

**LAPORAN PRAKTIKUM**  
**SESI 4**  
**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 untuk linkedlist
struct andrustacklink
{
    char title[200];
    char artist[200];
    struct andrustacklink *portal;
}*top, *roof, *scan;

//struct untuk array
struct track{
    char judul[200];
    char artis[200];
};

void line(){
    printf("_____\\n\\n");
}

int jumlah=0;
// linkedlist
void buat(){
    top=NULL;
}
//dari baris 30-
191, ditujukan untuk stack implementasi linkedlist, sedangkan baris 193-
285 ditujukan untuk stack implementasi array
void trackCount(){
    printf("\\n Jumlah track musik yang ada di dalam daftar : %d",jumlah);
}

void tambahData(char judul[], char musisi[]){
    if (top==NULL)
    {
        top=(struct andrustacklink *)malloc(1*sizeof(struct andrustacklink))
;
        top->portal=NULL;
        strcpy(top->title,judul);
        strcpy(top->artist,musisi);
    }else{
```

```

        scan=(struct andrustacklink *)malloc(1*sizeof(struct andrustacklink)
);
        scan->portal=top;
        strcpy(scan->title,judul);
        strcpy(scan->artist,musisi);
        top=scan;
    }
    jumlah++;
}

void showlist(){
    roof=top;
    line();
    printf("|      Musik      |      Artis      |\n");
    line();
    if (roof==NULL)
    {
        printf("Musik yang hanya bisa didengar dengan bahasa kalbu\n\n");
        return;
    }

    while (roof != NULL)
    {
        if (roof==top)
        {
            printf("|      %-9s      |      %-9s      |<--TOP\n",roof->title,roof->
artist);
        }else{
            printf("|      %-9s      |      %-9s      |\n",roof->title,roof->
artist);
        }

        /*printf("\nMusik : %s ",roof->title);
        printf("\nMusisi : %s ",roof->artist);*/

        roof=roof->portal;
    }
}

void hapus(){
    roof=top;

    if (roof==NULL)
    {

```

```

        printf("\n Anda menghapus musik dengan bahasa kalbu, luar biasa anak
indigo");
        return;
    }
    else
        roof = roof->portal;
    printf("\n\n Berhasil memainkan lagu : %s oleh %s\n\n", top->title, top-
>artist);
    free(top);
    top = roof;
    jumlah--;
}

void Judulmusikterbaru(){
    printf("\nJudul : %s",top->title);
}

void Artismusikterbaru(){
    printf("\nArtis : %s\n",top->artist);
}

void cekMusik(){
    if (top==NULL)
    {
        printf("\nTrack musik kosong");
    }else{
        printf("\nTrack musik berisi, terdapat %d track",jumlah);
    }
}

void HapusSemua(){
    roof = top;

    while (roof != NULL)
    {
        roof=top->portal;
        free(top);
        top=roof;
        roof=roof->portal;
    }
    free(roof);
    top=NULL;

    printf("\n Tracklist telah dikosongkan");
    jumlah=0;
}
}

```

```

void linkedlist(){
    int no, ch, e;
    char jdl[200]; char art[200];

    buat();

    while (1)
    {
        showlist();
        printf("\n 1 - Tambah Lagu");
        printf("\n 2 - Tampilkan Lagu Terbaru");
        printf("\n 3 - Mainkan Lagu");
        printf("\n 4 - Tampilkan Tracklist");
        printf("\n 5 - Kill Program");
        printf("\n Enter choice : ");
        scanf("%d", &ch);

        switch (ch)
        {
            case 1:
                system("clear");
                printf("Enter data : ");
                printf("\nJudul : "); scanf("%s", jdl);
                printf("Artis : "); scanf("%s", art);
                tambahData(jdl, art);
                break;
            case 2:
                system("clear");
                if (top == NULL)
                    printf("No elements in stack");
                else
                {
                    printf("track terbaru :\n");
                    Judulmusikterbaru();
                    Artismusikterbaru();
                }
                break;
            case 3:
                system("clear");
                hapus();
                break;
            case 4:
                system("clear");
                showlist();
                system("clear");

```

```

        break;
    case 5:
        exit(0);
    default :
        system("clear");
        printf(" Opsi tidak valid!! silahkan ulangi lagi\n ");
        break;
    }
}

void ProccesOfStackArray(struct track *list, int data){
    line();
    printf("|      Musik      |      Artis      |\n");
    line();
    if (data==0)
    {
        printf("\nMusik dengan bahasa kalbu\n\n");
    }else{
        for (int i = data-1; i>=0; i--)
        {
            //printf("%s      %s",list[i].judul, list[i].artis);
            if (i==data-1)
            {
                printf("|      %-9s      |      %-9s      |<---
Top\n",list[i].judul, list[i].artis);

