

# **LAPORAN PRAKTIKUM**

## **PERTEMUAN 5**

### **Pengenalan C++ : Single\_Linked\_List\_Bagian01**



**Nama :**

Ahmad Uffi Lestari Ma'ruf (2311104015)

**Dosen :**

YUDHA ISLAMI SULISTYA, S.Kom., M.Kom.

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## 1. Praktek Bersam

File singleLisr.cpp

```
#include <iostream>
using namespace std;

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

struct List {
    Elemen* first;
};

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

Elemen* alokasi(int x) {
    Elemen* P = new Elemen;
    if (P != NULL) {
        P->data = x;
        P->next = NULL;
    }
    return P;
}

void insertFirst(List& L, Elemen* P) {
    P->next = L.first;
    L.first = P;
}

void printInfo(List L) {
    Elemen* P = L.first;
    while (P != NULL) {
        cout << P->data << " ";
        P = P->next;
    }
    cout << endl;
}

Elemen* findElm(List L, int x) {
    Elemen* P = L.first;
    while (P != NULL)
    {
        if (P->data == x) {
            return P;
        }
        P = P->next;
    }
    return NULL;
}

int sumAllElements(List L) {
    int total = 0;
    Elemen* P = L.first;
    while (P != NULL) {
        total += P->data;
        P = P->next;
    }
    return total;
}
```

Main.cpp

```
#include "singlelist.cpp"

int main() {
    List L;
    Elemen* P1, *P2, *P3, *P4, *P5 = NULL;
    createList(L);

    P1 = alokasi(2);
    insertFirst(L, P1);

    P2 = alokasi(0);
    insertFirst(L, P2);

    P3 = alokasi(8);
    insertFirst(L, P3);

    P4 = alokasi(12);
    insertFirst(L, P4);

    P5 = alokasi(9);
    insertFirst(L, P5);

    printInfo(L);

    Elemen* found = findElm(L, 8);
    if (found != NULL) {
        cout << found->data << "Ditemukan dalam List" << endl;
    } else {
        cout << "Elemen Tidak Ditemukan dalam List" << endl;
    }

    int total = sumAllElements(L);
    cout << "Total Info dari kelima elemen adalah " << total << endl;

    return 0;
}
```

Maka outputnya adalah :

```
9 12 8 0 2
8Ditemukan dalam List
Total Info dari kelima elemen adalah 31
PS F:\ITTP\Modul Ittp\semester 3\prakti
```

## 2. Tugas Pendahuluan

### a. Soal satu

Pada soal pertama kita akan membuat program dimana user disuruh untuk menginputkan 6 angka lalu diakhir akan diberikan serch yaitu user bisa mencari angka yang tadi di inpu

## File soal1.cpp

```
// Soal 1: Program mencari elemen tertentu dalam Single Linked List
#include <iostream>
using namespace std;

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

typedef Node* address;
struct List {
    address head;
};

// Function untuk membuat node baru
address createNode(int value) {
    address newNode = new Node;
    newNode->info = value;
    newNode->next = NULL;
    return newNode;
}

// Function untuk mencari elemen
void searchElement(List L, int i) {
    address current = L.head;
    int position = 1;

    // Melakukan perulangan selama i belum ditemukan dan current belum
    NULLwhile (current != NULL && current->info != i) {
        current = current->next;
        position++;
    }

    // Jika elemen ditemukan
    if (current != NULL) {
        cout << "Elemen " << i << " ditemukan pada:" << endl;
        cout << "Alamat: " << current << endl;
        cout << "Posisi ke-" << position << endl;
    } else {
        cout << "Elemen " << i << " tidak ditemukan dalam list" << endl;
    }
}

int main() {
    List L;
    L.head = NULL;
    int value;

    // Input 6 elemen
    cout << "Masukkan 6 angka:\n";
    for(int i = 0; i < 6; i++) {
        cin >> value;
        address newNode = createNode(value);
        if (L.head == NULL) {
            L.head = newNode;
        } else {
            address temp = L.head;
            while (temp->next != NULL) {
                temp = temp->next;
            }
            temp->next = newNode;
        }
    }

    cout << "Masukkan nilai yang ingin dicari: ";
    cin >> value;
    searchElement(L, value);

    return 0;
}
```

Ketika program dijalankan akan muncul output seperti ini :

```
Masukkan 6 angka:
4
6
87
21
34
2
Masukkan nilai yang ingin dicari: 87
Elemen 87 ditemukan pada:
Alamat: 0x71ec70
Posisi ke-3
```

User akan disuruh untuk memasukan 6 angka yang akan dimasukan kedalam array. Ketika 6 angka sudah dimasukan maka akan diberikan serach dimana dia akan mencari angkanya. Si user mencari angka 87 maka akan muncul informasi mengenai angka 87.

