Nama : La Ode Muhammad Gazali
NIM : 222212696
Kelas : 2KS2

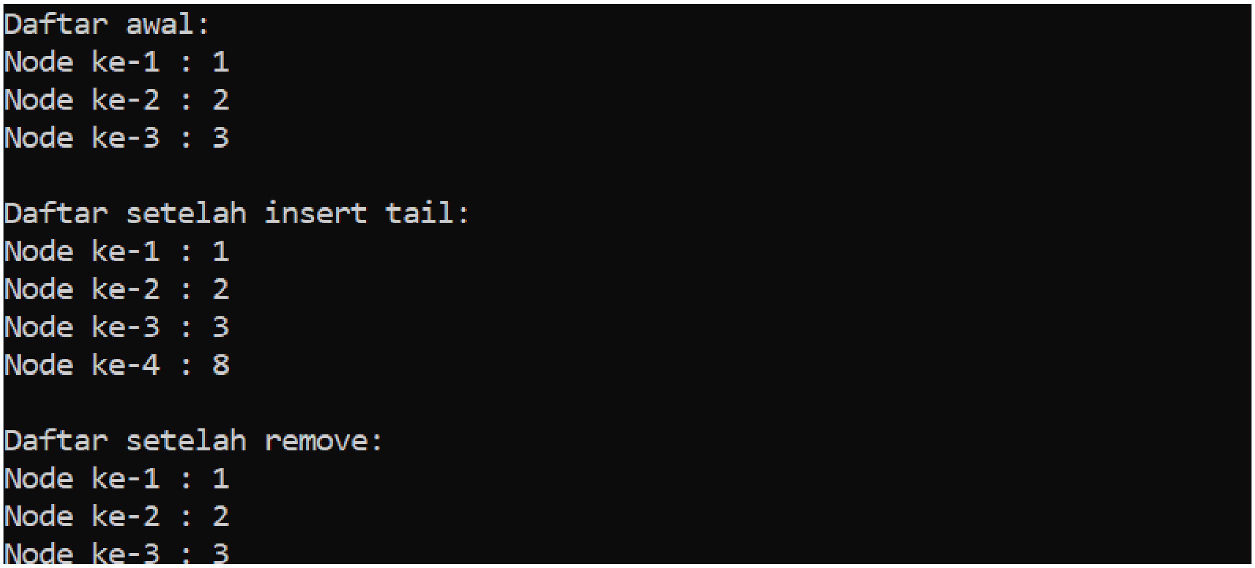
Modul 5 Praktikum Sturktur Data

1. Cobalah untuk memod**if**ikasi potongan program pada pembuatan simpul awal, insert kanan, dan delete kanan sehingga pointer tail dideklarasikan dan selalu diperbaharui isinya saat penambahan dan penghapusan simpul dari kanan.

Program :

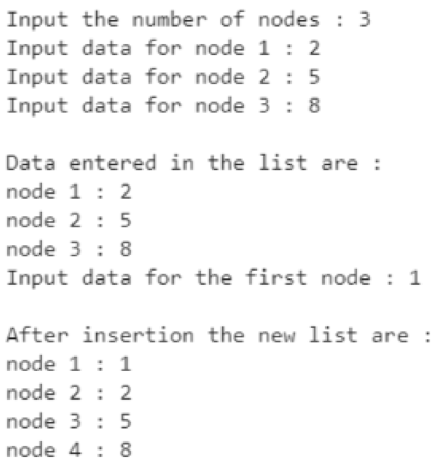
```
#include<stdio.h>
#include<stdlib.h>

struct node{
    int value;
    struct node *next;
    struct node *prev;
};

typedef struct node *ptrnode;

ptrnode createNode(int nilai){
    ptrnode p;
    p = (ptrnode)malloc(sizeof(struct node));
    p->value = nilai;
    p->next = NULL;
    p->prev = NULL;
    return p;
}
ptrnode insert_tail(ptrnode head, ptrnode *tail, int nilai) {
    ptrnode new_node = createNode(nilai);
    new_node->next = NULL;

    if (head == NULL) {
        new_node->prev = NULL;
        head = new_node;
        *tail = new_node;
     // Perbarui tail saat memasukkan elemen pertama
    } else {
        new_node->prev = *tail;
        (*tail)->next = new_node;
        *tail = new_node;
    }
    return head;
}

ptrnode remove_last(ptrnode head, ptrnode *tail) {
    if (head == NULL) {
        printf("Daftar kosong, tidak ada yang dapat dihapus\n");
        return head;
    }
    if (*tail == head) {
        free(head);
        *tail = NULL;
        return NULL;
    }
    ptrnode temp = *tail;
    *tail = (*tail)->prev;
    (*tail)->next = NULL;
    free(temp);

    return head;
}

ptrnode tampilnilai(ptrnode head) {
    ptrnode current = head;
    int i=1;
    while (current != NULL) {
        printf("Node ke-%d : %d",i, current->value);
        current = current->next;
        i++;
        printf("\n");
    }
    printf("\n");

    return head;
}
int main(){
    ptrnode head = NULL;
    ptrnode tail = NULL;

    head=insert_tail(head, &tail, 1);
    head=insert_tail(head, &tail, 2);
    head=insert_tail(head, &tail, 3);

    printf("Daftar awal:\n");
    tampilnilai(head);

    insert_tail(head,&tail, 8);
    printf("Daftar setelah insert tail:\n");
    tampilnilai(head);

    head = remove_last(head, &tail);
    printf("Daftar setelah remove:\n");
    tampilnilai(head);


    return 0;

}
```

