

**LAPORAN PRAKTIKUM**  
**PERTEMUAN 4**  
**SINGLE LINKED LIST**



**Nama :**

GAHA ZIDANE NURRAIHAN (2311104038)

**Dosen :**

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

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

## TP

1.membuat deklarasi tipe list :

```
1  #include <iostream>
2  #define first(L) L.first
3  #define next(P) P->next
4  #define info(P) P->info
5  using namespace std;
6  typedef int infotype;
7  typedef struct elmlist *address;
8
9  struct elmlist
10 {
11     infotype info;
12     address next;
13 }
14 };
15
16 struct List
17 {
18     address first;
19 };
20
```

List.cpp dan mengetik sintak :

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

2. Membuat list kosong, yaitu procedure createList :

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

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

```
void createlist(List &L){
    L.first = NIL;
}
```

3. Setelah list sudah ada, selanjutnya buatlah 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

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

    int input;
    std::cout << "Masukan angka: ";
    std::cin >> input;

    address P = allocate(input);

    insertFirst(L, P);

    printInfo(L);

    std::cout << "Masukan angka kedua: ";
    std::cin >> input;

    P = allocate(input);
    insertFirst(L, P);

    printInfo(L);

    std::cout << "Masukan angka ketiga: ";
    std::cin >> input;

    P = allocate(input);
    insertFirst(L, P);

    printInfo(L);

    return 0;
}
```

semua fungsi di panggil mulai dari createlist(), addres allocate(), insertfirst(), printinfo(), lalu sesuai nim saya :

```
PS C:\Users\VICTUS\OneDrive\LAPRAK STD\04_Single_Linked_List_Bagian 1\TP> g++ List.h List.cpp main.cpp -o program
PS C:\Users\VICTUS\OneDrive\LAPRAK STD\04_Single_Linked_List_Bagian 1\TP> ./program
Masukan angka: 0
0,
Masukan angka kedua: 3
3, 0,
Masukan angka ketiga: 8
8, 3, 0,
PS C:\Users\VICTUS\OneDrive\LAPRAK STD\04_Single_Linked_List_Bagian 1\TP>
```

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 :

list.h

```
void createList(List &L);
address allocate(int x);
void insertFirst(List &L, address P);
void insertLast(List &L, address P); // New function
void insertAfter(address Prec, address P);
void deleteFirst(List &L, address &P);
void deleteLast(List &L, address &P); // New function
void deleteAfter(address Prec, address &P);
```

list.cpp

```
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;
    }
}

// Search info: Function to search for an element in the list
address searchInfo(List L, infotype x) {
    address P = first(L);
    while (P != NULL) {
        if (info(P) == x) {
            return P;
        }
        P = next(P);
    }
    return NULL; // Return NULL if not found
}

// Delete last: Delete the last node from the list
void deletelast(List &L, address &P) {
    if (first(L) == NULL) {
        cout << "List is empty" << endl;
    } else if (next(first(L)) == NULL) {
        P = first(L);
        first(L) = NULL;
    } else {
        address temp = first(L);
        while (next(next(temp)) != NULL) {
            temp = next(temp);
        }
        P = next(temp);
        next(temp) = NULL;
    }
}
}
```

ii. Tambahkan Function searchInfo pada list.h dan list.cpp

list.h

```
address searchInfo(List L, infotype x); // New function
```

list.cpp

```
address searchInfo(List L, infotype x) {
    address P = first(L);
    while (P != NULL) {
        if (info(P) == x) {
            return P;
        }
        P = next(P);
    }
    return NULL; // Return NULL if not found
}
```

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):

main.cpp

```

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

    /* int input;
    std::cout << "Masukan angka: ";
    std::cin >> input;

    address P = allocate(input);

    insertFirst(L, P);

    printInfo(L);

    std::cout << "Masukan angka kedua: ";
    std::cin >> input;

    P = allocate(input);
    insertFirst(L, P);

    printInfo(L);

    std::cout << "Masukan angka ketiga: ";
    std::cin >> input;

    P = allocate(input);
    insertFirst(L, P);

    printInfo(L);
    */

    cout << ("\n");
    /* SOAL NO 7 */
    int nim, digit;
    cout << "Masukkan NIM per digit:" << endl;

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

        // Buat node baru dan masukkan di akhir list
        address P = allocate(digit);
        insertLast(L, P); // Gunakan insertLast agar urutan tidak terbalik
    }

    // Cetak isi list (NIM)
    cout << "Isi list: ";
    printInfo(L); // Menampilkan NIM yang diinput
    cout << endl;

    return 0;
}

```

output

```

Masukkan NIM per digit:
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: 3
Digit 10: 8
Isi list: 2, 3, 1, 1, 1, 0, 4, 0, 3, 8,

PS C:\Users\VICTUS\OneDrive\LAPRAK STD\04_Single_Linked_List_Bagian 1\TP>

```

makan output akan mengeluarkan nim saya.