## b. Soal 2

Pada soal 2 nantinya akan di suruh untuk memasukan 5 angka, setelah itu 5 angka itu akan di sorting dari kecil ke besar menggunakan bubble shorting.

```
// Soal 2: Program mengurutkan list menggunakan Bubble Sort
#include <iostream>
using namespace std;

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

typedef Node* address;
struct List {
    address head;
};

// Function untuk membuat node baru
address createNode(int value) {
    address newNode = new Node;
    newNode->info = value;
    newNode->next = NULL;
    return newNode;
}

// Procedure untuk bubble sort
void bubbleSort(List &L) {
    bool swapped;
    address current;

    do {
        swapped = false;
        current = L.head;

        while (current->next != NULL) {
            if (current->info > current->next->info) {
                // Tukar data
                int temp = current->info;
                current->info = current->next->info;
                current->next->info = temp;
                swapped = true;
            }
            current = current->next;
        }
    } while (swapped);
}
```

Maka output yang keluar akan seperti ini

```
Masukkan 5 angka:
2
4
5
1
9
List sebelum sorting: 2 -> 4 -> 5 -> 1 -> 9
List setelah sorting: 1 -> 2 -> 4 -> 5 -> 9
PS E:\ITTP\Modul_Itpp\semester_3\praktikum_std\SI
```

User memasukan angka secara angka lalu program angka mengurutkan dari kecil ke besar.

### c. Soal tiga

Pada soal ini nantinya insert akan dilakukan secara terurut dari bawah ke atas contoh program nya

```
// Soal 3: Program menambahkan elemen secara terurut
#include <iostream>
using namespace std;

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

typedef Node* address;
struct List {
    address head;
};

// Function untuk membuat node baru
address createNode(int value) {
    address newNode = new Node;
    newNode->info = value;
    newNode->next = NULL;
    return newNode;
}

// Procedure untuk insert sorted
void insertSorted(List &L, address P) {
    address Q = L.head;
    address Prev = NULL;
    bool found = false;

    // Melakukan perulangan selama found masih false dan Q masih menunjuk elemen pada list
    while (!found && Q != NULL) {
        // Melakukan pengecekan apakah info dari elemen yang ditunjuk memiliki nilai lebih kecil dari P
        if (Q->info < P->info) {
            Prev = Q;
            Q = Q->next;
        } else {
            found = true;
        }
    }

    // Melakukan pengecekan apakah Q elemen head
    if (Q == L.head) {
        // Tambahkan P sebagai head
        P->next = L.head;
        L.head = P;
    }

    // Melakukan pengecekan apakah Q berisi null
    else if (Q == NULL) {
        // Tambahkan P sebagai elemen terakhir
        Prev->next = P;
        P->next = NULL;
    }

    else {
        // Tambahkan P pada posisi diantara Prev dan Q
        Prev->next = P;
        P->next = Q;
    }
}
```

```

// Function untuk menampilkan list
void displayList(List L) {
    address current = L.head;
    while (current != NULL) {
        cout << current->info;
        if (current->next != NULL) cout << " -> ";
        current = current->next;
    }
    cout << endl;
}

int main() {
    List L;
    L.head = NULL;
    int value;

    // Input 4 elemen terurut
    cout << "Masukkan 4 angka secara terurut:\n";
    for(int i = 0; i < 4; i++) {
        cin >> value;
        address newNode = createNode(value);
        if (L.head == NULL) {
            L.head = newNode;
        } else {
            insertSorted(L, newNode);
        }
    }

    cout << "List saat ini: ";
    displayList(L);

    cout << "Masukkan nilai baru untuk disisipkan: ";
    cin >> value;
    address newNode = createNode(value);
    insertSorted(L, newNode);

    cout << "List setelah penyisipan: ";
    displayList(L);

    return 0;
}

