Nama : Haidar Fulca Kurniawan

NIM : 1203230077

Kelas : IF 03-02

TUGAS PRAKTIKUM OTH ALGORITMA STRUKTUR DATA

SOAL 1

- Source Code

```c
#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->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->next = aNext;
        b->prev = a;
        a->next = b;
        a->prev = bPrev;

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

        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;
    }

    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 yang ingin diinputkan: ");
    scanf("%d", &jumlah_data);
    printf("\n");

    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("\nData sebelum sorting:\n");
    cetak();

    sort_ascending();

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

    return 0;
}
```

- Output

```
Masukkan jumlah data yang ingin diinputkan: 5

Masukkan data ke-1: 5
Masukkan data ke-2: 3
Masukkan data ke-3: 8
Masukkan data ke-4: 1
Masukkan data ke-5: 6

Data sebelum sorting:
Address: 009E13B0, Data: 5
Address: 009E0DE8, Data: 3
Address: 009E0E00, Data: 8
Address: 009E0E18, Data: 1
Address: 009E0E30, Data: 6
Data setelah sorting:
Address: 009E0E18, Data: 1
Address: 009E0DE8, Data: 3
Address: 009E13B0, Data: 5
Address: 009E0E30, Data: 6
Address: 009E0E00, Data: 8
```

```
Masukkan jumlah data yang ingin diinputkan: 3

Masukkan data ke-1: 31
Masukkan data ke-2: 2
Masukkan data ke-3: 123

Data sebelum sorting:
Address: 00B013B0, Data: 31
Address: 00B00DE8, Data: 2
Address: 00B00E00, Data: 123

Data setelah sorting:
Address: 00B00DE8, Data: 2
Address: 00B013B0, Data: 31
Address: 00B00E00, Data: 123
```

- Penjelasan

- Program Code C ini bertujuan untuk membuat dan mengurutkan sebuah double circular linked list berdasarkan input dari pengguna. Pengguna dapat memasukkan sejumlah data yang kemudian akan diurutkan secara ascending.

- Deklarasi Struktur Node

- typedef struct Node *tnode, untuk Mendefinisikan tnode sebagai pointer ke struktur Node, mempermudah referensi ke node.
- struct Node ialah Struktur data dengan tiga anggota: int data (menyimpan nilai), tnode next (pointer ke node berikutnya), dan tnode prev (pointer ke node sebelumnya).
- *head = NULL, *tail = NULL ialah Inisiasasi pointer head dan tail sebagai NULL, menunjuk ke awal dan akhir dari linked list.

- Fungsi createNode

- tnode createNode, untuk Membuat node baru dengan nilai val.
  tnode temp = (tnode)malloc(sizeof(struct Node)), hal ini dilakukan untuk Mengalokasikan memori untuk node baru.
- temp->data = val; temp->next = NULL; temp->prev = NULL, untuk Mengisi data node, mengatur next dan prev sebagai NULL.
- return temp, untuk Mengembalikan pointer ke node baru.

- Fungsi insert_last

Fungsi ini berfungsi untuk menambahkan node baru di akhir linked list. Hal ini dilakukan karena agar data bisa terinput sesuai baris yang diinputkan (dari depan ke belakang).

- tnode temp = createNode(val), untuk membuat Membuat node baru dengan nilai val.
- if (head == NULL) { ... }, Jika linked list kosong, head dan tail menunjuk ke node baru, membuat linked list circular.
- else { ... }, Jika linked list tidak kosong, node baru ditambahkan di akhir, pointer next dan prev dari node terkait diatur.

- Fungsi swap_nodes

Fungsi ini berfungsi untuk Menukar posisi dua node dalam linked list.

- if (a == b) return, Jika node yang ditukar adalah node yang sama, maka tidak melakukan apa-apa.

Fungsi ini memiliki bagian yang berguna sebagai berikut :

- Menyimpan pointer prev dan next dari kedua node.
- Menukar pointer prev dan next dari node a dan b.
- Mengatur ulang head dan tail jika salah satu node yang ditukar adalah head atau tail.

- Fungsi sort_ascending

Fungsi ini berguna untuk mengurutkan linked list secara ascending menggunakan metode bubble sort.

Fungsi ini memiliki bagian yang berguna sebagai berikut :

- if (head == NULL) return, Jika linked list kosong, maka akan keluar dari fungsi.
- Menggunakan do-while loop untuk iterasi hingga tidak ada lagi penukaran yang diperlukan.
- Dalam setiap iterasi, membandingkan nilai data dari node saat ini dan node berikutnya, menukar posisi node jika diperlukan.

- Fungsi cetak

Fungsi ini untuk mencetak alamat dan data dari setiap node dalam linked list.

Fungsi ini memiliki bagian yang berguna sebagai berikut :

- if (head == NULL) return, Jika linked list kosong, maka keluar dari fungsi.
- Menggunakan do-while loop untuk mencetak data dari setiap node hingga kembali ke head.

- Fungsi main

Berikut bagian bagian pada main program :

- jumlah_data, i, nilai, Mendeklarasikan variabel untuk menyimpan jumlah data, indeks loop, dan nilai data dari pengguna.
- printf("Masukkan jumlah data yang ingin diinputkan: "); scanf("%d", &jumlah_data), Meminta pengguna memasukkan jumlah data.
- if (jumlah_data < 1 || jumlah_data > 10) { ... }, Memeriksa validitas jumlah data (antara 1 dan 10), keluar jika tidak valid.
- Lalu digunakan for loop untuk meminta pengguna memasukkan nilai data dan menambahkan data ke linked list dengan insert_last.
- printf("\nData sebelum sorting:\n"); cetak(), bmemanggil fungsi cetak() yang berfungsi untuk untuk Mencetak data sebelum diurutkan.

- sort_ascending(), untuk memanggil fungsi sort_ascending yang berfungsi Mengurutkan linked list secara ascending.
- printf("Data setelah sorting:\n"); cetak(), memanggil fungsi cetak() yang berfungsi untuk Mencetak data setelah diurutkan.