

Nama : Muhammad Zakaria Haniya

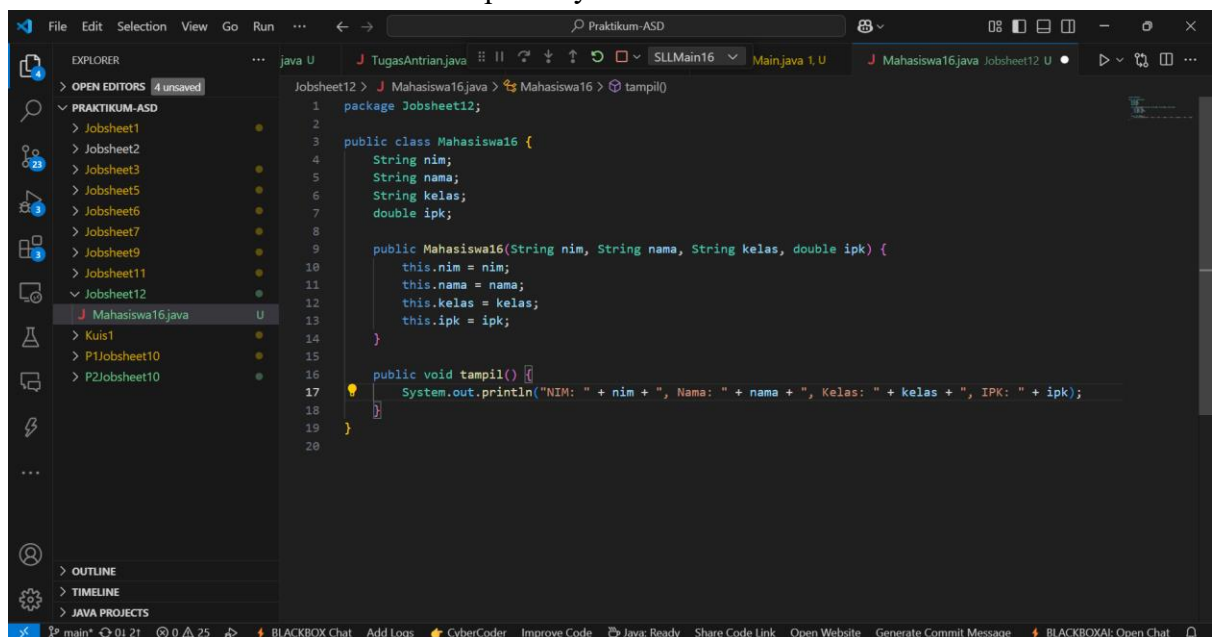
NIM : 244107020135

Kelas : TI-1B

Absen : 16

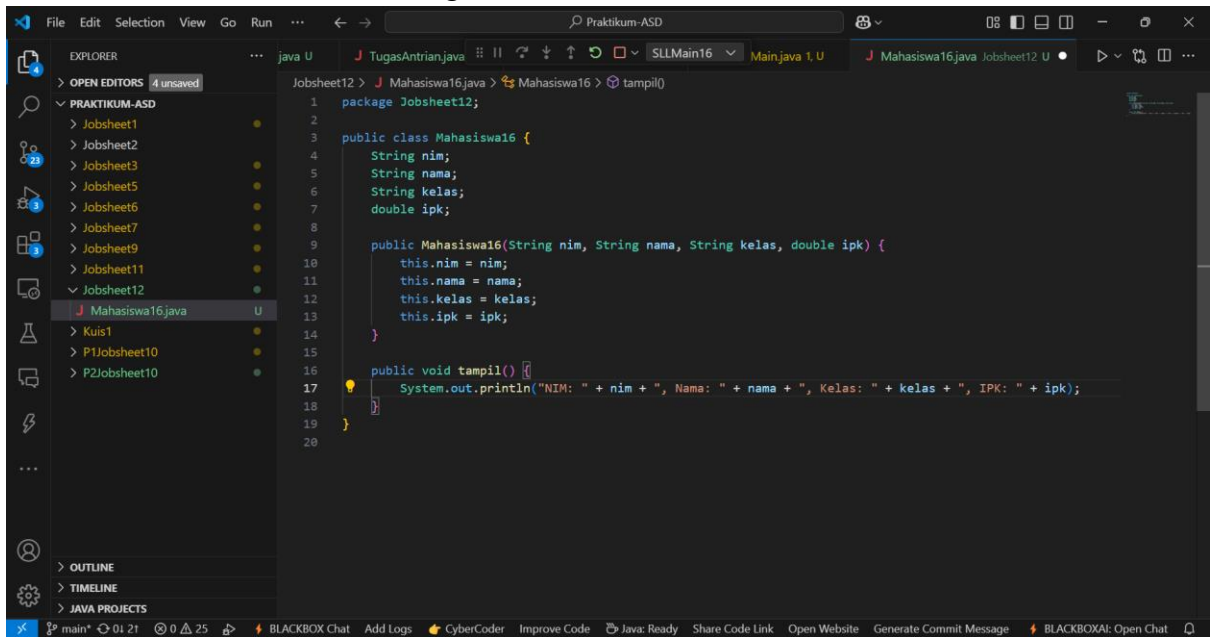
### Kegiatan Praktikum 1

1. Perhatikan diagram class Mahasiswa01, Node01 dan class DoublelinkedLists di bawah ini! Diagram class ini yang selanjutnya akan dibuat sebagai acuan dalam membuat kode program DoubleLinkedLists
2. Pada Project yang sudah dibuat pada Minggu sebelumnya, buat folder atau package baru bernama Jobsheet12 di dalam repository Praktikum ASD.



3. Buat class di dalam paket tersebut dengan nama Mahasiswa01. Di dalam class tersebut, deklarasikan atribut sesuai dengan diagram class di atas. Tambahkan juga

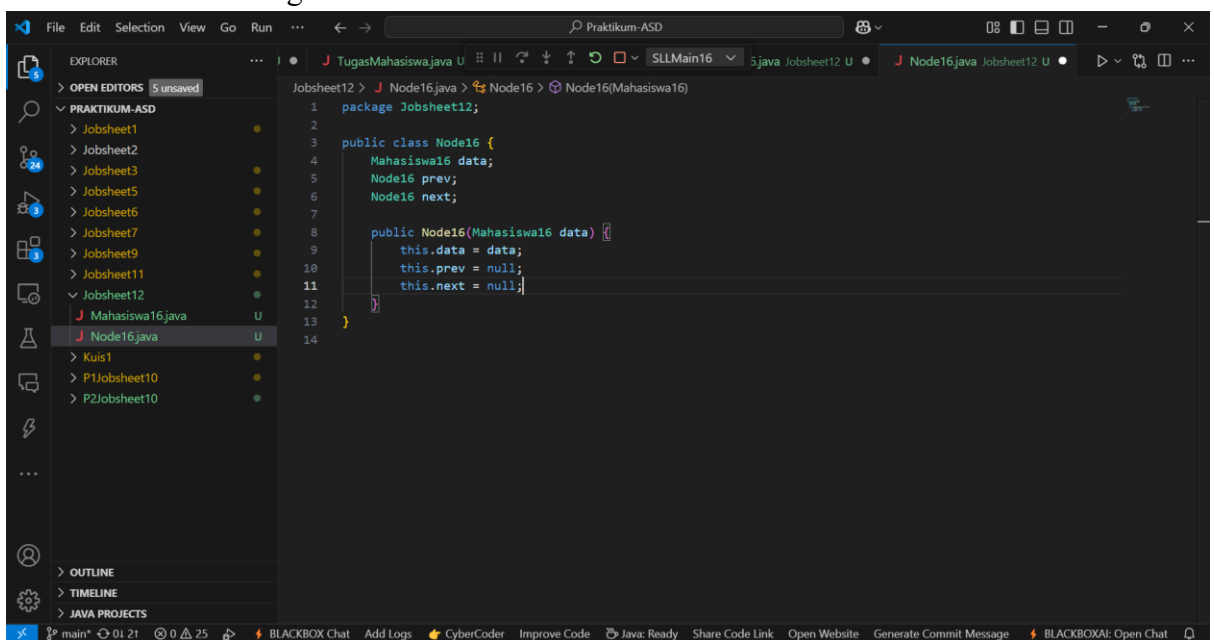
konstruktor dan method sesuai diagram di atas



The screenshot shows an IDE with the Explorer panel on the left displaying a project named 'PRAKTIKUM-ASD'. The file 'Mahasiswa16.java' is selected. The main editor displays the following Java code:

```
1 package Jobsheet12;
2
3 public class Mahasiswa16 {
4     String nim;
5     String nama;
6     String kelas;
7     double ipk;
8
9     public Mahasiswa16(String nim, String nama, String kelas, double ipk) {
10         this.nim = nim;
11         this.nama = nama;
12         this.kelas = kelas;
13         this.ipk = ipk;
14     }
15
16     public void tampil() {
17         System.out.println("NIM: " + nim + ", Nama: " + nama + ", Kelas: " + kelas + ", IPK: " + ipk);
18     }
19 }
20
```

4. Buat class di dalam paket tersebut dengan nama Node01. Di dalam class tersebut, deklarasikan atribut sesuai dengan diagram class di atas. Selanjutnya tambahkan konstruktor sesuai diagram di atas

