TP MODUL 6

Nama : **Ahmad Naufal Ramadhan**

NIM : **103012300239**

Kelas : **IF-47-02**

Kode Asprak : **YDA**

DLL.h

main.cpp | DLL.h | DLL.cpp

```
#ifndef DLL_H_INCLUDED
#define DLL_H_INCLUDED

#include <iostream>
using namespace std;

#define first(L) L.first
#define last(L) L.last
#define next(P) P->next
#define info(P) P->info
#define prev(P) P->prev

typedef struct Lagu infotype;
typedef struct elmList *address;

struct Lagu{
    string Penyanyi;
    string Judul;
};

struct elmList {
    infotype info;
    address next;
    address prev;
};

struct List{
    address first;
    address last;
};

bool isEmpty_103012300239(List L);
void createList_103012300239(List &L);
address createNewElmt_103012300239(infotype X, address P);
void insertFirst_103012300239(List &L, address P);
void insertAfter_103012300239(List &L, address P, address &prec);
void insertLast_103012300239(List &L, address P);
void deleteFirst_103012300239(List &L, address &P);
void deleteAfter_103012300239(List &L, address &P, address &prec);
void deleteLast_103012300239(List &L, address &P);
void concat_103012300239(List L1, List L2, List &L3);
address findLagu_103012300239(string Judul, List L);
void removeLagu_103012300239(string Judul, List &L);
void show_103012300239(List L);

#endif // DLL_H_INCLUDED

```

DLL.cpp

main.cpp | DLL.h | DLL.cpp

```cpp
#include "DLL.h"
using namespace std;

bool isEmpty_103012300239(List L){
    if (first(L) == NULL && last(L) == NULL){
        return true;
    } else {
        return false;
    }
}

void createList_103012300239(List &L){
    first(L) = NULL;
    last(L) = NULL;
}

address createNewElmt_103012300239(infotype X, address P){
    P = new elmList;

    if (P == NULL){
        return NULL;
    }

    info(P) = X;
    next(P) = NULL;
    prev(P) = NULL;
    return P;
}

void insertFirst_103012300239(List &L, address P){
    if (isEmpty_103012300239(L)){
        first(L) = P;
        last(L) = P;
    } else {
        next(P) = first(L);
        prev(next(P)) = P;
        first(L) = P;
    }
}

void insertAfter_103012300239(List &L, address P, address &prec){
    next(P) = next(prec);
    prev(P) = prec;
    prev(next(P)) = P;
    next(prec) = P;
}

void insertLast_103012300239(List &L, address P){
    if (isEmpty_103012300239(L)){
        first(L) = P;
        last(L) = P;
    } else {
        prev(P) = last(L);
        next(prev(P)) = P;
        last(L) = P;
    }
}

void deleteFirst_103012300239(List &L, address &P){
    if (first(L) == last(L)){
        P = first(L);
        first(L) = NULL;
        last(L) = NULL;
    } else {
        P = first(L);
        first(L) = next(P);
        prev(next(P)) = NULL;
        next(P) = NULL;
    }
}

void deleteAfter_103012300239(List &L, address &P, address &prec){
    if (next(prec) == last(L)) {
        P = next(prec);
        last(L) = prec;
        next(prec) = NULL;
        prev(P) = NULL;
    } else {
        P = next(prec);
        next(prec) = next(P);
        prev(next(P)) = prec;
        next(P) = NULL;
        prev(P) = NULL;
    }
}

void deleteLast_103012300239(List &L, address &P){
    if (first(L) == last(L)){
        P = last(L);
        first(L) = NULL;
        last(L) = NULL;
    } else {
        P = last(L);
        last(L) = prev(P);
        next(prev(P)) = NULL;
        prev(P) = NULL;
    }
}

void concat_103012300239(List L1, List L2, List &L3){
    first(L3) = first(L1);
    next(last(L1)) = first(L2);
    prev(first(L2)) = last(L1);
    last(L3) = last(L2);
}
```

