

Nama : Muhammad Zakaria Haniya

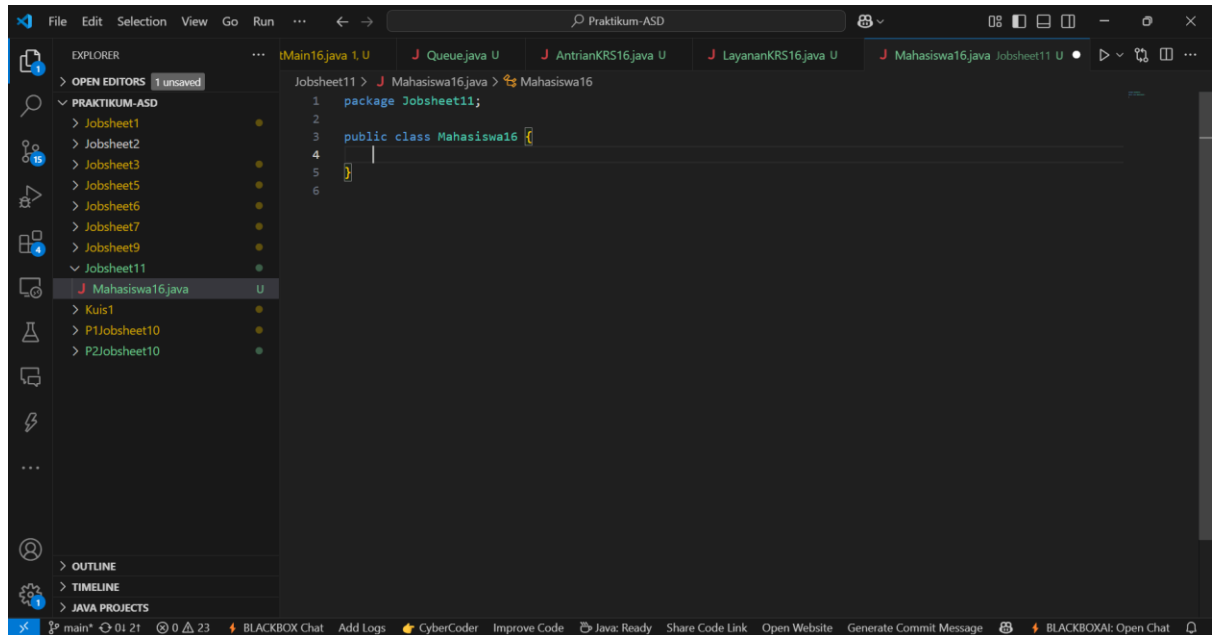
NIM : 244107020135

Kelas : TI-1B

Absen : 16

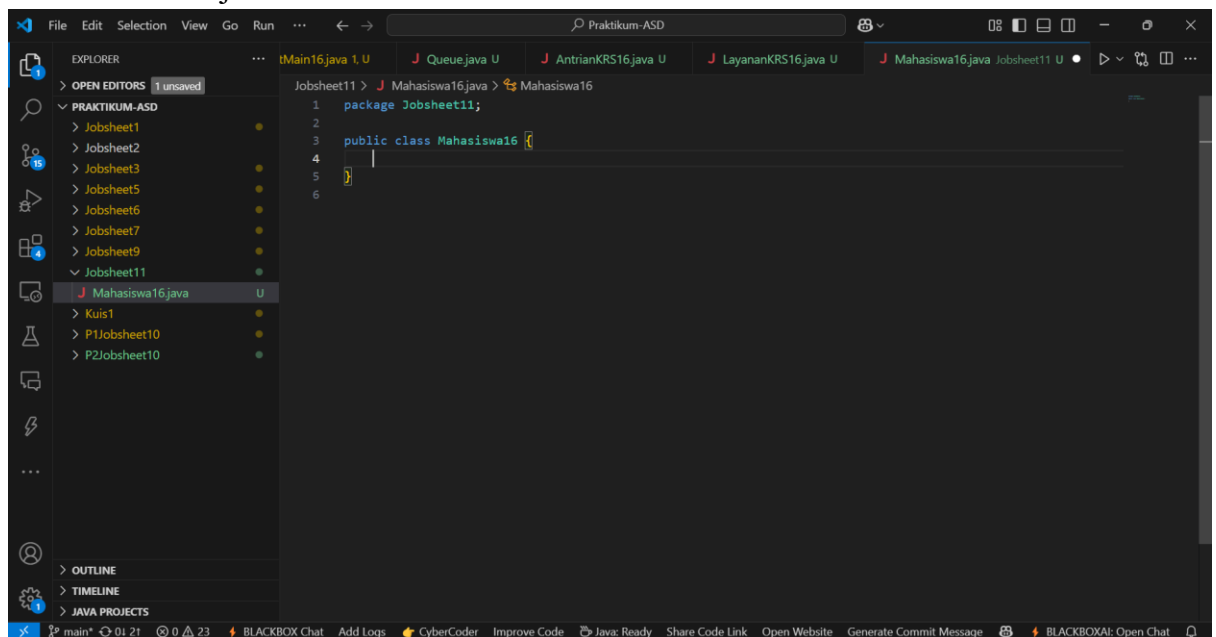
Pembuatan Single Linked List

1. Pada Project yang sudah dibuat pada Minggu sebelumnya. Buat folder atau package baru bernama Jobsheet11 di dalam repository Praktikum ASD.

