

Laporan Praktikum

Algoritma dan Struktur Data

Ganjil 2025/2026

Program Studi Teknik Informatika

Institut Teknologi Sumatera



Modul : Single Linked List

Nama : Muhammad Arfa Raditya

NIM : 124140015

Kelas (Kelas Asal) : RD

Instruksi sederhana :

- ❖ Disarankan untuk edit menggunakan Google Docs agar tidak berantakan,
- ❖ Silahkan mengganti nama modul baik yang ada pada **cover** maupun **header** sesuai dengan materi praktikum,
- ❖ Gunakan text styling seperti Heading 1, Normal Text yang telah terformat, atau text style lainnya untuk menjaga estetika laporan,
- ❖ Gunakan Syntax Highlighter untuk merapikan kode yang sudah anda buat ke dalam laporan.

Soal/Pertanyaan:

1. Latihan 1: Sistem Nilai Siswa Dinamis

Objektif: Membuat sistem penyimpanan nilai siswa menggunakan Single Linked List.

Spesifikasi:

Input: Nama siswa dan nilai (dapat ditambah kapan saja)

Output: Tampilkan semua data siswa

Latihan 2: Insert dengan Urutan Tersorting

Objektif: Implementasi insertion yang menjaga list tetap terurut.

Spesifikasi:

Setiap data baru diinsert pada posisi yang tepat. List selalu dalam kondisi terurut ascending.

```
void InsertSorted(List *L, infotype x) {  
    if (IsEmpty(*L) || (*L).first->info > x) {  
        InsertFirst(L, x);  
    } else {  
        address temp = (*L).first;  
  
        while (temp->next != NULL && temp->next->info < x) {  
            temp = temp->next;  
        }  
  
        InsertAfter(temp, x);  
    }  
}
```

Dasar Teori

Membuat sistem penyimpanan nilai siswa menggunakan Single Linked List. Pada program, kita bisa memilih untuk menambahkan, menampilkan, dan memberhentikan program.

Source Code

```
// Nama : M. Arfa Raditya
// NIM : 124140015
// Kelas : ASD RD
// Soal nomor 1

#include <iostream>
using namespace std;

typedef int infotype;

typedef struct TELmtList *address;

typedef struct TELmtList {
    string nama;           // nama siswa
    infotype info;         // nilai siswa
    address next;          // pointer ke node berikutnya
} ElmtList;

typedef struct TList {
    address first;
} List;

// Fungsi Dasar Linked List

// inisialisasi list kosong
void CreateEmpty(List* L) {
    (*L).first = NULL;
}

// mengecek apakah list kosong
bool IsEmpty(List L) {
    return (L.first == NULL);
}

// alokasi node baru
address Allocation(string nama, infotype nilai) {
    address newNode = new TELmtList;
    newNode->nama = nama;
    newNode->info = nilai;
    newNode->next = NULL;
}
```

```
    return newNode;
}

// insert di awal list
void InsertFirst(List* L, string nama, infotype nilai) {
    address newNode = Allocation(nama, nilai);
    newNode->next = (*L).first;
    (*L).first = newNode;
}

// insert di akhir list
void InsertLast(List* L, string nama, infotype nilai) {
    if (IsEmpty(*L)) {
        InsertFirst(L, nama, nilai);
    } else {
        address temp = (*L).first;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = Allocation(nama, nilai);
    }
}

// menampilkan semua data siswa
void ShowData(List L) {
    if (IsEmpty(L)) {
        cout << "List kosong." << endl;
    } else {
        address temp = L.first;
        cout << "Data Siswa:\n";
        while (temp != NULL) {
            cout << "- Nama : " << temp->nama << ", Nilai : " << temp->info << endl;
            temp = temp->next;
        }
    }
}

// dealokasi semua node

void DeallocateAll(List* L) {
    address temp;
    while ((*L).first != NULL) {
        temp = (*L).first;
```

```
        (*L).first = (*L).first->next;
        delete temp;
    }
}

// Program Utama

int main() {
    List siswa;
    CreateEmpty(&siswa);

    int pilihan;
    string nama;
    infotype nilai;

    do {
        cout << "\n=== Menu Sistem Nilai Siswa ===\n";
        cout << "1. Tambah Data Siswa\n";
        cout << "2. Tampilkan Semua Data\n";
        cout << "0. Keluar\n";
        cout << "Pilihan: ";
        cin >> pilihan;

        switch (pilihan) {
            case 1:
                cout << "Masukkan nama siswa: ";
                cin.ignore(); // membersihkan newline
                getline(cin, nama); // agar input nama bisa ada spasi nya
                cout << "Masukkan nilai siswa: ";
                cin >> nilai;
                InsertLast(&siswa, nama, nilai);
                break;

            case 2:
                ShowData(siswa);
                break;

            case 0:
                cout << "Terima kasih!\n";
                break;

            default:
                cout << "Pilihan tidak valid.\n";
        }
    } while (pilihan != 0);
}
```

```
    }  
    } while (pilihan != 0);  
  
    // dealokasi semua node sebelum keluar  
    DeallocateAll(&siswa);  
  
    return 0;  
}
```

