Tugas OTH Algoritma & Struktur Data Week 13

NAMA : AL FACHRI SAGIANTO

NIM : 1203230126

KELAS : IF-03-02

SOURCE CODE

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

typedef struct Node *tnode;

struct Node {
    int data;
    tnode next;
    tnode prev;
} *head = NULL, *tail = NULL;

tnode createNode(int val) {
    tnode temp = (tnode)malloc(sizeof(struct Node));
    temp->data = val;
    temp->next = NULL;
    temp->prev = NULL;
    return temp;
}

void insert_last(int val) {
    tnode temp = createNode(val);
```

```
    if (head == NULL) {
        head = tail = temp;
        head->next = head;
        head->prev = head;
    } else {
        temp->next = head;
        temp->prev = tail;
        tail->next = temp;
        head->prev = temp;
        tail = temp;
    }
}

void swap_nodes(tnode a, tnode b) {
    if (a == b) return;

    tnode aPrev = a->prev;
    tnode aNext = a->next;
    tnode bPrev = b->prev;
    tnode bNext = b->next;

    if (a->next == b) { // a is immediately before b
        a->next = bNext;
        a->prev = b;
        b->next = a;
        b->prev = aPrev;

        if (aPrev != NULL) aPrev->next = b;
```

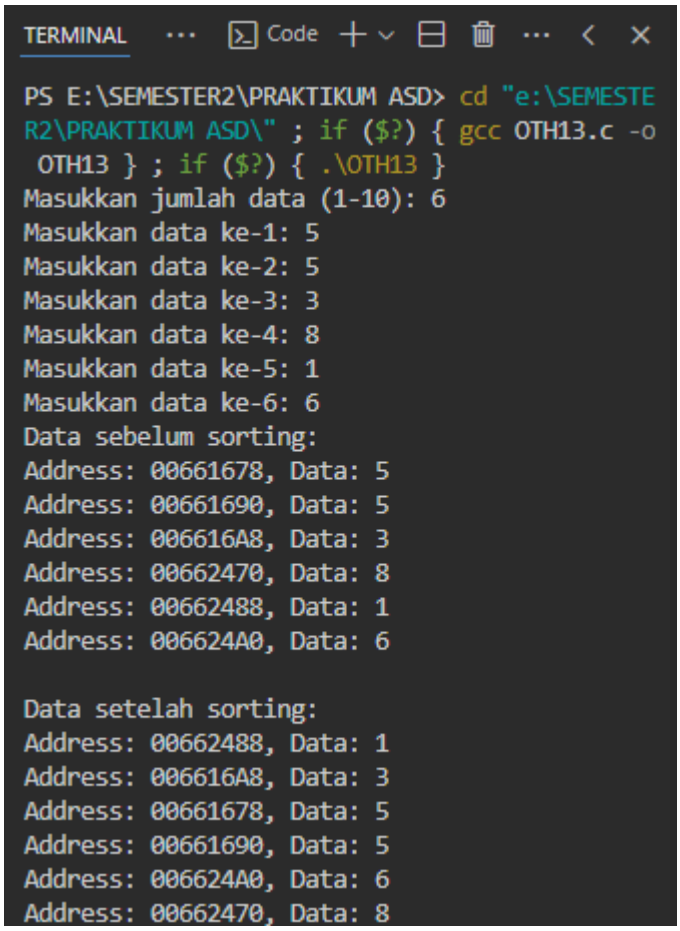
```
    if (bNext != NULL) bNext->prev = a;
} else if (b->next == a) { // b is immediately before a
    b->next = aNext;
    b->prev = a;
    a->next = b;
    a->prev = bPrev;

    if (bPrev != NULL) bPrev->next = a;
    if (aNext != NULL) aNext->prev = b;
} else {
    // Nodes are not adjacent
    a->next = bNext;
    a->prev = bPrev;
    b->next = aNext;
    b->prev = aPrev;

    if (aNext != NULL) aNext->prev = b;
    if (aPrev != NULL) aPrev->next = b;
    if (bNext != NULL) bNext->prev = a;
    if (bPrev != NULL) bPrev->next = a;
}