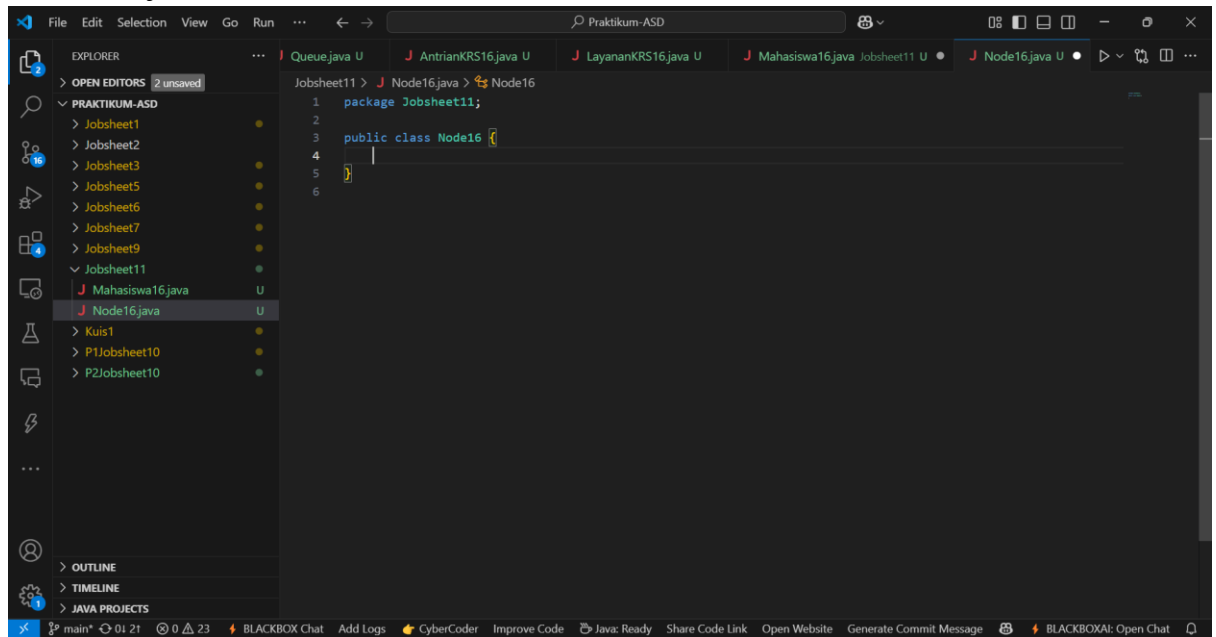


2. Tambahkan class-class berikut:

a. Mahasiswa00.java



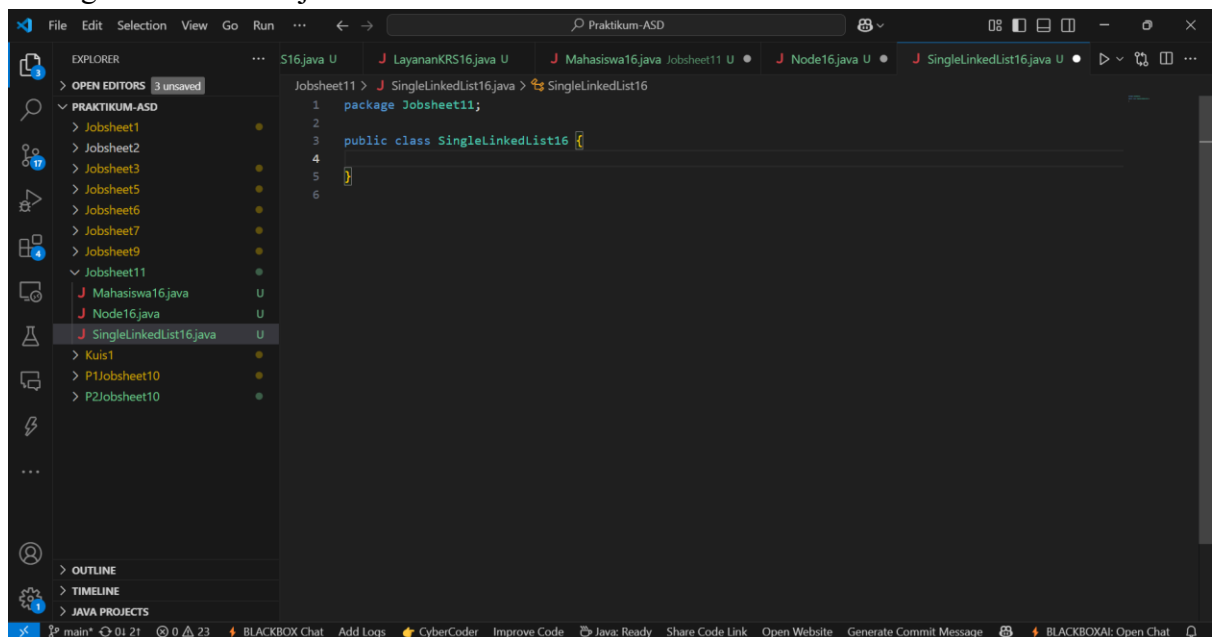
b. Node00.java



The screenshot shows an IDE with the Explorer on the left, displaying a project structure under 'PRAKTIKUM-ASD'. The file 'Node16.java' is selected. The main editor shows the following code:

```
1 package Jobsheet11;  
2  
3 public class Node16 {  
4  
5 }  
6
```

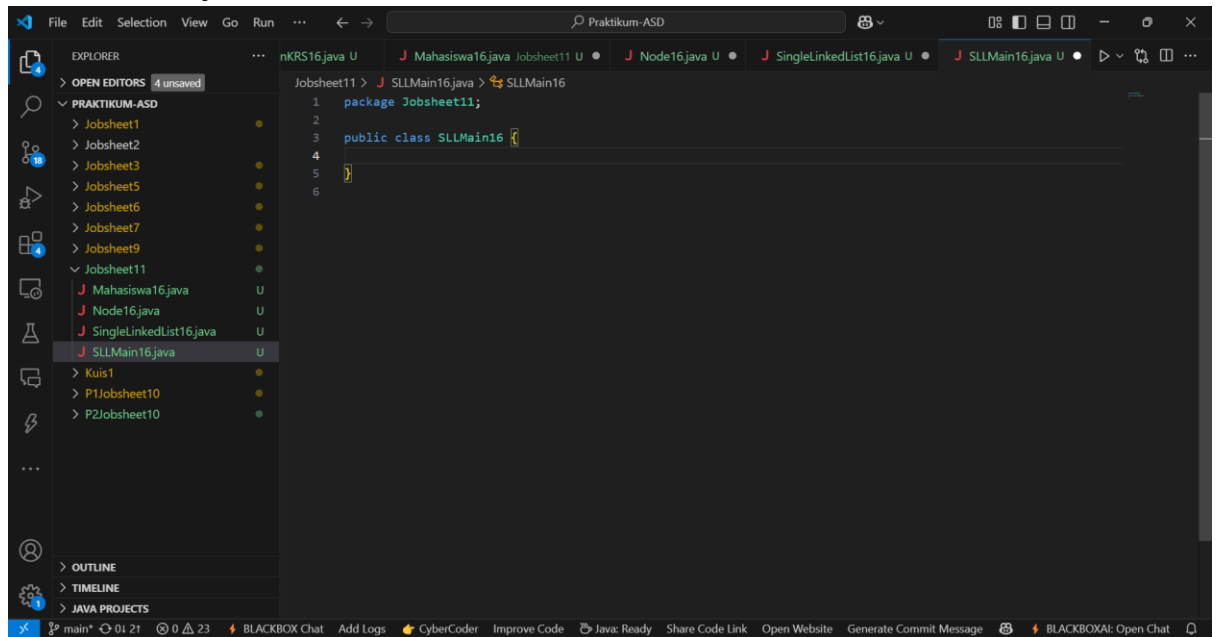
c. SingleLinkedList00.java



The screenshot shows the same IDE with the file 'SingleLinkedList16.java' selected. The main editor shows the following code:

```
1 package Jobsheet11;  
2  
3 public class SingleLinkedList16 {  
4  
5 }  
6
```

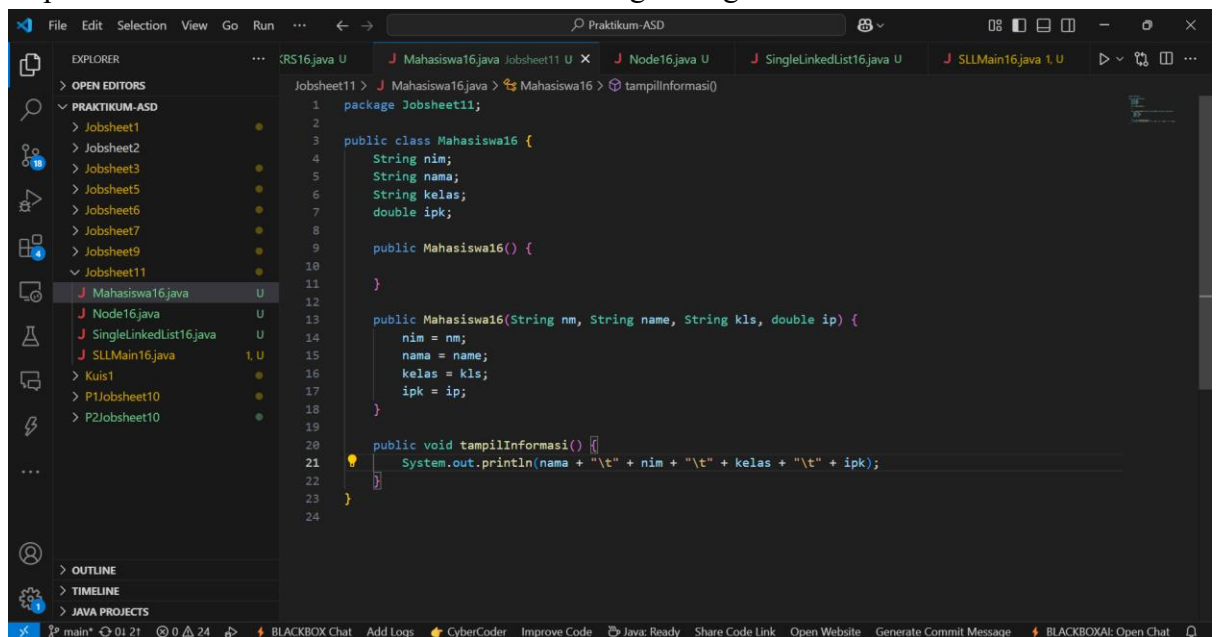
d. SLLMain00.java



```
1 package Jobsheet11;
2
3 public class SLLMain16 {
4
5 }
6
```

