

**LAPORAN PRAKTIKUM
MODUL 04
SINGLE LINKED LIST BAGIAN PERTAMA**



Nama :

Tegar Kang Ageng Gilang (2311104018)

Dosen :

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

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

TP (Tugas Pendahuluan)

Tujuan dari pembuatan program **Single Linked List** pada modul 4 ini adalah untuk mengimplementasikan struktur data *linked list* secara sederhana menggunakan bahasa C++.

1. Membuat Deklarasi Tipe Data dan Fungsi di list.h

Buat file header (list.h) yang mendefinisikan tipe data *List*, *address*, dan mendeklarasikan fungsi-fungsi yang dibutuhkan untuk operasi pada *Single Linked List*.

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

2. Membuat Implementasi Fungsi pada list.cpp

Implementasikan fungsi-fungsi yang telah dideklarasikan dalam file list.cpp.

```
C++ guided.cpp U  main U  guided_02 U  h list.h U  C++ list.cpp U X  C++ main.cpp U
TP > C++ list.cpp > insertLast(List &, address)
1  #include "list.h"
2  #include <iostream>
3
4  // Membuat list kosong
5  void createList(List &L) {
6      L.first = nullptr;
7  }
8
9  // Mengalokasikan elemen baru
10 address allocate(infotype X) {
11     address P = new Node;
12     P->info = X;
13     P->next = nullptr;
14     return P;
15 }
16
17 // Memasukkan elemen di awal list
18 void insertFirst(List &L, address P) {
19     P->next = L.first;
20     L.first = P;
21 }
22
23 // Memasukkan elemen di akhir list
24 void insertLast(List &L, address P) {
25     if (L.first == nullptr) {
26         insertFirst(L, P);
27     } else {
28         address last = L.first;
29         while (last->next != nullptr) {
30             last = last->next;
31         }
32         last->next = P;
33     }
34 }
35
36 // Menampilkan isi list
37 void printInfo(List L) {
38     address P = L.first;
39     while (P != nullptr) {
40         cout << P->info << " ";
41         P = P->next;
42     }
43     cout << endl;
44 }
```

3. Membuat Program Utama pada main.cpp

Buat file main.cpp untuk menguji operasi-operasi dasar pada Single Linked List. Pada percobaan ini, list akan berisi digit-digit terakhir dari NIM

```
TP > C++ main.cpp > main()
1  #include <iostream>
2  #include "list.cpp"
3
4
5  int main() {
6      List L;
7      createList(L);
8      // Input NIM 3 digit terakhir (misalnya: 713)
9      cout << "Masukkan 3 digit terakhir secara terbalik (misalnya 713):" << endl;
10     // Insert elemen satu per satu di awal (urutan terbalik)
11     insertFirst(L, allocate(3));
12     insertFirst(L, allocate(1));
13     insertFirst(L, allocate(7));
14     // Tampilkan isi list (akan terbalik karena insertFirst)
15     cout << "Isi list (terbalik): ";
16     printInfo(L);
17     // Menambahkan elemen menggunakan looping untuk NIM penuh 10 digit
18     cout << "\nMasukkan NIM penuh (10 digit) satu per satu:" << endl;
19     createList(L); // Reset list untuk input baru
20     for (int i = 1; i <= 10; i++) {
21         int digit;
22         cout << "Digit " << i << ": ";
23         cin >> digit;
24         insertLast(L, allocate(digit));
25     }
26     // Tampilkan isi list (urutan sesuai input karena menggunakan insertLast)
27     cout << "Isi list (NIM penuh): ";
28     printInfo(L);
29
30     return 0;
31 }
32
```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Masukkan NIM penuh (10 digit) satu per satu:
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: 1
Digit 10: 8
Isi list (NIM penuh): 2 3 1 1 1 0 4 0 1 8
TP git:(main) x
```