**Laporan Praktikum Algoritma dan Struktur Data**

**Jobsheet 12 - Double Linked Lists**

**Dosen Pengampu : Triana Fatmawati, S.T., M.T.**



**Nama : Annisa**

**Nim : 2341760032**

**Kelas : SIB 1E**

**Prodi : D-IV Sistem Informasi Bisnis**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2023/2024**

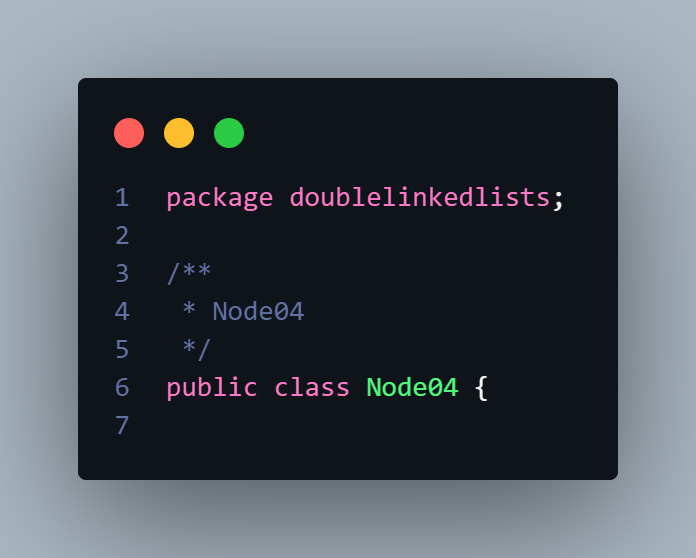
# Tujuan Praktikum

# Kegiatan Praktikum 1

# Percobaan 1

# Perhatikan diagram class Node dan class DoublelinkedLists di bawah ini! Diagram class ini yang selanjutnya akan dibuat sebagai acuan dalam membuat kode program DoubleLinkedLists.

* + 1. Buat paket baru dengan nama **doublelinkedlists**
    2. Buat class di dalam paket tersebut dengan nama **Node**

****

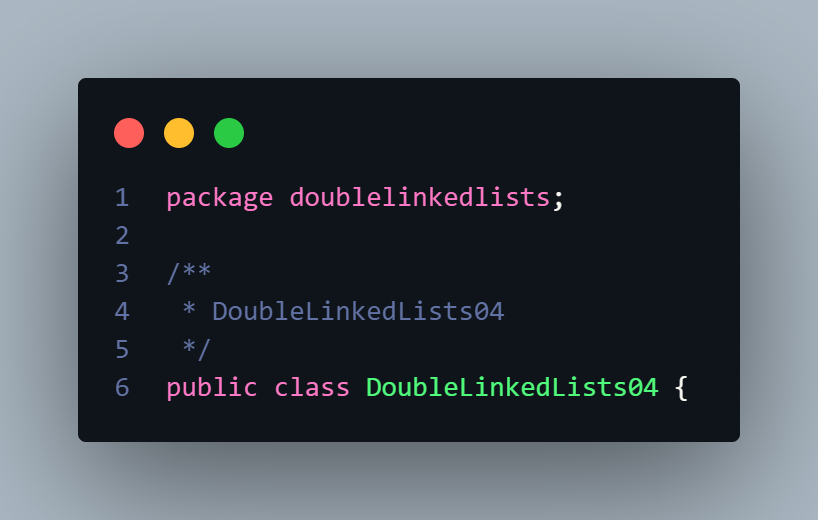
* + 1. Di dalam class tersebut, deklarasikan atribut sesuai dengan diagram class di atas.

****

* + 1. Selanjutnya tambahkan konstruktor default pada class Node sesuai diagram di atas.



* + 1. Buatlah sebuah class baru bernama DoubleLinkedLists pada package yang sama dengan node seperti gambar berikut:



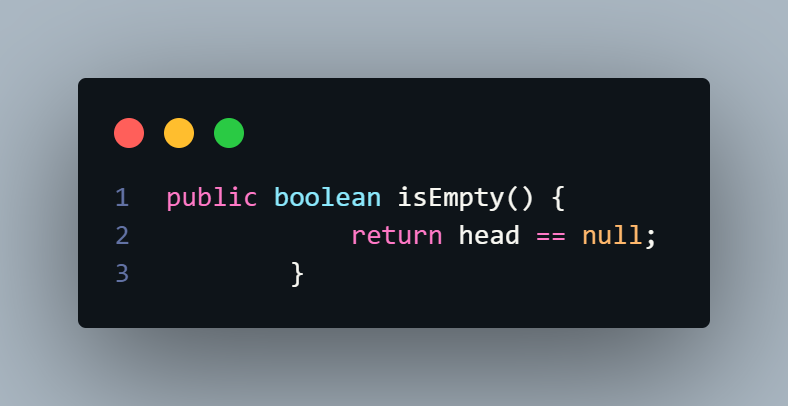
* + 1. Pada class DoubleLinkedLists tersebut, deklarasikan atribut sesuai dengan diagram class di atas.