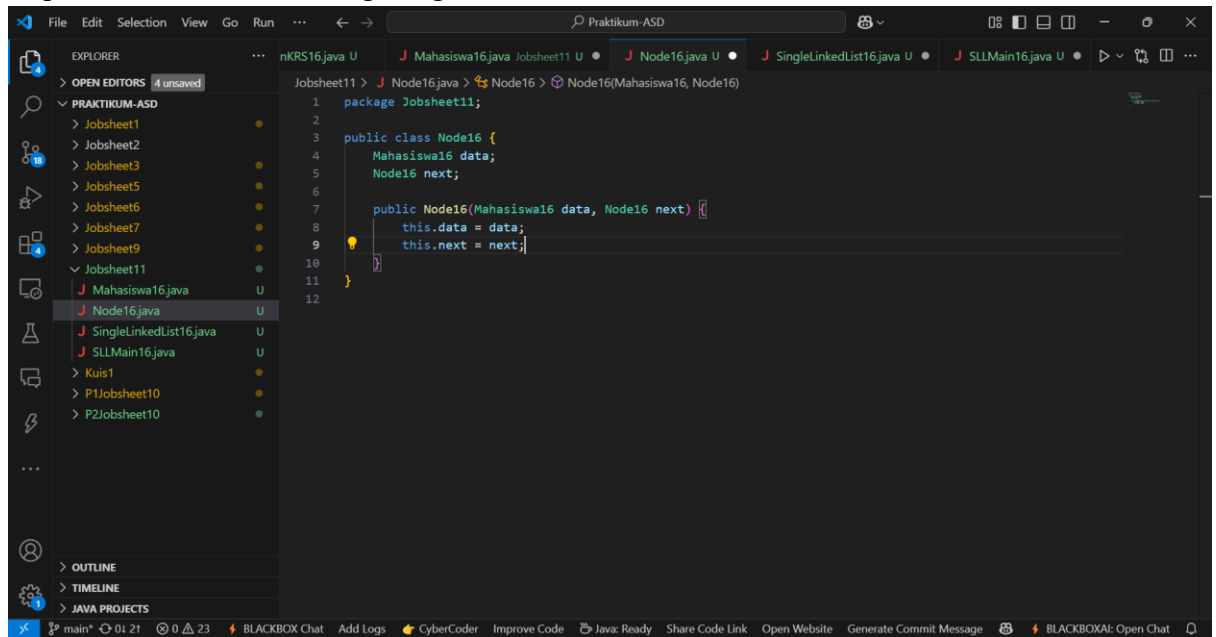
Ganti 00 dengan nomer Absen Anda

3. Implementasikan Class Mahasiswa00 sesuai dengan diagram class berikut ini :

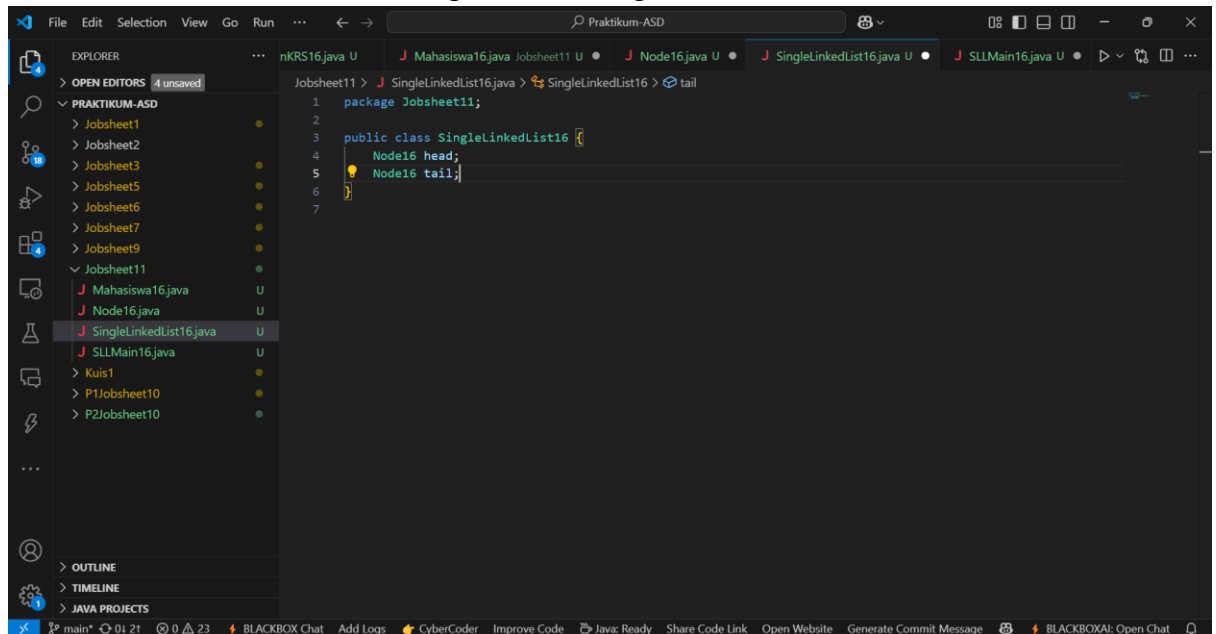


```
1 package Jobsheet11;
2
3 public class Mahasiswa16 {
4     String nim;
5     String nama;
6     String kelas;
7     double ipk;
8
9     public Mahasiswa16() {
10
11     }
12
13     public Mahasiswa16(String nm, String name, String kls, double ip) {
14         nim = nm;
15         nama = name;
16         kelas = kls;
17         ipk = ip;
18     }
19
20     public void tampilInformasi() {
21         System.out.println(nama + "\t" + nim + "\t" + kelas + "\t" + ipk);
22     }
23
24 }
```