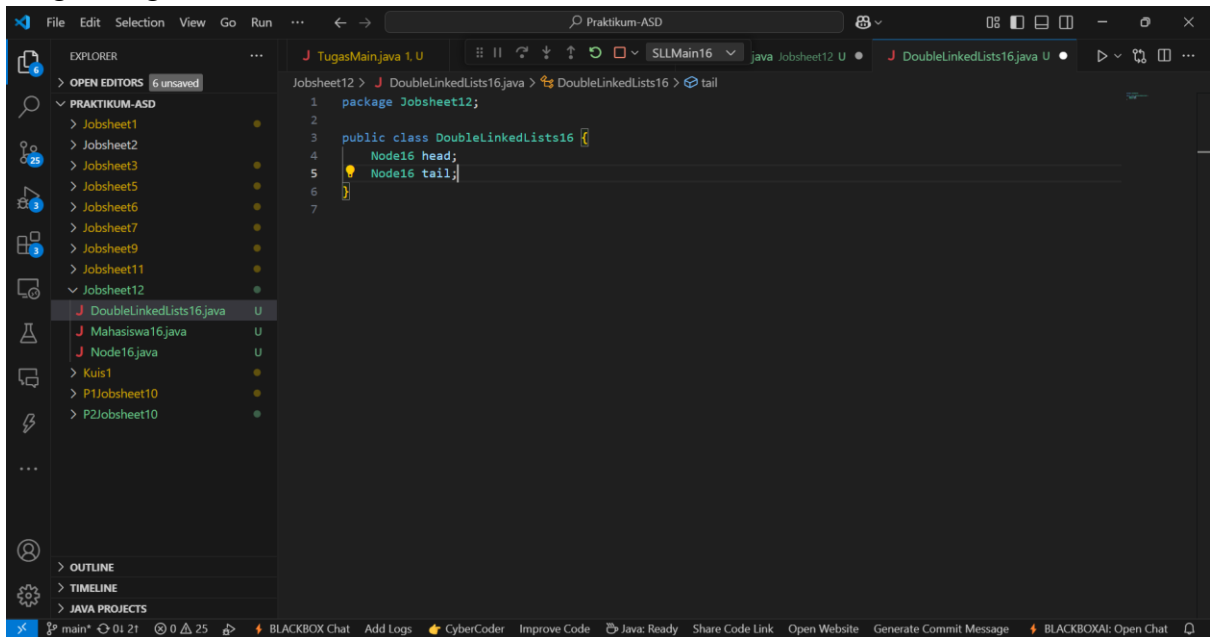


The screenshot shows the same IDE with the file 'Node16.java' selected. The main editor displays the following Java code:

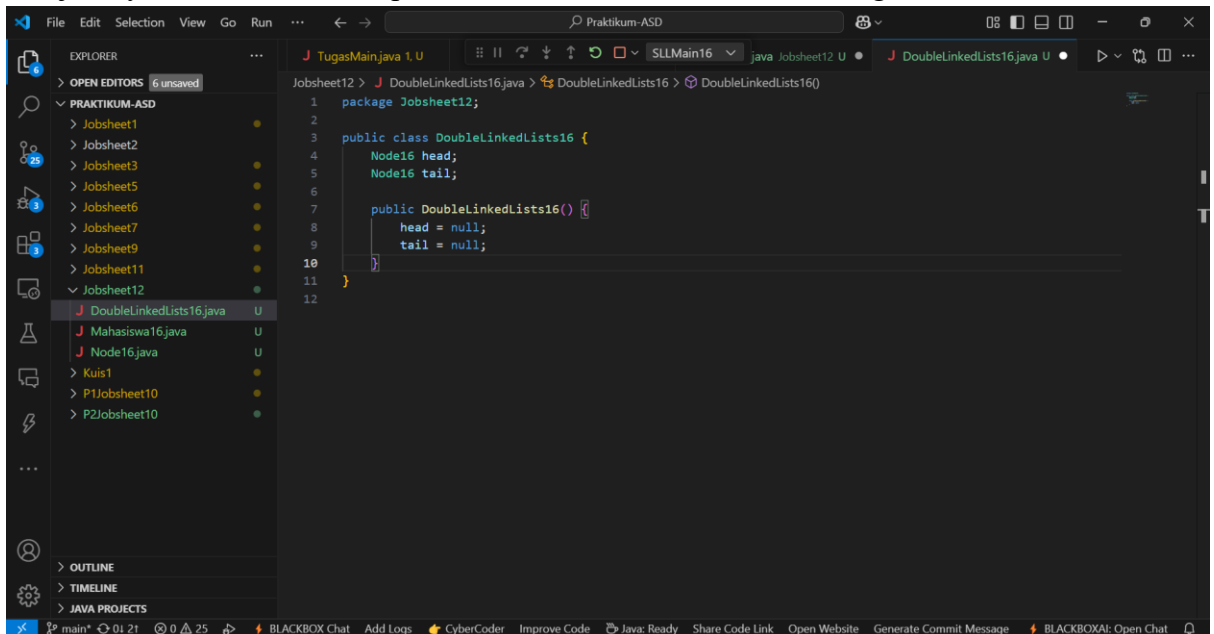
```
1 package Jobsheet12;
2
3 public class Node16 {
4     Mahasiswa16 data;
5     Node16 prev;
6     Node16 next;
7
8     public Node16(Mahasiswa16 data) {
9         this.data = data;
10        this.prev = null;
11        this.next = null;
12    }
13 }
14
```

5. Buatlah sebuah class baru bernama DoubleLinkedLists pada package yang sama dengan Node01. Pada class DoubleLinkedLists tersebut, deklarasikan atribut sesuai

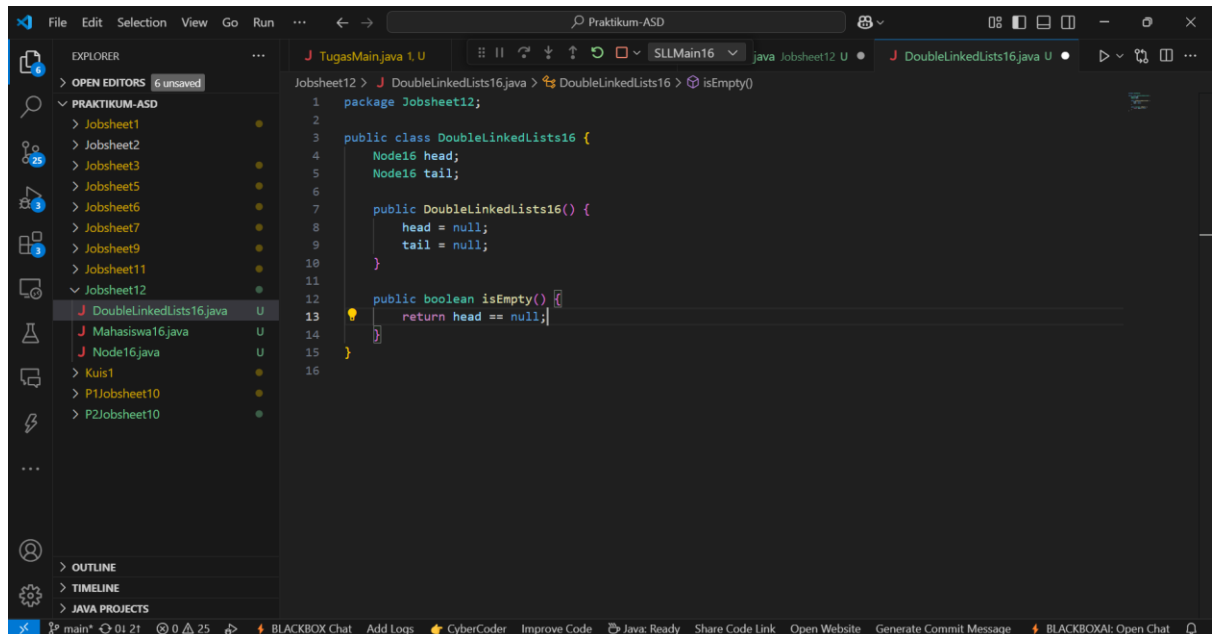
dengan diagram class di atas.



