simpul dalam list. Jika list kosong, mencetak [DATA KOSONG]. Loop berjalan dari pHead hingga mencapai pTail dan mencetak data dari setiap simpul.

Fungsi sortNode

sortNode(node *pWalker, node *pWalkerNext): Fungsi ini menukar dua simpul bertetangga jika data dalam pWalker lebih besar dari data dalam pWalkerNext. Fungsi ini hanya melakukan penukaran untuk dua simpul dan tidak mengurutkan seluruh list.

Fungsi viewWithAddress

viewWithAddress(): Fungsi ini mencetak alamat memori dan nilai data dari setiap simpul dalam list. Jika list kosong, mencetak [DATA KOSONG]. Loop berjalan dari pHead hingga mencapai pTail dan mencetak alamat serta data dari setiap simpul.

Fungsi main

main(): Fungsi utama program yang mengumpulkan data dari pengguna dan menyisipkannya ke dalam linked list. Setelah semua data dimasukkan, mencetak list awal, memanggil sortNode untuk mengurutkan dua simpul pertama (bukan seluruh list), lalu mencetak list setelah "pengurutan".