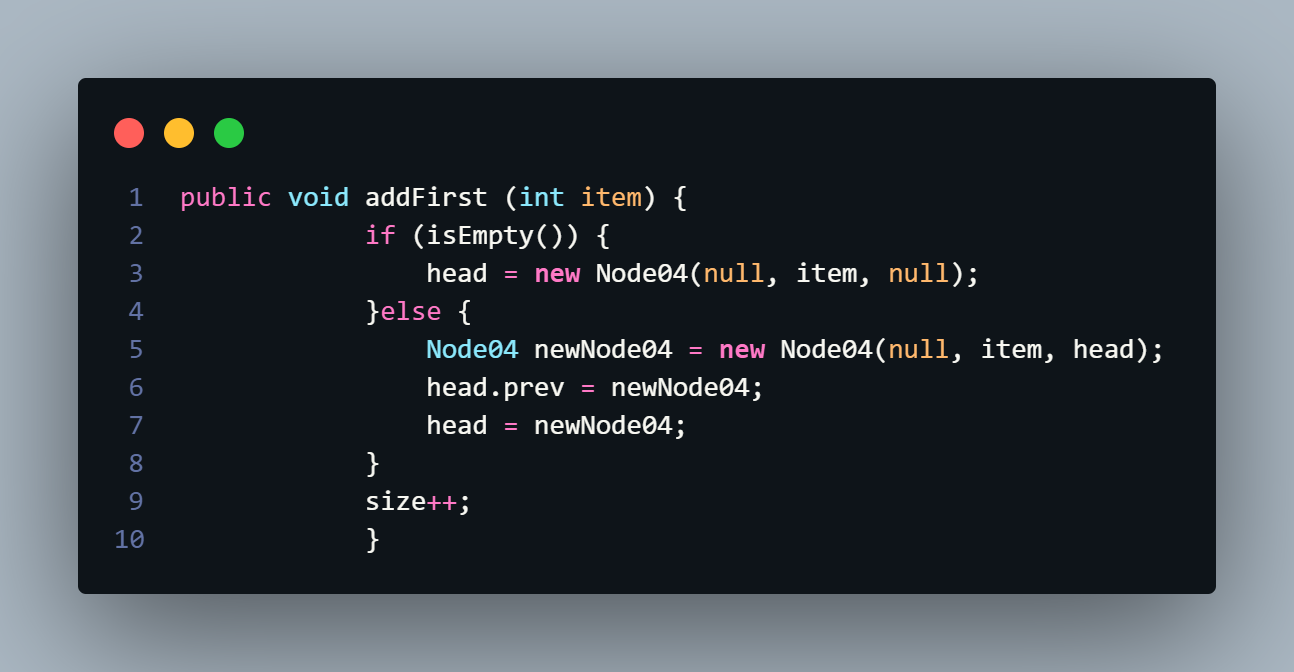
* + 1. Selajuntnya, buat konstruktor pada class DoubleLinkedLists sesuai gambar berikut.



* + 1. Buat method **isEmpty()**. Method ini digunakan untuk memastikan kondisi linked list kosong.



* + 1. Kemudian, buat method **addFirst().** Method ini akan menjalankan penambahan data di bagian depan linked list.



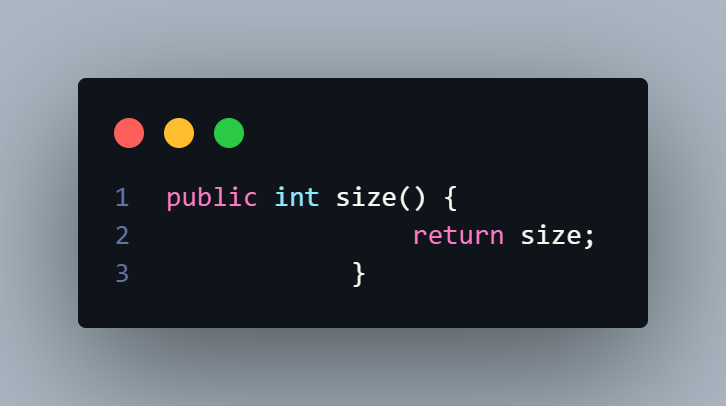
* + 1. Selain itu pembuatan method **addLast()** akan menambahkan data pada bagian belakang linked list.



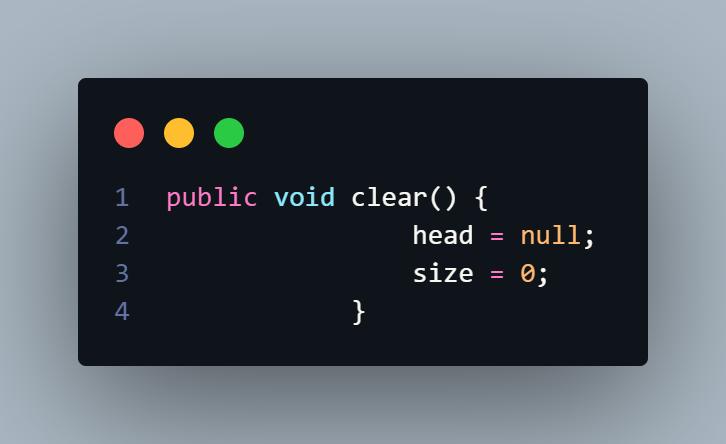
* + 1. Untuk menambakan data pada posisi yang telah ditentukan dengan indeks dapat dibuat dengan method **add(int item, int index)**



* + 1. Jumlah data yang ada di dalam linked lists akan diperbarui secara otomatis sehingga dapat dibuat method **size()** untuk mendapatkan nilai dari size.



