LAPORAN PRAKTIKUM
Modul 04
“Single Linked List Bagian Pertama”

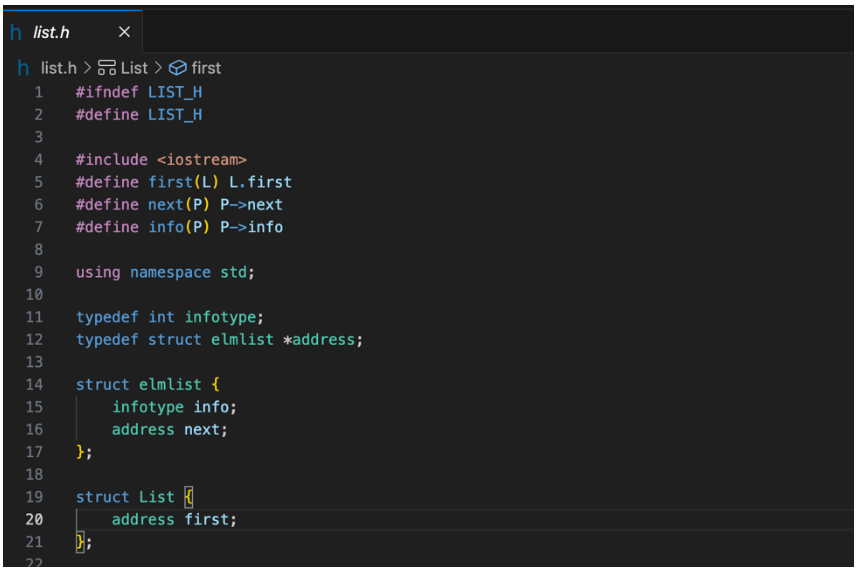
Disusun Oleh:
Yoga Eka Pratama - 2311104023
Kelas
SE-07-1
Dosen :
Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

Tugas Pendahuluan

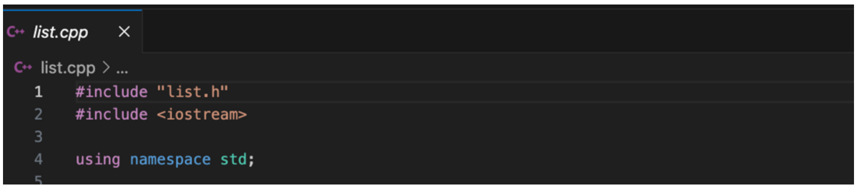
1. Membuat deklarasi tipe list

list.h



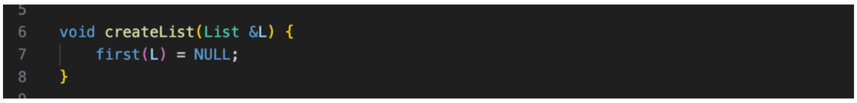
```cpp
#ifndef LIST_H
#define LIST_H

#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;
};
```

list.cpp



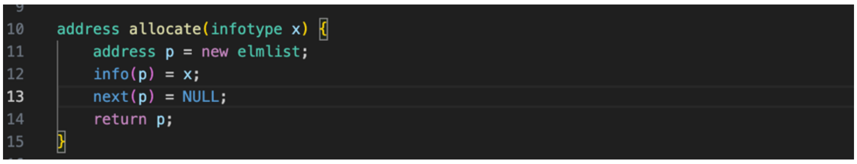
```cpp
#include "list.h"
#include <iostream>

using namespace std;
```

2. Membuat List kosong, yaitu procedure createList



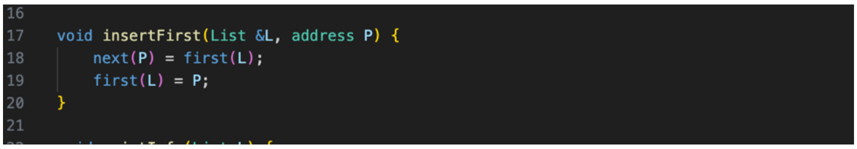
```cpp
void createList(List &L) {
    first(L) = NULL;
}
```

