

Laporan Praktikum

Algoritma dan Struktur Data

Ganjil 2025/2026

Program Studi Teknik Informatika

Institut Teknologi Sumatera



Modul : Single List

Nama : Andini

NIM : 124140192

Kelas (Kelas Asal) : RD



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

Silakan berikan penjelasan tentang teori terkait materi.

Di paragraf akhir, silahkan berikan informasi tambahan jika ada yang ingin disampaikan terlebih dahulu (misal error atau hal lainnya)

Source Code

NO.1

```
#include <iostream>
#include <string>

using namespace std;

struct Node {
    string nama;
    float nilai;
    Node* next;
};

Node* head = nullptr;

void tambahData(string nama, float nilai) {
    Node* newNode = new Node;
    newNode->nama = nama;
    newNode->nilai = nilai;
    newNode->next = nullptr;

    if (head == nullptr) {
        head = newNode;
    } else {
        Node* temp = head;
        while (temp->next != nullptr) {
            temp = temp->next;
        }
        temp->next = newNode;
    }

    cout << "Data siswa berhasil ditambahkan!\n";
}

void tampilkanData() {
    if (head == nullptr) {
        cout << "Belum ada data siswa.\n";
    }
}
```

```
        return;
    }

    Node* temp = head;
    cout << "\nDaftar Nilai Siswa:\n";
    cout << "-----\n";
    while (temp != nullptr) {
        cout << "Nama : " << temp->nama << endl;
        cout << "Nilai: " << temp->nilai << endl;
        cout << "-----\n";
        temp = temp->next;
    }
}

void hapusSemuaData() {
    Node* temp;
    while (head != nullptr) {
        temp = head;
        head = head->next;
        delete temp;
    }
}

int main() {
    int pilihan;
    string nama;
    float nilai;

    do {
        cout << "\nMenu:\n";
        cout << "1. Tambah Data Siswa\n";
        cout << "2. Tampilkan Semua Data\n";
        cout << "3. Keluar\n";
        cout << "Pilihan Anda: ";
        cin >> pilihan;
        cin.ignore();

        switch (pilihan) {
            case 1:
                cout << "Masukkan nama siswa: ";
                getline(cin, nama);
```

```
        cout << "Masukkan nilai siswa: ";
        cin >> nilai;
        tambahData(nama, nilai);
        break;
    case 2:
        tampilkanData();
        break;
    case 3:
        cout << "Terima kasih!\n";
        break;
    default:
        cout << "Pilihan tidak valid. Coba lagi.\n";
    }

} while (pilihan != 3);

hapusSemuaData();
return 0;
}
```

NO.2

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

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

struct List {
    Node* first;
};

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

void InsertFirst(List &L, int x) {
    Node* newNode = new Node;
    newNode->info = x;
```