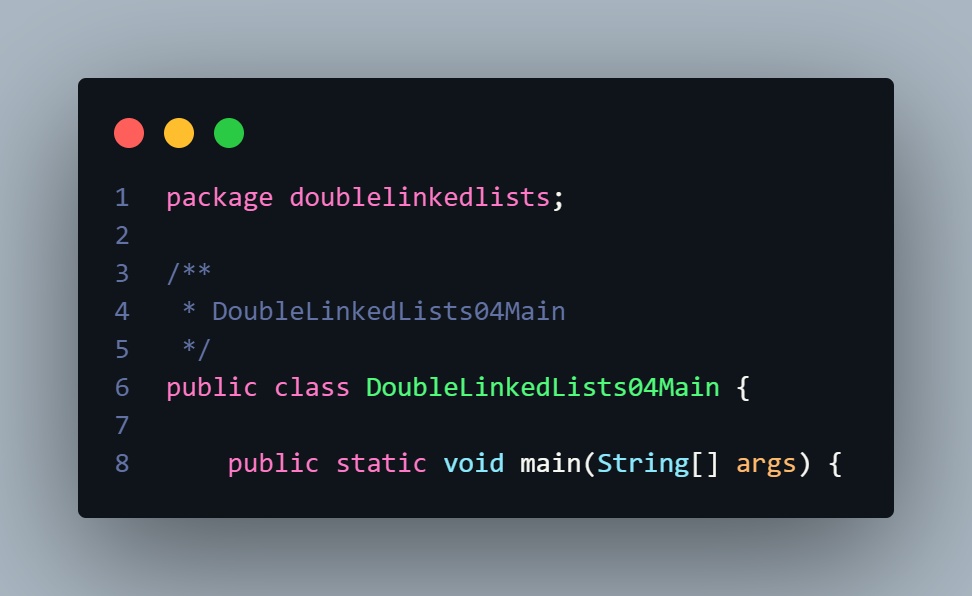
* + 1. Selanjutnya dibuat method **clear()** untuk menghapus semua isi linked lists, sehingga linked lists dalam kondisi kosong.



* + 1. Untuk mencetak isi dari linked lists dibuat method **print().** Method ini akan mencetak isi linked lists berapapun size-nya. Jika kosong akan dimunculkan suatu pemberitahuan bahwa linked lists dalam kondisi kosong.



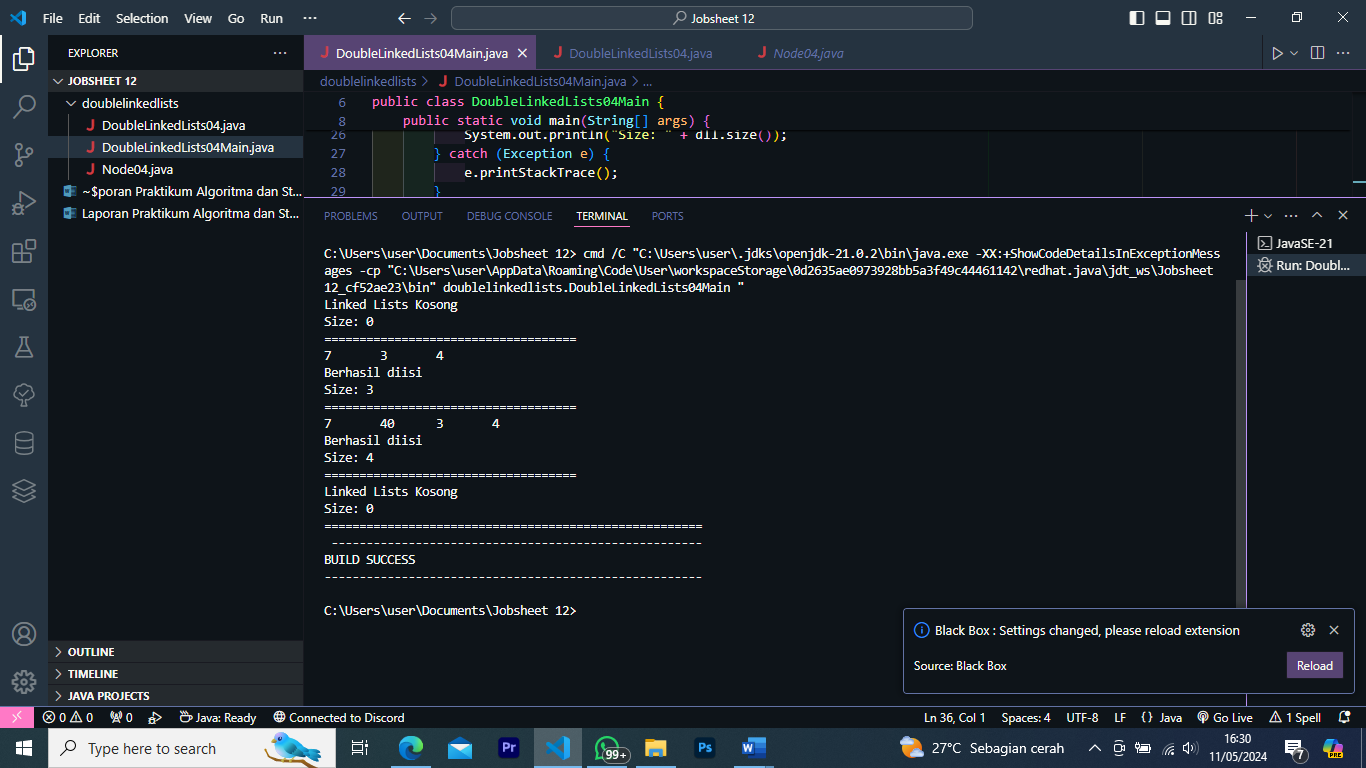
* + 1. Selanjutya dibuat class Main DoubleLinkedListsMain untuk mengeksekusi semua method yang ada pada class DoubleLinkedLists.