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



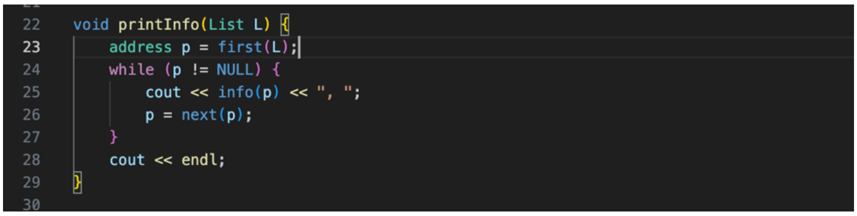
```cpp
address allocate(infotype x) {
    address p = new elmlist;
    info(p) = x;
    next(p) = NULL;
    return p;
}
```

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



```cpp

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

```

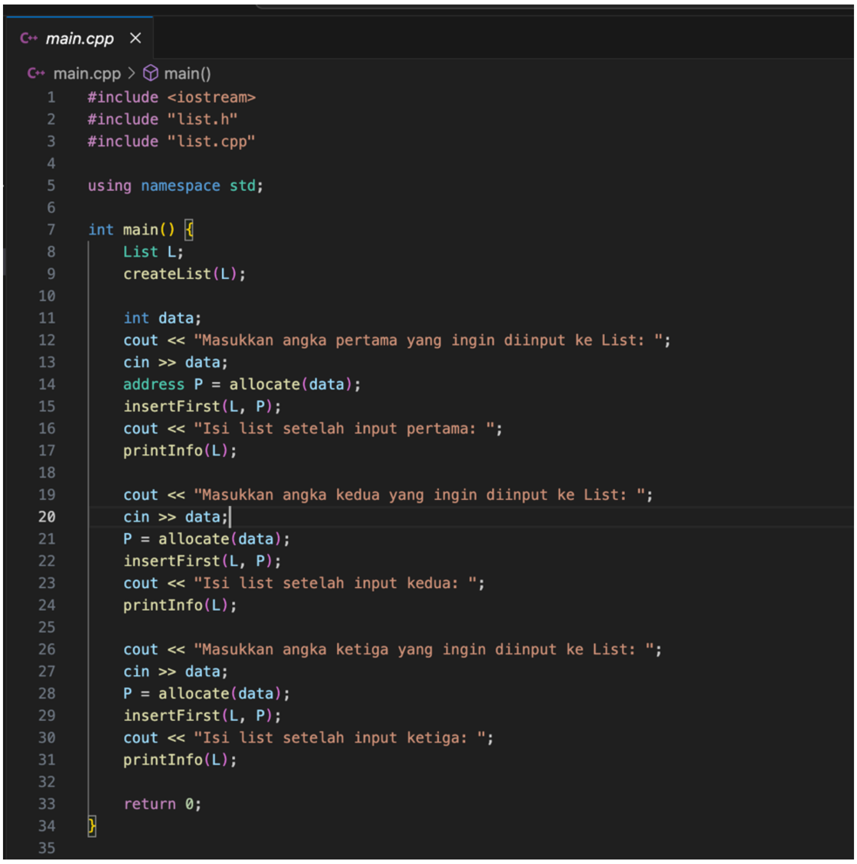
5. Setelah proses insert elemen, maka agar bisa mengetahui apakah elemen berhasil diinsertkan maka kita perlu menampilkan isi list



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

```

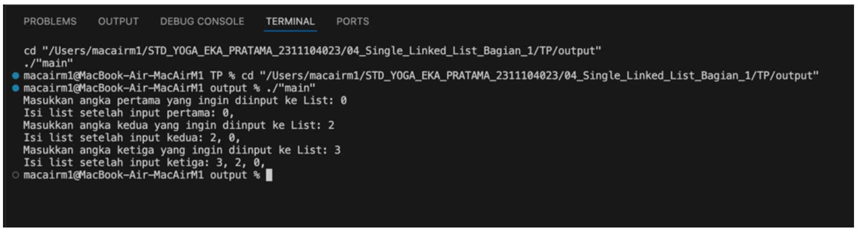
6. Sekarang, setelah ADT List sudah terisi dengan beberapa fungsi Procedure di atas, maka mari buat sebuah list berisi 3 elemen yang berisi 3 digit nim terakhir nim anda di main.cpp

Main.cpp



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

using namespace std;

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

    int data;
    cout << "Masukkan angka pertama yang ingin diinput ke List: ";
    cin >> data;
    address P = allocate(data);
    insertFirst(L, P);
    cout << "Isi list setelah input pertama: ";
    printInfo(L);

    cout << "Masukkan angka kedua yang ingin diinput ke List: ";
    cin >> data;
    P = allocate(data);
    insertFirst(L, P);
    cout << "Isi list setelah input kedua: ";
    printInfo(L);

    cout << "Masukkan angka ketiga yang ingin diinput ke List: ";
    cin >> data;
    P = allocate(data);
    insertFirst(L, P);
    cout << "Isi list setelah input ketiga: ";
    printInfo(L);

    return 0;
}

