**LAPORAN PRAKTIKUM STRUKTUR DATA**

SLL dan DLL



Oleh :

FAYI AMATULLAH AZHARA

NIM 2311537001

MATA KULIAH

STRUKTUR DATA

DOSEN PENGAMPU : Dr.WAHYUDI ST, MT.

**FAKULTAS TEKNOLOGI INFORMASI**

**DEPARTEMEN INFORMATIKA**

**UNIVERSITAS ANDALAS**

1. **PENDAHULUAN**

Singly Linked List (SLL) adalah struktur data di mana setiap elemen, atau node, berisi data dan pointer yang mengarah ke node berikutnya dalam daftar. Struktur ini memungkinkan penambahan dan penghapusan elemen di awal atau akhir daftar dengan cepat, tetapi traversal hanya bisa dilakukan dalam satu arah, dari node pertama ke node terakhir. Karena hanya memiliki satu pointer per node, SLL lebih hemat memori dibandingkan Doubly Linked List (DLL). Namun, kelemahan utamanya adalah sulit untuk menghapus node tertentu karena tidak ada referensi ke node sebelumnya, sehingga perlu traversal dari awal hingga menemukan node yang ingin dihapus.

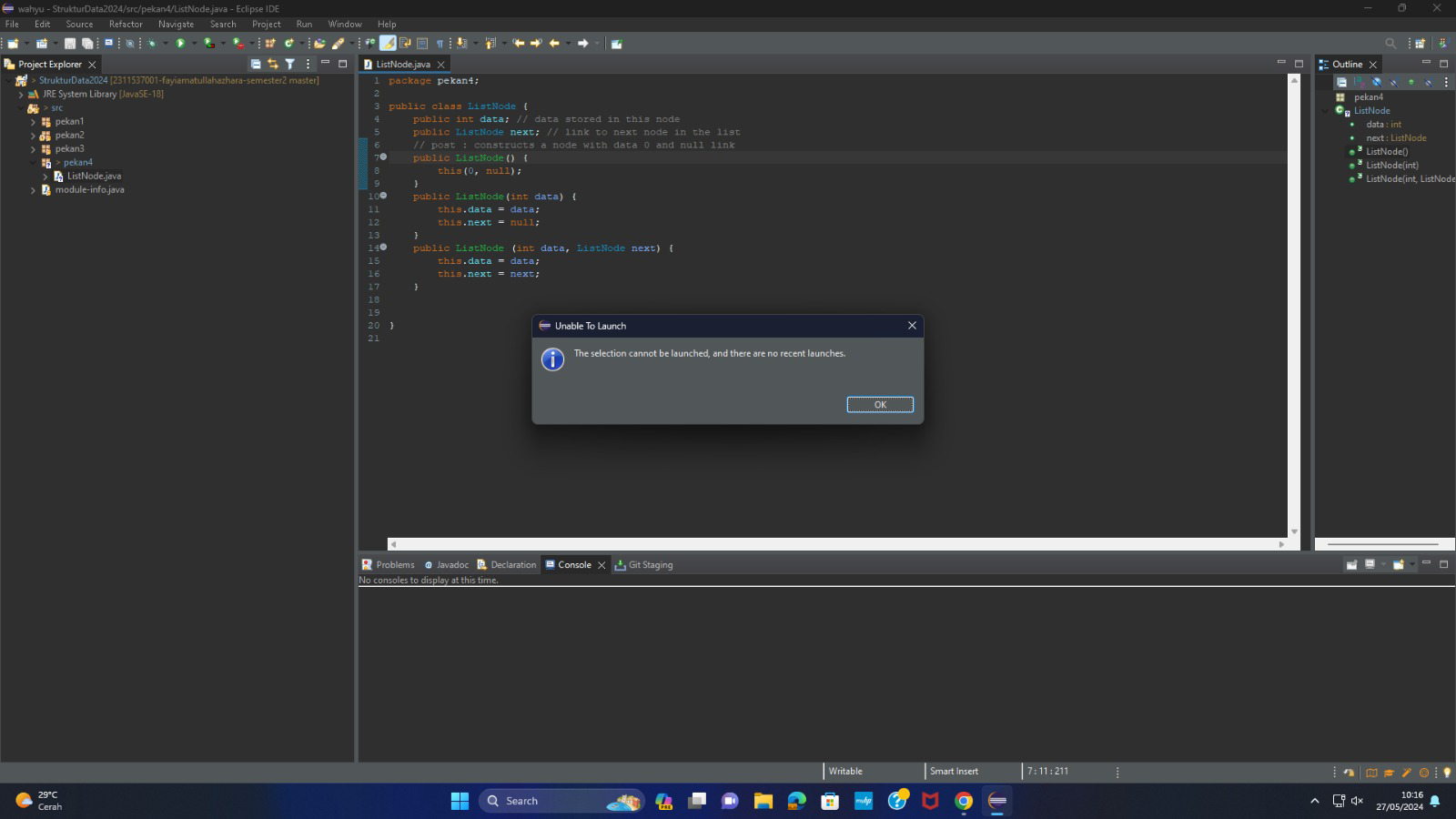
Doubly Linked List (DLL) adalah struktur data yang setiap nodenya berisi data dan dua pointer: satu mengarah ke node berikutnya dan satu lagi mengarah ke node sebelumnya. Dengan dua pointer ini, traversal dapat dilakukan dalam dua arah, memudahkan penambahan atau penghapusan elemen di mana saja dalam daftar. Meskipun lebih fleksibel, DLL membutuhkan lebih banyak memori karena setiap node memiliki dua pointer. Keunggulan utamanya adalah kemudahan dalam operasi penghapusan dan penambahan node di posisi manapun dalam daftar, membuatnya lebih efisien dalam banyak aplikasi dibandingkan SLL.