* + 1. Pada main class pada langkah 16 di atas buatlah object dari class DoubleLinkedLists kemudian eksekusi potongan program berikut ini



## **Verifikasi Hasil Percobaan**



## **Pertanyaan Percobaan**

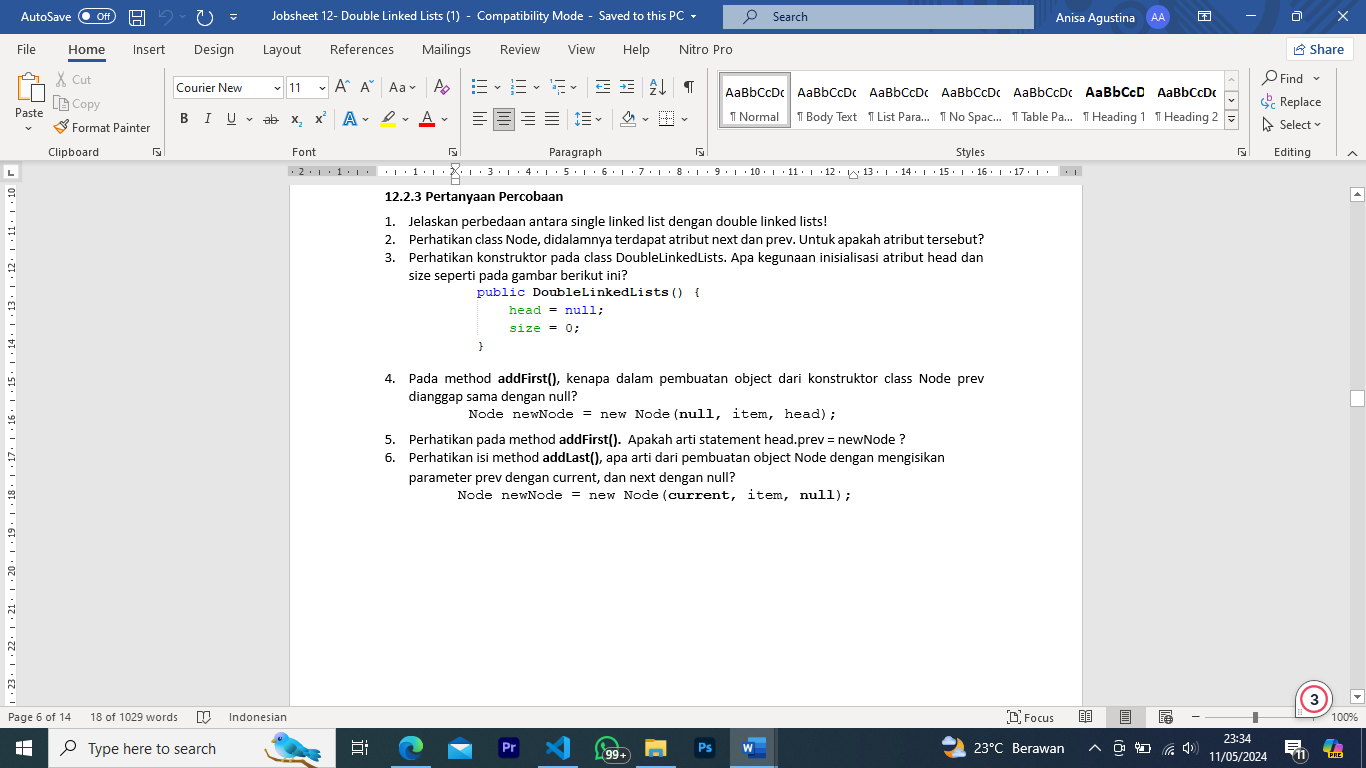
1. **Jelaskan perbedaan antara single linked list dengan double linked lists!**

Single linked list hanya memiliki satu pointer (next) per node yang menunjuk ke node berikutnya, sehingga hanya dapat diakses secara berurutan dari awal. Sementara double linked list memiliki dua pointer (next dan prev) per node, yang memungkinkan akses data dari dua arah (maju dan mundur), namun membutuhkan lebih banyak memori.

