**LAPORAN PRAKTIKUM 6**

**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 code

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

Kelas NodeDLL merupakan representasi dari sebuah simpul (*node*) dalam struktur data *doubly linked list* (DLL). Setiap node menyimpan tiga komponen utama, yaitu: data untuk menyimpan nilai atau informasi, next yang menunjuk ke simpul berikutnya dalam list, dan prev yang menunjuk ke simpul sebelumnya. Adanya dua pointer (next dan prev) memungkinkan traversal dua arah, baik maju maupun mundur. Konstruktor NodeDLL(int data) digunakan untuk menginisialisasi node baru dengan nilai tertentu, sekaligus mengatur pointer next dan prev ke null sebagai penanda bahwa node belum terhubung dengan node lain saat dibuat. Struktur dasar ini sangat penting dalam pengelolaan *doubly linked list*, karena memudahkan penambahan atau penghapusan node di berbagai posisi tanpa perlu traversal dari awal seperti pada *singly linked list*.

**2. Class InsertDLL**

****

Program di atas merupakan implementasi dari operasi penambahan simpul (*node*) dalam struktur data Doubly Linked List (DLL) menggunakan bahasa Java. Program ini terdiri dari beberapa metode yang masing-masing bertugas untuk menambahkan simpul di awal, akhir, atau posisi tertentu dari DLL.

Pertama-tama, pada metode insertBegin, program membuat simpul baru dengan data yang diberikan, lalu menghubungkannya ke simpul awal (head) dari DLL. Jika simpul awal tidak kosong, maka pointer prev dari simpul awal akan diarahkan ke simpul baru tersebut. Setelah proses ini selesai, simpul baru akan menjadi simpul pertama dalam DLL.

Selanjutnya, metode insertEnd digunakan untuk menambahkan simpul di akhir DLL. Jika head kosong, simpul baru akan menjadi head. Namun jika tidak, program akan menelusuri DLL hingga menemukan simpul terakhir, lalu menghubungkan simpul baru di akhir, serta mengatur pointer prev pada simpul baru agar menunjuk ke simpul sebelumnya.

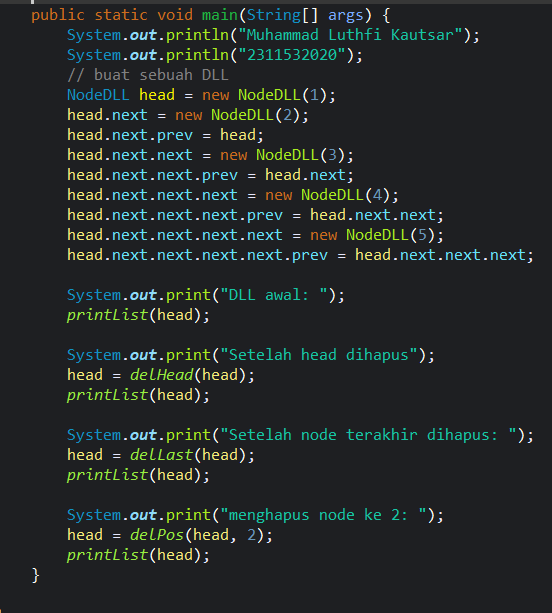
Metode insertAtPosition digunakan untuk menambahkan simpul pada posisi tertentu dalam DLL. Jika posisi yang dimaksud adalah 1, simpul baru akan dipasang seperti dalam metode insertBegin. Jika tidak, program akan menelusuri DLL hingga mencapai simpul sebelum posisi yang diinginkan. Jika posisi valid, maka simpul baru akan disisipkan di antara dua simpul, dengan pointer next dan prev disesuaikan agar struktur DLL tetap terhubung dua arah. Jika posisi yang dimasukkan melebihi panjang list, program akan menampilkan pesan "Posisi tidak ada".

Metode printList bertugas untuk mencetak seluruh isi DLL dari head hingga simpul terakhir, dengan format visualisasi yang menunjukkan hubungan dua arah antar simpul (<->).

Pada bagian main, program memulai dengan membuat sebuah DLL awal yang berisi tiga simpul: 2, 3, dan 5. Setelah itu, program menambahkan simpul 1 di awal list menggunakan insertBegin, menambahkan simpul 6 di akhir list menggunakan insertEnd, dan menyisipkan simpul 4 di posisi ke-4 menggunakan insertAtPosition. Setiap perubahan pada DLL dicetak menggunakan metode printList agar pengguna dapat melihat hasil dari setiap operasi.

A screen shot of a computer code

AI-generated content may be incorrect.**3. HapusDLL**

Program di atas merupakan implementasi penghapusan simpul (node) dalam struktur data Doubly Linked List (DLL) menggunakan bahasa Java. Program ini terdiri dari tiga fungsi utama yang masing-masing menangani penghapusan simpul di awal, akhir, dan posisi tertentu dalam DLL, serta satu fungsi tambahan untuk mencetak isi list.

Fungsi pertama, delHead, digunakan untuk menghapus simpul pertama (head) dari DLL. Jika head kosong, maka fungsi akan langsung mengembalikan null. Jika tidak kosong, head akan diarahkan ke simpul berikutnya, dan pointer prev dari simpul baru tersebut akan diset ke null, karena sekarang ia menjadi simpul pertama.

Fungsi kedua, delLast, digunakan untuk menghapus simpul terakhir dari DLL. Jika list kosong atau hanya berisi satu simpul, maka fungsi akan mengembalikan null, artinya list menjadi kosong. Jika ada lebih dari satu simpul, maka program akan menelusuri DLL hingga simpul terakhir. Setelah itu, pointer next dari simpul sebelumnya akan diset ke null, memutus simpul terakhir dari list.

Fungsi ketiga, delPos, digunakan untuk menghapus simpul pada posisi tertentu. Program akan menelusuri list hingga mencapai simpul pada posisi yang dimaksud. Jika posisi tersebut tidak ditemukan (karena melebihi panjang list), maka tidak ada perubahan yang dilakukan. Jika posisi valid, maka pointer next dari simpul sebelumnya dan prev dari simpul berikutnya akan diperbarui, sehingga simpul pada posisi tersebut dihapus dari rangkaian DLL. Jika simpul yang dihapus adalah head, maka head akan diperbarui ke simpul berikutnya. Fungsi printList hanya digunakan untuk menampilkan isi dari DLL saat ini, dengan mencetak nilai dari setiap simpul secara berurutan dari head ke tail.

Pada bagian main, program dimulai dengan membuat sebuah DLL yang terdiri dari lima simpul berisi angka 1 hingga 5. Setelah itu, program menghapus simpul pertama dengan delHead, lalu simpul terakhir dengan delLast, dan terakhir menghapus simpul pada posisi ke-2 menggunakan delPos. Setelah setiap operasi penghapusan, program mencetak isi DLL untuk menunjukkan perubahan yang terjadi.

Secara keseluruhan, program ini menggambarkan proses manipulasi dasar pada struktur Doubly Linked List, dengan perhatian khusus pada pengaturan pointer next dan prev agar list tetap terhubung secara konsisten dua arah setelah simpul dihapus.