1. **TUJUAN**

Memahami Konsep dan Implementasi SLL dan DLL

1. **LANGKAH-LANGKAH**
2. Membuka Eclipse IDE dan import proyek github dengan nama “STRUKTURDATA2024”
3. Membuat Package baru dengan nama “pekan4”
4. Membuat Kelas baru dengan nama “ListNode”

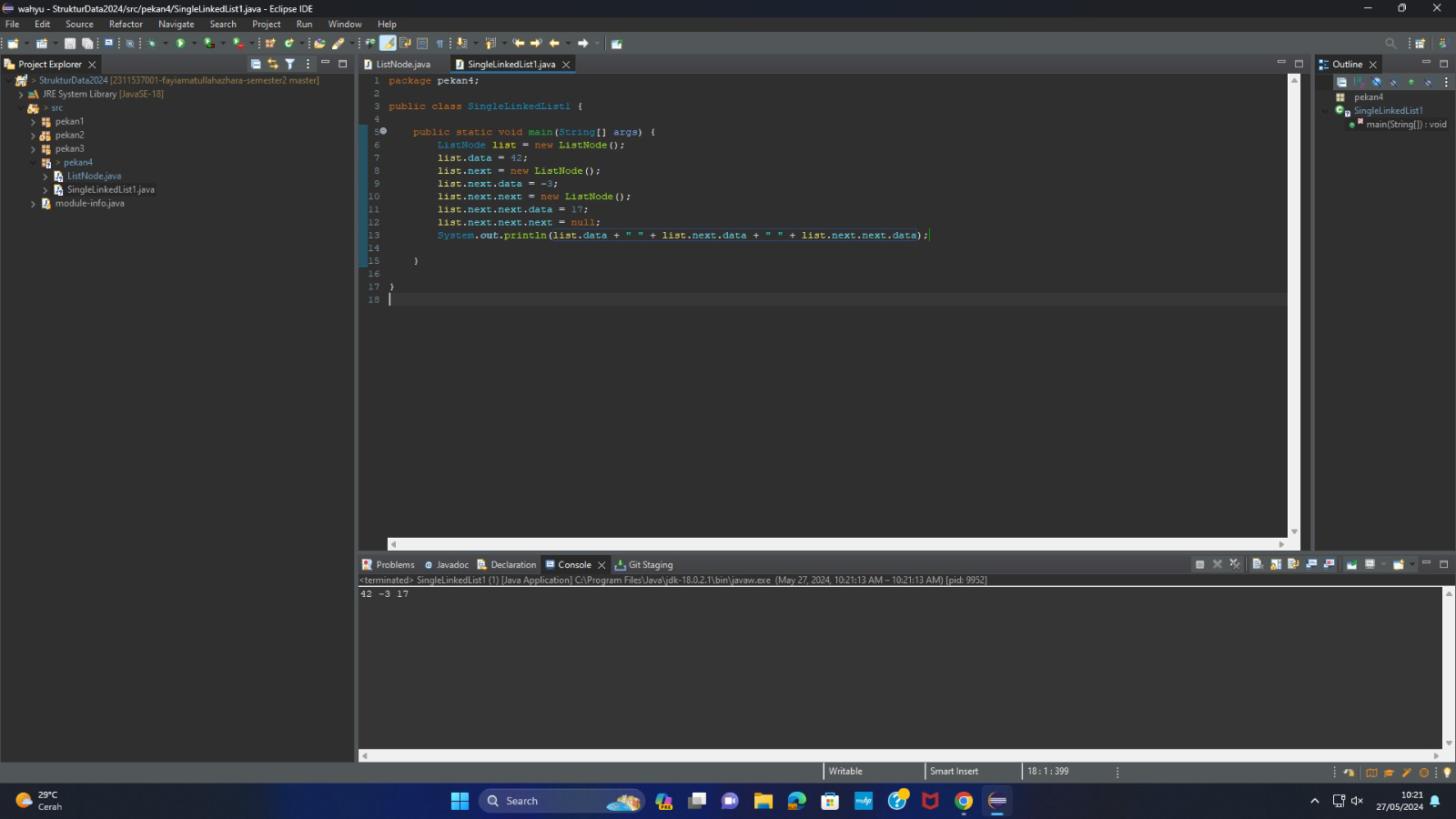
Pada kelas “ListNode”, masukkan kode seperti gambar di bawah