6. Selanjutnya, buat konstruktor pada class DoubleLinkedLists sesuai gambar berikut.

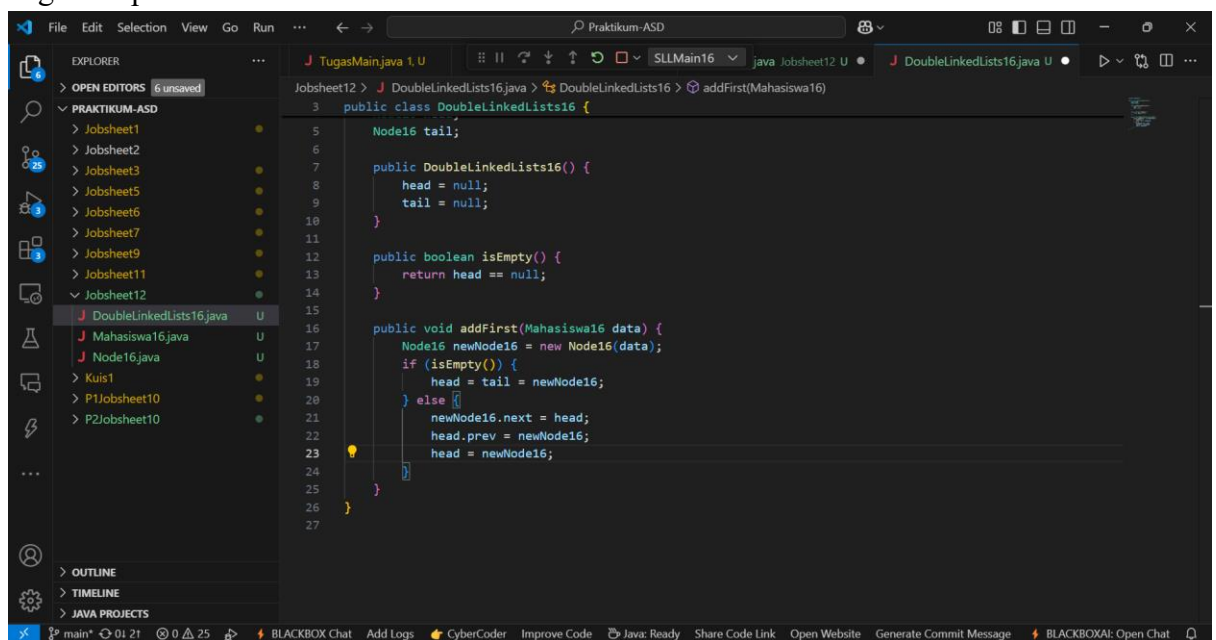


7. Buat method isEmpty(). Method ini digunakan untuk memastikan kondisi linked list kosong.



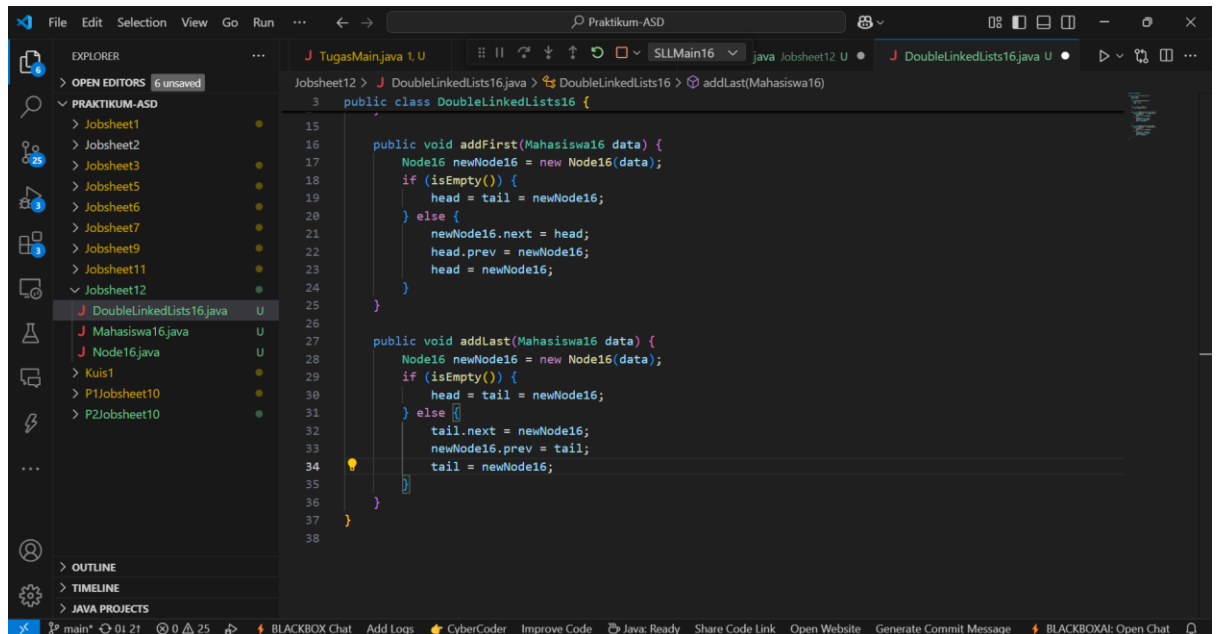
```
1 package Jobsheet12;
2
3 public class DoubleLinkedLists16 {
4     Node16 head;
5     Node16 tail;
6
7     public DoubleLinkedLists16() {
8         head = null;
9         tail = null;
10    }
11
12    public boolean isEmpty() {
13        return head == null;
14    }
15 }
16
```

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



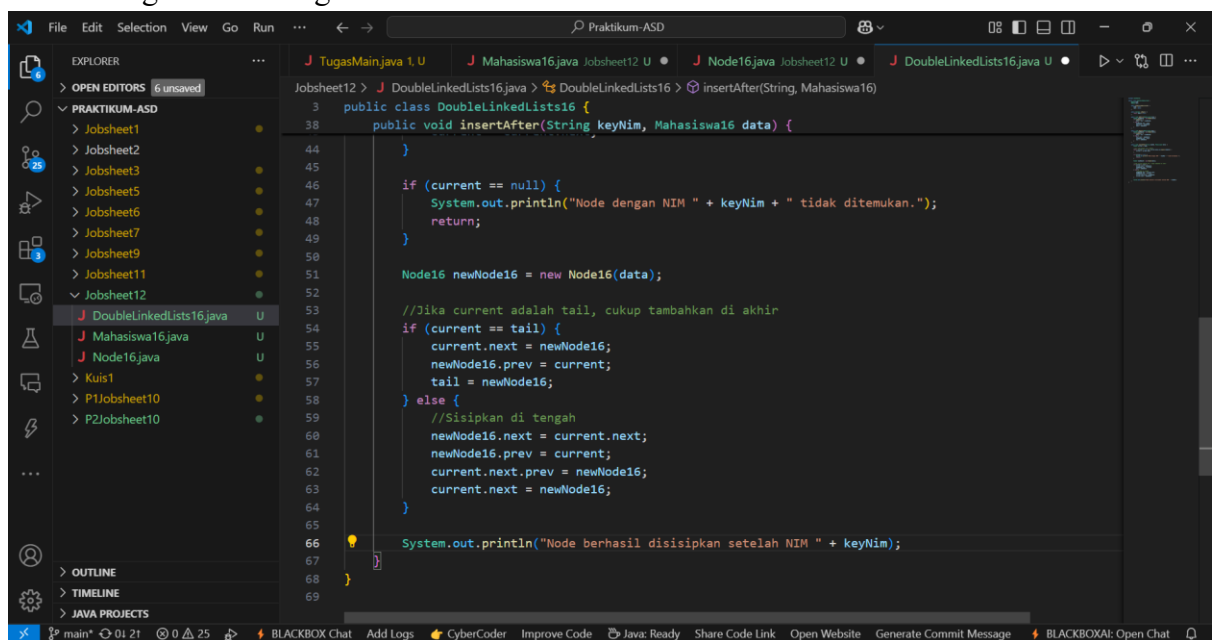
```
3 public class DoubleLinkedLists16 {
4     Node16 tail;
5
6     public DoubleLinkedLists16() {
7         head = null;
8         tail = null;
9     }
10
11    public boolean isEmpty() {
12        return head == null;
13    }
14
15    public void addFirst(Mahasiswa16 data) {
16        Node16 newNode16 = new Node16(data);
17        if (isEmpty()) {
18            head = tail = newNode16;
19        } else {
20            newNode16.next = head;
21            head.prev = newNode16;
22            head = newNode16;
23        }
24    }
25 }
26
27
```

9. Selain itu pembuatan method `addLast()` akan menambahkan data pada bagian belakang linked list.



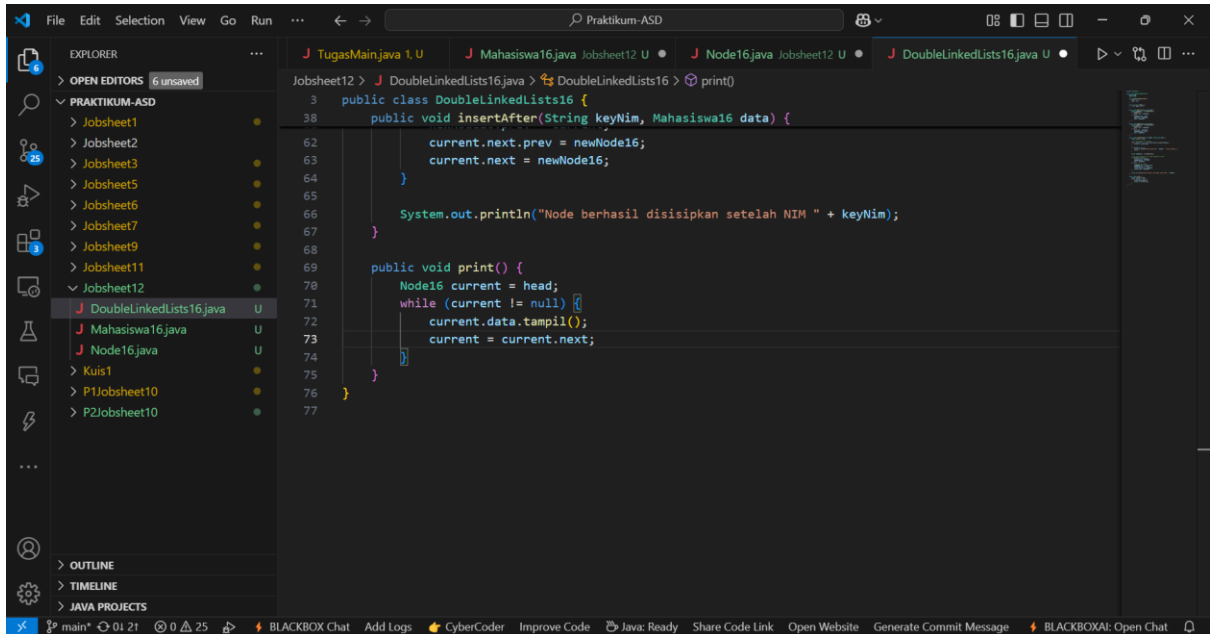
```
public class DoubleLinkedList16 {  
    15  
    16  
    17  
    18  
    19  
    20  
    21  
    22  
    23  
    24  
    25  
    26  
    27  
    28  
    29  
    30  
    31  
    32  
    33  
    34  
    35  
    36  
    37  
    38  
  
    public void addFirst(Mahasiswa16 data) {  
        Node16 newNode16 = new Node16(data);  
        if (isEmpty()) {  
            head = tail = newNode16;  
        } else {  
            newNode16.next = head;  
            head.prev = newNode16;  
            head = newNode16;  
        }  
    }  
  
    public void addLast(Mahasiswa16 data) {  
        Node16 newNode16 = new Node16(data);  
        if (isEmpty()) {  
            head = tail = newNode16;  
        } else {  
            tail.next = newNode16;  
            newNode16.prev = tail;  
            tail = newNode16;  
        }  
    }  
}
```

10. Untuk menambahkan data pada posisi setelah node yang menyimpan data key, dapat dibuat dengan cara sebagai berikut



```
public class DoubleLinkedList16 {  
    3  
    4  
    5  
    6  
    7  
    8  
    9  
    10  
    11  
    12  
    13  
    14  
    15  
    16  
    17  
    18  
    19  
    20  
    21  
    22  
    23  
    24  
    25  
    26  
    27  
    28  
    29  
    30  
    31  
    32  
    33  
    34  
    35  
    36  
    37  
    38  
    39  
    40  
    41  
    42  
    43  
    44  
    45  
    46  
    47  
    48  
    49  
    50  
    51  
    52  
    53  
    54  
    55  
    56  
    57  
    58  
    59  
    60  
    61  
    62  
    63  
    64  
    65  
    66  
    67  
    68  
    69  
  
    public void insertAfter(String keyNim, Mahasiswa16 data) {  
        if (current == null) {  
            System.out.println("Node dengan NIM " + keyNim + " tidak ditemukan.");  
            return;  
        }  
  
        Node16 newNode16 = new Node16(data);  
  
        //Jika current adalah tail, cukup tambahkan di akhir  
        if (current == tail) {  
            current.next = newNode16;  
            newNode16.prev = current;  
            tail = newNode16;  
        } else {  
            //Sisipkan di tengah  
            newNode16.next = current.next;  
            newNode16.prev = current;  
            current.next.prev = newNode16;  
            current.next = newNode16;  
        }  
  
        System.out.println("Node berhasil disisipkan setelah NIM " + keyNim);  
    }  
}
```

11. Untuk mencetak isi dari linked lists dibuat method print(). Method ini akan mencetak isi linked lists berapapun size-nya.