Output:



```
Daftar awal:
Node ke-1 : 1
Node ke-2 : 2
Node ke-3 : 3

Daftar setelah insert tail:
Node ke-1 : 1
Node ke-2 : 2
Node ke-3 : 3
Node ke-4 : 8

Daftar setelah remove:
Node ke-1 : 1
Node ke-2 : 2
Node ke-3 : 3
```

2. Buat sebuah program untuk menampilkan output di bawah ini menggunakan double linked list!

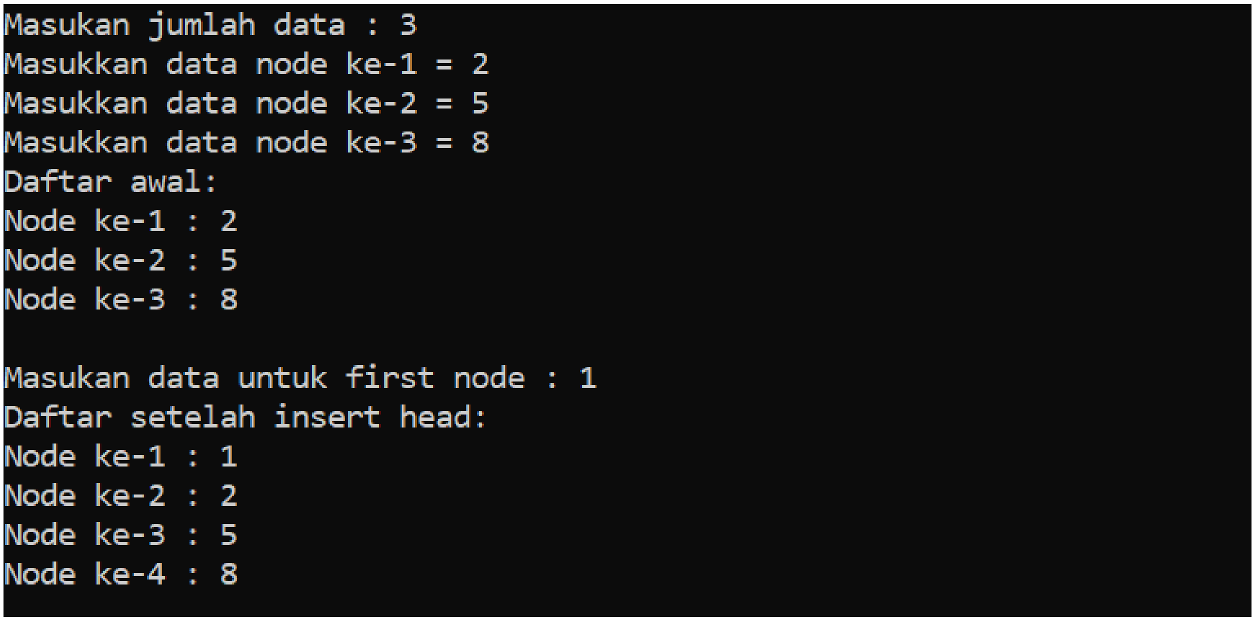


```
Input the number of nodes : 3
Input data for node 1 : 2
Input data for node 2 : 5
Input data for node 3 : 8

Data entered in the list are :
node 1 : 2
node 2 : 5
node 3 : 8
Input data for the first node : 1

After insertion the new list are :
node 1 : 1
node 2 : 2
node 3 : 5
node 4 : 8
```

Program :