main.cpp | DLL.h | DLL.cpp

```cpp

void concat_103012300239(List L1, List L2, List &L3){
    first(L3) = first(L1);
    next(last(L1)) = first(L2);
    prev(first(L2)) = last(L1);
    last(L3) = last(L2);
}

address findLagu_103012300239(string Judul, List L){
    address p = first(L);
    if (isEmpty_103012300239(L)){
        return NULL;
    } else {
        while (p != NULL && info(p).Judul != Judul){
            p = next(p);
        }
        if (p != NULL && info(p).Judul == Judul){
            return p;
        } else {
            return NULL;
        }
    }
}

void removeLagu_103012300239(string Judul, List &L){
    address p, q;
    if (isEmpty_103012300239(L)){
        cout << "List Kosong";
    } else {
        p = findLagu_103012300239(Judul, L);
        if (p == first(L)){
            deleteFirst_103012300239(L, q);
        } else if (p == last(L)){
            deleteLast_103012300239(L, q);
        } else {
            deleteAfter_103012300239(L, q, prev(p));
        }
    }
}

void show_103012300239(List L){
    address p = first(L);
    if (isEmpty_103012300239(L)){
        cout << "List Kosong" << endl;
    } else {
        while (p != NULL){
            cout << info(p).Penyanyi<< " - " << info(p).Judul << endl;
            p = next(p);
        }
    }
    cout << endl;
}
```

Main.cpp

main.cpp | DLL.h | DLL.cpp

```cpp
#include <iostream>
#include "DLL.h"

using namespace std;

int main(){
    List POP, ROCK, GABUNG;
    address p;
    int n, i;
    string judul;
    infotype x;

    createList_103012300239(POP);
    createList_103012300239(ROCK);
    createList_103012300239(GABUNG);

    //Inisiasi list ROCK 1
    x.Judul = "WelcomeToJungle";
    x.Penyanyi = "GunNRoses";
    p = createNewElmt_103012300239(x,p);
    insertFirst_103012300239(ROCK,p); //Insert First

    //Inisiasi list ROCK 2
    x.Judul = "HighwayToHell";
    x.Penyanyi = "ACDC";
    p = createNewElmt_103012300239(x,p);
    insertLast_103012300239(ROCK,p); //Insert Last

    //Inisiasi list ROCK 3
    x.Judul = "BohemianRhapsody";
    x.Penyanyi = "Queen";
    p = createNewElmt_103012300239(x,p);
    insertLast_103012300239(ROCK,p); //Insert Last

    cout << "List lagu ROCK: " << endl;
    show_103012300239(ROCK);

    cout << "Masukkan banyaknya Lagu POP yang ingin dimasukkan: ";
    cin >> n;

    for (i = 1; i <= n; i++){
        cout << "Masukkan lagu POP(Penyanyi Judul): ";
        cin >> x.Penyanyi >> x.Judul;

        p = createNewElmt_103012300239(x,p);
        insertLast_103012300239(POP,p); //Insert Last
    }
    cout << endl;

    cout << "List lagu POP: " << endl;
    show_103012300239(POP);

    cout << "Tambahkan lagu ROCK setelah elemen pertama(Penyanyi Judul): ";
    cin >> x.Penyanyi >> x.Judul;
```