1. **Perhatikan class Node, didalamnya terdapat atribut next dan prev. Untuk apakah atribut tersebut?**

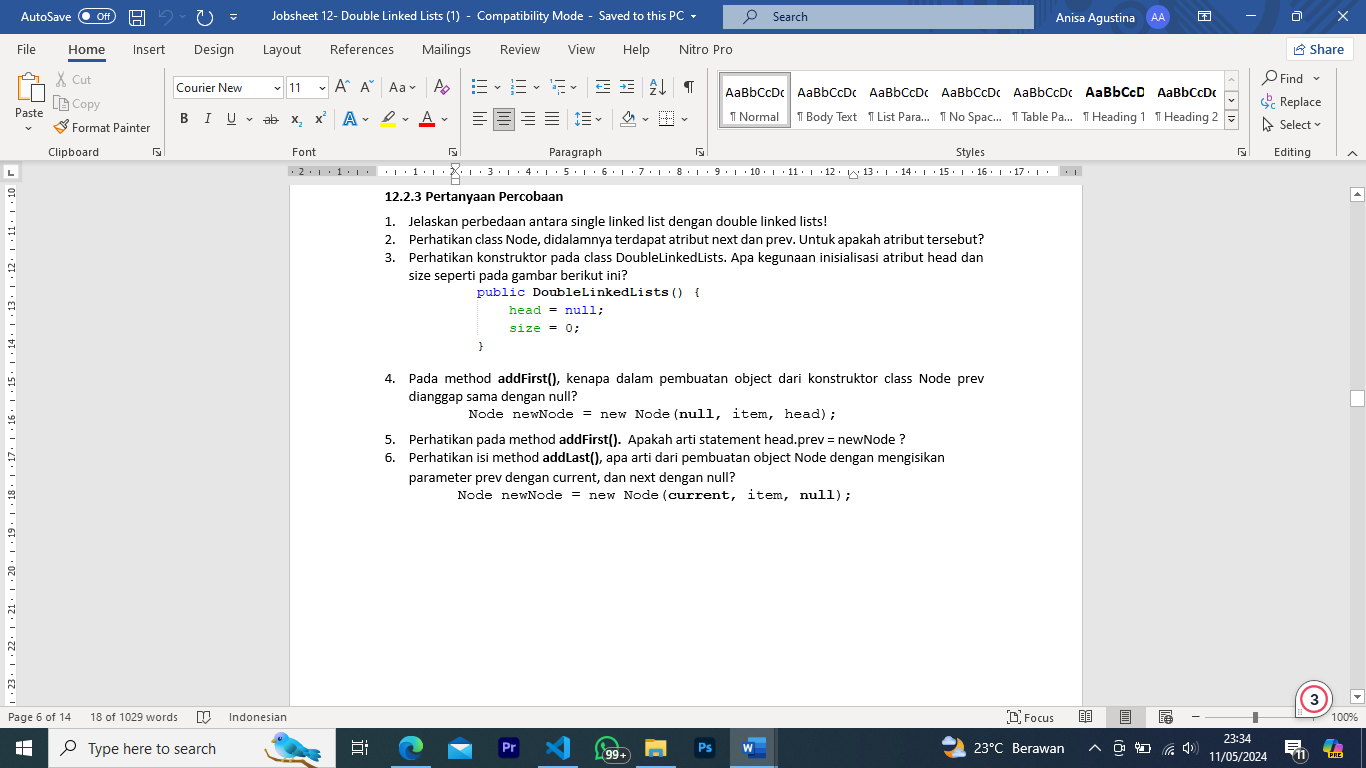
Dalam class Node04 atribut prev digunakan untuk menyimpan referensi atau pointer ke node sebelumnya, sementara atribut next digunakan untuk menyimpan referensi atau pointer ke node berikutnya. Dengan adanya dua atribut ini, setiap node dalam double linked list terhubung dengan node sebelum dan sesudahnya, memungkinkan traversal atau akses data dari dua arah

1. **Perhatikan konstruktor pada class DoubleLinkedLists. Apa kegunaan inisialisasi atribut head dan size seperti pada gambar berikut ini?**



Dalam konstruktor DoubleLinkedLists04(), inisialisasi atribut head dengan null dan size dengan 0 dilakukan untuk menyiapkan kondisi awal double linked list yang kosong. Head diinisialisasi null berarti belum ada node atau elemen di dalamnya, sedangkan size 0 mengonfirmasi bahwa jumlah node-nya adalah kosong.

1. **Pada method addFirst(), kenapa dalam pembuatan object dari konstruktor class Node prev dianggap sama dengan null?**

