LAPORAN PRAKTIKUM

PERTEMUAN IV

SINGLE LINKED LIST

Nama :

Mei sari mantiantini (2311104012)

Dosen :

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

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

Jawaban soal no 1-6

TP > C list.h > voidprintInfo(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);
```

List.cpp

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

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

address allocate(infotype x){
    address p = new elmList;
    p->info = x;
    p->next = NULL;
    return p;
}

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

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

TP > main.cpp > main()

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

int main() {
    List L;
    createList(L); // 1. Memanggil createList

    // 2. Memanggil allocate untuk 3 digit terakhir NIM dan masukkan ke dalam list
    address P1 = allocate(0);
    address P2 = allocate(1);
    address P3 = allocate(2);

    // 3. Memanggil insertFirst agar elemen masuk ke list di urutan terbalik
    insertFirst(L, P1);
    insertFirst(L, P2);
    insertFirst(L, P3);

    // 4. Memanggil procedure printInfo untuk menampilkan isi List
    cout << "Isi List (harus terbalik): ";
    printInfo(L);

    return 0;
}
```

Output:

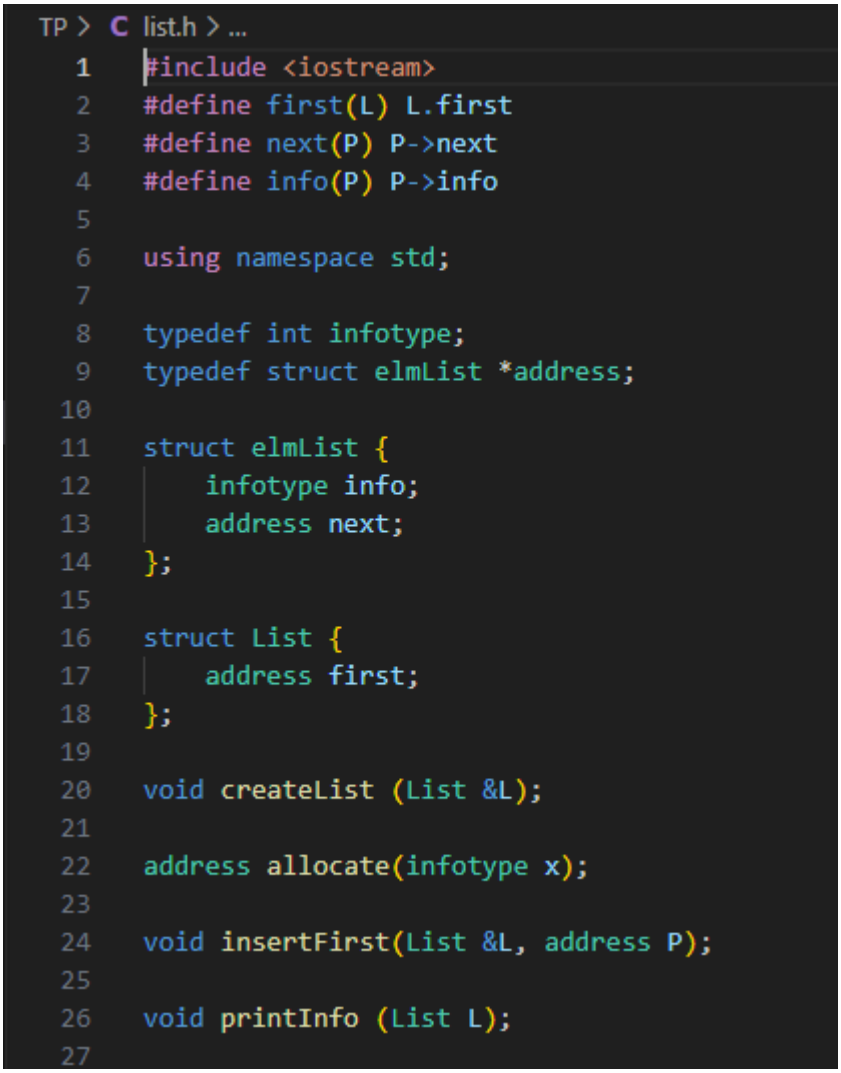
```
PS D:\Praktikum_04\TP> g++ list.cpp main.cpp -o list
PS D:\Praktikum_04\TP> g++ list.cpp main.cpp -o list
PS D:\Praktikum_04\TP> ./list
Isi List (harus terbalik): 2, 1, 0,
PS D:\Praktikum_04\TP>
```

Penjelasan:

