LAPORAN PRAKTIKUM

LINKEDLIST



Oleh :

MUHAMMAD GALID AVERO

NIM 2311532008

MATA KULIAH STRUKTUR DATA

DOSEN PENGAMPU : DR. WAHYUDI, S.T, M.T

ASISTEN LABORATORIUM : ZAKY ADIL HAKIM

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, MEI 2024

1. PENDAHULUAN

Linked list adalah salah satu struktur data linear yang terdiri dari sejumlah node yang dihubungkan secara berurutan. Setiap node dalam linked list terdiri dari dua bagian utama: data dan referensi (atau pointer) ke node berikutnya dalam urutan. Linked list dapat digunakan untuk menyimpan koleksi data yang dinamis, yang ukurannya dapat berubah-ubah saat program berjalan. Linked list sering digunakan dalam berbagai aplikasi seperti implementasi stack, queue, dan berbagai algoritma lainnya yang membutuhkan struktur data dinamis.

Berikut adalah beberapa karakteristik dan jenis linked list:

**Karakteristik Linked List**

1. **Node**: Unit dasar dari linked list yang menyimpan data dan pointer ke node berikutnya.
2. **Head**: Node pertama dari linked list.
3. **Tail**: Node terakhir dari linked list, yang pointer-nya mengarah ke null (menandakan akhir dari linked list).
4. **Dynamic Size**: Ukuran linked list dapat berubah-ubah, tidak seperti array yang memiliki ukuran tetap.

**Jenis-Jenis Linked List**

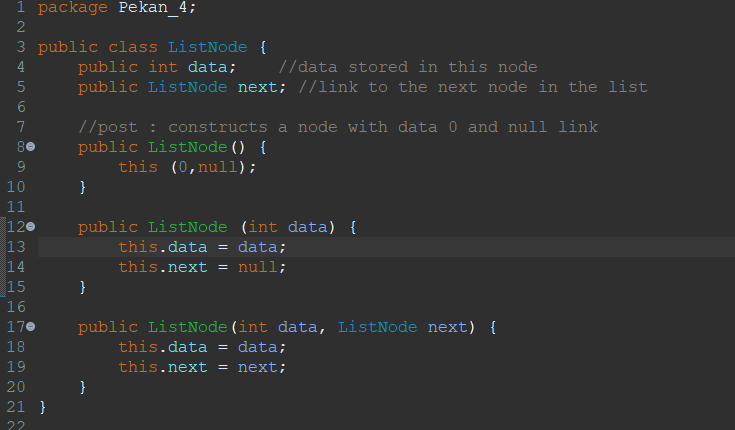
1. **Singly Linked List**: Setiap node hanya memiliki satu pointer yang mengarah ke node berikutnya.
2. **Doubly Linked List**: Setiap node memiliki dua pointer, satu mengarah ke node berikutnya dan satu lagi mengarah ke node sebelumnya.
3. **Circular Linked List**: Node terakhir terhubung kembali ke node pertama, membentuk lingkaran.
4. **Doubly Circular Linked List**: Sama seperti circular linked list, tetapi dengan node yang memiliki dua pointer (seperti pada doubly linked list).

**Kelebihan Linked List**

1. **Fleksibilitas Ukuran**: Mudah untuk menambah atau menghapus elemen tanpa harus mengalokasikan ulang seluruh struktur data.
2. **Efisiensi Insert/Delete**: Operasi penambahan dan penghapusan elemen lebih efisien dibandingkan dengan array jika posisi elemen diketahui.

**Kekurangan Linked List**

1. **Akses Lambat**: Waktu akses elemen lebih lambat karena harus melalui pointer satu per satu, berbeda dengan array yang memiliki akses langsung.
2. **Penggunaan Memori Tambahan**: Setiap node membutuhkan memori tambahan untuk menyimpan pointer.
3. TUJUAN PRAKTIKUM
4. Mengetahui cara membuat Lingle LinkedList
5. Mengetahui cara membuat Double LinkedList
6. Mengetahui cara menggunakan fungsi pada linkedlist
7. LANGKAH PRAKTIKUM
8. Penggunaan Fungsi untuk menambahkan linkedlist



Setiap node dalam linked list memiliki dua komponen:

data: sebuah variabel yang menyimpan nilai integer yang terkait dengan node tersebut.

next: sebuah referensi yang mengarah ke node berikutnya dalam daftar terkait.

