**LAPORAN PRAKTIKUM STRUKTUR DATA**

Node, BTree, TreeMain



Oleh :

RIDHO RADYA PRATAMA

NIM 2311533020

MATA KULIAH

STRUKTUR DATA

DOSEN PENGAMPU : Dr.WAHYUDI ST, MT.

**FAKULTAS TEKNOLOGI INFORMASI**

**DEPARTEMEN INFORMATIKA**

**UNIVERSITAS ANDALAS**

1. **PENDAHULUAN**

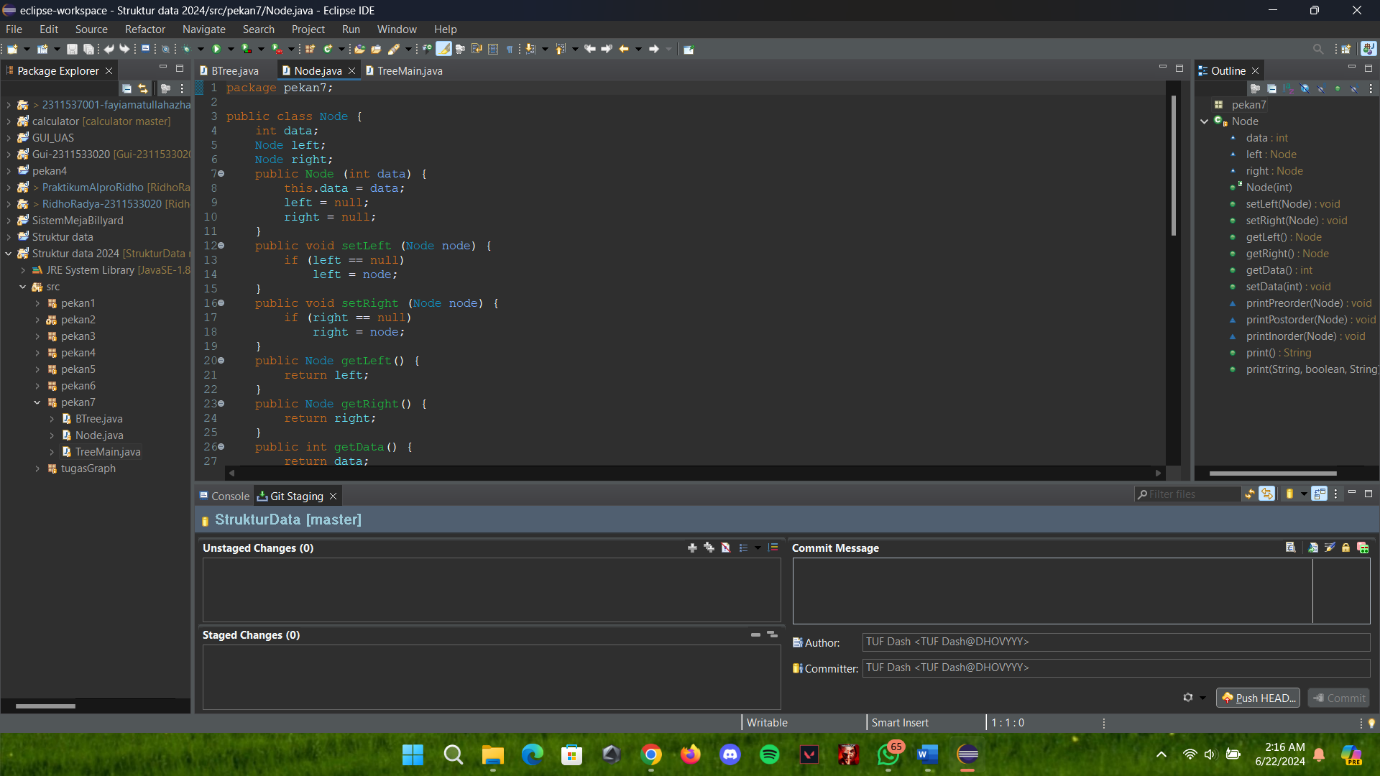
Node, BTree, dan TreeMain adalah komponen penting dalam struktur data dan algoritma yang sering digunakan dalam ilmu komputer untuk mengelola dan mengatur data secara efisien. Node adalah unit dasar dari struktur data seperti pohon dan grafik. Dalam konteks pohon, setiap node menyimpan data dan referensi (atau pointer) ke node lainnya, menciptakan hubungan hierarkis. Misalnya, dalam pohon biner, setiap node memiliki maksimal dua anak, yaitu anak kiri dan anak kanan. Node digunakan untuk menyusun berbagai jenis struktur data seperti pohon biner, pohon AVL, dan pohon B.

