LAPORAN PRAKTIKUM

MODUL 4

SINGLE LINKED LIST BAGIAN 1

Disusun Oleh:

Muhammad Fathi Rafa -2311104022

SE07A

Dosen :

Yudha Islami Prasetya, S.Kom., M.Cs.

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

INSTITUT TEKNOLOGI TELKOM

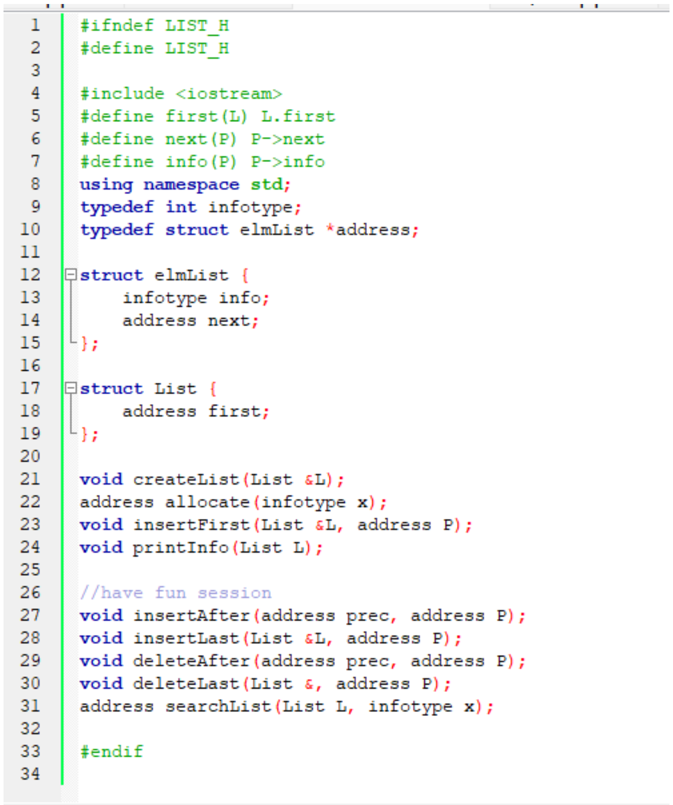
PURWOKERTO

2024

Soal Tugas Pendahuluan

1. File header

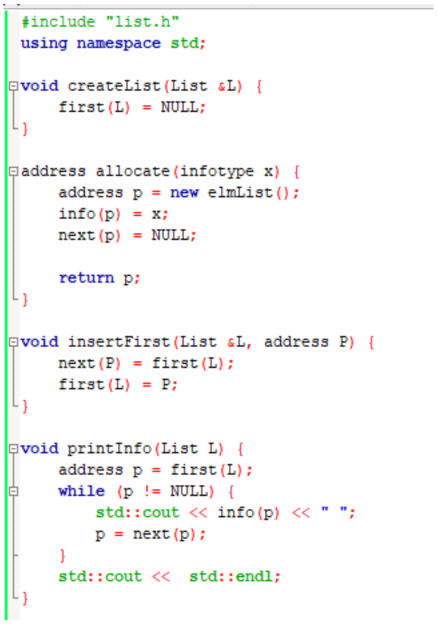
   File header mendeklarasikan struktur data dan prototipe fungsi yang akan digunakan di berbagai program. Seperti deklarasi elmlist dan list serta prototipe createList, allocate dan insertFirst.



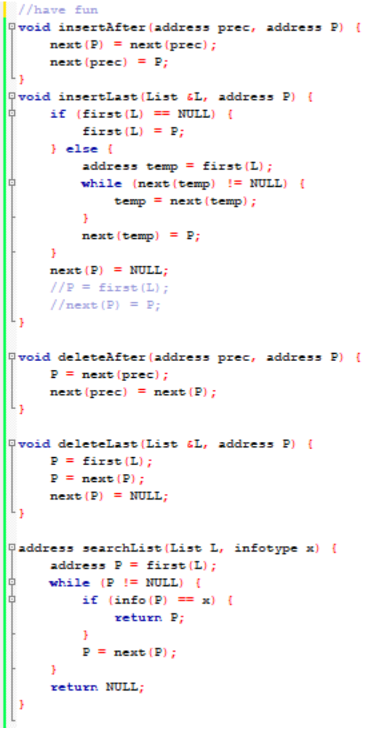
```
#ifndef LIST_H
#define LIST_H

#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;
};

void createList(List &L);
address allocate(infotype x);
void insertFirst(List &L, address P);
void printInfo(List L);

//have fun session
void insertAfter(address prec, address P);
void insertLast(List &L, address P);
void deleteAfter(address prec, address P);
void deleteLast(List &, address P);
address searchList(List L, infotype x);

#endif

```

2. File cpp

   File cpp berisi definisi dari fungsi dan kelas yang telah dideklarasikan dalam file header. File ini juga memisahkan deklarasi di file header dan implementasi membantu menjaga kode tetap terorganisir.