            }else{
                printf("|      %-9s      |      %-
9s      |\n",list[i].judul, list[i].artis);
            }

        }

    }

}

void arrayStack(){
    int maxmemory;
    int data=0;
    char title[50];
    char artist[50];

    printf("Isi jumlah kapasitas ruang yang anda butuhkan : ");
    scanf("%d",&maxmemory);

```

```

struct track list[maxmemory];
int opsi;
while (1)
{
    ProccesOfStackArray(list,data);
    printf("\n 1 - Tambah Lagu");
    printf("\n 2 - Tampilkan Lagu Terbaru");
    printf("\n 3 - Mainkan Lagu");
    printf("\n 4 - Kill Program");
    printf("\n Enter choice : ");
    scanf("%d",&opsi);
    switch (opsi)
    {
        case 1:
            if (data >= maxmemory)
            {
                printf("\nKapasitas melebihi maksimal!!!!\n");
            }else{
                printf("Input judul : ");
                scanf("%s",title);

                printf("Input artid : ");
                scanf("%s",artist);

                strcpy(list[data].judul, title);
                strcpy(list[data].artis, artist);
                data++;

            }
            printf("\n Berhasil menambahkan lagu : %s oleh %s\n\n", title,
artist);
            break;
        case 2:
            if(data == 0){
                printf("lagu gaib\n\n");
            }else{
                printf("Lagu terbaru : %s oleh %s\n", list[data-
1].judul,list[data-1].artis);
            }
            break;

        case 3:
            if (data==0)
            {
                printf("lagu gaib\n\n");
            }else{
                data--;
            }
    }
}

```

```

        printf("\n Berhasil memainkan lagu : %s oleh %s\n\n", title, artist);
        break;
    case 4:
        exit(0);
        break;
    default:
        printf("\npilihan tidak valid\n");
        break;
    }
}

int main(){
    int app;
    re:
    printf("Pilih aplikasi music player yang ingin anda gunakan : \n");
    printf("1. Groovy Player (aplikasi berbasis stack linkedlist)\n");
    printf("2. Graveyard Music (aplikasi berbasis stack array)\n ");
    scanf("%d", &app);
    if (app==1)
    {
        linkedlist();
    }else if (app==2)
    {
        arrayStack();
    }else{
        printf("\nOPSI TIDAK VALID\n");
        goto re;
    }
}

```



## B. Deskripsi Program

Ditampilan awal kan ditampilkan 2 pilihan aplikasi untuk user, jika use memilih opsi 1, maka user akan diarahkan untuk menuju ke program stack implementasi linkedlist, sedangkan jika memilih opsi 2, maka user diarahkan menuju program srack implementasi array

Di program stack implementasi linkedlist, user akan dipertemukan dengan sebuah tampilan opsi dari aksi apa saja yang akan dilakukan user terhadap program. Jika user memilih opsi 1, maka program akan memberikan sebuah fitur input kepada user, dan user bisa input judul lagu dan music yang ingin dia tambahkan. Setelah itu program akan menerima data inputan user dan menampilkan nya ke layar sebagai top. Jika user memilih opsi 2, maka program akan secara otomatis menampilkan data music yang berada pada posisi top. Jika user memilih opsi 3, maka program akan menghapus data yang ada pada posisi top. Jika user memilih opsi 4, maka program akan menampilkan data yang ada pada saat ini. Jika User memilih opsi 5, maka program akan memberhentikan proses nya.

Di program stack implementasi array, user akan di pertemuan dengan sebuah tampilan yang memerintahkan user untuk memberi kapasitas memori dari aplikasi tersebut. Jika user memilih opsi 1, maka program akan memberikan sebuah fitur input kepada user, dan user bisa input judul lagu dan music yang ingin ditambahkan(jika melebihi kapasitas, program akan memberikan peringatan kepada user). Setelah itu program akan menerima data inputan user dan menampilkan nya ke layar sebagai top. Jika user memilih opsi 2, maka program akan secara otomatis menampilkan data music yang berada pada posisi top. Jika user memilih opsi 3, maka program akan menghapus data yang ada pada posisi top. Jika user memilih opsi 4, maka program akan memberhentikan proses nya.

## C. Bukti Presentasi