BTree adalah jenis pohon yang dirancang khusus untuk menyimpan data di disk atau perangkat penyimpanan lainnya secara efisien, yang membuatnya sangat cocok untuk sistem basis data dan sistem berkas. BTree merupakan pohon seimbang (balanced tree) di mana setiap node dapat memiliki lebih dari dua anak, berbeda dengan pohon biner. Keunggulan utama BTree adalah kemampuannya untuk menjaga keseimbangan tinggi pohon bahkan ketika data sering dimasukkan dan dihapus, yang memastikan waktu pencarian, penyisipan, dan penghapusan tetap efisien. TreeMain, di sisi lain, biasanya merujuk pada kelas atau modul utama dalam sebuah aplikasi yang mengimplementasikan atau mengelola pohon (tree). TreeMain dapat mencakup metode untuk manipulasi pohon seperti menambahkan atau menghapus node, mencari node tertentu, dan menampilkan struktur pohon. Dalam banyak aplikasi, TreeMain bertindak sebagai titik pusat untuk mengelola dan berinteraksi dengan struktur pohon yang lebih kompleks.

1. **TUJUAN**

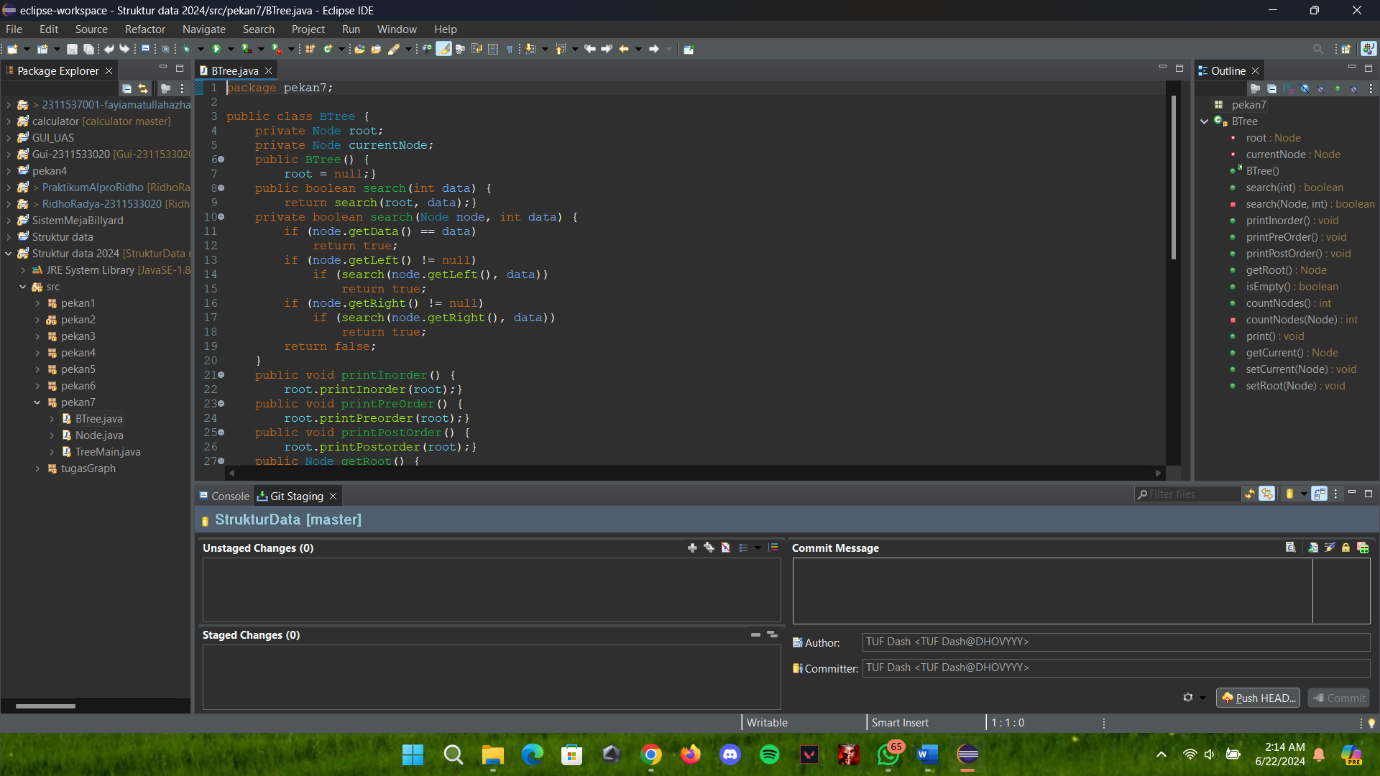
Memahami Konsep Node, Btree, dan TreeMain