```

maka output nya akan menjadi :

```

Masukkan 4 angka secara terurut:
2
3
4
5
List saat ini: 2 -> 3 -> 4 -> 5
Masukkan nilai baru untuk disisipkan: 5
List setelah penyisipan: 2 -> 3 -> 4 -> 5 -> 5

```

nah nilai baru 5 akan ditaruh belakang karena angka 5 memiliki nilai paling besar

### 3. Unguided

Pada program ini mengimplementasikan single linked list untuk menyimpan data mahasiswa, setiap nod nantinya akan menyimpan nim dan nama dari mahasiswa, tidak hanya itu program ini juga bisa membuat dan menambahkan data mahasiswa baru serta mencari data mahasiswa berdasarkan NIM.

```
#include <iostream>
#include <string>
using namespace std;

// Struktur data untuk menyimpan NIM dan Nama mahasiswa
struct Node {
    int nim;
    string nama;
    Node* next;
};

typedef Node* address;

struct List {
    address head;
};

// Function untuk membuat node baru
address createNode(int nim, string nama) {
    address newNode = new Node;
    newNode->nim = nim;
    newNode->nama = nama;
    newNode->next = NULL;
    return newNode;
}

// Function untuk menambahkan data mahasiswa ke dalam linked list
void tambahMahasiswa(List &L, int nim, string nama) {
    address newNode = createNode(nim, nama);

    // Jika list kosong
    if (L.head == NULL) {
        L.head = newNode;
    } else {
        // Tambahkan di akhir list
        address temp = L.head;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = newNode;
    }

    cout << "Data mahasiswa berhasil ditambahkan!" << endl;
}

// Function untuk mencari mahasiswa berdasarkan NIM
void cariMahasiswa(List L, int nim) {
    address current = L.head;
    bool found = false;

    while (current != NULL && !found) {
        if (current->nim == nim) {
            found = true;
        } else {
            current = current->next;
        }
    }

    if (found) {
        cout << "Mahasiswa ditemukan!" << endl;
        cout << "Nama: " << current->nama << endl;
    } else {
        cout << "Mahasiswa dengan NIM(" << nim << ") tidak ditemukan" << endl;
    }
}
```



```

// Function untuk menampilkan semua data mahasiswa
void tampilkanSemuaMahasiswa(List L) {
    address current = L.head;

    cout << "\nDaftar Mahasiswa:" << endl;
    cout << "======" << endl;

    if (current == NULL) {
        cout << "List kosong!" << endl;
        return;
    }

    while (current != NULL) {
        cout << "NIM: " << current->nim << endl;
        cout << "Nama: " << current->nama << endl;
        cout << "-----" << endl;
        current = current->next;
    }
}

int main() {
    List L;
    L.head = NULL;
    int pilihan, nim;
    string nama;

    do {
        cout << "\nProgram Data Mahasiswa" << endl;
        cout << "======" << endl;
        cout << "1. Tambah Data Mahasiswa" << endl;
        cout << "2. Cari Mahasiswa berdasarkan NIM" << endl;
        cout << "3. Tampilkan Semua Mahasiswa" << endl;
        cout << "4. Keluar" << endl;
        cout << "Pilihan: ";
        cin >> pilihan;

        switch(pilihan) {
            case 1:
                cout << "\nMasukkan data mahasiswa" << endl;
                cout << "NIM: ";
                cin >> nim;
                cout << "Nama: ";
                cin.ignore(); // Clear buffer
                getline(cin, nama);
                tambahMahasiswa(L, nim, nama);
                break;

            case 2:
                cout << "\nMasukkan NIM yang dicari: ";
                cin >> nim;
                cariMahasiswa(L, nim);
                break;

            case 3:
                tampilkanSemuaMahasiswa(L);
                break;

            case 4:
                cout << "Program selesai!" << endl;
                break;

            default:
                cout << "Pilihan tidak valid!" << endl;
        }
    } while (pilihan != 4);

    return 0;
}

```

Maka output yang dihasilkan seperti ini:

```
=====
1. Tambah Data Mahasiswa
2. Cari Mahasiswa berdasarkan NIM
3. Tampilkan Semua Mahasiswa
4. Keluar
Pilihan:
Masukkan data mahasiswa
NIM: Nama: Data mahasiswa berhasil ditambahkan!

Program Data Mahasiswa
=====
1. Tambah Data Mahasiswa
```