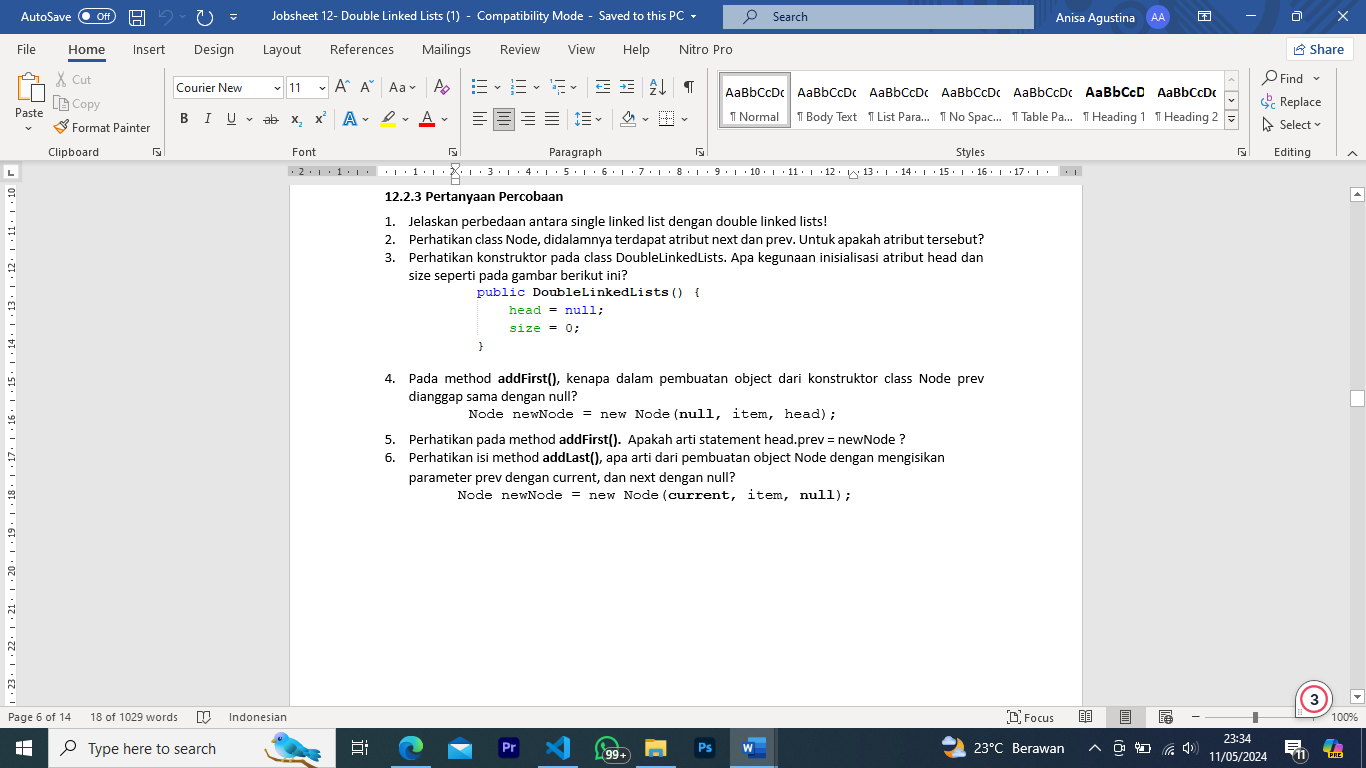


Pada pembuatan objek node baru newNode04 dalam method addFirst() **prev** diinisialisasi dengan nullkarena node baru tersebut akan menjadi node pertama (head) dalam double linked list, sehingga tidak memiliki node sebelumnya. Sedangkan **next** diinisialisasi dengan `head` untuk menghubungkan node baru dengan node pertama sebelumnya (jika ada) sehingga node baru menjadi head baru dalam struktur double linked list.

1. **Perhatikan pada method addFirst(). Apakah arti statement head.prev = newNode ?**

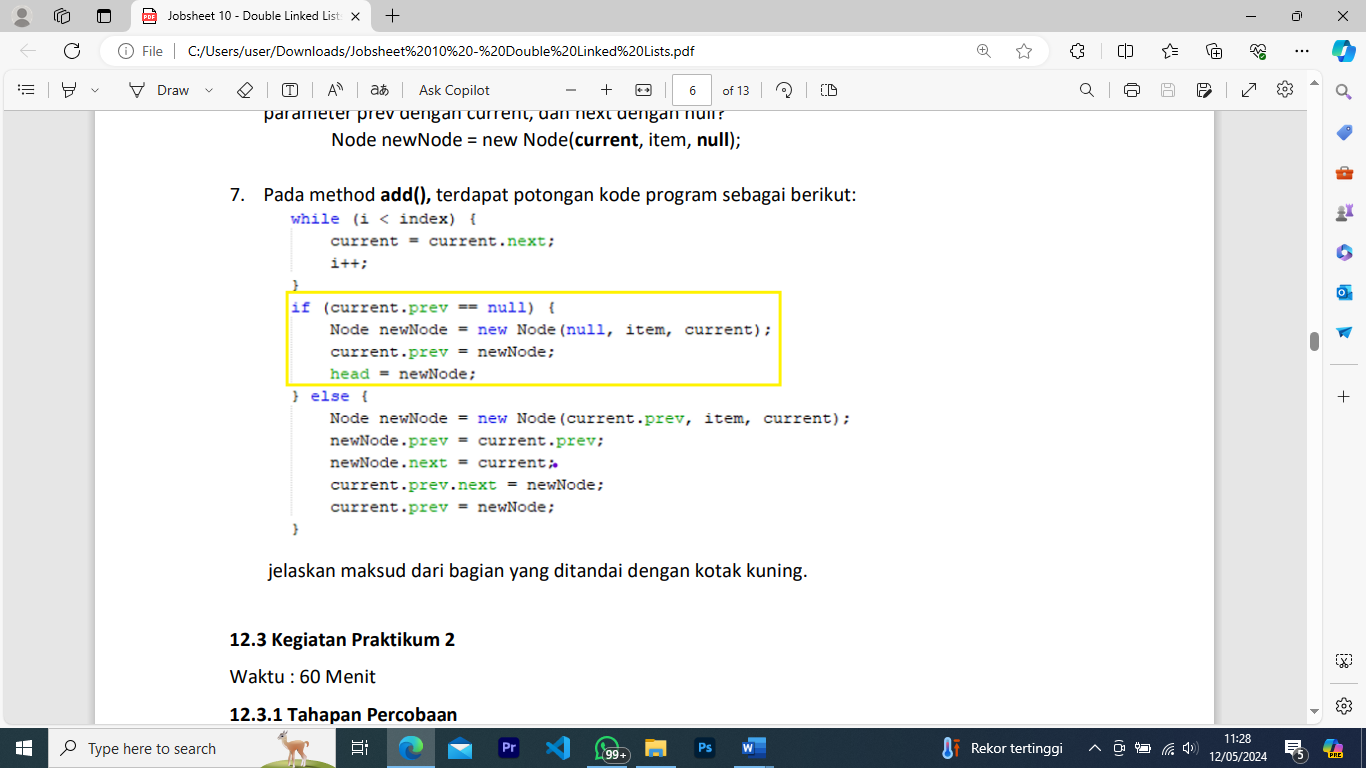
Statement head.prev = newNode04; dalam method addFirst() digunakan untuk memperbarui pointer prev dari node yang sebelumnya menjadi head (node pertama) dalam double linked list agar menunjuk ke node baru (newNode04) yang menjadi head baru.

