

**LAPORAN PRAKTIKUM**  
**MODUL 4**  
**SINGLE LINKED LIST BAGIAN 1**



**Nama :**

Reza Irawan(2311104035)

**Dosen :**

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

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

## I. TP

1) list.h

Code:

```
TP > C List.h > ...
1  #ifndef LIST_H
2  #define LIST_H
3
4  #include <iostream>
5  #define first(L) L.first
6  #define next(P) P->next
7  #define info(P) P->info
8
9  using namespace std;
10
11 typedef int infotype;
12 typedef struct elmlist *address;
13
14 struct elmlist {
15     infotype info;
16     address next;
17 };
18
19 struct List {
20     address first;
21 };
22
23 void createList(List &L);
24 address allocate(infotype x);
25 void insertFirst(List &L, address P);
26 void printInfo(List L);
27
28 #endif
```

2) list.cpp

Code:

```
TP > List.cpp > printInfo(List)
1  #include "list.h"
2  #include <iostream>
3
4  using namespace std;
5
6  void createList(List &L) {
7      first(L) = NULL;
8  }
9
10 address allocate(infotype x) {
11     address p = new elmlist;
12     info(p) = x;
13     next(p) = NULL;
14     return p;
15 }
16
17 void insertFirst(List &L, address P) {
18     next(P) = first(L);
19     first(L) = P;
20 }
21
22 void printInfo(List L) {
23     address p = first(L);
24     while (p != NULL) {
25         cout << info(p) << ", ";
26         p = next(p);
27     }
28     cout << endl;
29 }
```

### 3) main.cpp

Code:

```
TP > Main.cpp > main()
1  #include <iostream>
2  #include "list.h"
3  #include "list.cpp"
4
5  using namespace std;
6
7  int main() {
8      List L;
9      createList(L);
10
11      int data;
12      cout << "Masukkan NIM yang ingin diinput ke List: ";
13      cin >> data;
14      address P = allocate(data);
15      insertFirst(L, P);
16      cout << "Isi list setelah input pertama: ";
17      printInfo(L);
18
19      cout << "Masukkan NIM kedua yang ingin diinput ke List: ";
20      cin >> data;
21      P = allocate(data);
22      insertFirst(L, P);
23      cout << "Isi list setelah input kedua: ";
24      printInfo(L);
25
26      cout << "Masukkan NIM ketiga yang ingin diinput ke List: ";
27      cin >> data;
28      P = allocate(data);
29      insertFirst(L, P);
30      cout << "Isi list setelah input ketiga: ";
31      printInfo(L);
32
33      return 0;
34 }
```

Output:

```
PS D:\STRUKTUR DATA\04_Single_Lingked_List_Bagian_1> cd "d:\STRUKTUR DATA\04_Single_Lingked_List_Bagian_1\TP\"
ain } ; if ($?) { .\Main }
Masukkan NIM yang ingin diinput ke List: 172111
Isi list setelah input pertama: 172111,
Masukkan NIM kedua yang ingin diinput ke List: 172222
Isi list setelah input kedua: 172222, 172111,
Masukkan NIM ketiga yang ingin diinput ke List: 172444
Isi list setelah input ketiga: 172444, 172222, 172111,
PS D:\STRUKTUR DATA\04_Single_Lingked_List_Bagian_1\TP> █
```

## II. PENJELASAN

Implementasi Single Linked List ini terdiri dari tiga file utama: `list.h` yang mendefinisikan struktur data dan prototipe fungsi, `list.cpp` yang mengimplementasikan fungsi-fungsi tersebut, dan `main.cpp` sebagai program utama. Struktur data `elmList` digunakan untuk merepresentasikan node dalam list, dengan `info` menyimpan data (digit NIM) dan `next` sebagai pointer ke node berikutnya. Struct `List` memiliki pointer `first` yang menunjuk ke node pertama. Fungsi-fungsi kunci meliputi `createList()` untuk inisialisasi list kosong, `allocate()` untuk membuat node baru, `insertFirst()` untuk menyisipkan node di awal list, dan `printInfo()` untuk menampilkan isi list. Program utama meminta pengguna memasukkan NIM, menyimpannya dalam list menggunakan `insertFirst()`, dan menampilkan hasilnya. Karena penggunaan `insertFirst()`, output akan menampilkan digit-digit dalam urutan terbalik dari input, mendemonstrasikan prinsip dasar operasi pada Single Linked List seperti alokasi dinamis, penyisipan, dan traversal.