Kelas “ListNode” mendefinisikan node dari single linked list dengan dua atribut: data untuk menyimpan nilai dan next untuk mereferensikan node berikutnya. Kelas ini memiliki tiga konstruktor untuk membuat node dengan data tertentu dan referensi node berikutnya.

1. Membuat Kelas baru dengan nama “SingleLinkedList1”

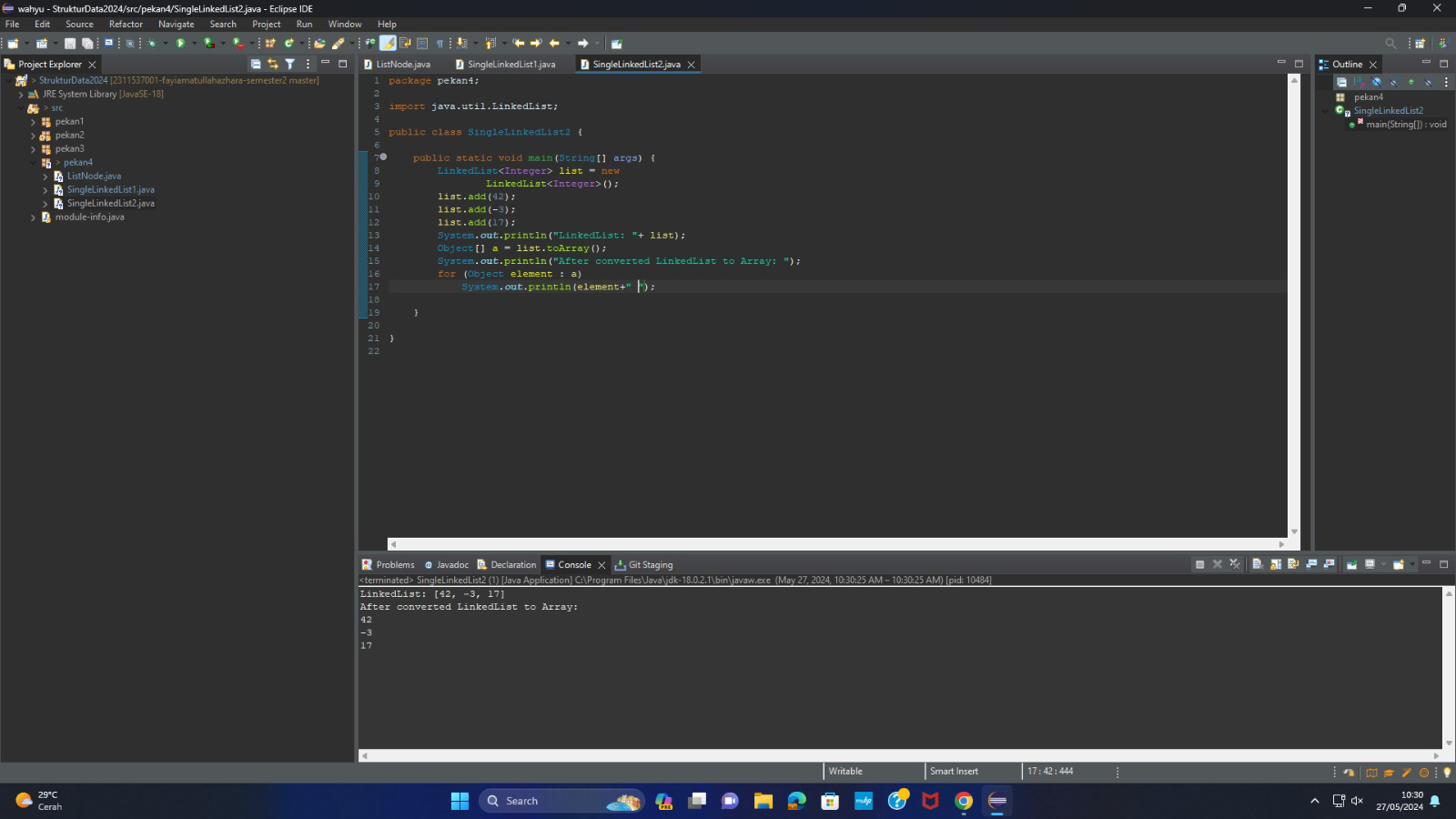
Pada kelas “SingleLinkedList1”, masukkan kode seperti gambar di bawah



kelas SingleLinkedList1 menggunakan ListNode untuk membuat linked list sederhana dengan tiga elemen: 42, -3, dan 17. Program ini mencetak nilai-nilai dari elemen-elemen linked list tersebut. Kelas ini menunjukkan cara manual untuk membangun dan mencetak single linked list.