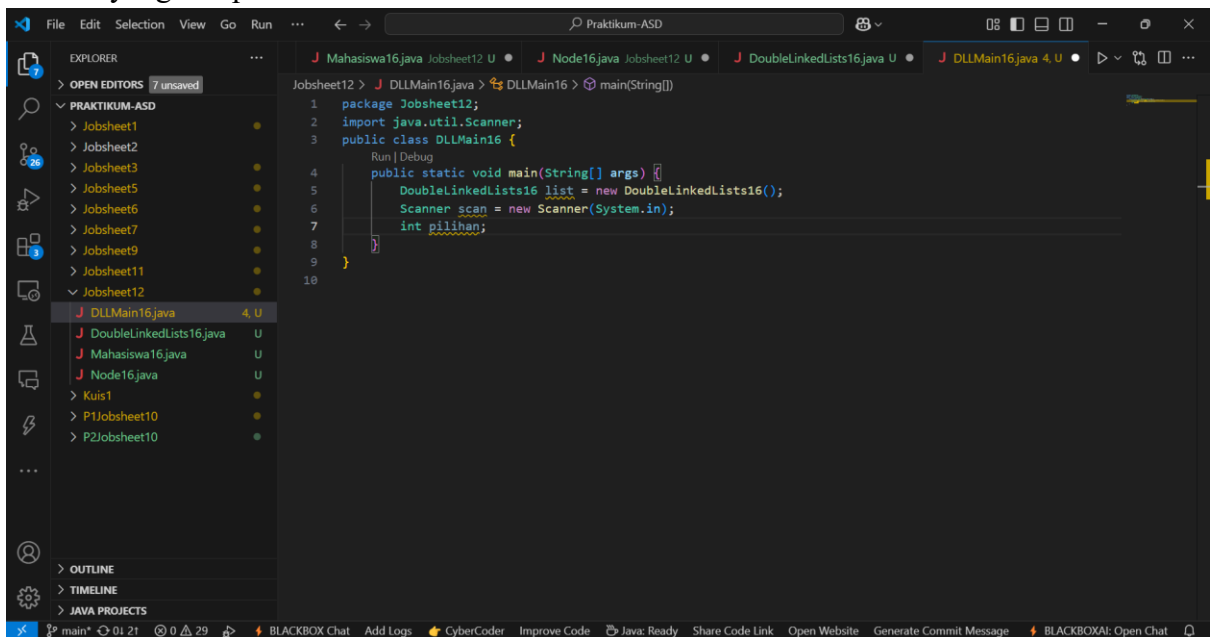


```
File Edit Selection View Go Run ... Praktikum-ASD
EXPLORER
  OPEN EDITORS 6 unsaved
  PRAKTIKUM-ASD
    Jobsheet1
    Jobsheet2
    Jobsheet3
    Jobsheet5
    Jobsheet6
    Jobsheet7
    Jobsheet9
    Jobsheet11
    Jobsheet12
    DoubleLinkedLists16.java U
    Mahasiswa16.java U
    Node16.java U
    Kuis1
    P1Jobsheet10
    P2Jobsheet10
  OUTLINE
  TIMELINE
  JAVA PROJECTS
  main* 01:21 0 0 25 BLACKBOX Chat Add Logs CyberCoder Improve Code Java: Ready Share Code Link Open Website Generate Commit Message BLACKBOXAI: Open Chat

Jobsheet12 > J DoubleLinkedLists16.java > DoubleLinkedLists16 > print()
3 public class DoubleLinkedLists16 {
38 public void insertAfter(String keyNim, Mahasiswa16 data) {
    current.next.prev = newNode16;
    current.next = newNode16;
}
    System.out.println("Node berhasil disisipkan setelah NIM " + keyNim);
}

    public void print() {
        Node16 current = head;
        while (current != null) {
            current.data.tampil();
            current = current.next;
        }
    }
}
```

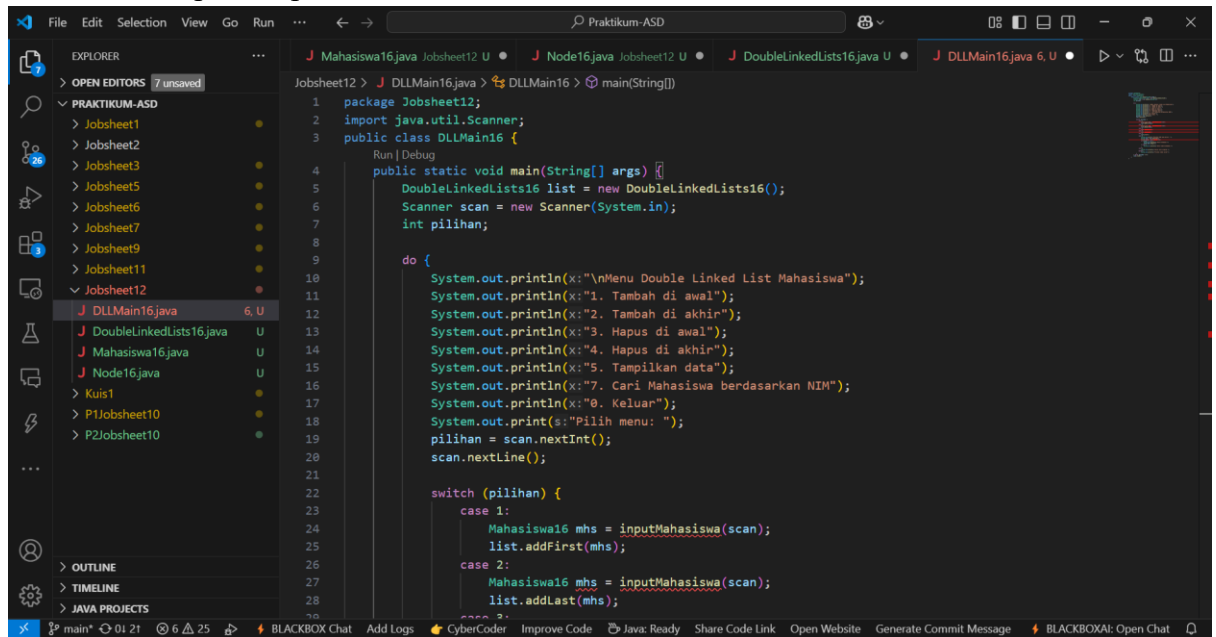
12. Selanjutnya dibuat class Main DoubleLinkedListsMain untuk mengeksekusi semua method yang ada pada class DoubleLinkedLists.



```
File Edit Selection View Go Run ... Praktikum-ASD
EXPLORER
  OPEN EDITORS 7 unsaved
  PRAKTIKUM-ASD
    Jobsheet1
    Jobsheet2
    Jobsheet3
    Jobsheet5
    Jobsheet6
    Jobsheet7
    Jobsheet9
    Jobsheet11
    Jobsheet12
    DLLMain16.java 4 U
    DoubleLinkedLists16.java U
    Mahasiswa16.java U
    Node16.java U
    Kuis1
    P1Jobsheet10
    P2Jobsheet10
  OUTLINE
  TIMELINE
  JAVA PROJECTS
  main* 01:21 0 0 29 BLACKBOX Chat Add Logs CyberCoder Improve Code Java: Ready Share Code Link Open Website Generate Commit Message BLACKBOXAI: Open Chat

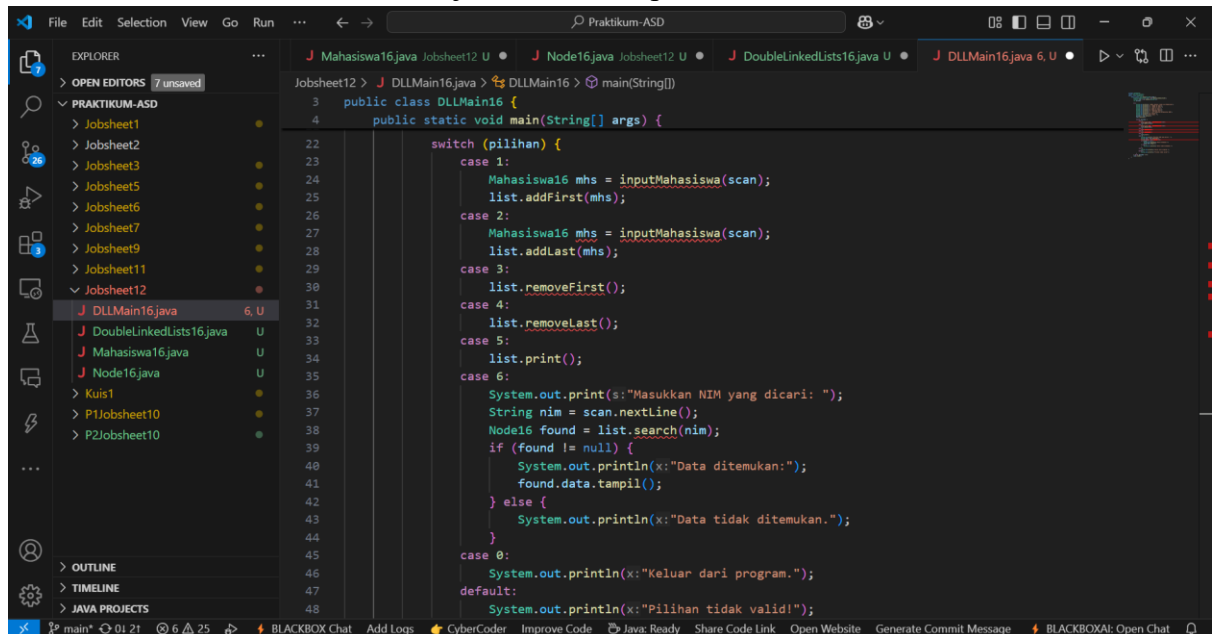
Jobsheet12 > J DLLMain16.java > DLLMain16 > main(String[])
1 package Jobsheet12;
2 import java.util.Scanner;
3 public class DLLMain16 {
    Run | Debug
4 public static void main(String[] args) {
5     DoubleLinkedLists16 list = new DoubleLinkedLists16();
6     Scanner scan = new Scanner(System.in);
7     int pilihan;
8 }
9
10 }
```