```

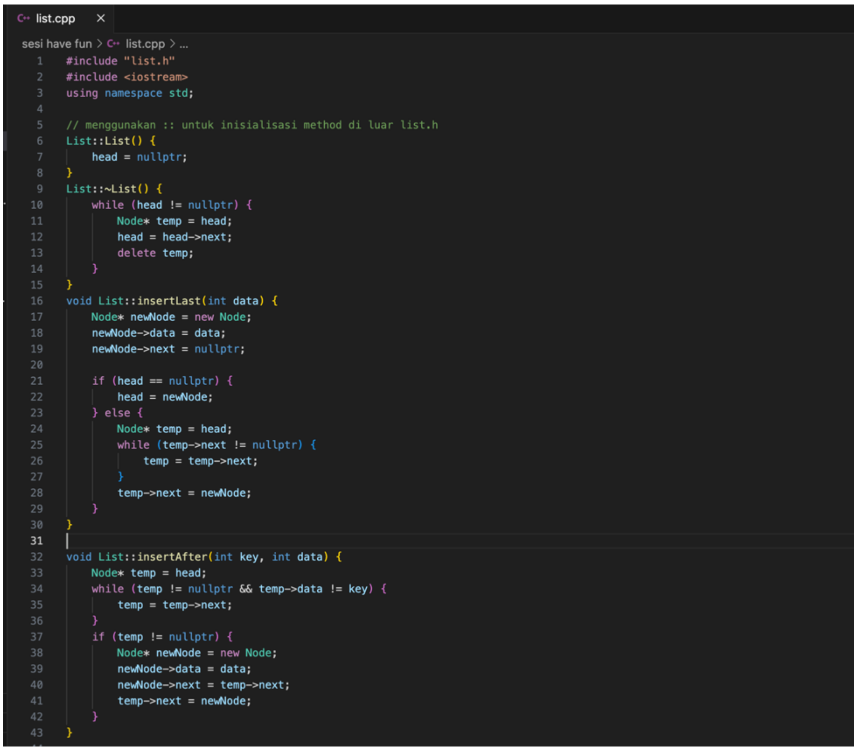
Output :