Dokumentasi Hasil Running

```
0. Keluar
Pilihan: 1
Masukkan nama siswa: arfa raditya
Masukkan nilai siswa: 100

=== Menu Sistem Nilai Siswa ===
1. Tambah Data Siswa
2. Tampilkan Semua Data
0. Keluar
Pilihan: 1
Masukkan nama siswa: budi santoso
Masukkan nilai siswa: 90

=== Menu Sistem Nilai Siswa ===
1. Tambah Data Siswa
2. Tampilkan Semua Data
0. Keluar
Pilihan: 1
Masukkan nama siswa: gibran raka
Masukkan nilai siswa: 40

=== Menu Sistem Nilai Siswa ===
1. Tambah Data Siswa
2. Tampilkan Semua Data
0. Keluar
Pilihan: 2
Data Siswa:
- Nama : arfa raditya, Nilai : 100
- Nama : budi santoso, Nilai : 90
- Nama : gibran raka, Nilai : 40

=== Menu Sistem Nilai Siswa ===
1. Tambah Data Siswa
2. Tampilkan Semua Data
0. Keluar
Pilihan: 0
Terima kasih!
```

Gambar 01. Program bisa menginputkan data secara terus menerus dan bisa menampilkan seluruh data yg sudah diinput

Dasar Teori

Membuat sistem sorting angka menggunakan Single Linked List, urutan angka dimulai dari angka yang terkecil hingga terbesar.

Source Code

```
// Nama : M. Arfa Raditya
// NIM : 124140015
// Kelas : ASD RD
// Soal nomor 2

#include <iostream>
using namespace std;

typedef int infotype;

// struktur node
typedef struct TElmtList *address;
typedef struct TElmtList {
    infotype info;
    address next;
} ElmtList;

// struktur list
typedef struct TList {
    address first;
} List;

// mengecek apakah list kosong
bool IsEmpty(List L) {
    return (L.first == nullptr);
}

// alokasi memori untuk node baru
address Allocation(infotype x) {
    address P = new ElmtList;
    P->info = x;
    P->next = nullptr;
```



```
    return P;
}

// dealokasi node
void Deallocate(address P) {
    delete P;
}

// menyisipkan node di awal list
void InsertFirst(List *L, infotype x) {
    address P = Allocation(x);
    P->next = L->first;
    L->first = P;
}

// menyisipkan node setelah node tertentu
void InsertAfter(address Prec, infotype x) {
    if (Prec != nullptr) {
        address P = Allocation(x);
        P->next = Prec->next;
        Prec->next = P;
    }
}

// menyisipkan node secara terurut menaik
void InsertSorted(List *L, infotype x) {
    // jika list kosong atau x lebih kecil dari elemen pertama
    if (IsEmpty(*L) || L->first->info > x) {
        InsertFirst(L, x);
    } else {
        // cari posisi yang tepat untuk sisipkan x
        address temp = L->first;
        while (temp->next != nullptr && temp->next->info < x) {
            temp = temp->next;
        }
        InsertAfter(temp, x);
    }
}

// menampilkan seluruh elemen list
void ShowData(List L) {
    address temp = L.first;
```

```
while (temp != nullptr) {
    cout << temp->info << " ";
    temp = temp->next;
}
cout << endl;
}

// menghapus seluruh elemen dalam list
void DeallocateAll(List *L) {
    address temp = L->first;
    while (temp != nullptr) {
        address nextNode = temp->next;
        Deallocate(temp);
        temp = nextNode;
    }
    L->first = nullptr;
}

int main() {
    List siswa;
    siswa.first = nullptr;

    int n;
    cout << "Masukkan jumlah data: ";
    cin >> n;

    for (int i = 0; i < n; i++) {
        infotype x;
        cout << "Data ke-" << i + 1 << ": ";
        cin >> x;
        InsertSorted(&siswa, x); // menyisipkan secara terurut
        cout << "List sementara: ";
        ShowData(siswa);
    }

    cout << "List akhir (terurut ascending): ";
    ShowData(siswa);

    DeallocateAll(&siswa); // menghapus semua elemen sebelum program selesai
    return 0;
}
```

Dokumentasi Hasil Running

```
Masukkan jumlah data: 7
Data ke-1: 100
List sementara: 100
Data ke-2: 12
List sementara: 12 100
Data ke-3: 45
List sementara: 12 45 100
Data ke-4: 75
List sementara: 12 45 75 100
Data ke-5: 24
List sementara: 12 24 45 75 100
Data ke-6: 90
List sementara: 12 24 45 75 90 100
Data ke-7: 58
List sementara: 12 24 45 58 75 90 100
List akhir (terurut ascending): 12 24 45 58 75 90 100
```

Gambar 02. Program berhasil menerima inputan angka dan langsung mengurutkan angka nya

Contoh Penggunaan:



Gambar 1. Situasi saya saat ini.

Link GitHub/GDB Online:

1. <https://github.com/arfakyu/Praktikum-ASD-RD>

Referensi

KALAU KALIAN PAKE GEN AI (CHAT GPT, GEMINI, CLAUDE, DLL. KALIAN BISA SHARE LINK PERCAKAPAN LINK GEN AI KALIAN DI SINI SEBAGAI BAHAN REFERENSI CODE KALIAN)