Penjelasan singkat: Kode di atas mengimplementasikan struktur data Single Linked List. Dimulai dengan mendefinisikan struktur data Daftar dan elemen-elemennya (nomor 1). Kemudian dibuat fungsi-fungsi dasar untuk operasi pada list, seperti createList untuk membuat list kosong (nomor 2), mengalokasikan untuk membuat elemen baru (nomor 3), insertFirst untuk menyisipkan elemen di awal list (nomor 4), dan printInfo untuk menampilkan isi daftar (nomor 5). Pada main.cpp (nomor 6), program membuat sebuah list, mengisinya dengan 3 digit terakhir NIM menggunakan loop, dan menampilkan hasilnya. Karena menggunakan insertFirst, urutan angka akan terbalik saat ditampilkan.

Jawaban soal no 7

List.h



```
28   // Tambahan prosedur
29   void insertLast(List &L, address P);
30   void insertAfter(address Prec, address P);
31   void deleteLast(List &L, address &P);
32   void deleteAfter(address Prec, address &P);
33
34   address searchInfo(List L, infotype x);
```

List.cpp

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

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

address allocate(infotype x){
    address p = new elmList;
    p->info = x;
    p->next = NULL;
    return p;
}

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

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);
        }
```

```
        last->next = P;
        }
}

void insertAfter(List &L, address P, infotype x) {
    address current = first(L);
    while (current != NULL && info(current) != x) {
        current = next(current);
    }
    if (current != NULL) { // Jika elemen ditemukan
        P->next = current->next;
        current->next = P;
    }
}

void deleteLast(List &L) {
    if (first(L) == NULL) {
        cout << "List kosong, tidak ada elemen untuk dihapus." << endl;
        return;

    }

    if (next(first(L)) == NULL) {
        delete first(L);
        first(L) = NULL;
    } else {
        address current = first(L);
        while (next(next(current)) != NULL) {
```

```
            current = next(current);
        }
        delete next(current);
        current->next = NULL;
    }
}

// Prosedur untuk menghapus elemen setelah elemen terte
void deleteAfter(List &L, infotype x) {
    address current = first(L);
    while (current != NULL && info(current) != x) {
        current = next(current);
    }
    if (current != NULL && next(current) != NULL) {
        address temp = next(current);
        current->next = next(temp);
        delete temp;
    }
}

// Fungsi untuk mencari elemen dalam list
address searchInfo(List L, infotype x) {
    address current = first(L);
    while (current != NULL) {
        if (info(current) == x) {
            return current;
        }
```

```
84              current = next(current);
85          }
86          return NULL;
87      }
88
89
90      void printInfo(List L){
91          address p = first(L);
92          while (p != NULL){
93              cout << info(p) << ", ";
94              p = next(p);
95          }
96          cout << endl;
97      }
```

Main.cpp

```
int main() {
    List L;
    createList(L);
    int nim[10] = {2, 3, 1, 1, 1, 0, 4, 0, 1, 2};

    // 2. Menggunakan looping untuk memasukkan setiap digit NIM
    for (int i = 0; i < 10; i++) {
        address P = allocate(nim[i]);
        insertLast(L, P);
    }

    // 3. Memanggil procedure printInfo untuk menampilkan isi List
    cout << "Isi List: ";
    printInfo(L); // Output: 2 3 1 1 1 0 4 0 1 2

    return 0;
}
```

Output:

```
PS D:\Praktikum_04> cd Tp
PS D:\Praktikum_04\Tp> g++ list.cpp main.cpp -o list
PS D:\Praktikum_04\Tp> ./list
Isi List: 2, 3, 1, 1, 1, 0, 4, 0, 1, 2,
PS D:\Praktikum_04\Tp>
```

Penjelasan:

terdapat beberapa saran untuk perluasan fungsionalitas dari Single Linked List yang telah dibuat. Pertama, disarankan untuk menambahkan prosedur insertLast dan insertAfter pada file list.h dan list.cpp. Prosedur insertLast memungkinkan pengguna untuk menyisipkan elemen baru di akhir list, sedangkan insertAfter memungkinkan

penyisipan elemen baru setelah elemen tertentu dalam list. Selain itu, disarankan juga untuk menambahkan fungsi searchInfo, yang berfungsi untuk mencari elemen tertentu dalam list, sehingga pengguna dapat mengetahui apakah elemen tersebut ada atau tidak. Terakhir, modifikasi main.cpp agar dapat menginput hingga 10 digit NIM dengan menggunakan loop, dan memastikan elemen yang ditambahkan ditampilkan dalam urutan yang sama dengan input (bukan vertikal), dengan menggunakan prosedur insertLast.