**LAPORAN PRAKTIKUM 5**

**STRUKTUR DATA**

**A logo of a university

AI-generated content may be incorrect.**

Dosen Pengampu:  
Dr. Wahyudi, S.T, M.Kom.

Disusun Oleh:  
Muhammad Luthfi Kautsar Rizata – 2311532020

**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**2025**

**A screen shot of a computer program

AI-generated content may be incorrect.1. Class TambahSLL**

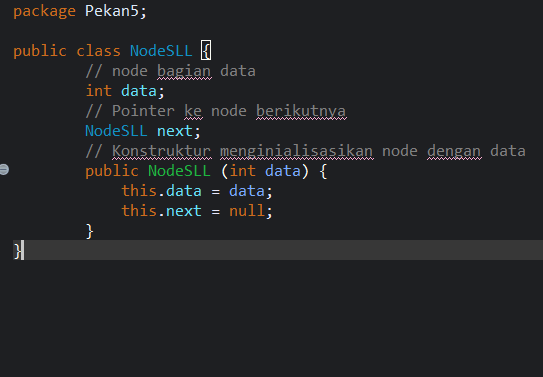
Program ini bertujuan untuk mengilustrasikan berbagai operasi penambahan simpul (node) pada struktur data *singly linked list* (SLL), yaitu penambahan di awal, di akhir, dan di posisi tertentu. Struktur NodeSLL digunakan sebagai representasi simpul, di mana setiap simpul menyimpan data bertipe int dan referensi ke simpul berikutnya.

Pertama, program membuat linked list awal dengan elemen 2 → 3 → 5 → 6. Kemudian, fungsi insertAtFront() dipanggil untuk menambahkan satu simpul dengan data 1 di awal list. Fungsi ini bekerja dengan membuat simpul baru, lalu menunjuk simpul tersebut ke head lama, dan akhirnya menjadikan simpul baru sebagai head yang baru. Hasilnya adalah list: 1→2→3→5→6.

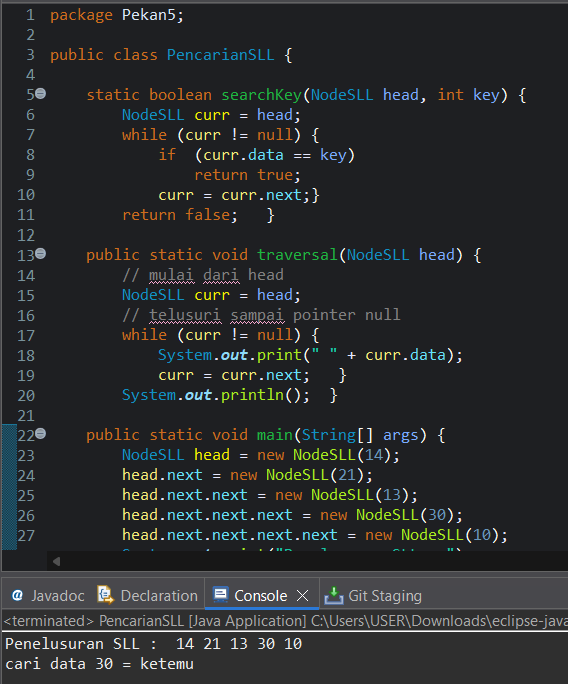
Selanjutnya, fungsi insertAtEnd() digunakan untuk menambahkan simpul dengan data 7 di akhir list. Fungsi ini menelusuri list hingga simpul terakhir, lalu menyambungkan simpul terakhir ke simpul baru. Hasilnya menjadi: 1→2→3→5→6→7.

Operasi ketiga adalah penambahan simpul di posisi tertentu, dalam hal ini di posisi ke-4 (antara simpul dengan data 3 dan 5). Fungsi insertPos() digunakan dengan cara menelusuri list hingga simpul sebelum posisi yang dituju. Setelah itu, simpul baru disisipkan dengan mengatur referensi next dari simpul sebelumnya dan simpul baru tersebut. Jika posisi yang diberikan tidak valid (kurang dari 1 atau melebihi panjang list), fungsi akan menampilkan pesan kesalahan. Dengan menambahkan simpul bernilai 4 di posisi ke-4, list berubah menjadi: 1→2→3→4→5→6→7.

Seluruh proses cetak list dilakukan menggunakan fungsi printList(), yang menampilkan isi list dengan panah --> sebagai pemisah antar simpul. Program ini memperlihatkan dasar manipulasi node dalam singly linked list dan sangat berguna untuk memahami cara kerja struktur data linier berbasis pointer.