Konstruktor

Kelas ListNode memiliki tiga konstruktor yang berbeda:

Konstruktor tanpa parameter

public ListNode() {

this(0, null);

}

Konstruktor ini akan membuat sebuah node dengan nilai data bernilai 0 dan next bernilai null. Artinya, node ini tidak memiliki nilai yang terkait dan tidak mengarah ke node lain.

Konstruktor dengan satu parameter

public ListNode(int data) {

this.data = data;

this.next = null;

}

Konstruktor ini akan membuat sebuah node dengan nilai data sesuai dengan parameter yang diberikan dan next bernilai null. Artinya, node ini memiliki nilai yang terkait tetapi tidak mengarah ke node lain.

Konstruktor dengan dua parameter

public ListNode(int data, ListNode next) {

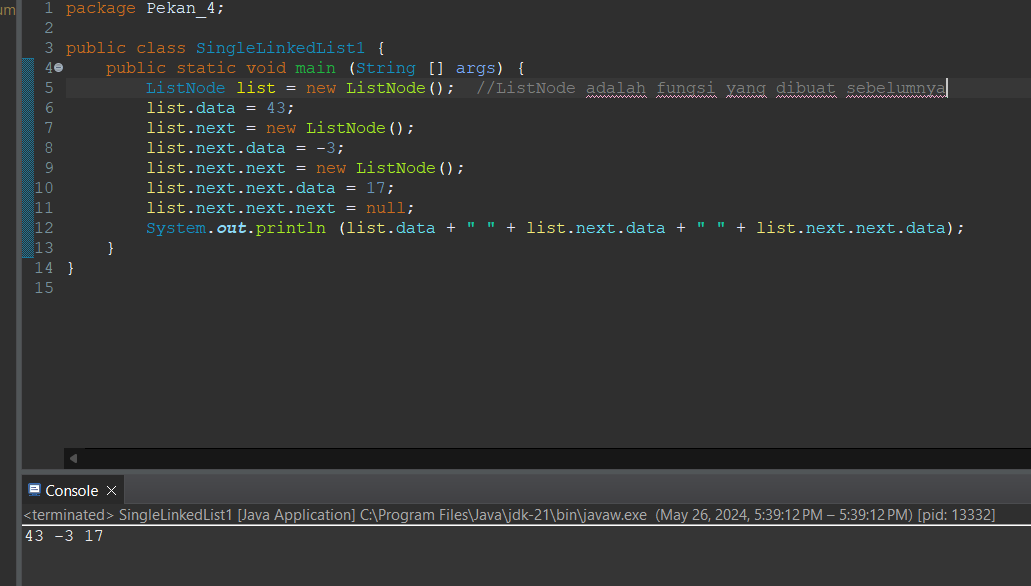
this.data = data;

this.next = next;

}

Konstruktor ini akan membuat sebuah node dengan nilai data sesuai dengan parameter pertama dan next sesuai dengan parameter kedua. Artinya, node ini memiliki nilai yang terkait dan mengarah ke node lain yang terkait dengan parameter kedua.

Berikut adalah contoh dari penggunaan kelas listnode di atas beserta hasil kode nya

