

Laporan Praktikum

Algoritma dan Struktur Data

Ganjil 2025/2026

Program Studi Teknik Informatika
Institut Teknologi Sumatera



Modul : Single Linked List

Nama : Juliani Leony Putri Melati
Manalu

NIM : 123140029

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 di insert pada posisi yang tepat. List selalu dalam kondisi terurut ascending

Dasar Teori

Pengertian

Single Linked List adalah struktur data dinamis yang terdiri dari rangkaian *node* yang saling terhubung melalui pointer. Setiap *node* menyimpan dua informasi utama, yaitu data dan alamat (pointer) yang menunjuk ke *node* berikutnya.

Struktur Node

Setiap *node* terdiri dari:

- Info → berisi data yang disimpan.
- Next → menunjuk ke alamat *node* berikutnya.

Linked List dimulai dari node pertama (first/head) dan berakhir di node terakhir, yang *next*-nya bernilai NULL.

Operasi Dasar pada Single Linked List

1. Inisialisasi – membuat list kosong.
2. Insert (penambahan data) – di awal, di tengah (setelah node tertentu), atau di akhir list.
3. Delete (penghapusan data) – menghapus node dari awal, tengah, atau akhir.
4. Traversal – menelusuri dan menampilkan isi list.
5. Search – mencari data tertentu di dalam list.

Source Code

Soal 1

```
1. #include <iostream>
2. using namespace std;
3.
4. //struktur node (tempat nyimpan data)
5. struct Node {
6.     string nama;      //nama siswa
7.     int nilai;        //nilai siswa
8.     Node* next;       //penunjuk ke data berikutnya
9. };
10.
11. Node* head = NULL;   //awal list (kosong dulu)
12.
13. //fungsi untuk nambah data di akhir
14. void tambahData(string nama, int nilai) {
15.     Node* baru = new Node; //buat node baru
16.     baru->nama = nama;      //isi nama
17.     baru->nilai = nilai;    //isi nilai
18.     baru->next = NULL;     //belum ada sambungan
19.
20.     if (head == NULL) {    //kalau list masih kosong
21.         head = baru;       //node baru jadi data pertama
22.     } else {
23.         Node* bantu = head; //mulai dari awal
24.         while (bantu->next != NULL) //cari data terakhir
25.             bantu = bantu->next;
26.         bantu->next = baru; //sambungkan di akhir
27.     }
28.
29.     cout << "Data siswa berhasil ditambah!\n";
```

```
30. }
31.
32. //fungsi untuk menampilkan semua data
33. void tampilData() {
34.     if (head == NULL) {
35.         cout << "Belum ada data!\n";
36.     } else {
37.         Node* bantu = head; //mulai dari data pertama
38.         cout << "\nDaftar Nilai Siswa:\n";
39.         while (bantu != NULL) { //selama masih ada data
40.             cout << bantu->nama << " - Nilai: " << bantu->nilai <<
endl;
41.             bantu = bantu->next; //lanjut ke data berikutnya
42.         }
43.     }
44. }
45.
46. int main() {
47.     int pilihan;
48.     string nama;
49.     int nilai;
50.
51.     do {
52.         cout << "\n=== MENU NILAI SISWA ===\n";
53.         cout << "1. Tambah Data\n";
54.         cout << "2. Lihat Semua Data\n";
55.         cout << "0. Keluar\n";
56.         cout << "Pilih menu: ";
57.         cin >> pilihan;
58.
59.         if (pilihan == 1) {
60.             cout << "Masukkan Nama Siswa: ";
61.             cin >> nama;
62.             cout << "Masukkan Nilai: ";
63.             cin >> nilai;
64.             tambahData(nama, nilai);
65.         }
66.         else if (pilihan == 2) {
67.             tampilData();
68.         }
69.
70.     } while (pilihan != 0);
```

```
71.  
72.     cout << "Program selesai!\n";  
73.     return 0;  
74. }  
75.
```

