

**TUGAS PENDAHULUAN**  
**MODUL 2**  
**“Single Linked List (Bagian Pertama)”**



**Disusun Oleh:**  
**Isabelle Putri Ardini - 2311104030**  
**SE-07-01**

**Dosen:**  
**Yudha Islami Sulistya**

**PROGRAM STUDI SOFTWARE ENGINEERING**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

## Soal Tugas Pendahuluan

### 1. Membuat deklarasi tipe list

Buat file list.h dan ketik sintak sesuai gambar berikut.

```
#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;
};
```

Buat file list.cpp dan ketik sintak berikut.

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

### 2. Membuat list kosong, yaitu procedure createList

Tambahkan pada list.h primitif dari procedure createList

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

Tambahkan pada list.cpp implementasi dari procedure createList, sintak c++ sebagai berikut:

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

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

Tambahkan pada list.h primitif dari fungsi allocate.

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

Tambahkan pada list.cpp implementasi dari fungsi allocate, sintak c++ sebagai berikut:

```

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. Proses insert dapat menggunakan procedure Insert First, procedure Insert Last, atau procedure insert After. Pada Tugas Pendahuluan kali ini, akan dicontohkan menggunakan insert first. Tambahkan pada list.h primitif procedure insertFirst.

```

void insertFirst(List &L, address P);

```

Tambahkan pada list.cpp implementasi dari procedure insertFirst sesuai sintak berikut:

```

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.

Tambahkan pada list.h primitif procedure printInfo

```

void printInfo(List L);

```

Tambahkan pada list.cpp implementasi dari proc printInfo, sintak C++ sebagai berikut:

```

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 Procedur di atas, maka mari buat sebuah List berisi 3 elemen yang berisi 3 digit nim terakhir Anda di main.cpp

Adapun gambaran isi dari main.cpp nya adalah sbb :

Tugas rekan-rekan adalah mengisi main.cpp di atas dengan sintak C++ sesuai dengan petunjuk. Buka kembali modul sebelumnya ya ^\_^. Setelah selesai, compile dan run. Lalu screen capture hasilnya.

Jika dilayar muncul 3 digit nim Anda yang muncul secara terurut TERBALIK dari urutan input.

**Jawaban:**

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

```

#include <iostream>
#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) {
        cout << info(p) << ", ";
        p = next(p);
    }
    cout << endl;
}

```

main.cpp

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

int main() {
    List L;
    infotype x;
    address p;

    // 1. Panggil createList untuk membuat list kosong
    createList(L);

    // 2. Input angka pertama dari user
    cout << "Enter first number: ";
    cin >> x;

    // 3. Alokasikan memori untuk elemen baru
    p = allocate(x);

    // 4. Masukkan elemen baru ke dalam list (insertFirst)
    insertFirst(L, p);

    // 5. Tampilkan isi list
    cout << "List after inserting the first number: ";
    printInfo(L);

    // 6. Input angka kedua dari user
    cout << "Enter second number: ";
    cin >> x;

    // 7. Alokasikan memori untuk elemen kedua
    p = allocate(x);

    // 8. Masukkan elemen kedua ke dalam list (insertFirst)
    insertFirst(L, p);
```

```

// 9. Tampilkan isi list setelah elemen kedua dimasukkan
cout << "List after inserting the second number: ";
printInfo(L);

// 10. Input angka ketiga dari user
cout << "Enter third number: ";
cin >> x;

// 11. Alokasikan memori untuk elemen ketiga
p = allocate(x);

// 12. Masukkan elemen ketiga ke dalam list (insertFirst)
insertFirst(L, p);

// 13. Tampilkan isi list setelah elemen ketiga dimasukkan
cout << "List after inserting the third number: ";
printInfo(L);

return 0;
}

```

Output:

```

Enter first number: 1
List after inserting the first number: 1,
Enter second number: 3
List after inserting the second number: 3, 1,
Enter third number: 5
List after inserting the third number: 5, 3, 1,

```

7. SESI HAVE FUN. Rekan-rekan dapat mencoba hal di bawah ini agar memudahkan saat praktikum:
  - i. Tambahkan procedure insertLast, insertAfter, deleteLast, deleteAfter pada list.h dan list.cpp
  - ii. Tambahkan Function searchInfo pada list.h dan list.cpp
  - iii. 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 (underscore adalah inputan user):

Masukkan NIM perdigit

Digit 1 : 1

Digit 2 : 1

Digit 3 : 3  
Digit 4 : 1  
Digit 5 : 9  
Digit 6 : 6  
Digit 7 : 4  
Digit 8 : 7  
Digit 9 : 4  
Digit 10 : 2  
Isi list : 1131964742

**Jawaban:**

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



```

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

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

address allocate(intotype 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) {
        cout << info(p);
        p = next(p);
    }
    cout << endl;
}

```

main.cpp

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

int main() {
    List L;
    infotype x;
    address p;

    cout << "Enter first number: ";
    cin >> x;
    p = allocate(x);
    insertFirst(L, p);

    cout << "Enter second number: ";
    cin >> x;
    p = allocate(x);
    insertFirst(L, p);

    cout << "Enter third number: ";
    cin >> x;
    p = allocate(x);
    insertFirst(L, p);

    cout << "Enter fourth number: ";
    cin >> x;
    p = allocate(x);
    insertFirst(L, p);

    cout << "Enter fifth number: ";
    cin >> x;
    p = allocate(x);
    insertFirst(L, p);
```

```
cout << "Enter sixth number: ";
cin >> x;
p = allocate(x);
insertFirst(L, p);

cout << "Enter seventh number: ";
cin >> x;
p = allocate(x);
insertFirst(L, p);

cout << "Enter eighth number: ";
cin >> x;
p = allocate(x);
insertFirst(L, p);

cout << "Enter ninth number: ";
cin >> x;
p = allocate(x);
insertFirst(L, p);

cout << "Enter tenth number: ";
cin >> x;
p = allocate(x);
insertFirst(L, p);

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

return 0;
}
```

Output:

```
Enter first number: 0
Enter second number: 3
Enter third number: 0
Enter fourth number: 4
Enter fifth number: 0
Enter sixth number: 1
Enter seventh number: 1
Enter eighth number: 1
Enter ninth number: 3
Enter tenth number: 2
Isi list: 2311104030
```