4. Implementasi class Node seperti gambar berikut ini

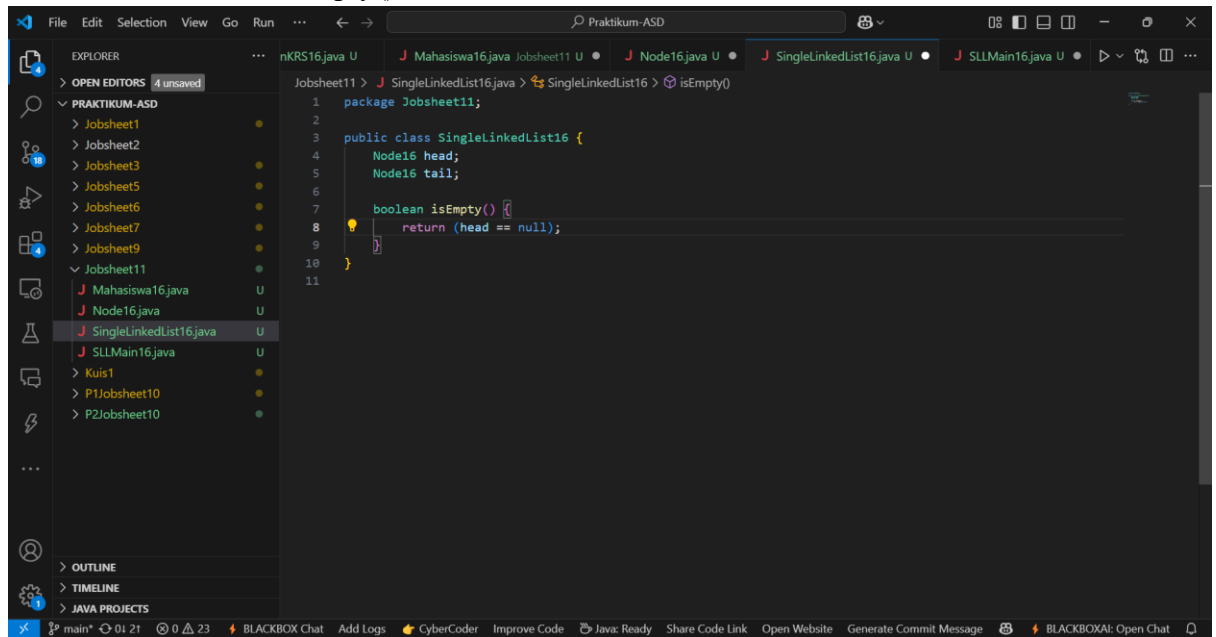


5. Tambahkan attribute head dan tail pada class SingleLinkedList



6. Sebagai langkah berikutnya, akan diimplementasikan method-method yang terdapat pada SingleLinkedList.

7. Tambahkan method isEmpty().

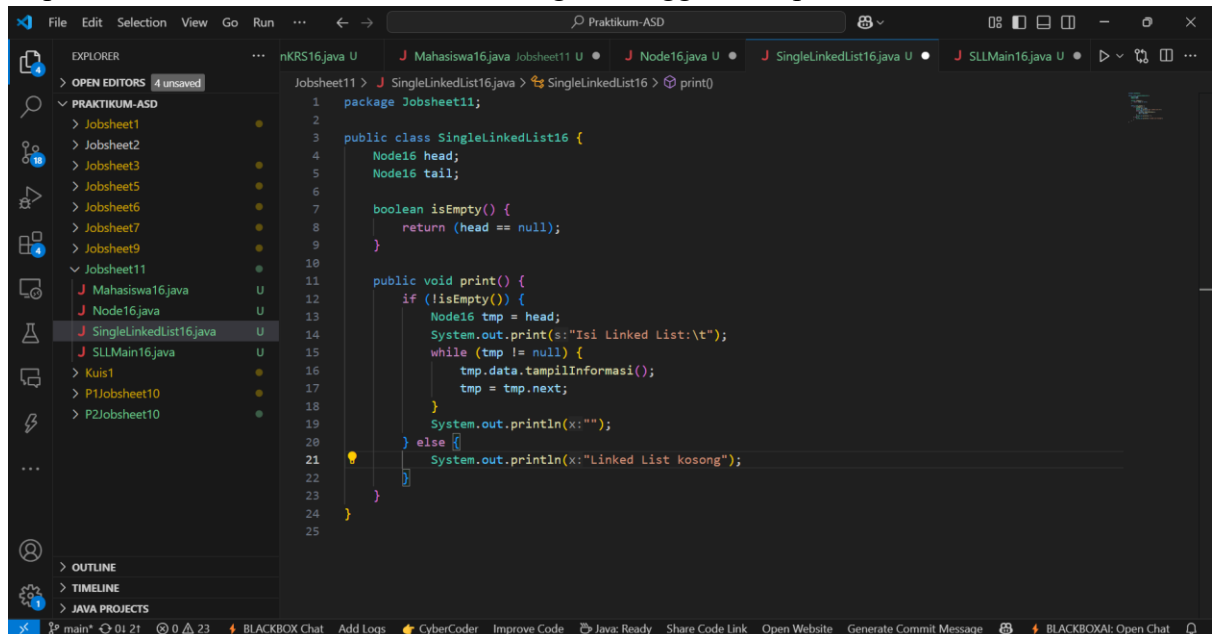


The screenshot shows an IDE with the following code in the `SingleLinkedList16.java` file:

```
1 package Jobsheet11;
2
3 public class SingleLinkedList16 {
4     Node16 head;
5     Node16 tail;
6
7     boolean isEmpty() {
8         return (head == null);
9     }
10 }
11
```

The Explorer panel on the left shows the project structure with files like `Jobsheet1` through `Jobsheet11`, `Mahasiswa16.java`, `Node16.java`, `SingleLinkedList16.java`, and `SLLMain16.java`.

8. Implementasi method untuk mencetak dengan menggunakan proses traverse

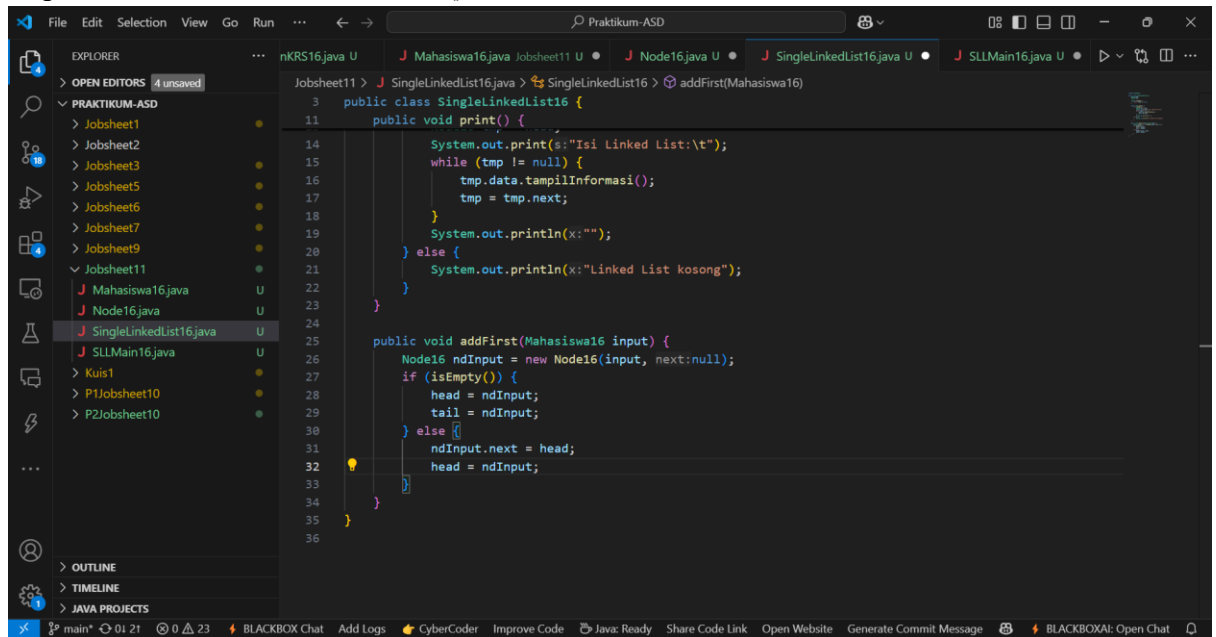


The screenshot shows the same IDE with the `print()` method added to the `SingleLinkedList16` class. The code is as follows:

```
1 package Jobsheet11;
2
3 public class SingleLinkedList16 {
4     Node16 head;
5     Node16 tail;
6
7     boolean isEmpty() {
8         return (head == null);
9     }
10
11     public void print() {
12         if (!isEmpty()) {
13             Node16 tmp = head;
14             System.out.print(s:"Isi Linked List:\t");
15             while (tmp != null) {
16                 tmp.data.tampilInformasi();
17                 tmp = tmp.next;
18             }
19             System.out.println(x:"");
20         } else {
21             System.out.println(x:"Linked List kosong");
22         }
23     }
24 }
25
```

The Explorer panel on the left remains the same, showing the project structure.