```
newNode->next = L.first;
L.first = newNode;
}
```

```
void InsertAfter(Node* prevNode, int x) {
    if (prevNode == nullptr) return;
    Node* newNode = new Node;
    newNode->info = x;
    newNode->next = prevNode->next;
    prevNode->next = newNode;
}
```

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

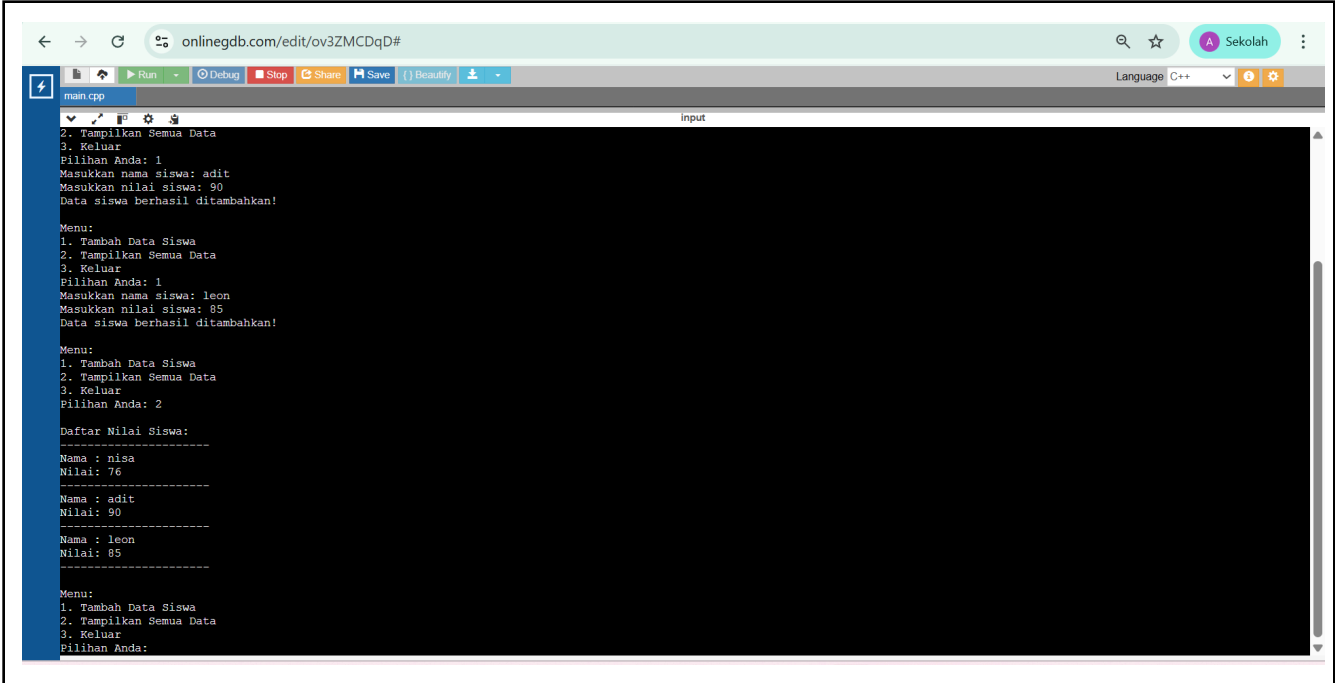
```
void PrintList(List L) {
    Node* temp = L.first;
    while (temp != nullptr) {
        cout << temp->info << " ";
        temp = temp->next;
    }
    cout << endl;
}
```

```
int main() {
    List L;
    L.first = nullptr;