### 13. Buatlah menu pilihan pada class main



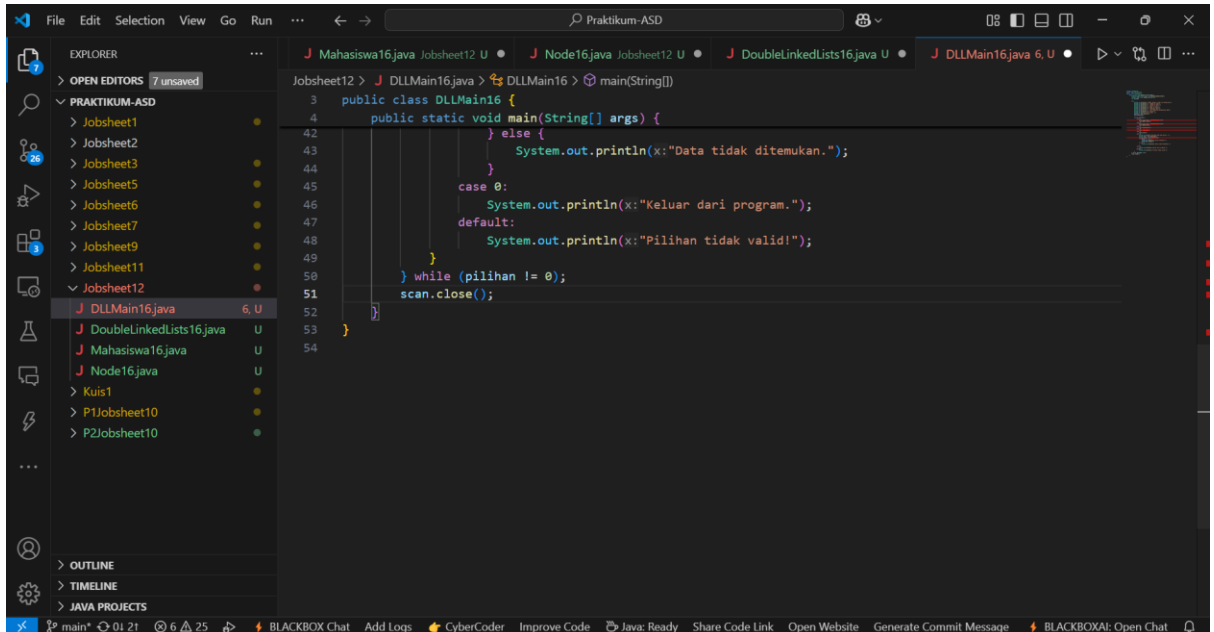
```
1 package Jobsheet12;
2 import java.util.Scanner;
3 public class DLLMain16 {
4
5     public static void main(String[] args) {
6         DoubleLinkedLists16 list = new DoubleLinkedLists16();
7         Scanner scan = new Scanner(System.in);
8         int pilihan;
9
10        do {
11            System.out.println(x:"\nMenu Double Linked List Mahasiswa");
12            System.out.println(x:"1. Tambah di awal");
13            System.out.println(x:"2. Tambah di akhir");
14            System.out.println(x:"3. Hapus di awal");
15            System.out.println(x:"4. Hapus di akhir");
16            System.out.println(x:"5. Tampilkan data");
17            System.out.println(x:"7. Cari Mahasiswa berdasarkan NIM");
18            System.out.println(x:"0. Keluar");
19            System.out.print(s:"Pilih menu: ");
20            pilihan = scan.nextInt();
21            scan.nextLine();
22
23            switch (pilihan) {
24                case 1:
25                    Mahasiswa16 mhs = inputMahasiswa(scan);
26                    list.addFirst(mhs);
27                case 2:
28                    Mahasiswa16 mhs = inputMahasiswa(scan);
29                    list.addLast(mhs);
30            }
31        }
32    }
33 }
```

### 14. Tambahkan switch case untuk menjalankan menu pilihan di atas



```
22 switch (pilihan) {
23     case 1:
24         Mahasiswa16 mhs = inputMahasiswa(scan);
25         list.addFirst(mhs);
26     case 2:
27         Mahasiswa16 mhs = inputMahasiswa(scan);
28         list.addLast(mhs);
29     case 3:
30         list.removeFirst();
31     case 4:
32         list.removeLast();
33     case 5:
34         list.print();
35     case 6:
36         System.out.print(s:"Masukkan NIM yang dicari: ");
37         String nim = scan.nextLine();
38         Node16 found = list.search(nim);
39         if (found != null) {
40             System.out.println(x:"Data ditemukan:");
41             found.data.tampil();
42         } else {
43             System.out.println(x:"Data tidak ditemukan.");
44         }
45     case 0:
46         System.out.println(x:"Keluar dari program.");
47     default:
48         System.out.println(x:"Pilihan tidak valid!");
49 }
```

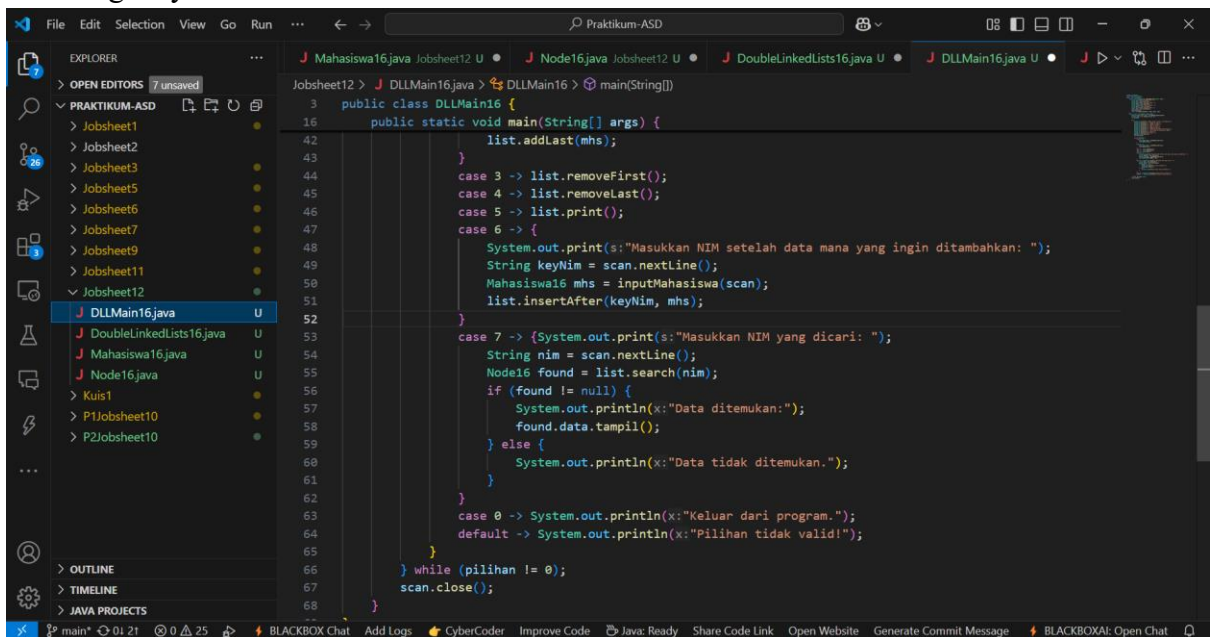
15. Jangan lupa tambahkan while di bawah switch case dan close untuk menutup object scanner



The screenshot shows an IDE with a project named 'PRAKTIKUM-ASD'. The Explorer panel on the left shows a folder 'PRAKTIKUM-ASD' containing several files, including 'DLLMain16.java'. The main editor displays the code for 'DLLMain16.java'. The code is as follows:

```
public class DLLMain16 {  
    public static void main(String[] args) {  
        } else {  
            System.out.println(x:"Data tidak ditemukan.");  
        }  
        case 0:  
            System.out.println(x:"Keluar dari program.");  
        default:  
            System.out.println(x:"Pilihan tidak valid!");  
    }  
    } while (pilihan != 0);  
    scan.close();  
}
```

16. Ada satu karakter yang perlu ditambahkan agar code bisa berjalan. Silakan dianalisis kekurangannya dan ditambahkan sendiri.

