

Nama : Tangguh Laksana

NIM 1302210025

main.cpp

```
#include "SLL.h"
#include "SLL.cpp"
int main()
{
    List A;
    cout << "first(A) sebelum createList : " << ((A).first) << endl;
    createList(A);
    cout << "first(A) sebelum createList : " << ((A).first) << endl;
    cout << endl;

    //new element
    adr P = newElement('T'); *x: 'T'
    cout << "Info(P) : " << (P)->info << endl;
    cout << "Next(P) : " << (P)->next << endl;
    cout << endl;

    //insert element
    cout << "first(A) sebelum insertFirst : " << ((A).first) << endl;
    insertFirst(A, P); *p: P
    cout << "first(A) sebelum insertFirst : " << ((A).first) << endl;
    cout << "info First(A) : " << ((A).first)->info << endl;

    P = newElement('A'); *x: 'A'
    insertFirst(A,P); *p: P
    cout << "info First(A) : " << ((A).first)->info << endl;

    P = newElement('L'); *x: 'L'
    insertFirst(A,P); *p: P
    cout << "info First(A) : " << ((A).first)->info << endl;

    adr myafter = P;

    P = newElement('N'); *x: 'N'
    insertFirst(A,P); *p: P
    cout << "info First(A) : " << ((A).first)->info << endl;
    cout << endl;
    //show
    show(A);

    //deleteLast(A);
    //show(A);

    //insertAfter
    P = newElement('V'); *x: 'V'
    insertAfter(A,P,myafter); *p: P *after: myafter
    cout << "info Insert "<< (P)->info << " after " << (myafter)->info << " : " << endl;
    show(A);
    cout << endl;
}
```

SLL.cpp

```
#include "SLL.h"
#include <iostream>

void createList(List &A){
    ((A).first) = NULL;
}

adr newElement(intotype x){
    adr P = new element;
    (P)->info = x;
    (P)->next = NULL;
    return P;
}

void insertFirst(List &A,adr p){
    if(((A).first) == NULL){
        ((A).first) = p;
    }else{
        (p)->next = ((A).first);
        ((A).first) = p;
    }
}

void show(List A){
    if((A).first != NULL){
        adr p= (A).first;
        while (p!=NULL) {
            cout << (p)->info << " ";
            p = (p)->next;
        }
        cout << endl;
    }else{
        cout << "List Kosong" << endl;
    }
}

adr deleteLast(List &A){
    adr p,q;
    if(((A).first) == NULL){
        p = NULL;
        cout << "List Kosong" << endl;
    }else if(((A).first)->next == NULL){
        p = (A).first;
        (A).first = NULL;
    }else{
        q = (A).first;
        p = (A).first;
        while ((p)->next != NULL) {
            q = p;
            p = (p)->next;
        }
        (q)->next = NULL;
    }
    return p;
}

void insertAfter(List &A,adr p,adr after){
    (p)->next = (after)->next;
    (after)->next = p;
}
```

SLL.h

```
#ifndef SLL_H_INCLUDED
#define SLL_H_INCLUDED

#include <iostream>
using namespace std;

typedef char infotype;
typedef struct element *adr;

struct element {
    infotype info;
    adr next;
};

struct List{
    adr first;
};

void createList(List &A);
adr newElement(infotype x);
void insertFirst(List &A,adr p);
void show(List A);
adr deleteLast(List &A);
void insertAfter(List &A,adr p,adr after);

#endif
```

Output

```
first(A) sebelum createList : 0x10
first(A) sebelum createList : 0

Info(P) : T
Next(P) : 0

first(A) sebelum insertFirst : 0
first(A) sebelum insertFirst : 0x751710
info First(A) : T
info First(A) : A
info First(A) : L
info First(A) : N

N L A T
info Insert V after L :
N L V A T
```