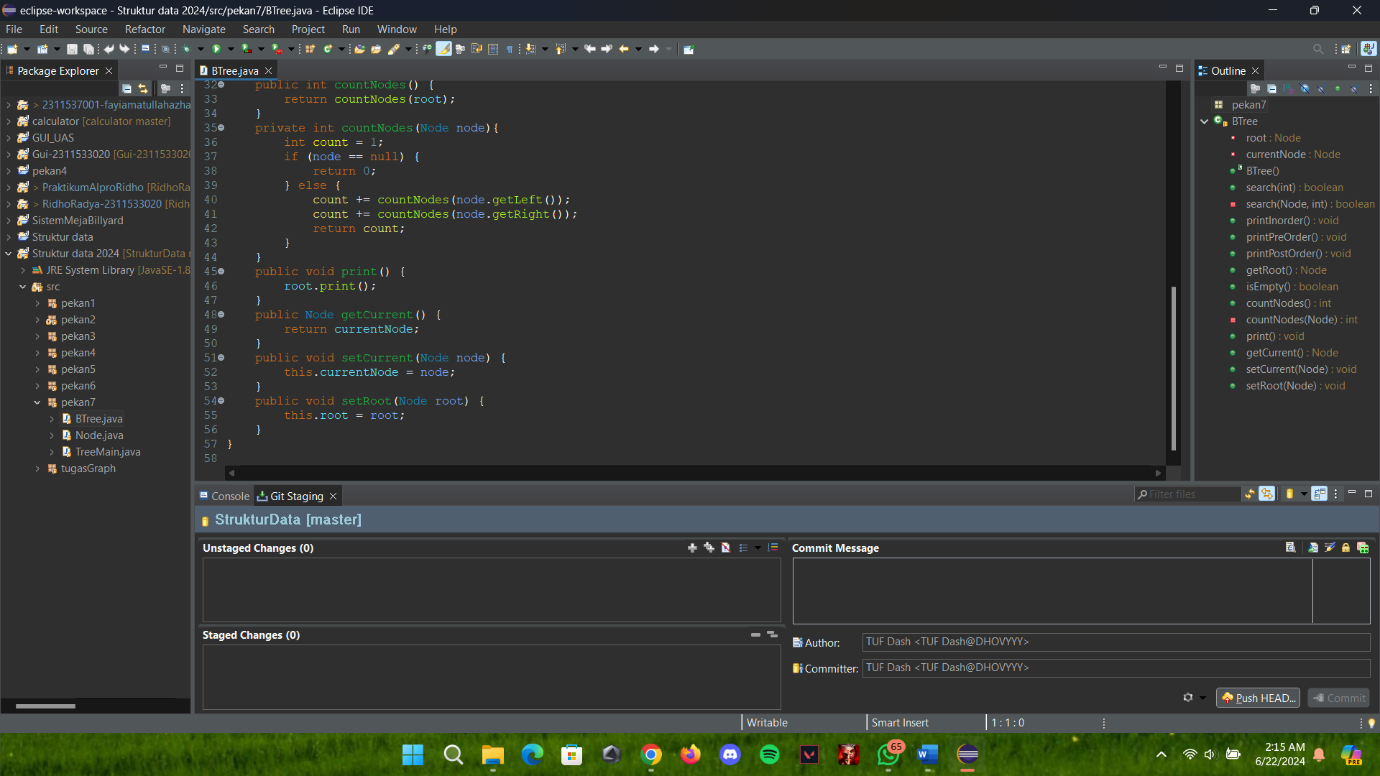
1. **LANGKAH-LANGKAH**
2. Membuka Eclipse IDE dan import proyek github dengan nama “STRUKTURDATA2024”
3. Membuat Package baru dengan nama “pekan7”
4. Membuat Kelas baru dengan nama “Node”
5. Pada kelas “Node”, masukkan kode seperti gambar di bawah