9. Implementasikan method addFirst().

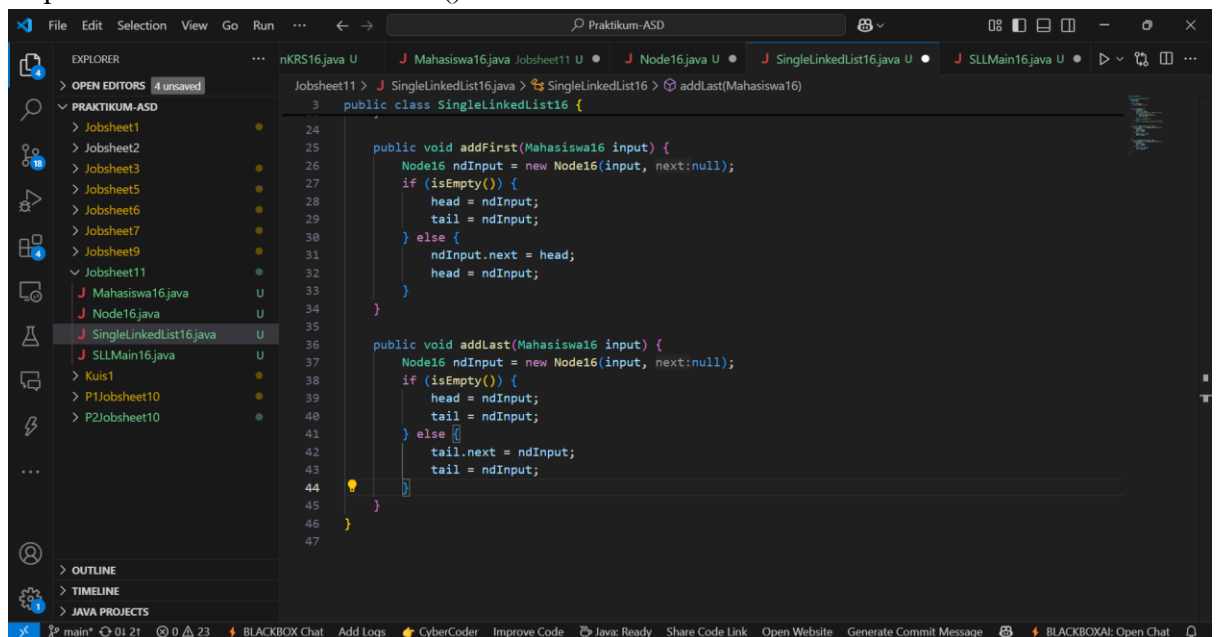


The screenshot shows an IDE with the following code in `SingleLinkedList16.java`:

```
3 public class SingleLinkedList16 {
11     public void print() {
14         System.out.print(s:"Isi Linked List:\t");
15         while (tmp != null) {
16             tmp.data.tampilInformasi();
17             tmp = tmp.next;
18         }
19         System.out.println(x:"");
20     } else {
21         System.out.println(x:"Linked List kosong");
22     }
23 }

25 public void addFirst(Mahasiswa16 input) {
26     Node16 ndInput = new Node16(input, next:null);
27     if (isEmpty()) {
28         head = ndInput;
29         tail = ndInput;
30     } else {
31         ndInput.next = head;
32         head = ndInput;
33     }
34 }
35 }
36 }
```

10. Implementasikan method addLast().

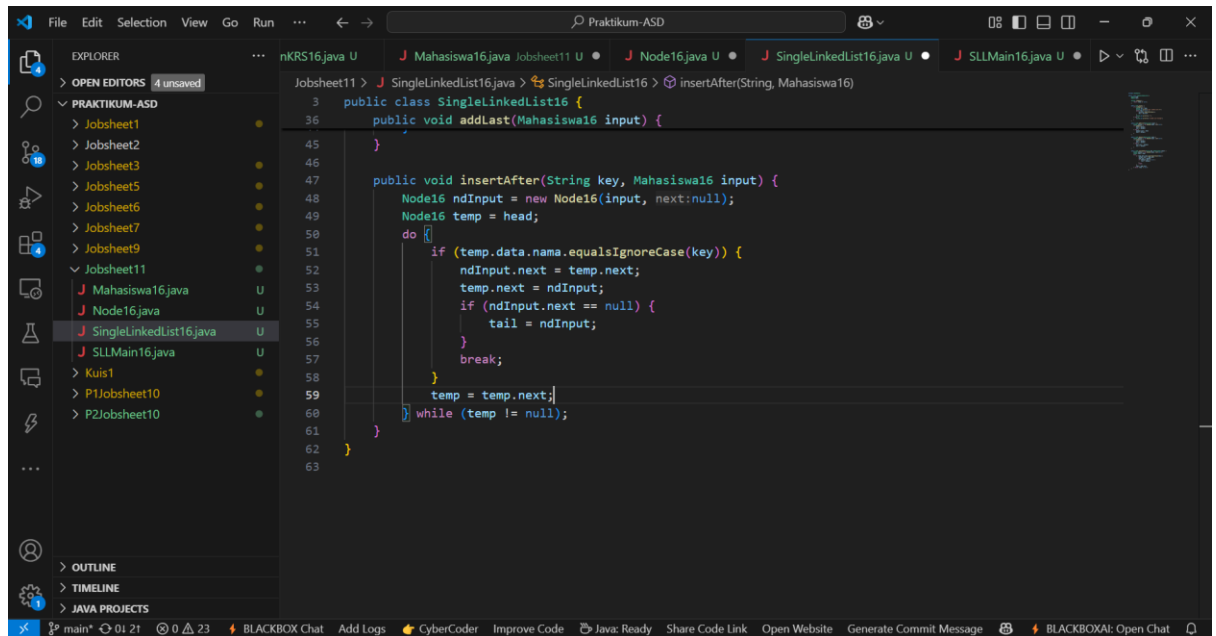


The screenshot shows an IDE with the following code in `SingleLinkedList16.java`:

```
24
25
26 public void addFirst(Mahasiswa16 input) {
27     Node16 ndInput = new Node16(input, next:null);
28     if (isEmpty()) {
29         head = ndInput;
30         tail = ndInput;
31     } else {
32         ndInput.next = head;
33         head = ndInput;
34     }
35 }

36 public void addLast(Mahasiswa16 input) {
37     Node16 ndInput = new Node16(input, next:null);
38     if (isEmpty()) {
39         head = ndInput;
40         tail = ndInput;
41     } else {
42         tail.next = ndInput;
43         tail = ndInput;
44     }
45 }
46 }
47 }
```