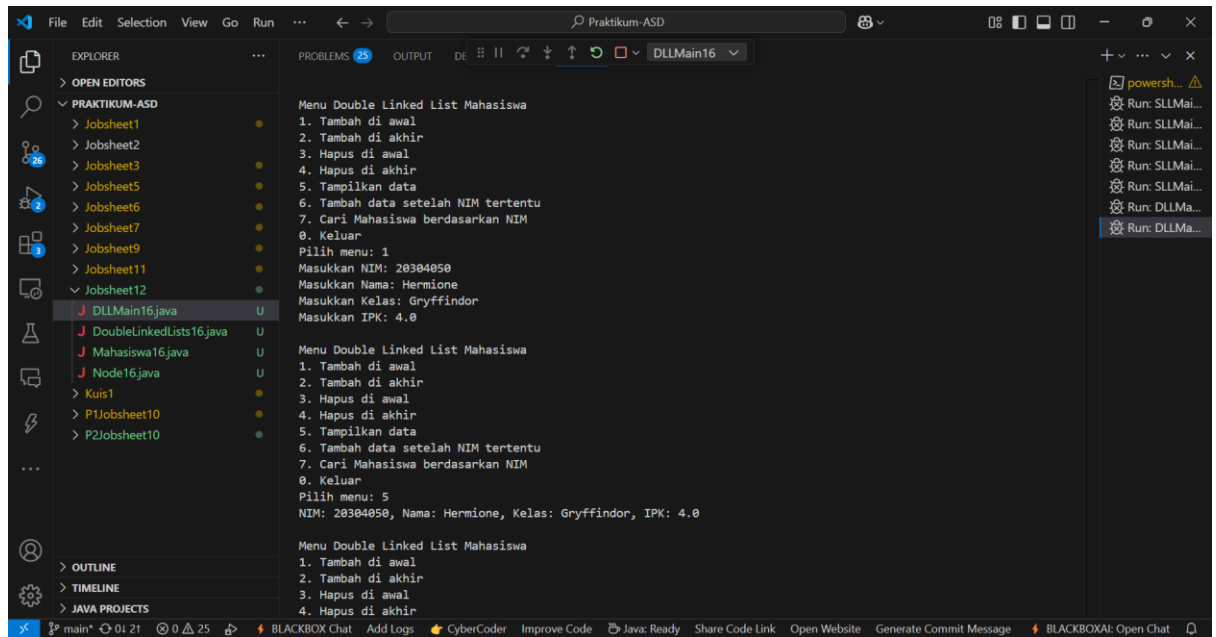


The screenshot shows the same IDE with the same project. The Explorer panel on the left shows the same files. The main editor displays the code for 'DLLMain16.java'. The code is as follows:

```
public class DLLMain16 {  
    public static void main(String[] args) {  
        list.addLast(mhs);  
    }  
    case 3 -> list.removeFirst();  
    case 4 -> list.removeLast();  
    case 5 -> list.print();  
    case 6 -> {  
        System.out.print(s:"Masukkan NIM setelah data mana yang ingin ditambahkan: ");  
        String keyNim = scan.nextLine();  
        Mahasiswa16 mhs = inputMahasiswa(scan);  
        list.insertAfter(keyNim, mhs);  
    }  
    case 7 -> {System.out.print(s:"Masukkan NIM yang dicari: ");  
        String nim = scan.nextLine();  
        Node16 found = list.search(nim);  
        if (found != null) {  
            System.out.println(x:"Data ditemukan:");  
            found.data.tampil();  
        } else {  
            System.out.println(x:"Data tidak ditemukan.");  
        }  
    }  
    case 0 -> System.out.println(x:"Keluar dari program.");  
    default -> System.out.println(x:"Pilihan tidak valid!");  
    }  
    } while (pilihan != 0);  
    scan.close();  
}
```



## 17. Verifikasi hasil



### Pertanyaan

1. Jelaskan perbedaan antara single linked list dengan double linked lists!
  - Single linked list adalah struktur data yang lebih sederhana dan hemat memori karena hanya menggunakan satu pointer untuk menunjuk ke node berikutnya, namun hanya bisa ditelusuri satu arah. Sebaliknya, double linked list menggunakan dua pointer untuk menunjuk ke node sebelumnya dan berikutnya, memungkinkan traversal dua arah serta manipulasi data yang lebih fleksibel, meskipun membutuhkan lebih banyak memori dan implementasinya lebih kompleks
2. Perhatikan class Node01, di dalamnya terdapat atribut next dan prev. Untuk apakah atribut tersebut?
  - Atribut next dan prev memungkinkan node dalam double linked list untuk terhubung dua arah, sehingga kita bisa menelusuri data dari depan ke belakang maupun dari belakang ke depan. Inilah yang membedakan double linked list dari single linked list (yang hanya memiliki next)
3. Perhatikan konstruktor pada class DoubleLinkedLists. Apa kegunaan dari konstruktor tersebut?
  - Konstruktor tersebut berguna untuk mengatur kondisi awal dari linked list agar siap digunakan. Dengan head dan tail diset ke null, ini berarti linked list masih kosong, dan siap untuk ditambahkan node-node baru
4. Pada method addFirst(), apa maksud dari kode berikut?
  - Kode head = tail = newNode; digunakan untuk menginisialisasi node pertama dalam double linked list yang kosong, di mana node tersebut menjadi baik head maupun tail dari list

5. Perhatikan pada method `addFirst()`. Apakah arti statement `head.prev = newNode` ?
  - `head.prev = newNode` berarti menghubungkan node lama yang berada di awal list ke node baru yang ditambahkan di depannya, agar hubungan dua arah dalam double linked list tetap terjaga
6. Modifikasi code pada fungsi `print()` agar dapat menampilkan warning/ pesan bahwa linked lists masih dalam kondisi.

```

3 public class DoubleLinkedLists16 {
38     public void insertAfter(String keyNim, Mahasiswa16 data) {
39         //Sisipkan di tengah
60         newNode16.next = current.next;
61         newNode16.prev = current;
62         current.next.prev = newNode16;
63         current.next = newNode16;
64     }
65
66     System.out.println("Node berhasil disisipkan setelah NIM " + keyNim);
67 }
68
69 public void print() {
70     Node16 current = head;
71     if (isEmpty()) {
72         System.out.println("x: Linked List masih kosong!");
73     }
74     while (current != null) {
75         current.data.tampil();
76         current = current.next;
77     }
78 }
79
80 public void add(int item, int index) {
81 }
82
83
84 public void removeFirst() {
85     if (isEmpty()) {
86         System.out.println("x: List Kosong, tidak dapat menghapus.");
87     }
88 }

```

7. Pada `insertAfter()`, apa maksud dari kode berikut ?  
`current.next.prev = newNode;`
  - `current.next.prev = newNode;` berfungsi untuk menghubungkan node setelah `current` ke node baru (`newNode`) pada bagian belakangnya (`prev`), sehingga hubungan dua arah dalam double linked list tetap terjaga saat node disisipkan di tengah list

8. Modifikasi menu pilihan dan switch-case agar fungsi insertAfter() masuk ke dalam menu pilihan dan dapat berjalan dengan baik.

```
Pilih menu: 5
NIM: 20304050, Nama: Hermione, Kelas: Gryffindor, IPK: 4.0

Menu Double Linked List Mahasiswa
1. Tambah di awal
2. Tambah di akhir
3. Hapus di awal
4. Hapus di akhir
5. Tampilkan data
6. Tambah data setelah NIM tertentu
7. Cari Mahasiswa berdasarkan NIM
0. Keluar

Pilih menu: 6
Masukkan NIM setelah data mana yang ingin ditambahkan: 20304050
Masukkan NIM: 244107020135
Masukkan Nama: Zakaria
Masukkan Kelas: 1B
Masukkan IPK: 4.0
Node berhasil disisipkan setelah NIM 20304050

Menu Double Linked List Mahasiswa
1. Tambah di awal
2. Tambah di akhir
3. Hapus di awal
4. Hapus di akhir
5. Tampilkan data
6. Tambah data setelah NIM tertentu
7. Cari Mahasiswa berdasarkan NIM
0. Keluar

Pilih menu: 5
NIM: 20304050, Nama: Hermione, Kelas: Gryffindor, IPK: 4.0
NIM: 244107020135, Nama: Zakaria, Kelas: 1B, IPK: 4.0
```

## Kegiatan Praktikum 2

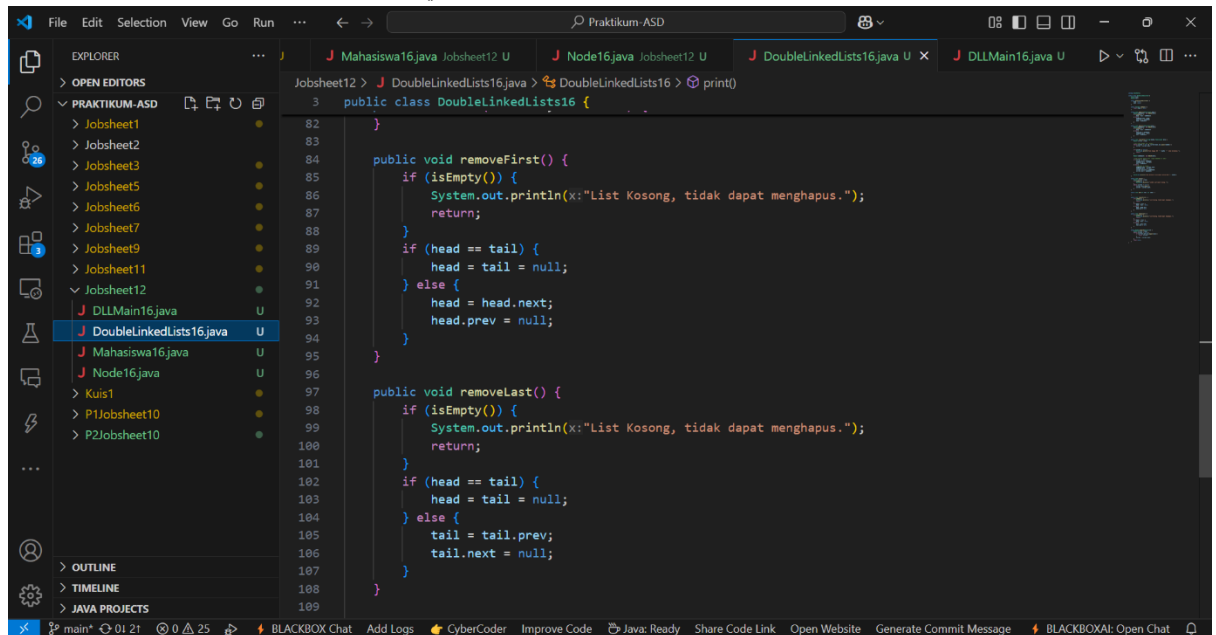
1. Buatlah method removeFirst() di dalam class DoubleLinkedLists.

```
public class DoubleLinkedLists16 {
    // ... other methods ...

    public void removeFirst() {
        if (isEmpty()) {
            System.out.println(x:"List Kosong, tidak dapat menghapus.");
            return;
        }
        if (head == tail) {
            head = tail = null;
        } else {
            head = head.next;
            head.prev = null;
        }
    }

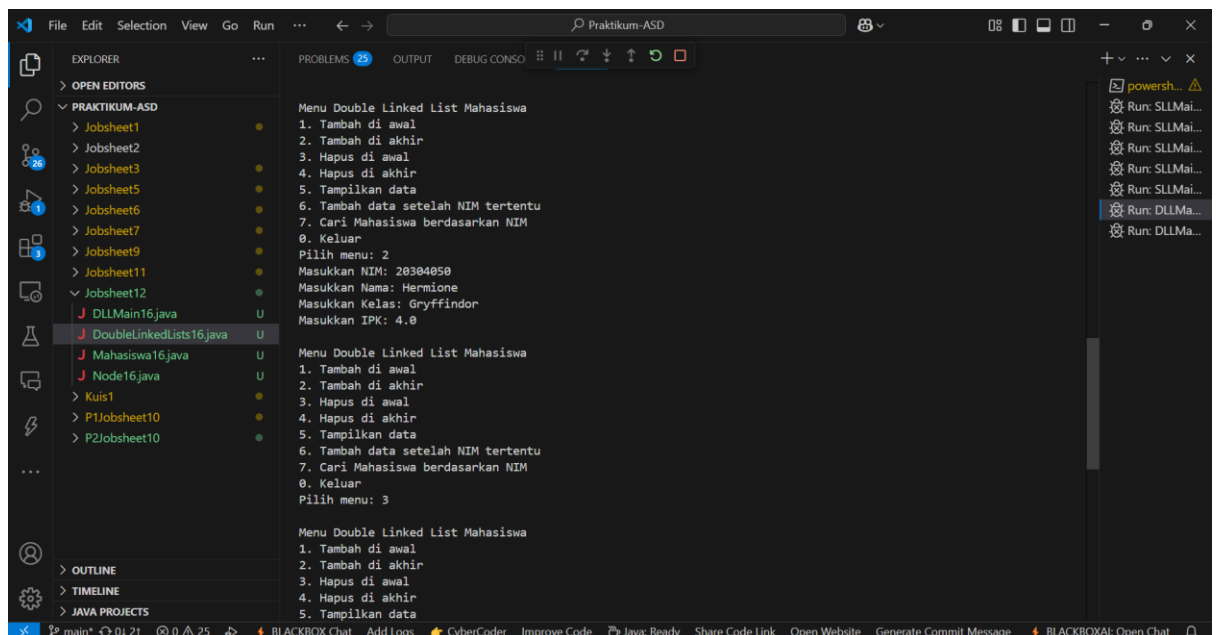
    public void removeLast() {
        if (isEmpty()) {
            System.out.println(x:"List Kosong, tidak dapat menghapus.");
            return;
        }
        if (head == tail) {
            head = tail = null;
        } else {
            tail = tail.prev;
            tail.next = null;
        }
    }
}
```