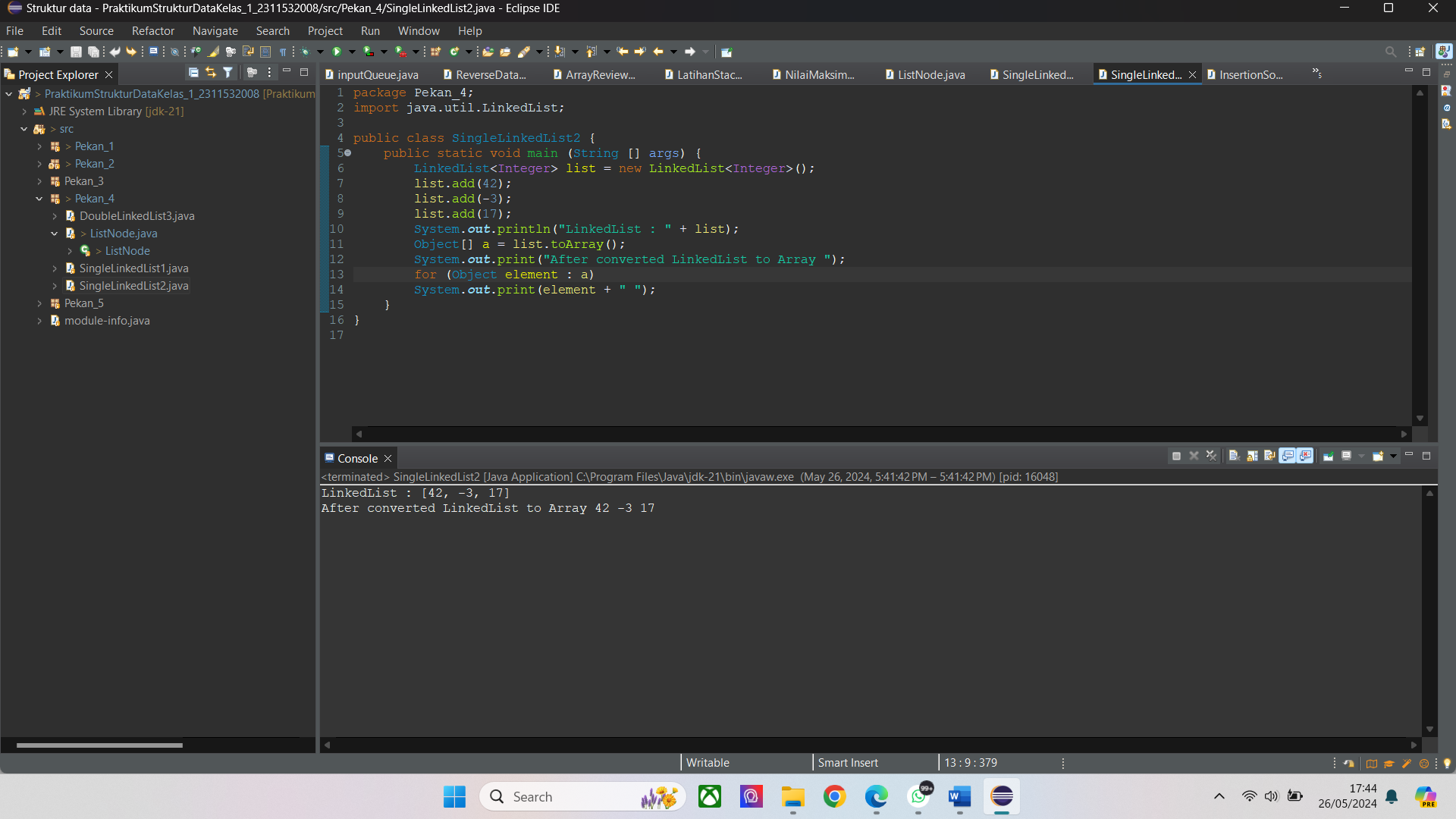


Dalam kelas ini, terdapat sebuah variabel list bertipe ListNode. ListNode adalah kelas yang merepresentasikan node dalam linked list. Setiap node memiliki dua komponen: data dan next. data adalah nilai yang disimpan dalam node, sedangkan next adalah referensi ke node selanjutnya.

Dalam metode main, kita membuat instance dari kelas ListNode dan menyimpan referensinya ke variabel list. Kemudian, kita mengisi nilai dari data pada node pertama dengan angka 43. Setelah itu, kita membuat node baru dan menyimpannya ke next dari node pertama. Node baru ini memiliki nilai data dengan angka -3. Kita melakukan hal yang sama untuk node ketiga dengan nilai data dengan angka 17.

Selanjutnya, kita mencetak nilai dari data pada setiap node dengan menggunakan perintah System.out.println.