Soal 2

```
1. #include <iostream>  
2. using namespace std;  
3.  
4. //struktur node sederhana  
5. struct Node {  
6.     int nilai;           //data yang disimpan  
7.     Node* next;         //penunjuk ke data berikutnya  
8. };  
9.  
10. Node* head = NULL;    //awal list (masih kosong)  
11.  
12. //fungsi untuk menambah data biar tetap urut kecil ke besar  
13. void tambahTerurut(int nilaiBaru) {  
14.     Node* baru = new Node;    //buat node baru  
15.     baru->nilai = nilaiBaru;  //isi nilai  
16.     baru->next = NULL;        //belum nyambung ke mana-mana  
17.  
18.     //kalau list kosong atau nilai baru lebih kecil dari nilai  
    pertama  
19.     if (head == NULL || nilaiBaru < head->nilai) {  
20.         baru->next = head;    //node baru jadi paling depan  
21.         head = baru;  
22.     } else {  
23.         Node* bantu = head;  
24.         //cari posisi yang tepat buat disisipin  
25.         while (bantu->next != NULL && bantu->next->nilai <  
            nilaiBaru) {  
26.             bantu = bantu->next;  
27.         }  
28.         //sambungkan node baru di posisi yang pas  
29.         baru->next = bantu->next;  
30.         bantu->next = baru;  
31.     }  
32.     cout << "Data " << nilaiBaru << " berhasil dimasukkan secara  
    urut!\n";
```

```
33. }
34.
35. //fungsi untuk menampilkan isi list
36. void tampilData() {
37.     if (head == NULL) {
38.         cout << "Belum ada data!\n";
39.     } else {
40.         Node* bantu = head;
41.         cout << "\nData (Ascending): ";
42.         while (bantu != NULL) {
43.             cout << bantu->nilai << " ";
44.             bantu = bantu->next;
45.         }
46.         cout << endl;
47.     }
48. }
49.
50. int main() {
51.     int nilai;
52.     char lanjut;
53.
54.     do {
55.         cout << "Masukkan nilai: ";
56.         cin >> nilai;
57.         tambahTerurut(nilai);
58.         tampilData();
59.
60.         cout << "Tambah lagi? (y/n): ";
61.         cin >> lanjut;
62.     } while (lanjut == 'y' || lanjut == 'Y');
63.
64.     cout << "Program selesai!\n";
65.     return 0;
66. }
```

Dokumentasi Hasil Running

Soal 1

```
=== MENU NILAI SISWA ===
1. Tambah Data
2. Lihat Semua Data
0. Keluar
Pilih menu: 1
Masukkan Nama Siswa: Juliani
Masukkan Nilai: 90
Data siswa berhasil ditambah!

=== MENU NILAI SISWA ===
1. Tambah Data
2. Lihat Semua Data
0. Keluar
Pilih menu: 1
Masukkan Nama Siswa: Leony
Masukkan Nilai: 100
Data siswa berhasil ditambah!

=== MENU NILAI SISWA ===
1. Tambah Data
2. Lihat Semua Data
0. Keluar
Pilih menu: 2

Daftar Nilai Siswa:
Juliani - Nilai: 90
Leony - Nilai: 100

=== MENU NILAI SISWA ===
1. Tambah Data
2. Lihat Semua Data
0. Keluar
Pilih menu: 0
Program selesai!
```

Gambar 1. Saat program dijalankan, pengguna memilih menu Tambah Data, lalu memasukkan nama dan nilai siswa. Data tersimpan secara berurutan sesuai waktu input. Ketika memilih menu Lihat Semua Data, program menampilkan seluruh daftar siswa beserta nilainya dari awal hingga akhir list.

Soal 2

```
Masukkan nilai: 90
Data 90 berhasil dimasukkan secaraurut!

Data (Ascending): 90
Tambah lagi? (y/n): y
Masukkan nilai: 100
Data 100 berhasil dimasukkan secaraurut!

Data (Ascending): 90 100
Tambah lagi? (y/n): n
Program selesai!
```

Gambar 1. Setiap kali pengguna memasukkan nilai baru, program langsung menempatkannya di posisi yang sesuai agar data tetap terurut dari kecil ke besar. Hasil tampilan menunjukkan semua nilai tersusun secara otomatis tanpa perlu diurutkan ulang.

Link GitHub/GDB Online:

Link GitHub:

<https://github.com/juliani07/Laprak-ASD-RD-123140029.git>

Referensi

<https://chatgpt.com/share/68e1f57a-392c-8011-bd73-efe7db6d29f8>