Program di atas mengimplementasikan kelas Node yang merepresentasikan sebuah node dalam pohon biner. Input yang diberikan ke dalam program berupa data integer yang disimpan dalam setiap node melalui konstruktor Node(int data). Metode setLeft dan setRight digunakan untuk menetapkan node anak kiri dan kanan, masing-masing, jika mereka belum diatur. Untuk output, program ini menyediakan beberapa metode untuk melakukan traversal dan mencetak data dari node-node dalam pohon. Metode printPreorder, printPostorder, dan printInorder mencetak data dari node dalam urutan preorder, postorder, dan inorder. Selain itu, metode print dan print(String prefix, boolean isTail, String sb) digunakan untuk mencetak representasi visual dari pohon biner, menampilkan struktur hierarkis pohon dengan node dan cabangnya.

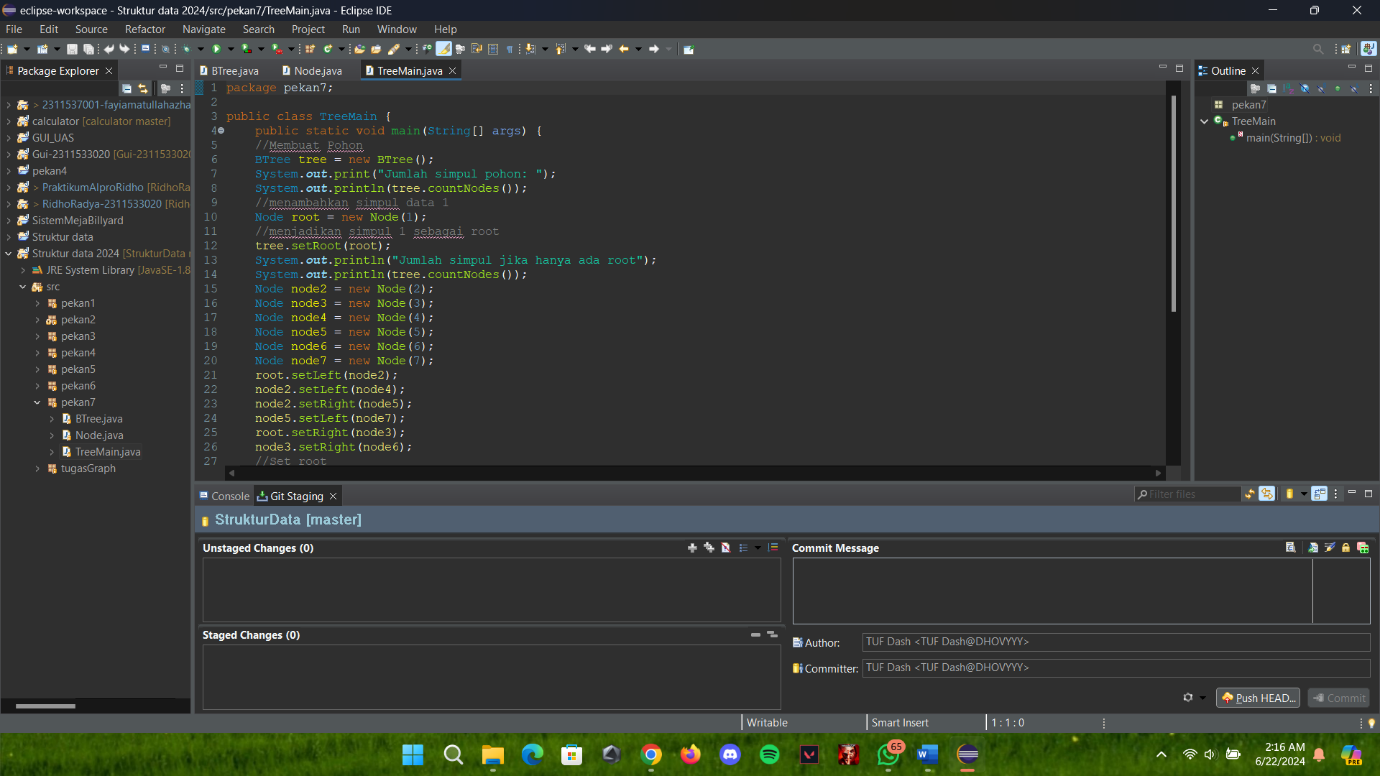
1. Membuat Kelas baru dengan nama “BTree”