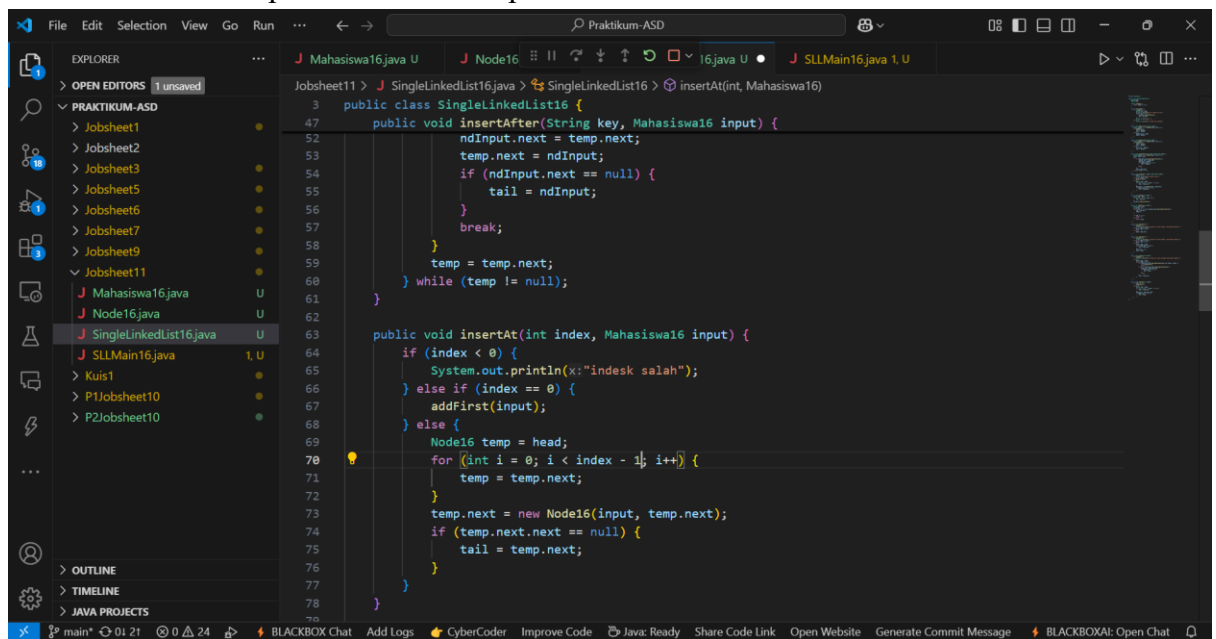
11. Implementasikan method `insertAfter`, untuk memasukkan node yang memiliki data input setelah node yang memiliki data key.



```
File Edit Selection View Go Run ... Praktikum-ASD
EXPLORER
> OPEN EDITORS 4 unsaved
> PRAKTIKUM-ASD
  > Jobsheet1
  > Jobsheet2
  > Jobsheet3
  > Jobsheet5
  > Jobsheet6
  > Jobsheet7
  > Jobsheet9
  > Jobsheet11
  J Mahasiswa16.java U
  J Node16.java U
  J SingleLinkedList16.java U
  J SLLMain16.java U
  > Kuis1
  > P1Jobsheet10
  > P2Jobsheet10
  ...
  > OUTLINE
  > TIMELINE
  > JAVA PROJECTS
    main* 01:21 0 23 BLACKBOX Chat Add Logs CyberCoder Improve Code Java: Ready Share Code Link Open Website Generate Commit Message BLACKBOXAI: Open Chat

Jobsheet11 > J SingleLinkedList16.java > SingleLinkedList16 > insertAfter(String, Mahasiswa16)
3 public class SinglyLinkedList16 {
36 public void addLast(Mahasiswa16 input) {
45 }
46
47 public void insertAfter(String key, Mahasiswa16 input) {
48     Node16 ndInput = new Node16(input, next:null);
49     Node16 temp = head;
50     do {
51         if (temp.data.nama.equalsIgnoreCase(key)) {
52             ndInput.next = temp.next;
53             temp.next = ndInput;
54             if (ndInput.next == null) {
55                 tail = ndInput;
56             }
57             break;
58         }
59         temp = temp.next;
60     } while (temp != null);
61 }
62 }
63 }
```

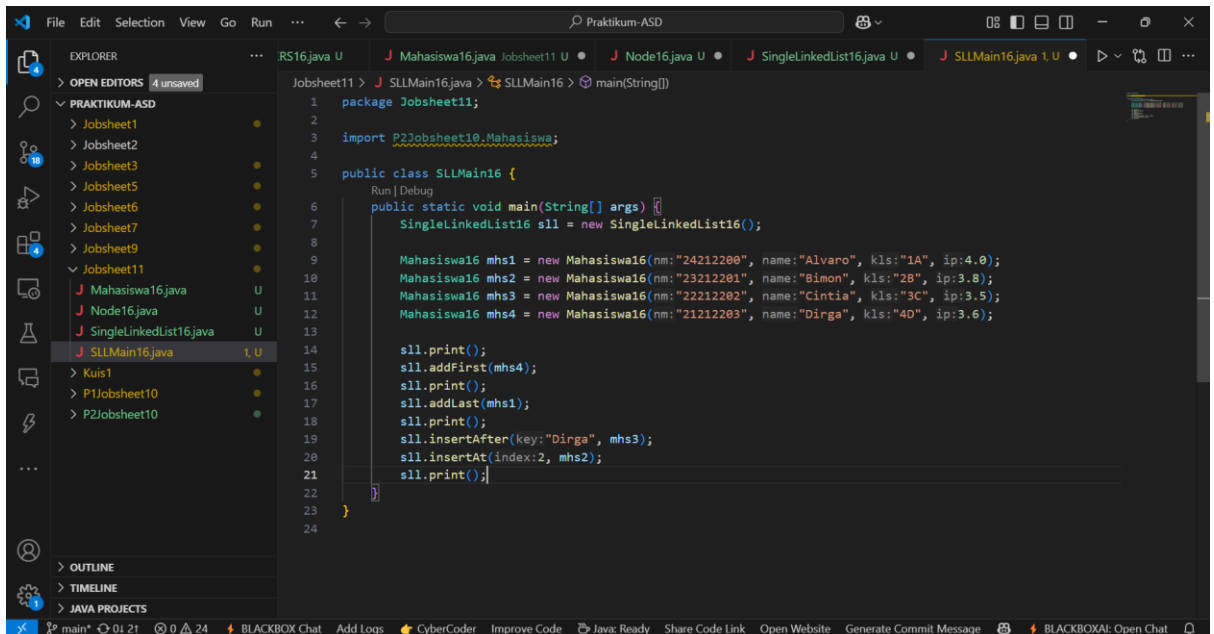
12. Tambahkan method penambahan node pada indeks tertentu.



```
File Edit Selection View Go Run ... Praktikum-ASD
EXPLORER
> OPEN EDITORS 1 unsaved
> PRAKTIKUM-ASD
  > Jobsheet1
  > Jobsheet2
  > Jobsheet3
  > Jobsheet5
  > Jobsheet6
  > Jobsheet7
  > Jobsheet9
  > Jobsheet11
  J Mahasiswa16.java U
  J Node16.java U
  J SingleLinkedList16.java U
  J SLLMain16.java 1 U
  > Kuis1
  > P1Jobsheet10
  > P2Jobsheet10
  ...
  > OUTLINE
  > TIMELINE
  > JAVA PROJECTS
    main* 01:21 0 24 BLACKBOX Chat Add Logs CyberCoder Improve Code Java: Ready Share Code Link Open Website Generate Commit Message BLACKBOXAI: Open Chat

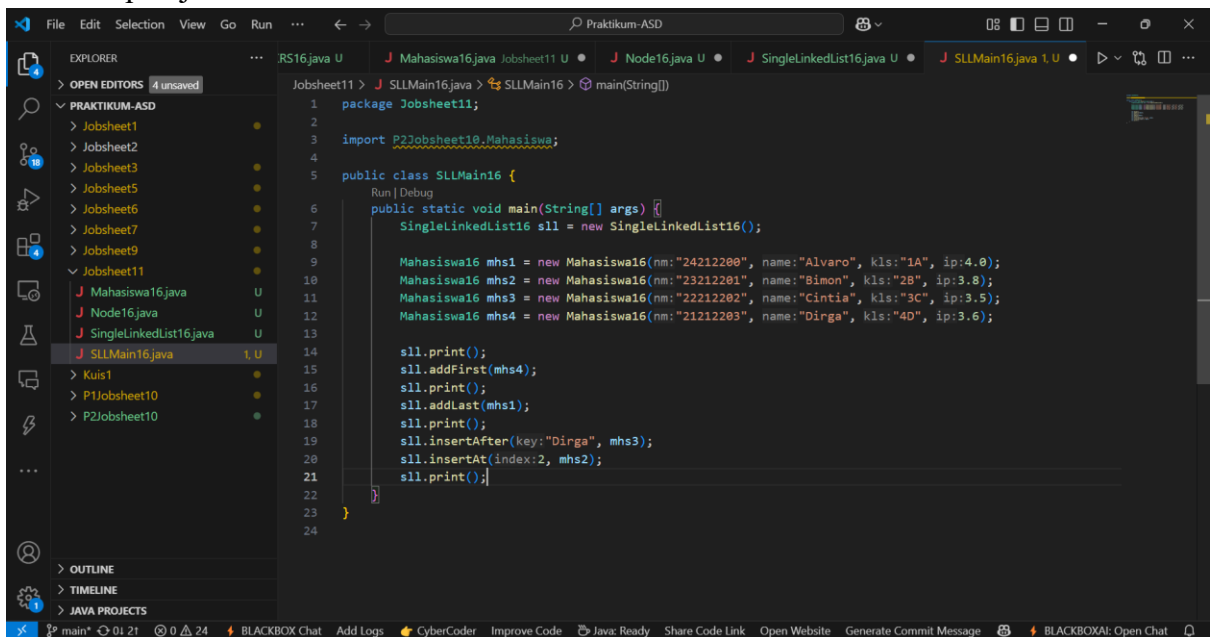
Jobsheet11 > J SingleLinkedList16.java > SingleLinkedList16 > insertAt(int, Mahasiswa16)
3 public class SinglyLinkedList16 {
47 public void insertAfter(String key, Mahasiswa16 input) {
52     ndInput.next = temp.next;
53     temp.next = ndInput;
54     if (ndInput.next == null) {
55         tail = ndInput;
56     }
57     break;
58 }
59 temp = temp.next;
60 } while (temp != null);
61 }
62
63 public void insertAt(int index, Mahasiswa16 input) {
64     if (index < 0) {
65         System.out.println("Indeks salah");
66     } else if (index == 0) {
67         addFirst(input);
68     } else {
69         Node16 temp = head;
70         for (int i = 0; i < index - 1; i++) {
71             temp = temp.next;
72         }
73         temp.next = new Node16(input, temp.next);
74         if (temp.next.next == null) {
75             tail = temp.next;
76         }
77     }
78 }
79 }
```