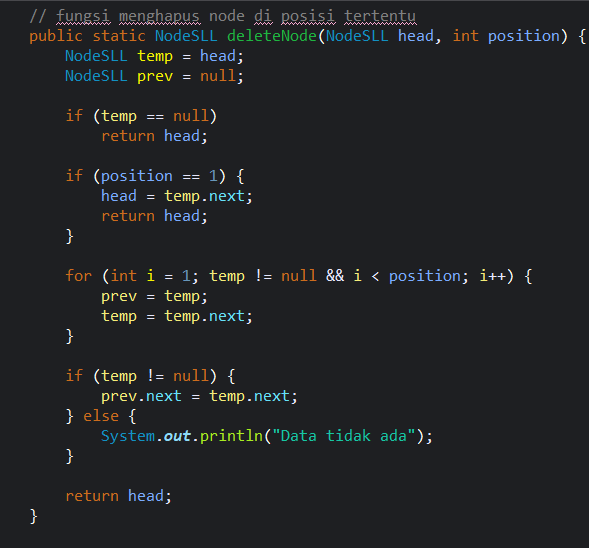
**2. Class NodeSLL**

Kelas NodeSLL merupakan representasi dasar dari sebuah simpul (node) dalam struktur data *singly linked list*. Di dalam kelas ini terdapat dua atribut utama, yaitu data bertipe int yang digunakan untuk menyimpan nilai pada simpul, serta next yang merupakan referensi (pointer) ke simpul berikutnya dalam list. Atribut next bertipe NodeSLL, memungkinkan struktur data ini saling terhubung satu sama lain secara berantai. Konstruktor dalam kelas ini berfungsi untuk menginisialisasi objek simpul dengan nilai data tertentu yang diterima sebagai parameter, sekaligus menyetel pointer next ke null, yang menandakan bahwa simpul ini belum terhubung ke simpul lainnya saat pertama kali dibuat. Kelas ini menjadi komponen fundamental dalam operasi-operasi yang dilakukan pada *singly linked list*, seperti penambahan, penghapusan, maupun penelusuran data.

**3. Class PencarianSLL**

Program ini merupakan implementasi sederhana dari proses penelusuran dan pencarian nilai dalam struktur data *singly linked list* (SLL). Fungsi traversal() digunakan untuk menelusuri dan mencetak isi dari list, dimulai dari simpul pertama (head) hingga simpul terakhir, dengan cara mengakses setiap simpul melalui pointer next. Sementara itu, fungsi searchKey() bertugas untuk mencari apakah sebuah nilai tertentu (key) terdapat di dalam list. Pencarian dilakukan dengan membandingkan nilai data dari setiap simpul dengan nilai key yang dicari. Jika ditemukan kecocokan, fungsi akan segera mengembalikan nilai true. Jika seluruh simpul telah ditelusuri dan tidak ditemukan nilai yang dicari, maka akan mengembalikan false.

Pada fungsi main(), sebuah list dibuat secara manual dengan lima simpul berisi data 14, 21, 13, 30, dan 10. Program kemudian menampilkan isi list melalui pemanggilan fungsi traversal(), dan selanjutnya melakukan pencarian terhadap nilai 30 menggunakan fungsi searchKey(). Karena nilai tersebut ada di dalam list, maka output yang ditampilkan adalah “ketemu”. Program ini menggambarkan bagaimana konsep pencarian linear diterapkan dalam konteks linked list.

**4. HapusSLL**

Program ini mendemonstrasikan proses penghapusan simpul (node) dalam struktur data *singly linked list* (SLL), baik dari bagian awal, akhir, maupun dari posisi tertentu dalam list. Fungsi deleteHead() menghapus simpul pertama dengan cukup memindahkan head ke simpul berikutnya. Fungsi removeLastNode() menghapus simpul terakhir dengan menelusuri hingga simpul kedua terakhir, kemudian mengatur pointer next-nya menjadi null agar memutus referensi ke simpul terakhir. Sedangkan deleteNode() digunakan untuk menghapus simpul di posisi tertentu, dengan logika: jika posisi adalah 1, maka fungsi akan memanggil penghapusan head. Jika tidak, fungsi akan menelusuri simpul-simpul hingga mencapai posisi yang diinginkan, lalu mengubah referensi next dari simpul sebelumnya agar melewati simpul yang ingin dihapus.

Pada bagian main(), program terlebih dahulu membuat list dengan data 1->2->3->4->5->6. Kemudian, program melakukan tiga kali penghapusan secara berurutan: pertama, menghapus head, sehingga list menjadi 2->3->4->5->6. Kedua, menghapus simpul terakhir, mengubah list menjadi 2->3->4->5. Ketiga, menghapus simpul pada posisi ke-2 (dalam hal ini nilai 3), sehingga list akhir menjadi 2->4->5. Dengan pendekatan ini, program memberikan gambaran yang jelas tentang bagaimana manipulasi pointer bekerja dalam penghapusan simpul pada SLL.