

LAPORAN PRAKTIKUM
SESI 7
PRAKTIKUM COMP6362 – DATA STRUCTURES
KELAS BE20



Oleh :
2440008600 – Andru Baskara Putra

SEMESTER GENAP 2020/2021
BINA NUSANTARA UNIVERSITY
MALANG

A. Kode Program

Source Code

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

struct data{
    struct data *prev;
    int rate;
    char genre[200];
    char title[200];
    struct data *portal;
};

typedef struct data node;
node *first =NULL;
node *last=NULL;

void tambahDepan(char judul[],char jenis[],int nilai){
    node *nodebaru = (node *)malloc(sizeof(node));

    strcpy(nodebaru->title, judul);
    strcpy(nodebaru->genre, jenis);
    nodebaru->rate=nilai;

    nodebaru->portal=nodebaru;
    nodebaru->prev=nodebaru;

    if (first==NULL)
    {
        first=nodebaru;
        last=nodebaru;
        printf("\n---Film telah di tambahkan---\n");
    }else{
        nodebaru->portal=first;
        nodebaru->prev=last;
        first->prev=nodebaru;
        last->portal=nodebaru;
        first=nodebaru;
        printf("\n---Film telah di tambahkan---\n");
    }
}

void tambahAkhir(char judul[],char jenis[],int nilai){
    node *nodebaru = (node *)malloc(sizeof(node));
```

```

strcpy(nodebaru->title, judul);
strcpy(nodebaru->genre, jenis);
nodebaru->rate=nilai;

nodebaru->portal=nodebaru;
nodebaru->prev=nodebaru;

if (first==NULL)
{
    first=nodebaru;
    last=nodebaru;
    printf("\n---Film telah di tambahkan---\n");
}else{
    last->portal=nodebaru;
    nodebaru->portal=first;
    nodebaru->prev=last;
    last=nodebaru;
    first->prev=last;
    printf("\n---Film telah di tambahkan---\n");
}
}

void tambahKeinginan(char judul[],char jenis[],int nilai){
    int pos;
    printf("Enter a Position : ");
    scanf("%d",&pos);
    if (pos==1)
    {
        tambahDepan(judul,jenis,nilai);
        return;
    }else if (pos>1 && first!=NULL)
    {
        node *curr =first;
        node *scan = (node *)malloc(sizeof(node));
        int count =0;

        do
        {
            count++;
            scan=curr;
            curr=curr->portal;
        } while (curr->portal != first && count<pos-1);

        if (count==pos-1)
        {
            if (scan==last)
            {

```

```

        tambahAkhir(judul,jenis,nilai);
    }else
    {
        node *nodebaru = (node *)malloc(sizeof(node));
        strcpy(nodebaru->title, judul);
        strcpy(nodebaru->genre, jenis);
        nodebaru->rate=nilai;

        scan->portal=nodebaru;
        nodebaru->portal=curr;
        nodebaru->prev=scan;
        curr->prev=nodebaru;
        printf("\n---Film telah di tambahkan---\n");
    }
    return;

}

}

printf("\nPOSISI TIDAK DITEMUKAN!!!\n");

}

void hapusDepan(){
    if (first==NULL)
    {
        printf("\nKOSONG!!!\n");
        return;
    }else if (first==last)
    {
        printf("\n--- Film telah di tonton ---\n");
        free(first);
        first=last=NULL;
    }else
    {
        node *scan = first;

        last->portal = first->portal;
        first = first->portal;
        first->prev=last;
        printf("\n--- Film telah di tonton ---\n");
        free(scan);
    }
}

```

```

}

void hapusAkhir(){
    if (first==NULL)
    {
        printf("\nKOSONG!!!\n");
        return;
    }

    if (first==last)
    {
        printf("\n--- Film telah di tonton ---\n");
        free(first);
        first=last=NULL;
    }

    node *scan = first;
    node *curr = first;

    while (curr->portal != first)
    {
        scan = curr;
        curr =curr->portal;
    }

    scan->portal=first;
    last=scan;
    first->prev=last;
    printf("\n--- Film telah di tonton ---\n");
    free(curr);
}

void hapusKeinginan(){
    int pos;
    printf("Enter a Position : ");
    scanf("%d",&pos);
    if (!first)
    {
        printf("\nKOSONG!!!\n");
        return;
    }

    if (pos==1)
    {
        hapusDepan();
    }
}

```

```

        return;
    }

    node *curr =first;
    node *scan;
    int count = 0;

    do
    {
        count++;
        scan=curr;
        curr=curr->portal;
    } while (curr->portal != first && count<pos-1);

    if (count==pos-1)
    {
        if (curr==last)
        {
            hapusAkhir();
            return;
        }
        scan->portal=curr->portal;
        curr->portal->prev=scan;
        printf("\n--- Film telah di tonton ---\n");
        free(curr);
        return;
    }

    printf("\nPosisi tidak ditemukan!!!\n");
}

```

```

void showlist(){
    if (first==NULL)
    {

        printf("+-----+-----+-----+-----+-----+
+\\n" );
        printf("| No. |   Name of Movie   |           Genre           |           Rate
|\\n");
        printf("+-----+-----+-----+-----+-----+
+\\n" );
        printf("|
|\\n");

```