13. Pada class SLLMain00, buatlah fungsi main, kemudian buat object dari class SingleLinkedList.



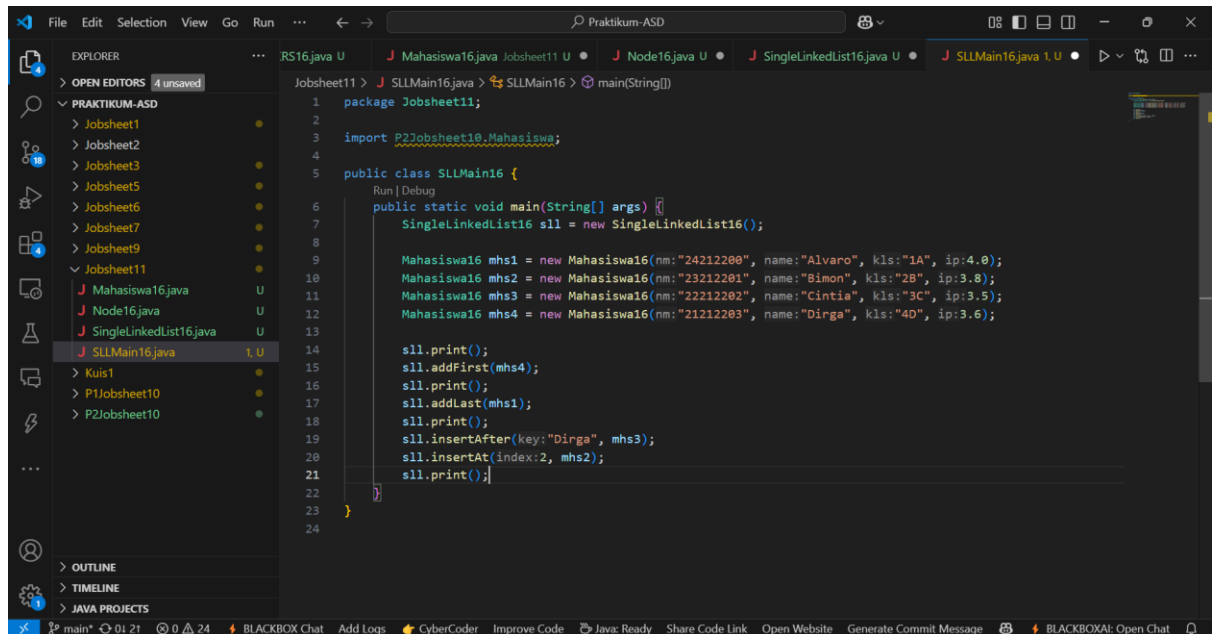
```
1 package Jobsheet11;
2
3 import P2Jobsheet10.Mahasiswa;
4
5 public class SLLMain16 {
6     public static void main(String[] args) {
7         SingleLinkedList16 sll = new SingleLinkedList16();
8
9         Mahasiswa16 mhs1 = new Mahasiswa16(nm:"24212200", name:"Alvaro", kls:"1A", ip:4.0);
10        Mahasiswa16 mhs2 = new Mahasiswa16(nm:"23212201", name:"Bimon", kls:"2B", ip:3.8);
11        Mahasiswa16 mhs3 = new Mahasiswa16(nm:"22212202", name:"Cintia", kls:"3C", ip:3.5);
12        Mahasiswa16 mhs4 = new Mahasiswa16(nm:"21212203", name:"Dirga", kls:"4D", ip:3.6);
13
14        sll.print();
15        sll.addFirst(mhs4);
16        sll.print();
17        sll.addLast(mhs1);
18        sll.print();
19        sll.insertAfter(key:"Dirga", mhs3);
20        sll.insertAt(index:2, mhs2);
21        sll.print();
22    }
23 }
24
```

14. Buat empat object mahasiswa dengan nama mhs1, mhs2, mhs3, mhs4 kemudian isi data setiap object melalui konstruktor.



```
1 package Jobsheet11;
2
3 import P2Jobsheet10.Mahasiswa;
4
5 public class SLLMain16 {
6     public static void main(String[] args) {
7         SingleLinkedList16 sll = new SingleLinkedList16();
8
9         Mahasiswa16 mhs1 = new Mahasiswa16(nm:"24212200", name:"Alvaro", kls:"1A", ip:4.0);
10        Mahasiswa16 mhs2 = new Mahasiswa16(nm:"23212201", name:"Bimon", kls:"2B", ip:3.8);
11        Mahasiswa16 mhs3 = new Mahasiswa16(nm:"22212202", name:"Cintia", kls:"3C", ip:3.5);
12        Mahasiswa16 mhs4 = new Mahasiswa16(nm:"21212203", name:"Dirga", kls:"4D", ip:3.6);
13
14        sll.print();
15        sll.addFirst(mhs4);
16        sll.print();
17        sll.addLast(mhs1);
18        sll.print();
19        sll.insertAfter(key:"Dirga", mhs3);
20        sll.insertAt(index:2, mhs2);
21        sll.print();
22    }
23 }
24
```