## 2. Tambahkan method removeLast() di dalam class DoubleLinkedLists.



```
public class DoubleLinkedLists16 {  
    ...  
    public void removeFirst() {  
        if (isEmpty()) {  
            System.out.println("List Kosong, tidak dapat menghapus.");  
            return;  
        }  
        if (head == tail) {  
            head = tail = null;  
        } else {  
            head = head.next;  
            head.prev = null;  
        }  
    }  
    public void removeLast() {  
        if (isEmpty()) {  
            System.out.println("List Kosong, tidak dapat menghapus.");  
            return;  
        }  
        if (head == tail) {  
            head = tail = null;  
        } else {  
            tail = tail.prev;  
            tail.next = null;  
        }  
    }  
}
```

## 3. Verifikasi hasil



```
Menu Double Linked List Mahasiswa  
1. Tambah di awal  
2. Tambah di akhir  
3. Hapus di awal  
4. Hapus di akhir  
5. Tampilkan data  
6. Tambah data setelah NIM tertentu  
7. Cari Mahasiswa berdasarkan NIM  
0. Keluar  
Pilih menu: 2  
Masukkan NIM: 20304050  
Masukkan Nama: Hermione  
Masukkan Kelas: Gryffindor  
Masukkan IPK: 4.0  
  
Menu Double Linked List Mahasiswa  
1. Tambah di awal  
2. Tambah di akhir  
3. Hapus di awal  
4. Hapus di akhir  
5. Tampilkan data  
6. Tambah data setelah NIM tertentu  
7. Cari Mahasiswa berdasarkan NIM  
0. Keluar  
Pilih menu: 3  
  
Menu Double Linked List Mahasiswa  
1. Tambah di awal  
2. Tambah di akhir  
3. Hapus di awal  
4. Hapus di akhir  
5. Tampilkan data
```

## Pertanyaan

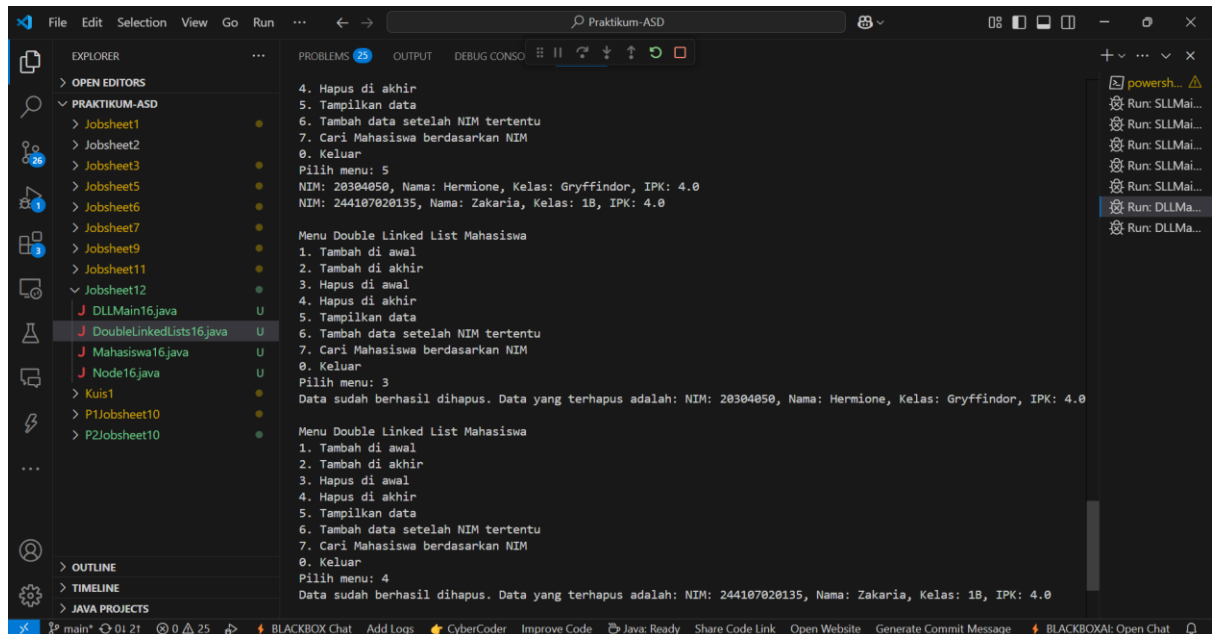
### 1. Apakah maksud statement berikut pada method removeFirst()?

head = head.next;

head.prev = null;

- Statement head = head.next; head.prev = null; digunakan untuk menghapus node pertama pada double linked list dan mengatur ulang pointer agar node baru yang menjadi head tidak lagi memiliki referensi ke node yang dihapus, menjaga integritas struktur data linked list

2. Modifikasi kode program untuk menampilkan pesan “Data sudah berhasil dihapus.  
Data yang terhapus adalah ... “



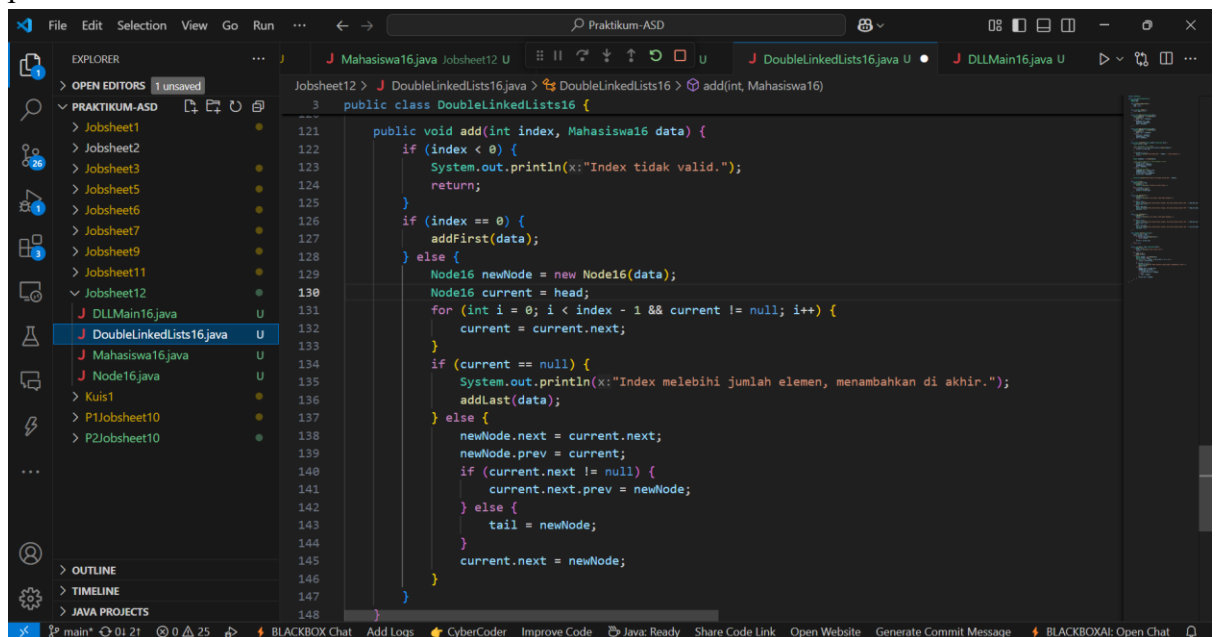
```
4. Hapus di akhir
5. Tampilkan data
6. Tambah data setelah NIM tertentu
7. Cari Mahasiswa berdasarkan NIM
0. Keluar
Pilih menu: 5
NIM: 20304050, Nama: Hermione, Kelas: Gryffindor, IPK: 4.0
NIM: 244107020135, Nama: Zakaria, Kelas: 1B, IPK: 4.0

Menu Double Linked List Mahasiswa
1. Tambah di awal
2. Tambah di akhir
3. Hapus di awal
4. Hapus di akhir
5. Tampilkan data
6. Tambah data setelah NIM tertentu
7. Cari Mahasiswa berdasarkan NIM
0. Keluar
Pilih menu: 3
Data sudah berhasil dihapus. Data yang terhapus adalah: NIM: 20304050, Nama: Hermione, Kelas: Gryffindor, IPK: 4.0

Menu Double Linked List Mahasiswa
1. Tambah di awal
2. Tambah di akhir
3. Hapus di awal
4. Hapus di akhir
5. Tampilkan data
6. Tambah data setelah NIM tertentu
7. Cari Mahasiswa berdasarkan NIM
0. Keluar
Pilih menu: 4
Data sudah berhasil dihapus. Data yang terhapus adalah: NIM: 244107020135, Nama: Zakaria, Kelas: 1B, IPK: 4.0
```

