

LAPORAN PRAKTIKUM
Modul 4
“Single Linked List (Bagian I)”



Disusun Oleh:
Doni Wicaksono - 21104062
SE-05-02

Dosen:
Yudha Islami Sulistya, S.Kom., M.Cs

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

1. Membuat deklarasi tipe List

```
#include <iostream>

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

using namespace std;
typedef int infotype;
typedef struct elmList *address;

struct elmList {
    infotype info;
    address next;
};

struct List{
    address first;
};
```

```
#include <iostream>
#include "list.h"
using namespace std;
```

2. Membuat list kosong, yaitu procedure createList

```
void createList(List &L);
```

```
void createList (List &L) {
    first(L)= NULL;
}
```

3. Setelah list sudah ada, selanjutnya buatlah elemen dengan menggunakan fungsi allocate

```
address allocate(infotype x);
```

```
address allocate(infotype x){
    address p = new elmList;
    info(p) = x;
    next(p) = NULL;

    return p;
}
```

4. Setelah List dan elemen sudah ada, maka selanjutnya elemen tersebut harus diinsert ke List agar bisa menjadi elemen list

```
void insertLast(List &L, address P);
```

```
void insertFirst(List &L, address P){  
    next(P) = first(L);  
    first(L) = P;  
}
```

5. Setelah proses insert elemen, maka agar bisa mengetahui apakah elemen berhasil diinsertkan, maka kita perlu menampilkan isi list.

```
void printInfo(List L);
```

```
void printInfo (List L){  
    address p = first(L);  
    while (p != NULL){  
        cout << info(p) << ", ";  
        p = next(p);  
    }  
    cout << endl;  
}
```

6. Sekarang setelah ADT List sudah terisi dengan beberapa faktor fungsi Procedur di atas, maka mari buat sebuah List berisi 3 elemen yang berisi 3 digit nim terakhir Anda di main.cpp

```
#include <iostream>
#include "list.h"
using namespace std;

int main()
{
    List L;
    createList(L);

    int input;
    address p;

    cout << "Masukkan angka pertama: ";
    cin >> input;
    p = allocate(input);
    insertFirst(L, p);
    printInfo(L);

    cout << "Masukkan angka kedua: ";
    cin >> input;
    p = allocate(input);
    insertFirst(L, p);
    printInfo(L);

    cout << "Masukkan angka ketiga: ";
    cin >> input;
    p = allocate(input);
    insertFirst(L, p);
    printInfo(L);

    return 0;
}
```

```
@slashedzer0 →/workspaces/STD_Doni_Wicaksono_21104062/04_Single_Linked_List_Bagian_1/TP
Masukkan angka pertama: 0
0,
Masukkan angka kedua: 6
6, 0,
Masukkan angka ketiga: 2
2, 6, 0,
```

7. SESI HAVE FUN

```
void insertLast(List &L, address P) {
    if (first(L) == NULL) {
        first(L) = P;
    } else {
        address last = first(L);
        while (next(last) != NULL) {
            last = next(last);
        }
        next(last) = P;
    }
}

void insertAfter(address Prec, address P) {
    if (Prec != NULL) {
        next(P) = next(Prec);
        next(Prec) = P;
    }
}

void deleteLast(List &L, address &P) {
    if (first(L) == NULL) {
        P = NULL;
    } else if (next(first(L)) == NULL) {
        P = first(L);
        first(L) = NULL;
    } else {
        address last = first(L);
        address prevLast = NULL;
        while (next(last) != NULL) {
            prevLast = last;
            last = next(last);
        }
        P = last;
        next(prevLast) = NULL;
    }
}

void deleteAfter(address Prec, address &P) {
    if (Prec != NULL && next(Prec) != NULL) {
        P = next(Prec);
        next(Prec) = next(next(Prec));
    }
}
```

```
void insertLast(List &L, address P);
void insertAfter(address Prec, address P);
void deleteLast(List &L, address &P);
void deleteAfter(address Prec, address &P);
```

```

address searchInfo(List L, infotype x) {
    address p = first(L);
    while (p != NULL) {
        if (info(p) == x) {
            return p;
        }
        p = next(p);
    }
    return NULL;
}

```

```

address searchInfo(List L, infotype x);

```

```

int main() {
    List L;
    createList(L);

    int digit;
    address p;
    cout << "Masukkan NIM per- digit\n";

    for (int i = 1; i <= 10; i++) {
        cout << "Digit " << i << ": ";
        cin >> digit;

        p = allocate(digit);

        insertLast(L, p);
    }

    cout << "Isi list : ";
    printInfo(L);

    return 0;
}

```

```

● @slashedzer0 →/workspaces/STD_Doni_Wicaksono_21104062/04_Single_Linked_List_Bagian_1/TP
Masukkan NIM per- digit
Digit 1: 0
Digit 2: 0
Digit 3: 2
Digit 4: 1
Digit 5: 1
Digit 6: 0
Digit 7: 4
Digit 8: 0
Digit 9: 6
Digit 10: 2
Isi list : 0, 0, 2, 1, 1, 0, 4, 0, 6, 2,

```