1. **Perhatikan isi method addLast(), apa arti dari pembuatan object Node dengan mengisikan parameter prev dengan current, dan next dengan null?**

****

Pada pembuatan objek node baru newNode04 dalam method addLast(), parameter prev diisi dengan current yang merujuk pada node terakhir yang sudah ada dalam double linked list. Ini untuk menghubungkan node baru tersebut sebagai node berikutnya dari node terakhir sebelumnya melalui pointer prev. Sedangkan parameter next diisi dengan null karena node baru tersebut akan menjadi node terakhir dalam double linked list, sehingga tidak ada node berikutnya lagi.

1. **Pada method add(), terdapat potongan kode program sebagai berikut:**

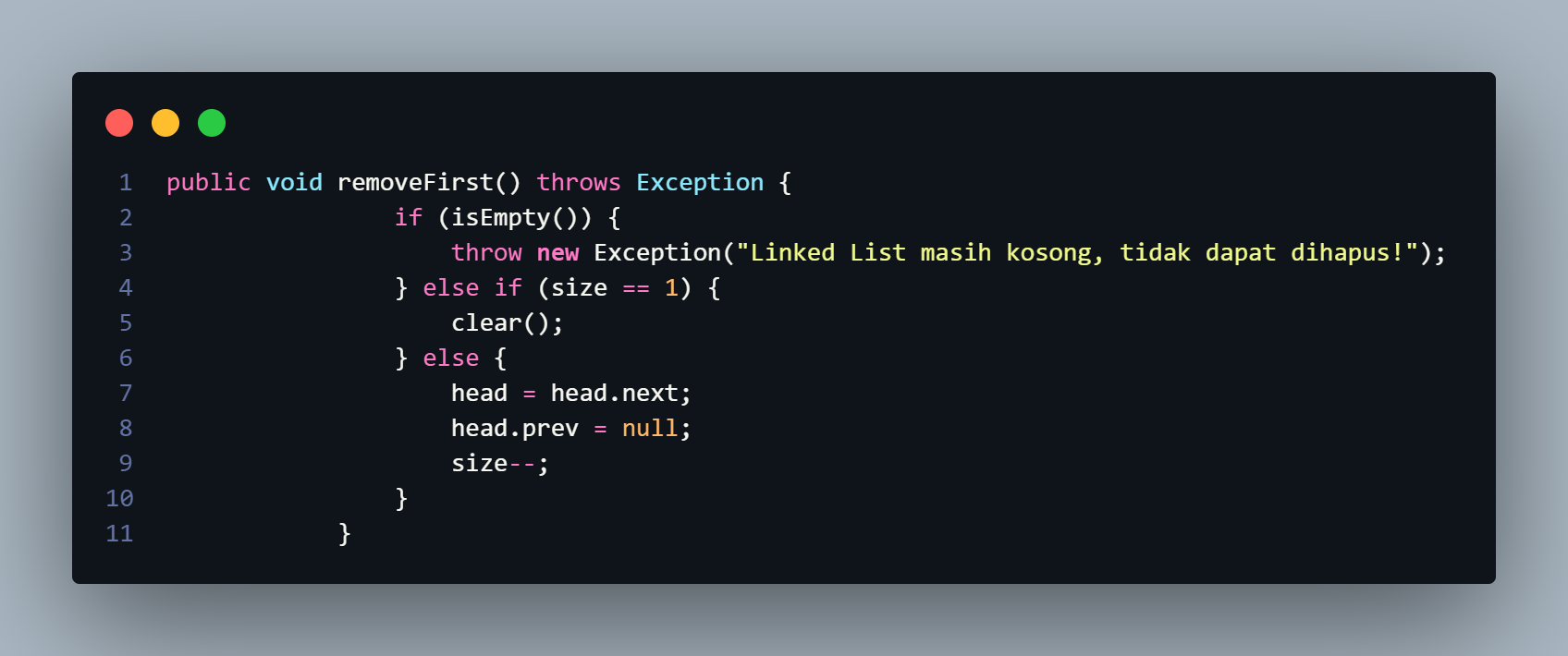
****

**jelaskan maksud dari bagian yang ditandai dengan kotak kuning.**

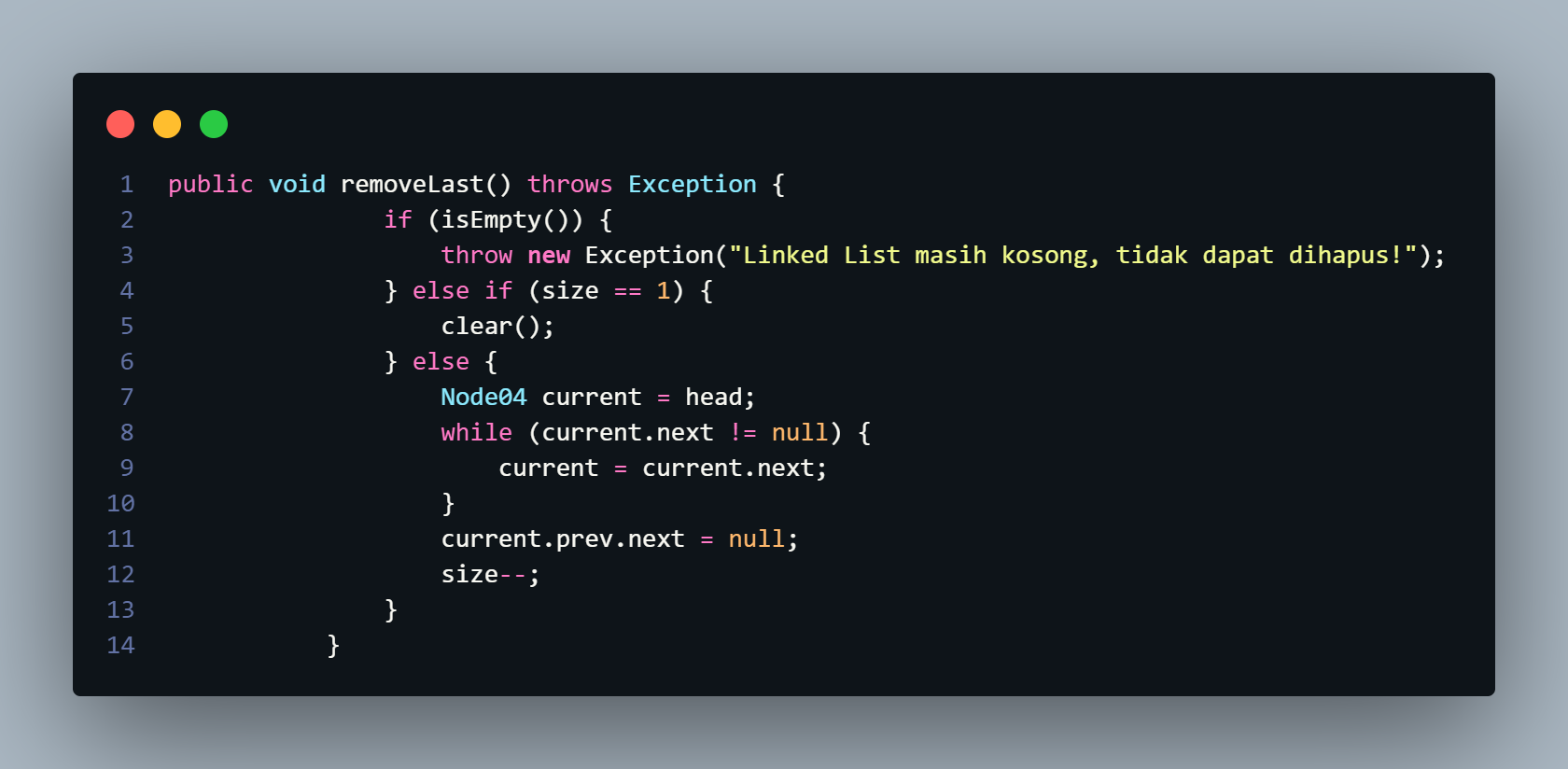
Pertama, kode memeriksa apakah current.prev bernilai null, yang mengindikasikan linked list masih kosong. Jika kondisi tersebut terpenuhi, kode akan membuat instance baru dari objek Node dengan nilai item yang akan ditambahkan, dan menghubungkannya dengan current. Kemudian, current.prev dan head (node pertama dalam linked list) diatur menunjuk ke node baru yang baru saja dibuat. Dengan demikian, node baru tersebut menjadi node pertama dan satu-satunya dalam linked list yang sebelumnya masih kosong.

Praktikum 2

1. Buatlah method removeFirst() di dalam class DoubleLinkedLists.

****

1. Tambahkan method removeLast() di dalam class DoubleLinkedLists.

****

1. Tambahkan pula method remove(int index) pada class DoubleLinkedLists dan amati hasilnya.

****

1. Untuk mengeksekusi method yang baru saja dibuat, tambahkan potongan kode program berikut pada main class.

****

**Verifikasi**

****