1. Membuat Kelas baru dengan nama “SingleLinkedList2”

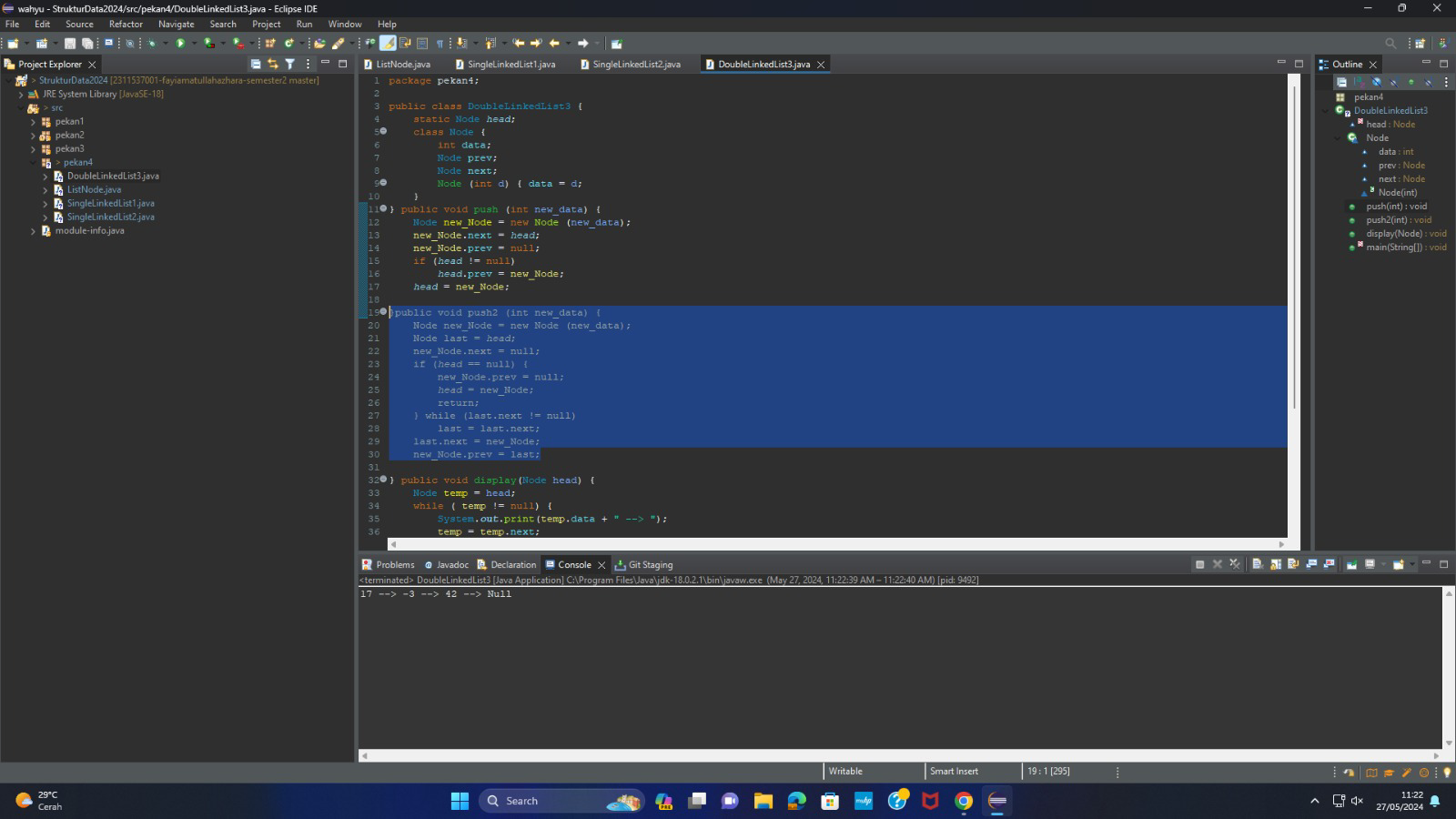
Pada kelas “SingleLinkedList2”, masukkan kode seperti gambar di bawah