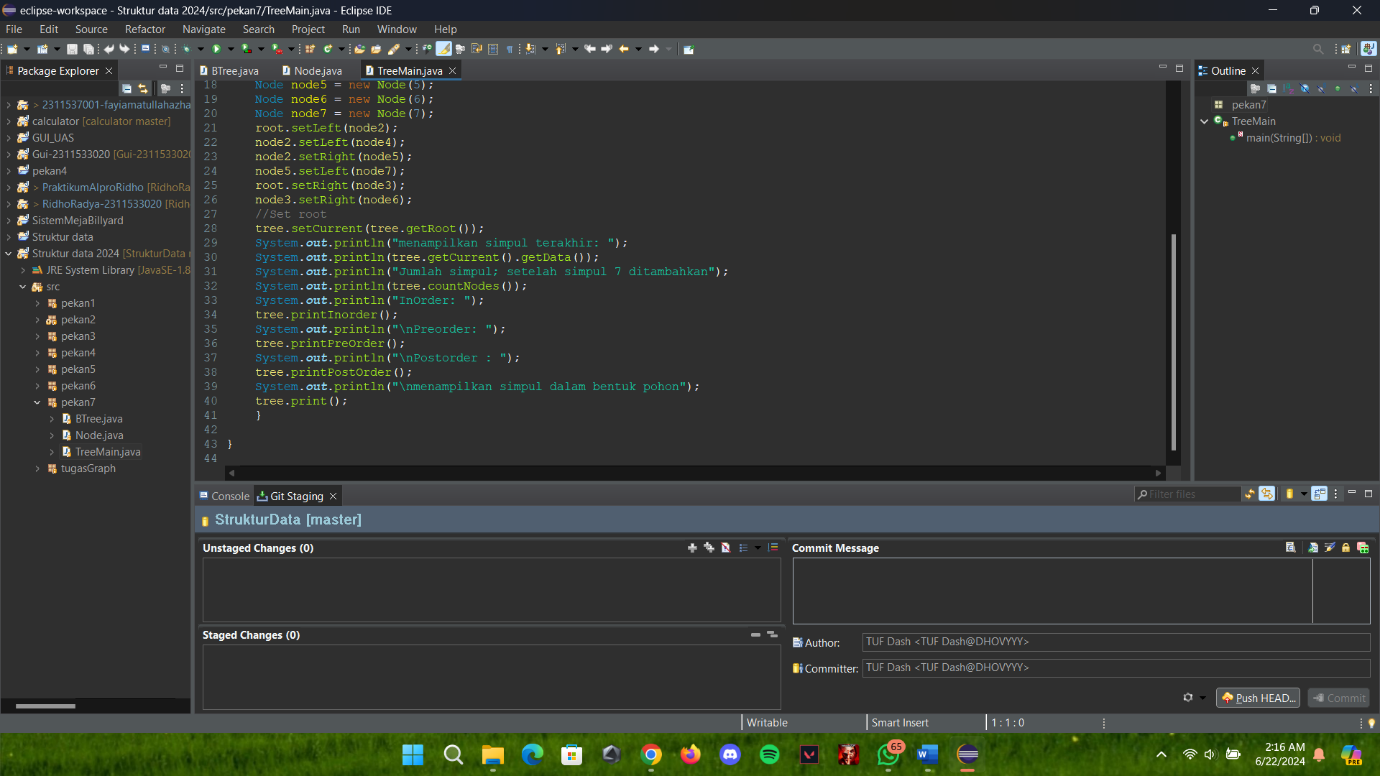
Pada kelas “BTree”, masukkan kode seperti gambar di bawah