// Update head and tail if needed
if (head == a) {
    head = b;
} else if (head == b) {
    head = a;
}
```

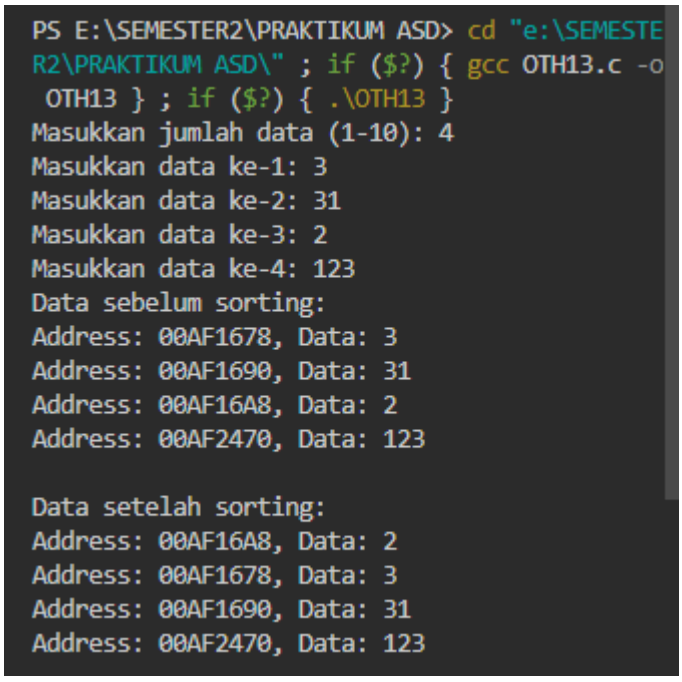
```
    if (tail == a) {
        tail = b;
    } else if (tail == b) {
        tail = a;
    }
}

void sort_ascending() {
    if (head == NULL) return;

    int swapped;
    tnode ptr1;
    tnode lptr = NULL;

    do {
        swapped = 0;
        ptr1 = head;

        do {
            if (ptr1->next != head && ptr1->data > ptr1->next->data) {
                swap_nodes(ptr1, ptr1->next);
                swapped = 1;
            }
            ptr1 = ptr1->next;
        } while (ptr1->next != head);

        lptr = ptr1;
```

```
    } while (swapped);
}

void cetak() {
    if (head == NULL) return;

    tnode temp = head;
    do {
        printf("Address: %p, Data: %d\n", (void*)temp, temp->data);
        temp = temp->next;
    } while (temp != head);
    printf("\n");
}

int main() {
    int jumlah_data, i, nilai;

    printf("Masukkan jumlah data (1-10): ");
    scanf("%d", &jumlah_data);

    if (jumlah_data < 1 || jumlah_data > 10) {
        printf("Jumlah data harus antara 1 dan 10.\n");
        return 1;
    }

    for (i = 0; i < jumlah_data; i++) {
        printf("Masukkan data ke-%d: ", i + 1);
        scanf("%d", &nilai);
```

```
        insert_last(nilai);
    }

    printf("Data sebelum sorting:\n");
    cetak();

    sort_ascending();

    printf("Data setelah sorting:\n");
    cetak();

    return 0;
}
```

OUTPUT


TERMINAL
Code
PS E:\SEMESTER2\PRAKTIKUM ASD> cd "e:\SEMESTE
R2\PRAKTIKUM ASD\" ; if ($?) { gcc OTH13.c -o
OTH13 } ; if ($?) { .\OTH13 }
Masukkan jumlah data (1-10): 6
Masukkan data ke-1: 5
Masukkan data ke-2: 5
Masukkan data ke-3: 3
Masukkan data ke-4: 8
Masukkan data ke-5: 1
Masukkan data ke-6: 6
Data sebelum sorting:
Address: 00661678, Data: 5
Address: 00661690, Data: 5
Address: 006616A8, Data: 3
Address: 00662470, Data: 8
Address: 00662488, Data: 1
Address: 006624A0, Data: 6
Data setelah sorting:
Address: 00662488, Data: 1
Address: 006616A8, Data: 3
Address: 00661678, Data: 5
Address: 00661690, Data: 5
Address: 006624A0, Data: 6
Address: 00662470, Data: 8


PS E:\SEMESTER2\PRAKTIKUM ASD> cd "e:\SEMESTE
R2\PRAKTIKUM ASD\" ; if ($?) { gcc OTH13.c -o
OTH13 } ; if ($?) { .\OTH13 }
Masukkan jumlah data (1-10): 4
Masukkan data ke-1: 3
Masukkan data ke-2: 31
Masukkan data ke-3: 2
Masukkan data ke-4: 123
Data sebelum sorting:
Address: 00AF1678, Data: 3
Address: 00AF1690, Data: 31
Address: 00AF16A8, Data: 2
Address: 00AF2470, Data: 123
Data setelah sorting:
Address: 00AF16A8, Data: 2
Address: 00AF1678, Data: 3
Address: 00AF1690, Data: 31
Address: 00AF2470, Data: 123