LAPORAN PRAKTIKUM
MODUL 4

SINGLE LINKED LIST BAGIAN PERTAMA

Disusun Oleh:

Prajna paramitha - 2311104016

SE 07 01

Dosen :

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

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO

2024

A. Soal Tugas Pendahuluan

A. Soal praktek

1. Membuat deklarasi tipe list

```
# include <iostream>
# define first(L) L.first
# define next(P) P -> next
# define info(P) P -> info

// buat misahin

using namespace std;

typedef int infotype;
typedef struct elmlist *address;

struct elmlist{
    infotype info;
    address next;
};

struct List{
    address first;
};
```

List.cpp

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

using namespace std;
```

2. Membuat list kosong

List.h

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

Penjelasan : Procedure untuk nantinya kita membuat list baru

List.cpp

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

Penjelasan : pemanggilan yang di list.h, dan nantinya di list cpp ini yang akan mengelola dan mengexecute

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

List.h

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

Penjelasan : allocate address nya dengan parameter infotype data x(datanya)

List.cpp

```
// implementasi dari fungsi allocate di file list.h
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

List.h

```
// proses insert dapat menggunkana pro
void insertFirst(List &L, address P);
```

List.cpp

```
// implementasi dari procedure insertfirst
void insertFirst(List &L,address P){
    next (P) = first(L);
    first(L) = P;
}
```

Penjelasan : procedure proses menambahkan data ke dalam list.

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

```
// show data
void printInfo(List L);
```

   List.cpp

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

   Penjelasan : untuk menampilkan isi list, sekaligus kita melakukan cheking apakah data yang kita inputkan berhasil masuk ke dalam list.
6. Membuat list berisi 3 elemen yang berisi 3 digit nim terakhir

```
#include <iostream>
#include "list.h" //include file list.h

using namespace std;

int main(){
    // 1. panggil create list
    //pendeklarasian sesuai dgn yg kita buat di list
    //jadi p -> menyimpan alamat
    // x -> data
    List L;
    address P;
    infotype x;

    createList(L);

    // 2. buat syntax tanya angka pertama yg mau diinputkan user ke list
    cout <<"Masukkan angka pertama yang akan dimasukkan:" <<endl;
    cin >> x;

    // 3. panggil fungsi allocate agar data tsb dijadikan elemen
    P = allocate(x);

    // 4. panggil procedure insert first yg dibuat
    insertFirst(L, P);

    // 5. panggil procedure show info untuk cek apakah angka tsb berhasil menjadi elemen
    cout <<"Isi list:";
    //menampilkan data dengan parameter L : datanya
    printInfo(L);

    // 6. buat kembali syntaxt no 2 s/d no 5 untuk data angka kedua dari user -> untuk input data
    cout <<"Masukkan angka kedua yang akan dimasukkan:"<<endl;
    cin>> x;
    P = allocate(x);
    insertFirst(L,P);
     cout <<"Isi list:";
    //menampilkan data dengan parameter L : datanya
    printInfo(L);
```

```
    // 7. buat kembali syntax no 2 s/d no 5 untuk data angka ketiga dari user -> untuk input data
    cout <<"Masukkan angka ketiga yang akan dimasukkan:"<<endl;
    cin>> x;
    P = allocate(x);
    insertFirst(L,P);
     cout <<"Isi list:";
    //menampilkan data dengan parameter L : datanya
    printInfo(L);
    return 0;
}
```

   Penjelasan :
   - Address P : artinya p nantinya akan menyimpan alamat
   - Infotype x : artinya x akan menyimpan data
   - createList : pemanggilan procedure yang sudah kita buat untuk menambah data ke dalam list
   - cout <<”masukkan angka pertama…:”<<endl : meminta user untuk menginputkan angka pertama yang nantinya akan dimasukkan kedalam list
   - cin >> x : data yang diinputkan akan disimpan ke dalam variable x
   - P = allocate(x) : memanggil fungsi allocate, yang nantinya akan mengalokasikan memori untuk elemen baru sesuai nilai x (sebagai parameter) dan menyimpan alamat elemen baru kedalam var P
   - insertFirst(L,P) : Untuk menambahkan elemen baru
   - cout <<”isi list” : untuk menampilkan ouput tulisan isi list
   - printInfo(L) : untuk menampilkan isi list

B. Sesi have fun

1. Tambahkan procedure insertLast, insertAfter, deleteLast, deleteAfter pada list.h dan list.cpp
2. Tambahkan Function searchInfo pada list.h dan list.cpp

List.h

```
// createList dengan parameter alamat L
void createList(List &L);
// untuk mengallocate
address allocate(int x);
// insert first
void insertFirst(List &L, address P);
// insertLast
void insertLast(List &L, address P);
// InsertAfter
void insertAfter(List &L, address Prec, address P);
// delete last
void deleteLast(List &L, address &P);
// delete after
void deleteAfter(List &L, address Prec, address &P);
// menambahkan function sesuai dengan ketentuan
address searchInfo(List L, int x);
void printInfo(List L);
```

