

Nama : Muhammad Rizky Ramadhan

NIM : 1203230055

Kelas : Informatika

Mata Kuliah : Algoritma Sistem Data

## 1. 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);
}
//fungsi ini digunakan untuk mengalokasikan memori baru sebesar ukuran sebuah
node.

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

    if (pNew == NULL)
    {
        printf("\n[ALOKASI TIDAK BERHASILL]");
    }
    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;
    }
}

//fungsi ini digunakan untuk menyisipkan data ke dalam antrian. Jika alokasi
memori gagal, maka akan menampilkan pesan "[ALOKASI TIDAK BERHASIL]".

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

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

```

//fungsi ini digunakan untuk menampilkan semua data yang ada dalam antrian. Jika antrian kosong, maka akan menampilkan pesan "[DATA TIDAK ADA]".

```
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;
```

```
    }
```

```
}
```

//fungsi ini digunakan untuk mengurutkan data yang ada dalam antrian dengan metode bubble sort.

```
void viewWithAddress()
```

```
{
```

```
    node *pWalker = pHead;
```

```
    int i = 1;
```

```
    if (pWalker == NULL)
```

```
    {
```

```
        printf("\n[DATA TIDAK ADA]");
```

```

    }
    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");
}
//fungsi ini digunakan untuk menampilkan semua data beserta alamat memori yang
dimilikinya. Jika antrian kosong, maka akan menampilkan pesan "[DATA TIDAK ADA]".

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();
    //Dalam main(), program meminta pengguna untuk memasukkan jumlah data yang akan
    disimpan dalam antrian.
    //Setelah itu, program akan meminta pengguna untuk memasukkan data-data tersebut.
    //Setelah data-data telah disimpan, program akan menampilkan data-data awal dan
    data-data setelah diurutkan.

    return 0;
}

```

Penjelasan :

Program ini akan meminta input jumlah data dan masing-masing data dari user, lalu menambahkan data ke linked list. Setelah itu, program akan mencetak linked list sebelum dan sesudah diurutkan secara ascending

Output :

```
Masukan jumlah data : 3
```

```
Inputkan data ke-1: 31
```

```
Inputkan data ke-2: 2
```

```
Inputkan data ke-3: 123
```

```
Address: 0x16afac0, Data: 31
```

```
Address: 0x16afae0, Data: 2
```

```
Address: 0x16afb00, Data: 123
```

```
Address: 0x16afae0, Data: 2
```

```
Address: 0x16afac0, Data: 31
```

```
Address: 0x16afb00, Data: 123
```

Masukan jumlah data : 5

Inputkan data ke-1: 5

Inputkan data ke-2: 3

Inputkan data ke-3: 8

Inputkan data ke-4: 1

Inputkan data ke-5: 6

Address: 0x101cac0, Data: 5

Address: 0x101cae0, Data: 3

Address: 0x101cb00, Data: 8

Address: 0x101cb20, Data: 1

Address: 0x101cb40, Data: 6

Address: 0x101cb20, Data: 1

Address: 0x101cae0, Data: 3

Address: 0x101cac0, Data: 5

Address: 0x101cb40, Data: 6

Address: 0x101cb00, Data: 8