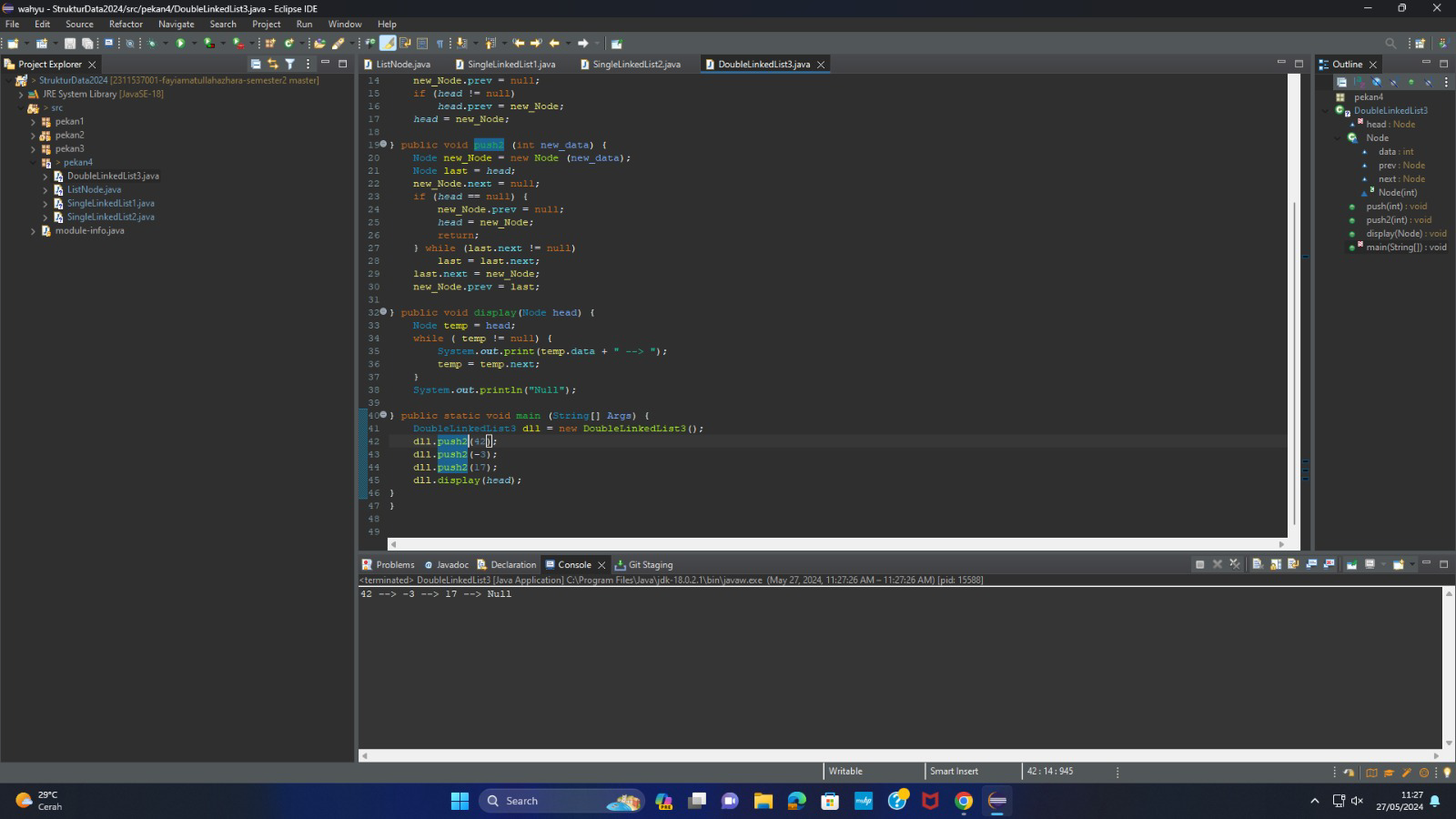


Kelas SingleLinkedList2 menggunakan LinkedList dari Java Collections Framework untuk membuat linked list dan mengonversinya ke array. Program ini menambahkan tiga elemen (42, -3, dan 17) ke dalam linked list, mencetaknya, kemudian mengonversinya ke array dan mencetak elemen-elemen dari array tersebut. Ini menunjukkan cara alternatif menggunakan koleksi Java untuk mengelola linked list.

1. Membuat Kelas baru dengan nama “DoubleLinkedList3”

Pada kelas “DoubleLinkedList3”, masukkan kode seperti gambar di bawah



 kelas DoubleLinkedList3 mengimplementasikan double linked list dengan mendefinisikan node yang memiliki referensi ke node sebelumnya dan berikutnya. Kelas ini menyediakan metode untuk menambah elemen di awal (push) dan di akhir (push2) dari linked list. Program ini menambahkan tiga elemen (42, -3, dan 17) ke dalam double linked list dan mencetaknya dari kepala sampai akhir.

**D. PENUTUP**

Dalam laporan praktikum ini, program ini menunjukkan berbagai cara untuk mengimplementasikan linked list dalam Java, baik secara manual maupun menggunakan Java Collections Framework, serta perbedaan antara single dan double linked list. Implementasi manual menggunakan kelas ListNode dan SingleLinkedList1 memberikan pemahaman mendalam tentang struktur data dasar dan cara mengelola node secara individual. Kelas SingleLinkedList2 memperlihatkan kemudahan dan efisiensi yang ditawarkan oleh Java Collections Framework dalam mengelola linked list dengan metode built-in yang lebih intuitif. Sementara itu, kelas DoubleLinkedList3 mengilustrasikan struktur data double linked list yang lebih kompleks, memungkinkan navigasi dua arah melalui node dengan penggunaan referensi ke node sebelumnya dan berikutnya.