List.cpp

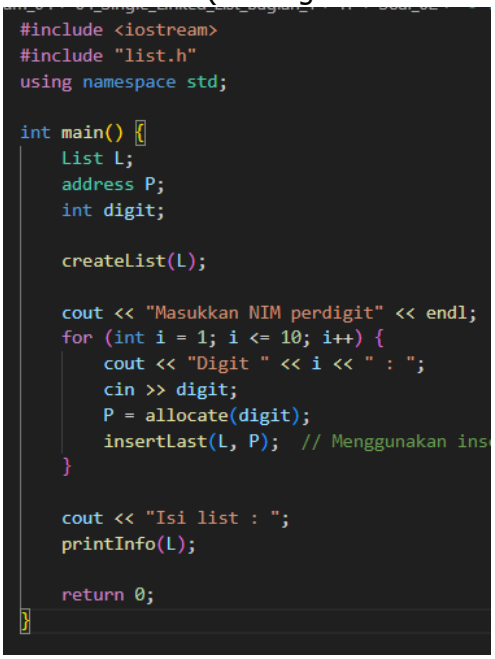
```
// insertLast, insertAfter, deleteLast, deleteAfter
#include "list.h"
#include <iostream>
using namespace std;

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

address allocate(int x) {
    address P = new Element;
    P->info = x;
    P->next = nullptr;
    return P;
}

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

void insertLast(List &L, address P) {
    if (L.first == nullptr) {
        L.first = P;
    } else {
        address Q = L.first;
        while (Q->next != nullptr) {
            Q = Q->next;
        }
        Q->next = P;
    }
}

void insertAfter(List &L, address Prec, address P) {
    if (Prec != nullptr) {
        P->next = Prec->next;
        Prec->next = P;
    }
}
```

```
void deleteLast(List &L, address &P) {
    if (L.first != nullptr) {
        if (L.first->next == nullptr) {
            P = L.first;
            L.first = nullptr;
        } else {
            address Q = L.first;
            while (Q->next->next != nullptr) {
                Q = Q->next;
            }
            P = Q->next;
            Q->next = nullptr;
        }
    }
}

void deleteAfter(List &L, address Prec, address &P) {
    if (Prec != nullptr && Prec->next != nullptr) {
        P = Prec->next;
        Prec->next = P->next;
        P->next = nullptr;
    }
}

address searchInfo(List L, int x) {
    address P = L.first;
    while (P != nullptr) {
        if (P->info == x) {
            return P;
        }
        P = P->next;
    }
    return nullptr;
}

void printInfo(List L) {
    address P = L.first;
    while (P != nullptr) {
        cout << P->info;
        P = P->next;
    }
    cout << endl;
}
```

Penjelasan :

- Struct Element : Mendefinisikan struktur elementnya.
- Int info : menyimpan data
- Elemen *next : pointrt yg nunjuk ke elemen selanjutnya
- createList : bikin list
- allocate : mengalokasikan memori
- insertfirst : menambahkan elemen dipaling pertama
- insertlast : menambahkan elemen dipaling terakhir, ini agar nanti hasil outpur nim itu tidak terbalik, karena dia menambahkannya dibelakang
- insertAfter : menyisipkan list
- deleteLast : menghapus elemen dri belakang
- deletetAfter : menghapus elemen setelah jdi bukan di belakang/didepan bisa ditengah”
- searchInfo : mencari data
- Printinfo : menampilkan isi listnya

3. Ubah main.cpp agar proses insert N data tidak satu persatu, tapi sesuai dengan jumlah digit NIM yaitu 10 data (clue : gunakan looping). Dan NIM yang diinput, saat di show tidak boleh terurut terbalik (clue : gunakan insert Last) Tampilan



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

int main() {
    List L;
    address P;
    int digit;

    createList(L);

    cout << "Masukkan NIM perdigit" << endl;
    for (int i = 1; i <= 10; i++) {
        cout << "Digit " << i << " : ";
        cin >> digit;
        P = allocate(digit);
        insertLast(L, P);  // Menggunakan ins
    }

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

    return 0;
}
```

Penjelasan :

- Include list.h : memanggil si file list.h
- Address P : artinya p akan menyimpan alamat
- Digit : datanya
- createList : memanggil procedure createList untuk membuat list baru yang nantinya akan dimasukkan data sesuai dengan yang user inputkan
- cout <<”Masukkan NIM perdigit” << endl : menampilkan pesan untuk input
- for (int I =1…) perulangan untuk memasukkan nim perdigit nya
- cinn >> digit : untuk menyimpan setiap digit yg diinputkan user ke dalam var digit
- insertLast(L,P) : menggunakan metode insertLast, jadi setiap data yang diinputkan sama user ditaruh dibelakang, agar nanti output NIM nya tidak terbalik.
- Cout >>”isi list” : menampilkan output isi list
- printInfo(L) : menampilkan isi list, memanggil procedure yg udah dibuat