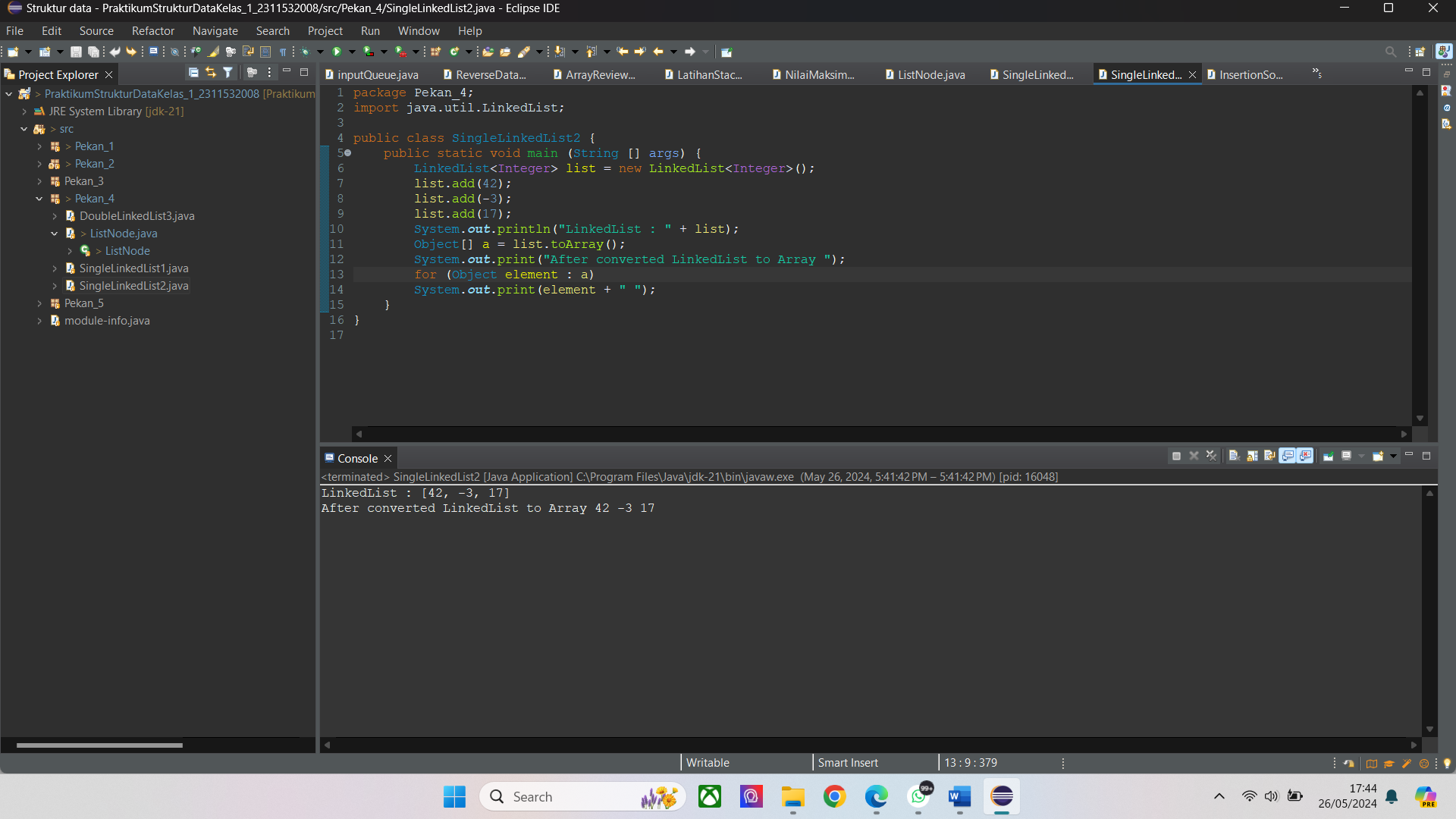
1. Mengubah bentuk dari LinkedList menjadi Array