```

        printf("|
        --- No Movies Data Available --
        |\n");
        printf("|\n");
        printf("+-----+-----+-----+-----+-----+
+\\n" );
        return;
    }

    node *curr = first;
    int i=1;

    printf("+-----+-----+-----+-----+-----+
+\\n" );
    printf("| No. |   Name of Movie   |           Genre           |           Rate
    |\n");
    printf("+-----+-----+-----+-----+-----+
+\\n" );
    do
    {
        printf("| %d | %-12s | %-16s | %-6d |\\n",i++,curr-
>title,curr->genre,curr->rate);
        curr = curr->portal;
    } while (curr != first);
    printf("+-----+-----+-----+-----+-----+
+\\n" );
}

void urutData(){
    node *curr =first;
    node *index=NULL;
    int scan;
    char scan_1[200];
    char scan_2[200];

    if (first==NULL)
    {
        printf("\\n Gaada yg bisa diurut\\n");
        return;
    }else
    {
        do
        {
            index=curr->portal;
            while (index != first)
            {

```

```

        if (strcmpi(curr->title,index->title)>0)
        {
            strcpy(scan_1,curr->title);
            strcpy(curr->title,index->title);
            strcpy(index->title,scan_1);
            strcpy(scan_2,curr->genre);
            strcpy(curr->genre,index->genre);
            strcpy(index->genre,scan_2);
            scan =curr->rate;
            curr->rate= index->rate;
            index->rate = scan;
        }
        index= index->portal;
    }
    curr =curr->portal;
} while (curr->portal != first);

}

}

int main(){
    int opsi; char jud[200]; char gen[200]; int rat; int letak;
    while(1){
        showlist();
        printf("\n1.Tambah Depan\n2.Tambah Akhir\n3.Tambah Keinginan\n4.Tonton D
        epan\n5.Tonton Akhir\n6.Tonton keinginan\n7.urut berdasarkan judul\n8.Kill p
        rogram\n");
        printf(">>>>");scanf("%d",&opsi);

        switch (opsi)
        {
            case 1:
                printf("\nMasukan Judul : " ); scanf("%s",jud);
                printf("\nMasukan genre : "); scanf("%s",gen);
                re1:
                printf("\nMasukan Rating penilaian : "); scanf("%d",&rat);
                if (rat<1 || rat>10)
                {
                    printf("\nHanya bisa 1-10!\n");
                    goto re1;
                }

                tambahDepan(jud,gen,rat);

```



```

        break;

    case 2:
        printf("\nMasukan Judul : " ); scanf("%s",jud);
        printf("\nMasukan genre : "); scanf("%s",gen);
        re2:
        printf("\nMasukan Rating penilaian : "); scanf("%d",&rat);
        if (rat<1 || rat>10)
        {
            printf("\nHanya bisa 1-10!\n");
            goto re2;
        }
        tambahAkhir(jud,gen,rat);
        break;

    case 3:
        printf("\nMasukan Judul : " ); scanf("%s",jud);
        printf("\nMasukan genre : "); scanf("%s",gen);
        re3:
        printf("\nMasukan Rating penilaian : "); scanf("%d",&rat);
        if (rat<1 || rat>10)
        {
            printf("\nHanya bisa 1-10!\n");
            goto re3;
        }
        tambahKeinginan(jud,gen,rat);
        break;

    case 4:
        hapusDepan();
        break;

    case 5:
        hapusAkhir();
        break;

    case 6:

        hapusKeinginan(letak);
        break;

    case 7:
        urutData();
        break;

    case 8:
        exit(0);
        break;

```

```
        default:
            printf("\nSalah gan\n");
            break;
    }
}
return 0;
}
```

B. Deskripsi Program

1. Struct data

Berisikan kumpulan data-data yang akan digunakan dalam program nanti

2. Void tambahDepan()

Merupakan fungsi dimana program akan menambah node di bagian depan rangkaian linked list

3. Void tambahAkhir()

Merupakan fungsi dimana program akan menambah node dibagian belakang rangkaian linkedlist

4. Void tambahKeinginan()

Merupakan fungsi dimana program akan menambah data sesuai dengan posisi yang diinginkan user

5. Void hapusDepan()

Merupakan fungsi dimana program akan menghapus data yang ada dibagian depan rangkaian linkedlist

6. Void hapusAkhir()

Merupakan fungsi dimana program akan menghapus data yang ada dibagian akhir/belakang rangkaian data

7. Void hapusKeinginan()

Merupakan fungsi dimana program akan menghapus data yang dimana posisi penghapusan nya dapat ditentukan user.

8. Void showlist()

Fungsi ini akan memunculkan data-data yang telah dibuat ke layar

9. Void urutData()

Fungsi dimana data(nama) akan diurutkan berdasarkan alfabet, dari a-z.

10. Void main

Berisikan proses utama dari program, user akan diperlihatkan beberapa pilihan untuk menambah data, menghapus data, mengurut data dan juga memberhentikan jalan nya program.

Bukti Presentasi

Khusus lab 7 saya tidak dapat presentasi, dikarenakan saya terkena demam.