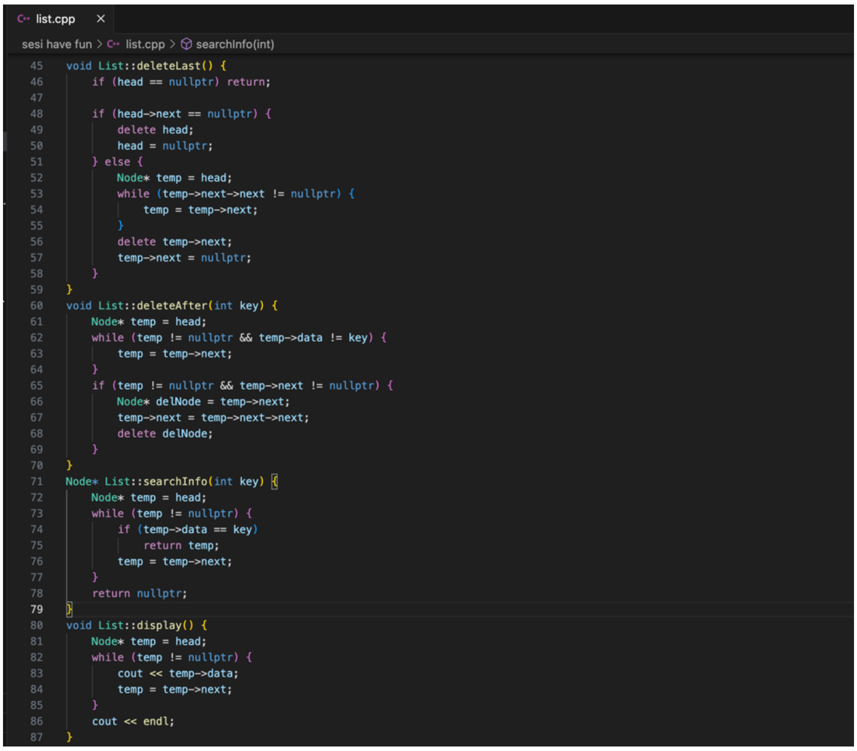
```
cd "/Users/macairm1/STD_YOGA_EKA_PRATAMA_2311104023/04_Single_Linked_List_Bagian_1/TP/output"
./"main"
macairm1@MacBook-Air-MacAirM1 TP % cd "/Users/macairm1/STD_YOGA_EKA_PRATAMA_2311104023/04_Single_Linked_List_Bagian_1/TP/output"
macairm1@MacBook-Air-MacAirM1 output % ./"main"
Masukkan angka pertama yang ingin diinput ke List: 0
Isi list setelah input pertama: 0,
Masukkan angka kedua yang ingin diinput ke List: 2
Isi list setelah input kedua: 2, 0,
Masukkan angka ketiga yang ingin diinput ke List: 3
Isi list setelah input ketiga: 3, 2, 0,
macairm1@MacBook-Air-MacAirM1 output %
```

7. Sesi Have Fun

List.cpp

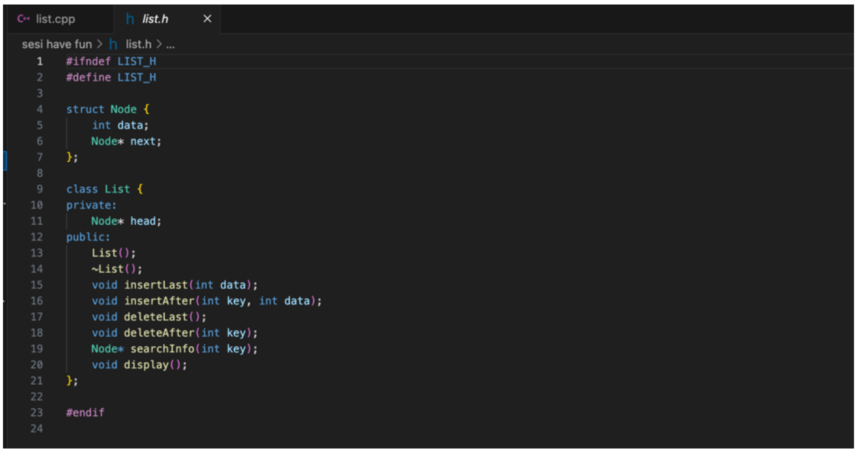


```cpp
#include "list.h"
#include <iostream>
using namespace std;

// menggunakan :: untuk inisialisasi method di luar list.h
List::List() {
    head = nullptr;
}
List::~List() {
    while (head != nullptr) {
        Node* temp = head;
        head = head->next;
        delete temp;
    }
}
void List::insertLast(int data) {
    Node* newNode = new Node;
    newNode->data = data;
    newNode->next = nullptr;

    if (head == nullptr) {
        head = newNode;
    } else {
        Node* temp = head;
        while (temp->next != nullptr) {
            temp = temp->next;
        }
        temp->next = newNode;
    }
}

void List::insertAfter(int key, int data) {
    Node* temp = head;
    while (temp != nullptr && temp->data != key) {
        temp = temp->next;
    }
    if (temp != nullptr) {
        Node* newNode = new Node;
        newNode->data = data;
        newNode->next = temp->next;
        temp->next = newNode;
    }
}
```