Kelas BTree merepresentasikan sebuah pohon biner dan menyediakan berbagai metode untuk mengelola serta berinteraksi dengan struktur pohon ini. Input untuk program ini meliputi operasi-operasi seperti pencarian data (dengan metode search), penambahan node ke dalam pohon, dan pengaturan node akar (setRoot). Metode search memeriksa keberadaan data tertentu dalam pohon dengan mengiterasi node secara rekursif mulai dari akar. Output dari program ini termasuk hasil dari traversal pohon dalam urutan inorder, preorder, dan postorder, yang masing-masing ditangani oleh metode printInorder, printPreorder, dan printPostorder. Selain itu, program juga dapat mencetak jumlah total node dalam pohon menggunakan metode countNodes, dan menampilkan struktur visual pohon melalui metode print. Metode tambahan seperti getRoot, isEmpty, getCurrent, dan setCurrent menyediakan fungsionalitas untuk mengakses dan memodifikasi atribut-atribut penting dari pohon biner.

1. Pada kelas “TreeMain”, masukkan kode seperti gambar di bawah





Program TreeMain menerima input berupa operasi-operasi untuk membangun dan memanipulasi sebuah pohon biner menggunakan kelas BTree dan Node. Program ini memulai dengan membuat objek BTree dan mencetak jumlah simpul awal yang adalah nol. Node baru dengan data dari 1 hingga 7 kemudian dibuat dan diatur ke dalam pohon, dengan node 1 sebagai akar, dan diikuti dengan menambahkan node-node lainnya sebagai anak-anak kiri dan kanan dari node yang sesuai. Setelah setiap penambahan, program mencetak jumlah total simpul dalam pohon. Untuk output, program menampilkan data node saat ini, jumlah total simpul setelah semua node ditambahkan, serta hasil dari traversal inorder, preorder, dan postorder. Selain itu, program juga mencetak struktur visual dari pohon biner, menampilkan hubungan hierarkis antara node-node yang ada.

**D. PENUTUP**

Melalui praktikum ini, saya berhasil memahami konsep dasar dan implementasi struktur data pohon biner menggunakan kelas Node, BTree, dan TreeMain. Saya mempelajari cara membangun pohon biner, menambahkan node, serta melakukan traversal dalam berbagai urutan (inorder, preorder, postorder). Praktikum ini juga menyoroti pentingnya pohon BTree dalam pengelolaan data yang efisien pada sistem penyimpanan. Dengan melakukan berbagai operasi pada pohon, saya dapat melihat bagaimana struktur data ini mempermudah proses pencarian, penambahan, dan penghapusan data. Selain itu, implementasi visualisasi struktur pohon memberikan pemahaman yang lebih mendalam tentang hubungan hierarkis antar node. Kesimpulannya, praktikum ini memperkuat pemahaman saya tentang struktur data pohon dan aplikasinya dalam ilmu komputer, serta mengasah keterampilan saya dalam pemrograman dan manajemen data secara efisien.