

**Modul :** Single LinkedList

**Nama :** Aulia Putri Sayidina

**NIM :** 122140060

**Kelas (Kelas Asal) :** RC

Instruksi sederhana :

* Disarankan kepada **Praktikan Algoritma Struktur Data** untuk mengeditnya menggunakan Google Docs agar tidak berantakan dan rapi,
* Silahkan mengganti **Nama Modul** baik yang ada pada **Cover** dan **Header** sesuai dengan materi praktikum,
* Gunakan text styling seperti **Heading 1**, **Normal Text** yang telah terformat / Text Style lainnya yang digunakan untuk menjaga estetika laporan,
* Gunakan [**Syntax Highlighter**](https://highlight.hohli.com/index.php)untuk merapikan kode yang sudah Praktikan buat ke dalam Laporan Praktikum.

**Dirancang oleh Kevin Simorangkir Terinspirasi dari Format Tata Tulis Laporan Embedded**

# Materi Praktikum

PENDAHULUAN:

Dalam struktur data, Linked List adalah suatu struktur yang terdiri dari elemen-elemen data yang disusun dalam urutan linear. Setiap elemen dalam Linked List disebut sebagai "node," dan setiap node memiliki dua bagian utama: data dan pointer yang menunjuk ke node berikutnya. Salah satu jenis Linked List yang umum digunakan adalah Single Linked List.

FUNGSI LINKED LIST:

Linked List memberikan fleksibilitas dalam menyimpan dan mengelola data, terutama ketika ukuran data tidak diketahui pada awalnya atau membutuhkan penambahan dan penghapusan data secara dinamis.

Single linkedlist terdiri dari value dan pointer next.

# Link repl.it Source Code

<https://onlinegdb.com/C8UCUmv06>

# Source Code

1. #include <iostream>
2. using namespace std;
4. // Struktur untuk node
5. **struct Node {**
6. int data;
7. Node\* next;
8. };
10. **// Fungsi untuk membuat node baru**
11. Node\* createNode(int value) {
12. Node\* newNode = new Node;
13. newNode->data = value;
14. newNode->next = nullptr;
15. **return newNode;**
16. }
18. // Fungsi untuk menambahkan node ke dalam linked list terurut
19. void insertSorted(Node\*& head, int value) {
20. **Node\* newNode = createNode(value);**
22. // Kasus jika linked list masih kosong atau nilai baru lebih kecil dari head
23. if (head == nullptr || value < head->data) {
24. newNode->next = head;
25. **head = newNode;**
26. return;
27. }
29. // Menemukan tempat yang tepat untuk menyisipkan node
30. **Node\* current = head;**
31. while (current->next != nullptr && current->next->data < value) {
32. current = current->next;
33. }
35. **// Menyisipkan node baru setelah current**
36. newNode->next = current->next;
37. current->next = newNode;
38. }
40. **// Fungsi untuk menampilkan isi linked list**
41. void displayList(Node\* head) {
42. Node\* current = head;
43. while (current != nullptr) {
44. cout << current->data << " ";
45. **current = current->next;**
46. }
47. cout << endl;
48. }
50. **int main() {**
51. Node\* head = nullptr;
52. int input;
54. cout << "Masukkan angka (integer) atau -1 untuk selesai: ";
56. while (true) {
57. cin >> input;
59. // Selesai jika input -1
60. **if (input == -1) {**
61. break;
62. }
64. // Memasukkan data ke dalam linked list secara terurut
65. **insertSorted(head, input);**
67. // Menampilkan isi linked list
68. cout << "Linked List saat ini: ";
69. displayList(head);
71. cout << "Masukkan angka (integer) atau -1 untuk selesai: ";
72. }
74. return 0;
75. **}**

# Dokumentasi Hasil Running

**Gambar 1.** Hasil Running Code Single LinkedList

# Referensi

Modul perkuliahan Algoritma Struktur Data – Single LinkedList dari ITERA