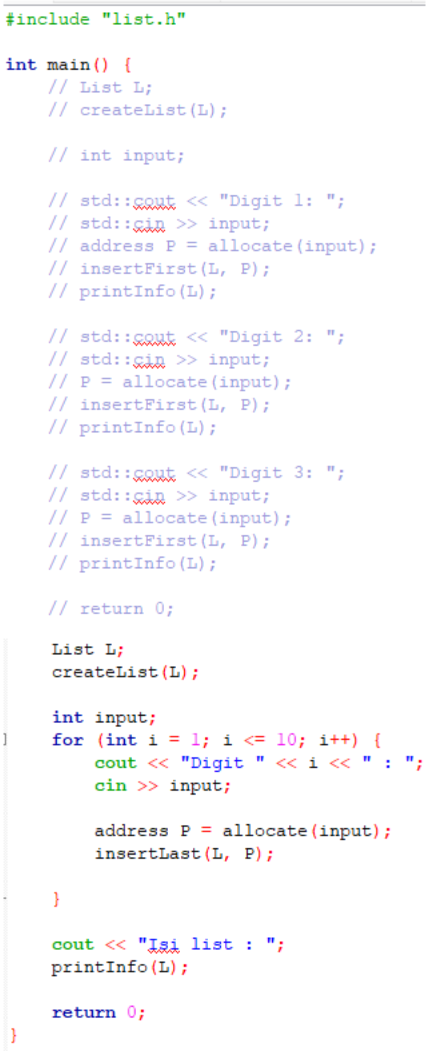
```cpp
#include "list.h"
using namespace std;

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

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

    return p;
}

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

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



```cpp
//have fun
void insertAfter(address prec, address P) {
    next(P) = next(prec);
    next(prec) = P;
}
void insertLast(List &L, address P) {
    if (first(L) == NULL) {
        first(L) = P;
    } else {
        address temp = first(L);
        while (next(temp) != NULL) {
            temp = next(temp);
        }
        next(temp) = P;
    }
    next(P) = NULL;
    //P = first(L);
    //next(P) = P;
}

void deleteAfter(address prec, address P) {
    P = next(prec);
    next(prec) = next(P);
}

void deleteLast(List &L, address P) {
    P = first(L);
    P = next(P);
    next(P) = NULL;
}

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

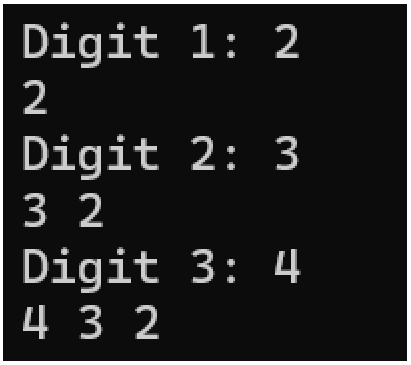
3. File main

   Pada file main gigunakan menguji fungsi dan kelas yang telah di implementasikan. Ini memungkinkan pengembang untuk melihat bagaimana berbagai bagian bekerja bersama

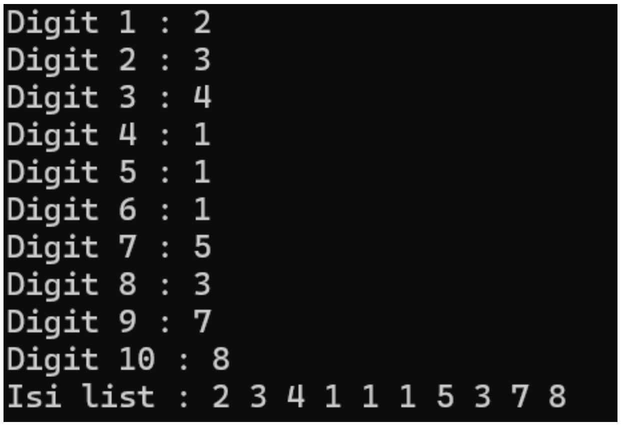


```cpp
#include "list.h"

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

    // int input;

    // std::cout << "Digit 1: ";
    // std::cin >> input;
    // address P = allocate(input);
    // insertFirst(L, P);
    // printInfo(L);

    // std::cout << "Digit 2: ";
    // std::cin >> input;
    // P = allocate(input);
    // insertFirst(L, P);
    // printInfo(L);

    // std::cout << "Digit 3: ";
    // std::cin >> input;
    // P = allocate(input);
    // insertFirst(L, P);
    // printInfo(L);

    // return 0;

    List L;
    createList(L);

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

        address P = allocate(input);
        insertLast(L, P);

    }

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

    return 0;
}
```

Output:



```
Digit 1: 2
2
Digit 2: 3
3 2
Digit 3: 4
4 3 2
```



```
Digit 1 : 2
Digit 2 : 3
Digit 3 : 4
Digit 4 : 1
Digit 5 : 1
Digit 6 : 1
Digit 7 : 5
Digit 8 : 3
Digit 9 : 7
Digit 10 : 8
Isi list : 2 3 4 1 1 1 5 3 7 8
```