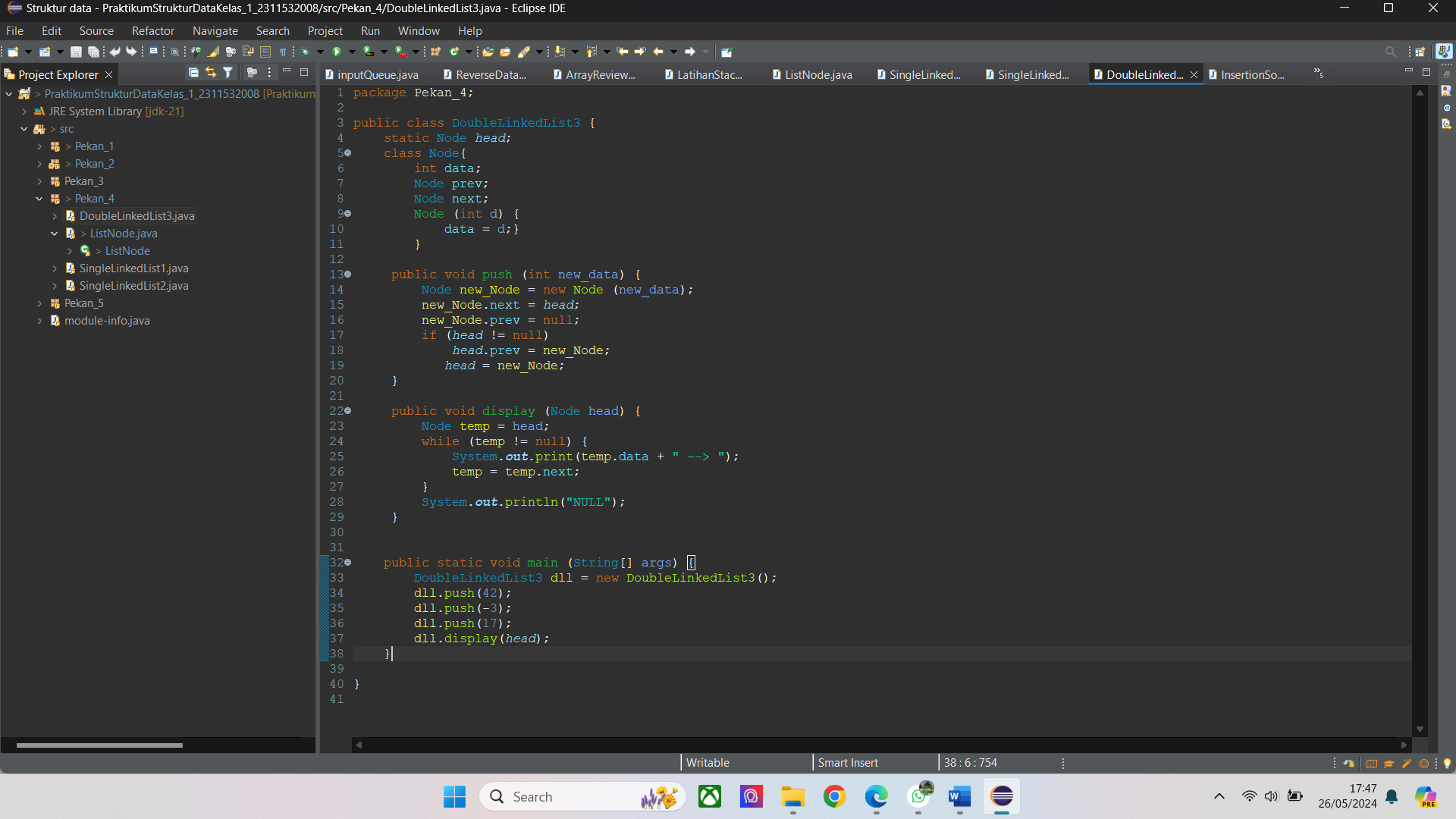
Dalam kelas ini, terdapat sebuah variabel list bertipe LinkedList<Integer>. LinkedList adalah kelas yang disediakan oleh koleksi Java dan dapat digunakan sebagai linked list. Kita menginisialisasi variabel list dengan membuat instance dari kelas LinkedList dan menggunakan tipe data generic Integer.

Dalam metode main, kita menambahkan tiga elemen ke linked list menggunakan metode add. Kemudian, kita mencetak linked list menggunakan metode println dan menampilkan nilai dari setiap elemen dengan menggunakan tanda spasi. Hasilnya adalah:

Setelah itu, kita mengubah linked list menjadi array menggunakan metode toArray. Metode ini mengembalikan array dengan tipe data yang sama dengan tipe data generic yang digunakan pada linked list. Kemudian, kita mencetak setiap elemen dari array menggunakan perulangan for-each. Hasilnya adalah:



1. Penggunaan Double LinkedList untuk memasukkan suatu nilai integer



Dalam kelas ini, terdapat dua komponen utama: Node dan DoubleLinkedList3 itu sendiri.