## Tugas

1. Tambahkan fungsi add() pada kelas DoubleLinkedList untuk menambahkan node pada indeks tertentu



```
public class DoubleLinkedLists16 {
    ...
    public void add(int index, Mahasiswa16 data) {
        if (index < 0) {
            System.out.println("Index tidak valid.");
            return;
        }
        if (index == 0) {
            addFirst(data);
        } else {
            Node16 newNode = new Node16(data);
            Node16 current = head;
            for (int i = 0; i < index - 1 && current != null; i++) {
                current = current.next;
            }
            if (current == null) {
                System.out.println("Index melebihi jumlah elemen, menambahkan di akhir.");
                addLast(data);
            } else {
                newNode.next = current.next;
                newNode.prev = current;
                if (current.next != null) {
                    current.next.prev = newNode;
                } else {
                    tail = newNode;
                }
                current.next = newNode;
            }
        }
    }
}
```

2. Tambahkan `removeAfter()` pada kelas `DoubleLinkedList` untuk menghapus node setelah data key.

```

149
150
151
152     if (isEmpty()) {
153         System.out.println(x:"List kosong. Tidak ada yang bisa dihapus.");
154         return;
155     }
156     while (current != null && !current.data.nim.equals(keyNim)) {
157         current = current.next;
158     }
159     if (current == null) {
160         System.out.println("Node dengan NIM " + keyNim + " tidak ditemukan.");
161         return;
162     }
163     if (current.next != null) {
164         Node16 removeAfter = current.next;
165         current.next = removeAfter.next;
166         if (removeAfter.next != null) {
167             removeAfter.next.prev = current;
168         } else {
169             tail = current;
170         }
171         System.out.println("Node setelah NIM " + keyNim + " berhasil dihapus.");
172     } else {
173         System.out.println("Tidak ada node setelah NIM " + keyNim + ".");
174     }
175 }
176

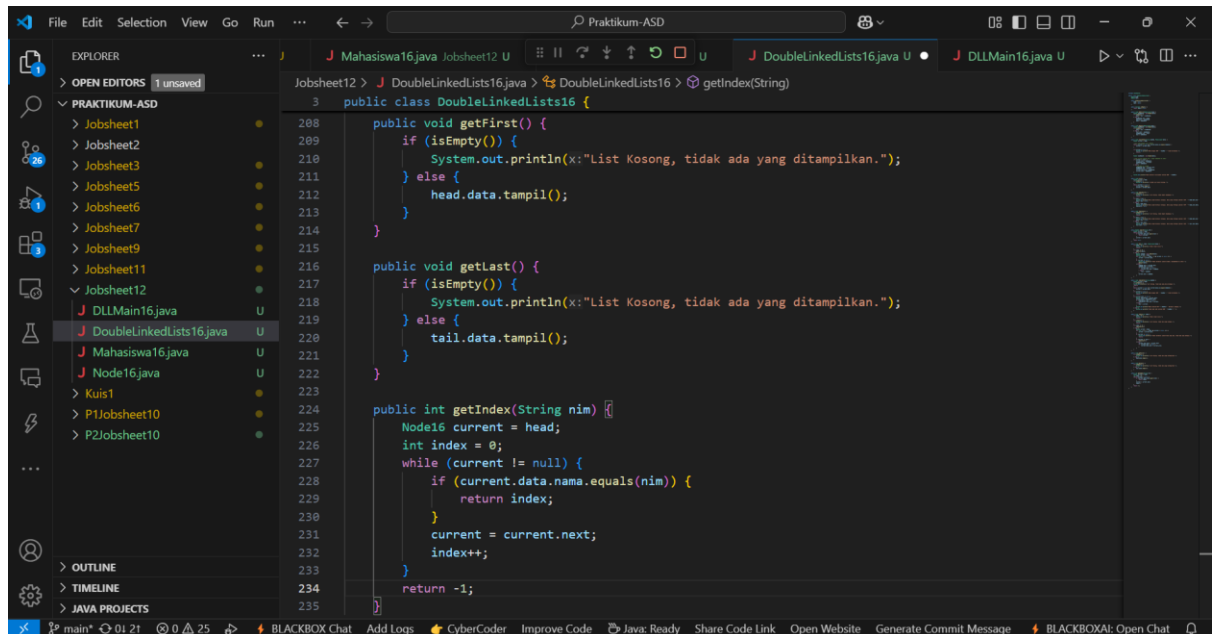
```

3. Tambahkan fungsi `remove()` pada kelas `DoubleLinkedList` untuk menghapus node pada indeks tertentu.

```

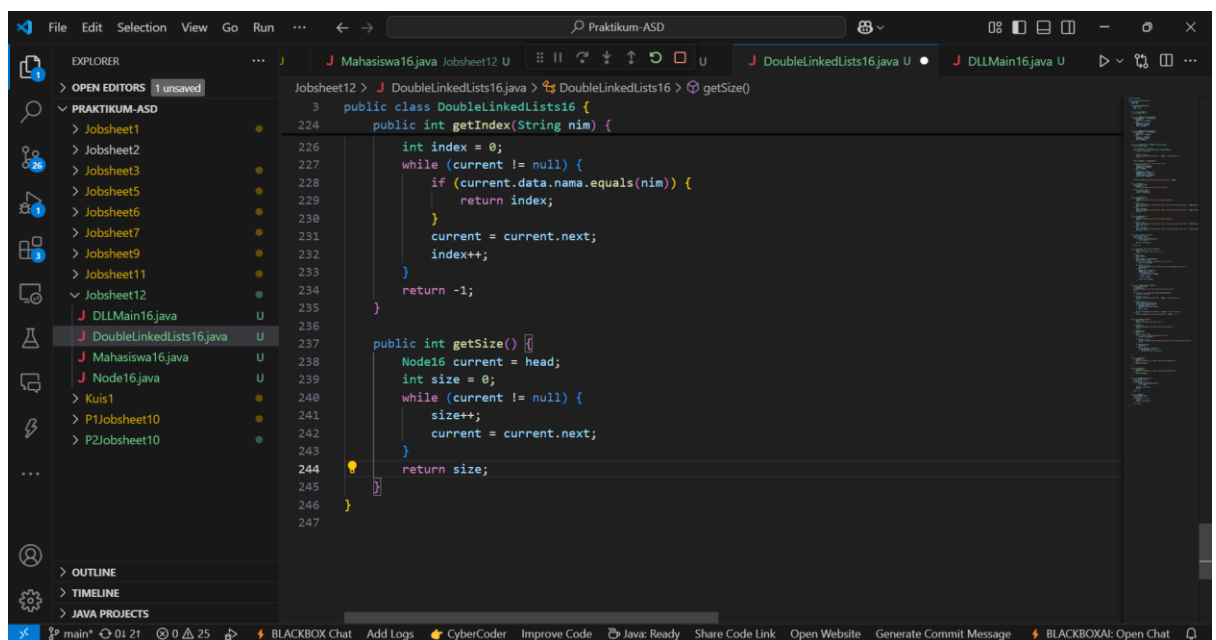
3 public class DoubleLinkedLists16 {
177     public void remove(int index) {
178         if (index < 0) {
179             System.out.println(x:"Index tidak valid.");
180             return;
181         }
182         if (isEmpty()) {
183             System.out.println(x:"List Kosong, tidak ada yang dihapus.");
184             return;
185         }
186         if (index == 0) {
187             removeFirst();
188         } else {
189             Node16 current = head;
190             for (int i = 0; i < index && current != null; i++) {
191                 current = current.next;
192             }
193             if (current == null) {
194                 System.out.println(x:"Index melebihi jumlah data yang ada, tidak ada yang dihapus.");
195                 return;
196             }
197             if (current == tail) {
198                 removeLast();
199             } else {
200                 current.prev.next = current.next;
201                 if (current.next != null) {
202                     current.next.prev = current.prev;
203                 }
204             }
205         }
206     }
207 }
  
```

4. Tambahkan fungsi `getFirst()`, `getLast()` dan `getIndex()` untuk menampilkan data pada node head, node tail dan node pada indeks tertentu.



```
public class DoubleLinkedList16 {  
    public void getFirst() {  
        if (isEmpty()) {  
            System.out.println(x: "List Kosong, tidak ada yang ditampilkan.");  
        } else {  
            head.data.tampil();  
        }  
    }  
  
    public void getLast() {  
        if (isEmpty()) {  
            System.out.println(x: "List Kosong, tidak ada yang ditampilkan.");  
        } else {  
            tail.data.tampil();  
        }  
    }  
  
    public int getIndex(String nim) {  
        Node16 current = head;  
        int index = 0;  
        while (current != null) {  
            if (current.data.nama.equals(nim)) {  
                return index;  
            }  
            current = current.next;  
            index++;  
        }  
        return -1;  
    }  
}
```

5. tambahkan kode program dan fungsi agar dapat membaca size/ jumlah data pada Double Linked List



```
public class DoubleLinkedList16 {  
    public int getIndex(String nim) {  
        int index = 0;  
        while (current != null) {  
            if (current.data.nama.equals(nim)) {  
                return index;  
            }  
            current = current.next;  
            index++;  
        }  
        return -1;  
    }  
  
    public int getSize() {  
        Node16 current = head;  
        int size = 0;  
        while (current != null) {  
            size++;  
            current = current.next;  
        }  
        return size;  
    }  
}
```