main.cpp | DLL.h | DLL.cpp

```
52
53      cout << "Tambahkan lagu ROCK setelah elemen pertama(Penyanyi Judul): ";
54      cin >> x.Penyanyi >> x.Judul;
55      p = createNewElmt_103012300239(x,p);
56      insertAfter_103012300239(ROCK, p, first(ROCK)); //insert After
57      cout << endl;
58
59      cout << "List lagu ROCK: " << endl;
60      show_103012300239(ROCK);
61
62      //Menggabungkan 2 List
63      cout << "Hasil penggabungan 2 List" << endl << endl;
64      concat_103012300239(ROCK, POP, GABUNG);
65      cout << "List semua lagu: " << endl;
66      show_103012300239(GABUNG);
67
68
69      cout << "Masukkan judul lagu yang dicari: ";
70      cin >> judul;
71      p = findLagu_103012300239(judul, GABUNG);
72      if (p != NULL){
73          cout << info(p).Penyanyi<< " - " << info(p).Judul << endl;
74      } else {
75          cout << "Lagu " << judul << " tidak ditemukan" << endl;
76      }
77      cout << endl;
78
79       cout << "Masukkan judul lagu yang dicari: ";
80      cin >> judul;
81      p = findLagu_103012300239(judul, GABUNG);
82      if (p != NULL){
83          cout << info(p).Penyanyi<< " - " << info(p).Judul << endl;
84      } else {
85          cout << "Lagu " << judul << " tidak ditemukan" << endl;
86      }
87      cout << endl;
88
89      cout << "Masukkan lagu yang ingin di hapus: ";
90      cin >> judul;
91      removeLagu_103012300239(judul, GABUNG);
92      cout << "List semua lagu: " << endl;
93      show_103012300239(GABUNG);
94
95      cout << "Masukkan lagu yang ingin di hapus: ";
96      cin >> judul;
97      removeLagu_103012300239(judul, GABUNG);
98      cout << "List semua lagu: " << endl;
99      show_103012300239(GABUNG);
100
101     cout << "Masukkan lagu yang ingin di hapus: ";
102     cin >> judul;
103     removeLagu_103012300239(judul, GABUNG);
104     cout << "List semua lagu: " << endl;
105     show_103012300239(GABUNG);
106
107     return 0;
108 }
```

Running

```
List lagu ROCK:
GunNRoses - WelcomeToJungle
ACDC - HighwayToHell
Queen - BohemianRhapsody

Masukkan banyaknya Lagu POP yang ingin dimasukkan: 3
Masukkan lagu POP(Penyanyi Judul): FosterThePeople Imagination
Masukkan lagu POP(Penyanyi Judul): TheCorrs Radio
Masukkan lagu POP(Penyanyi Judul): Maroon5 Sugar

List lagu POP:
FosterThePeople - Imagination
TheCorrs - Radio
Maroon5 - Sugar

Tambahkan lagu ROCK setelah elemen pertama(Penyanyi Judul): GunsNRoses SweetChildO'Mine

List lagu ROCK:
GunNRoses - WelcomeToJungle
GunsNRoses - SweetChildO'Mine
ACDC - HighwayToHell
Queen - BohemianRhapsody

Hasil penggabungan 2 List

List semua lagu:
GunNRoses - WelcomeToJungle
GunsNRoses - SweetChildO'Mine
ACDC - HighwayToHell
Queen - BohemianRhapsody
FosterThePeople - Imagination
TheCorrs - Radio
Maroon5 - Sugar

Masukkan judul lagu yang dicari: HighwayToHell
ACDC - HighwayToHell

Masukkan judul lagu yang dicari: SatuBulan
Lagu SatuBulan tidak ditemukan

Masukkan lagu yang ingin di hapus: BohemianRhapsody
List semua lagu:
GunNRoses - WelcomeToJungle
GunsNRoses - SweetChildO'Mine
ACDC - HighwayToHell
FosterThePeople - Imagination
TheCorrs - Radio
Maroon5 - Sugar

Masukkan lagu yang ingin di hapus: WelcomeToJungle
List semua lagu:
GunsNRoses - SweetChildO'Mine
ACDC - HighwayToHell
FosterThePeople - Imagination
TheCorrs - Radio
Maroon5 - Sugar

Masukkan lagu yang ingin di hapus: Sugar
List semua lagu:
GunsNRoses - SweetChildO'Mine
ACDC - HighwayToHell
FosterThePeople - Imagination
TheCorrs - Radio


Process returned 0 (0x0)   execution time : 76.212 s
Press any key to continue.
```