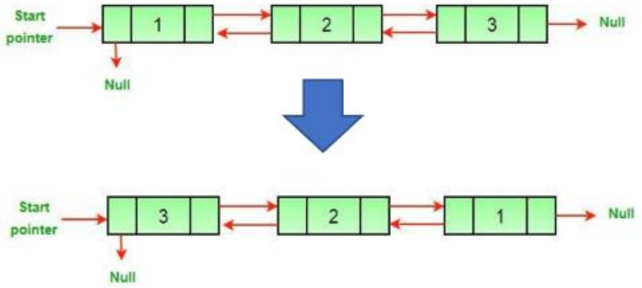
```
#include<stdio.h>
#include<stdlib.h>

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

typedef struct node* mynode;

mynode createNode(int nilai){
    mynode temp;
    temp = (mynode)malloc(sizeof(struct node));;
    temp->data = nilai;
    temp->next = NULL;
    temp->prev = NULL;
}

void insert_head(mynode* head, int data) {
    mynode newNode = createNode(data);
    if (*head == NULL) {
        *head = newNode;
    } else {
        newNode->next = *head;
        (*head)->prev = newNode;
        *head = newNode;
    }
}

mynode insert_tail(mynode head, mynode *tail, int data) {
    mynode new_node = createNode(data);
    new_node->next = NULL;

    if (head == NULL) {
        new_node->prev = NULL;
        head = new_node;
        *tail = new_node;
// Perbarui tail saat memasukkan elemen pertama
    } else {
        new_node->prev = *tail;
        (*tail)->next = new_node;
        *tail = new_node;
    }
    return head;
}


mynode tampilnilai(mynode head) {
    mynode current = head;
    int i=1;
    while (current != NULL) {
        printf("Node ke-%d : %d",i, current->data);
        current = current->next;
        i++;
        printf("\n");
    }
    printf("\n");

    return head;
}

int main(){
    int data,n;
    mynode head = NULL;
    mynode tail = NULL;

    printf("Masukan jumlah data : ");scanf("%d",&n);
    for (int i = 0; i < n; i++) {
        printf("Masukkan data node ke-%d = ", i + 1);
        scanf("%d", &data);
        head = insert_tail(head, &tail, data);
    }

    printf("Daftar awal:\n");
    tampilnilai(head);

    printf("Masukan data untuk first node : ");
    scanf("%d", &data);
    insert_head(&head, data);

    printf("Daftar setelah insert head:\n");
    tampilnilai(head);


return 0;

}
```

Output:



```
Masukan jumlah data : 3
Masukkan data node ke-1 = 2
Masukkan data node ke-2 = 5
Masukkan data node ke-3 = 8
Daftar awal:
Node ke-1 : 2
Node ke-2 : 5
Node ke-3 : 8

Masukan data untuk first node : 1
Daftar setelah insert head:
Node ke-1 : 1
Node ke-2 : 2
Node ke-3 : 5
Node ke-4 : 8
```

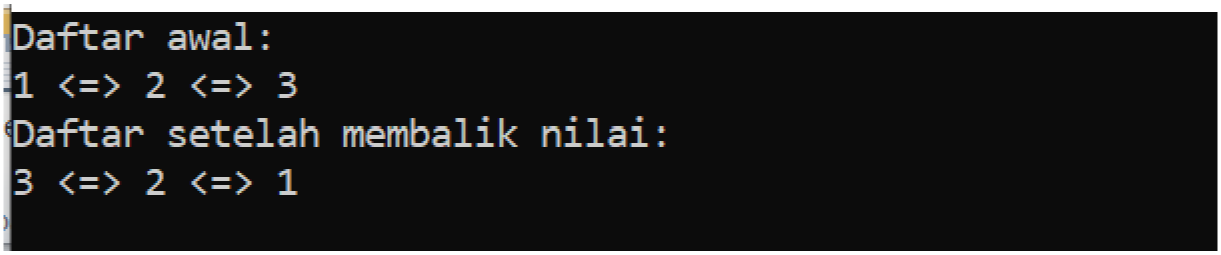
3. Bagaimana untuk membalik nilai-nilai dalam double linked list (tail ke head)?



Program:

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

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

typedef struct node* mynode;

mynode createNode(int nilai){
    mynode temp;
    temp = (mynode)malloc(sizeof(struct node));;
    temp->data = nilai;
    temp->next = NULL;
    temp->prev = NULL;
}

void insert_head(mynode* head, int data) {
    mynode newNode = createNode(data);
    if (*head == NULL) {
        *head = newNode;
    } else {
        newNode->next = *head;
        (*head)->prev = newNode;
        *head = newNode;
    }
}

void membalik_nilai(mynode* head) {
    mynode current = NULL;
    mynode temp = NULL;

    while (*head != NULL) {
        temp = (*head)->prev;
        (*head)->prev = (*head)->next;
        (*head)->next = temp;
        current = *head;
        *head = (*head)->prev;
    }

    if (current != NULL) {
        *head = current;
    }
}

void tampilnilai(mynode head) {
    mynode current = head;
    while (current != NULL) {
        printf("%d", current->data);
        current = current->next;
        if (current != NULL)
            printf(" <=> ");
    }
    printf("\n");
}

int main(){
    int data,n;
    char pilih;
    mynode head = NULL;
    mynode tail = NULL;

    insert_head(&head, 3);
    insert_head(&head, 2);
    insert_head(&head, 1);

    printf("Daftar awal:\n");
    tampilnilai(head);

    membalik_nilai(&head);
    printf("Daftar setelah membaik nilai:\n");
    tampilnilai(head);

    return 0;

}
```

Output:



```
Daftar awal:
1 <=> 2 <=> 3
Daftar setelah membalik nilai:
3 <=> 2 <=> 1
```