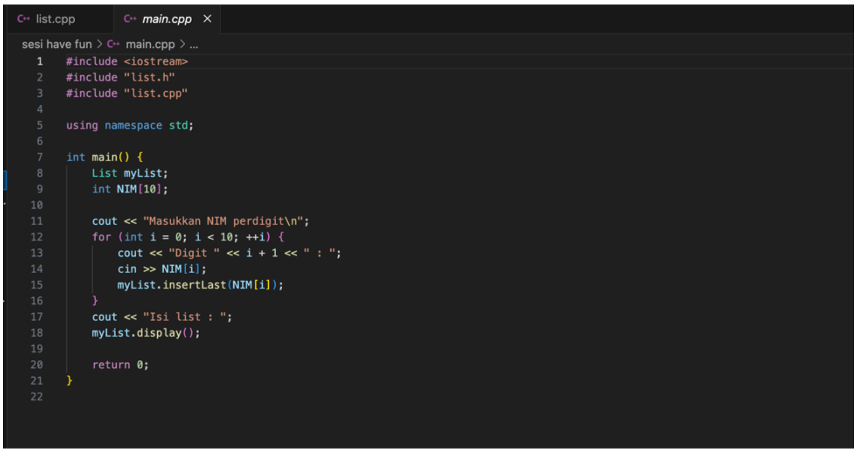
```cpp
void List::deleteLast() {
    if (head == nullptr) return;

    if (head->next == nullptr) {
        delete head;
        head = nullptr;
    } else {
        Node* temp = head;
        while (temp->next->next != nullptr) {
            temp = temp->next;
        }
        delete temp->next;
        temp->next = nullptr;
    }
}
void List::deleteAfter(int key) {
    Node* temp = head;
    while (temp != nullptr && temp->data != key) {
        temp = temp->next;
    }
    if (temp != nullptr && temp->next != nullptr) {
        Node* delNode = temp->next;
        temp->next = temp->next->next;
        delete delNode;
    }
}
Node* List::searchInfo(int key) {
    Node* temp = head;
    while (temp != nullptr) {
        if (temp->data == key)
            return temp;
        temp = temp->next;
    }
    return nullptr;
}
void List::display() {
    Node* temp = head;
    while (temp != nullptr) {
        cout << temp->data;
        temp = temp->next;
    }
    cout << endl;
}
```

List.h



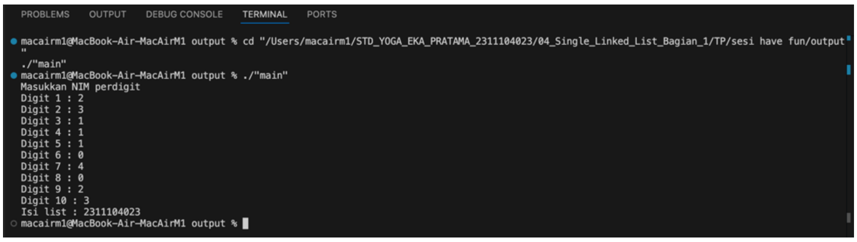
```cpp
#ifndef LIST_H
#define LIST_H

struct Node {
    int data;
    Node* next;
};

class List {
private:
    Node* head;
public:
    List();
    ~List();
    void insertLast(int data);
    void insertAfter(int key, int data);
    void deleteLast();
    void deleteAfter(int key);
    Node* searchInfo(int key);
    void display();
};

#endif
```

Main.cpp



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

using namespace std;

int main() {
    List myList;
    int NIM[10];

    cout << "Masukkan NIM perdigit\n";
    for (int i = 0; i < 10; ++i) {
        cout << "Digit " << i + 1 << " : ";
        cin >> NIM[i];
        myList.insertLast(NIM[i]);
    }
    cout << "Isi list : ";
    myList.display();

    return 0;
}
```

Output :



```
macairm1@MacBook-Air-MacAirM1 output % cd "/Users/macairm1/STD_YOGA_EKA_PRATAMA_2311104023/04_Single_Linked_List_Bagian_1/TP/sesi have fun/output
"
./"main"
macairm1@MacBook-Air-MacAirM1 output % ./"main"
Masukkan NIM perdigit
Digit 1 : 2
Digit 2 : 3
Digit 3 : 1
Digit 4 : 1
Digit 5 : 1
Digit 6 : 0
Digit 7 : 4
Digit 8 : 0
Digit 9 : 2
Digit 10 : 3
Isi list : 2311104023
macairm1@MacBook-Air-MacAirM1 output % 
```