    InsertSorted(L, 5);
    InsertSorted(L, 2);
}
```

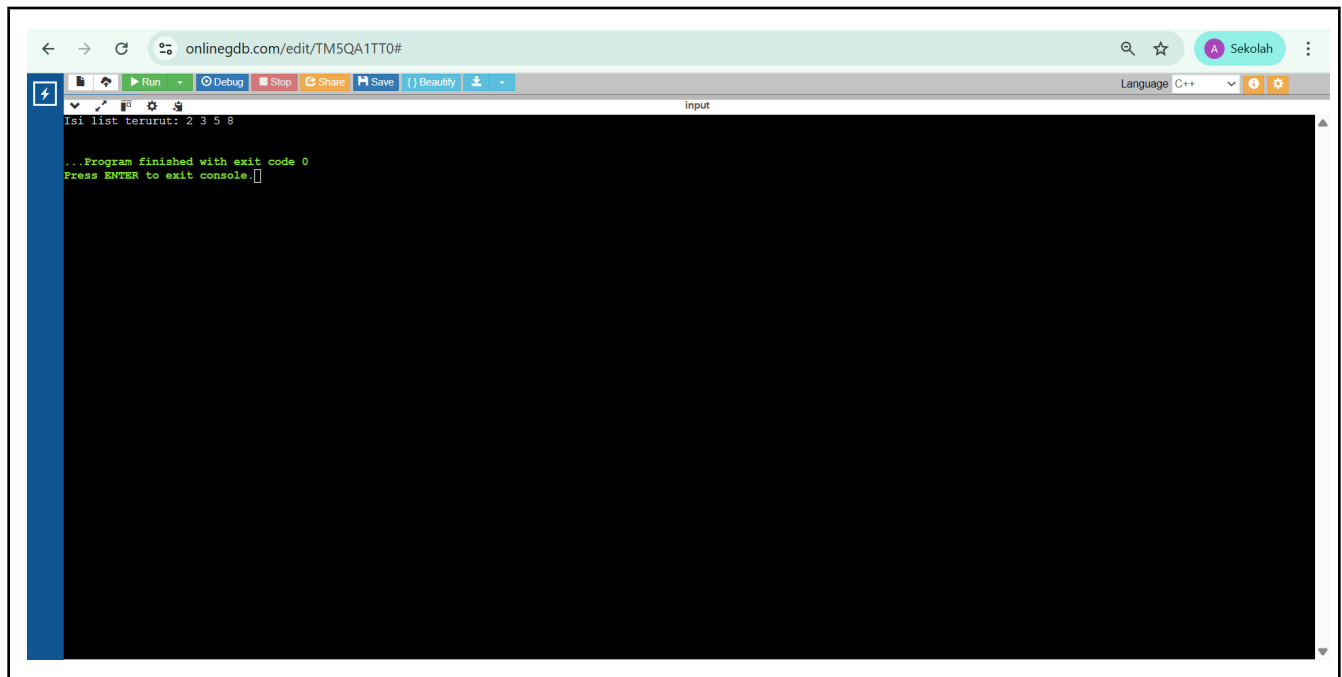
```
InsertSorted(L, 8);  
InsertSorted(L, 3);  
  
cout << "Isi list terurut: ";  
PrintList(L);  
return 0;  
}
```

Dokumentasi Hasil Running



```
onlinegdb.com/edit/ov3ZMCDqD#  
main.cpp  
2. Tampilkan Semua Data  
3. Keluar  
Pilihan Anda: 1  
Masukkan nama siswa: adit  
Masukkan nilai siswa: 90  
Data siswa berhasil ditambahkan!  
  
Menu:  
1. Tambah Data Siswa  
2. Tampilkan Semua Data  
3. Keluar  
Pilihan Anda: 1  
Masukkan nama siswa: leon  
Masukkan nilai siswa: 85  
Data siswa berhasil ditambahkan!  
  
Menu:  
1. Tambah Data Siswa  
2. Tampilkan Semua Data  
3. Keluar  
Pilihan Anda: 2  
  
Daftar Nilai Siswa:  
-----  
Nama : nisa  
Nilai: 76  
-----  
Nama : adit  
Nilai: 90  
-----  
Nama : leon  
Nilai: 85  
-----  
  
Menu:  
1. Tambah Data Siswa  
2. Tampilkan Semua Data  
3. Keluar  
Pilihan Anda:
```

Gambar 1. Kode akan menjalankan program sesuai dengan keinginan user, ditampilkan pilihan yang dimana setiap nomor nya memiliki output berbeda.



The screenshot shows a web browser window with the URL `onlinegdb.com/edit/TM5QA1TT0#`. The browser's address bar and tabs are visible. Below the browser window, there is a dark-themed code editor with a blue sidebar on the left. The code editor contains the following text:

```
Isi list terurut: 2 3 5 8
...Program finished with exit code 0
Press ENTER to exit console.
```

Gambar 2. Program akan mengurutkan nilai dari yang terkecil sampai yang terbesar

Contoh Penggunaan:

1. Bisa digunakan dalam sistem pendidikan, contoh nya sistem rekap nilai dinamis.
2. Daftar ranking nilai otomatis tanpa perlu sorting manual.
3. Sistem daftar transaksi berdasarkan nominal.

Link GitHub/GDB Online:

1. <https://github.com/andini124140192-oss/LAPRAK-ASD-RD-192.git>

Referensi

<https://chatgpt.com/share/68ea817d-5d84-800f-86ab-2bfa551ecbfd>