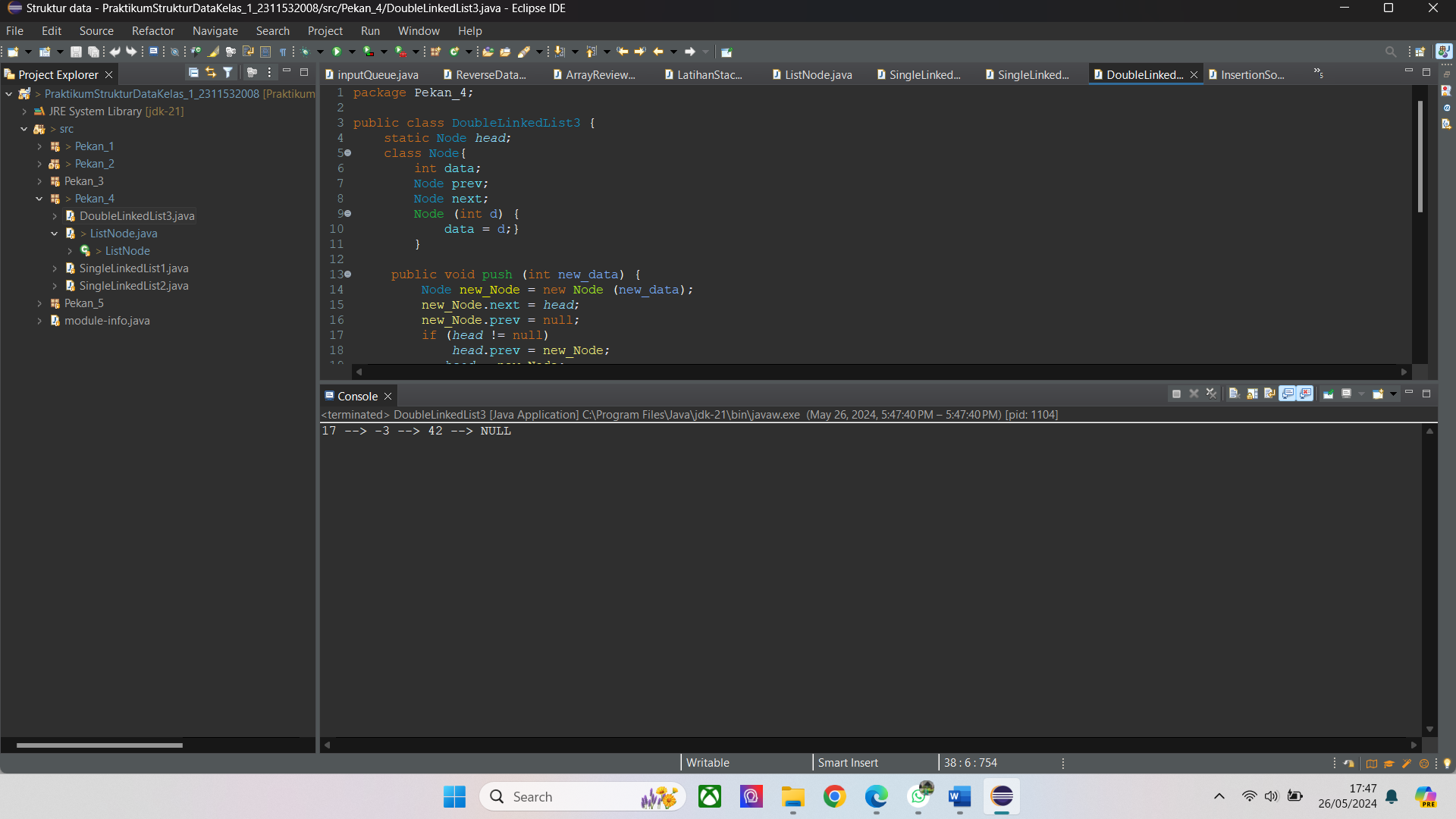
Node adalah komponen yang menyimpan data dan referensi ke node sebelumnya (prev) dan node selanjutnya (next).

DoubleLinkedList3 adalah kelas yang mengelola node-node tersebut. Kelas ini memiliki dua metode: push dan display.

Metode push digunakan untuk menambahkan node baru ke awal daftar. Node baru ini akan menjadi kepala daftar, dan node sebelumnya akan menjadi node selanjutnya dari node baru ini.

Metode display digunakan untuk menampilkan isi dari daftar. Metode ini akan mencetak nilai dari setiap node dalam daftar, diikuti oleh tanda panah (-->) untuk menunjukkan hubungan antar node.

Dalam metode main, kita membuat instance dari kelas DoubleLinkedList3 dan menambahkan tiga node ke daftar menggunakan metode push. Kemudian, kita menampilkan isi dari daftar menggunakan metode display. Untuk hasil dari kode nya adalah sebagai berikut



1. KESIMPULAN

Kesimpulan yang dapat diambil oleh penulis dari praktikum kali ini adalah pada penggunaan linkedlist dan juga double linkedlist adalah bisa untuk berbagai aplikasi seperti implementasi stack, queue, dan berbagai algoritma lainnya yang membutuhkan struktur data dinamis. Selain itu juga dapat dikombinasikan penggunaannya seperti pengubahan tipe data dari integer menjadi array dan juga berbagai jenis kombinasi lainnya.