15. Tambahkan Method penambahan data dan pencetakan data di setiap penambahannya agar terlihat perubahannya.



```
1 package Jobsheet11;
2
3 import P2Jobsheet10.Mahasiswa;
4
5 public class SLLMain16 {
6     public static void main(String[] args) {
7         SingleLinkedList16 sll = new SingleLinkedList16();
8
9         Mahasiswa16 mhs1 = new Mahasiswa16(nm:"24212200", name:"Alvaro", kls:"1A", ip:4.0);
10        Mahasiswa16 mhs2 = new Mahasiswa16(nm:"23212201", name:"Bimon", kls:"2B", ip:3.8);
11        Mahasiswa16 mhs3 = new Mahasiswa16(nm:"22212202", name:"Cintia", kls:"3C", ip:3.5);
12        Mahasiswa16 mhs4 = new Mahasiswa16(nm:"21212203", name:"Dirga", kls:"4D", ip:3.6);
13
14        sll.print();
15        sll.addFirst(mhs4);
16        sll.print();
17        sll.addLast(mhs1);
18        sll.print();
19        sll.insertAfter(key:"Dirga", mhs3);
20        sll.insertAt(index:2, mhs2);
21        sll.print();
22    }
23 }
24 }
```

16. Verifikasi Hasil Percobaan

```
Linked List kosong
Isi Linked List:
Dirga    21212203    4D    3.6

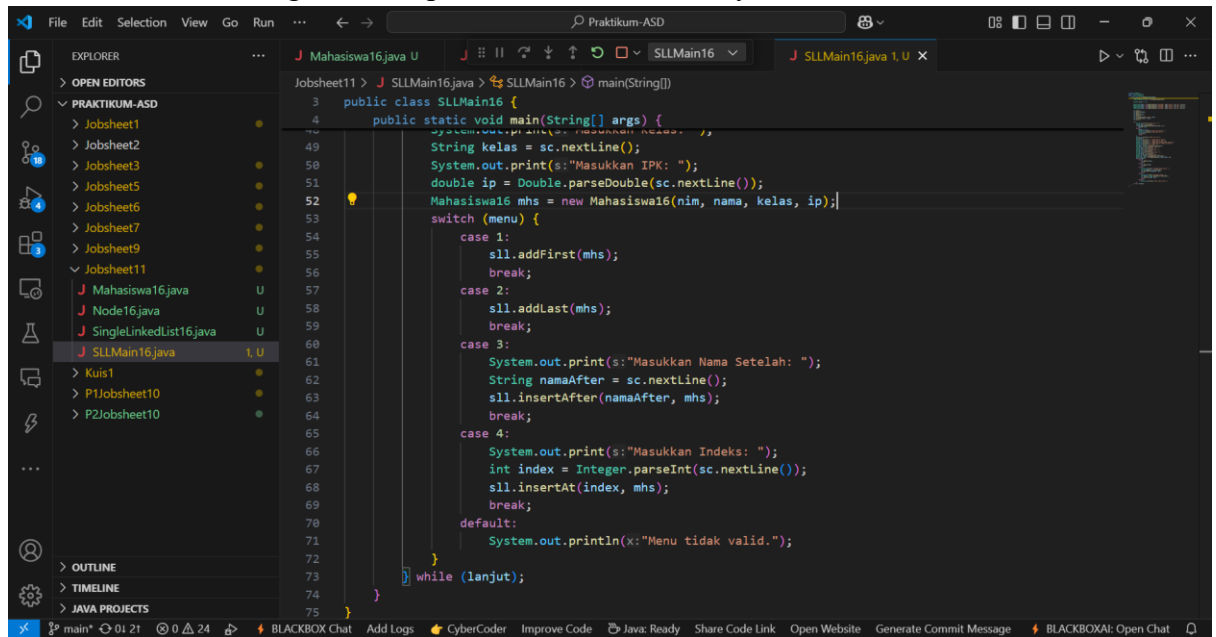
Isi Linked List:
Dirga    21212203    4D    3.6
Alvaro   24212200    1A    4.0

Isi Linked List:
Dirga    21212203    4D    3.6
Cintia   22212202    3C    3.5
Bimon    23212201    2B    3.8
Alvaro   24212200    1A    4.0
```

Pertanyaan

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?
 - Output “Linked List kosong” muncul karena program memanggil print() sebelum menambahkan node apapun ke linked list
2. Jelaskan kegunaan variable temp secara umum pada setiap method!
 - temp adalah variabel penunjuk (pointer) yang digunakan untuk menjelajahi atau mengakses node-node di dalam linked list, dimulai dari head

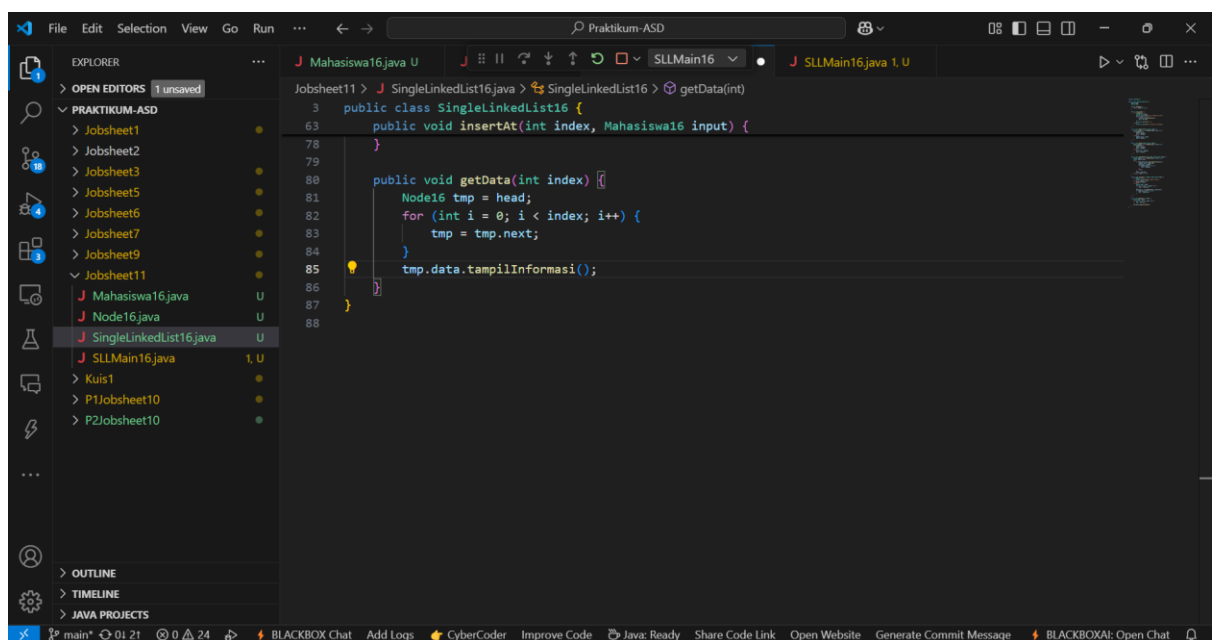
3. Lakukan modifikasi agar data dapat ditambahkan dari keyboard!



```
3 public class SLLMain16 {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.println("Masukkan kelas: ");
7         String kelas = sc.nextLine();
8         System.out.print(s:"Masukkan IPK: ");
9         double ip = Double.parseDouble(sc.nextLine());
10        Mahasiswa16 mhs = new Mahasiswa16(nim, nama, kelas, ip);
11        switch (menu) {
12            case 1:
13                sll.addFirst(mhs);
14                break;
15            case 2:
16                sll.addLast(mhs);
17                break;
18            case 3:
19                System.out.print(s:"Masukkan Nama Setelah: ");
20                String namaAfter = sc.nextLine();
21                sll.insertAfter(namaAfter, mhs);
22                break;
23            case 4:
24                System.out.print(s:"Masukkan Indeks: ");
25                int index = Integer.parseInt(sc.nextLine());
26                sll.insertAt(index, mhs);
27                break;
28            default:
29                System.out.println(x:"Menu tidak valid.");
30        }
31        while (lanjut);
32    }
33 }
```

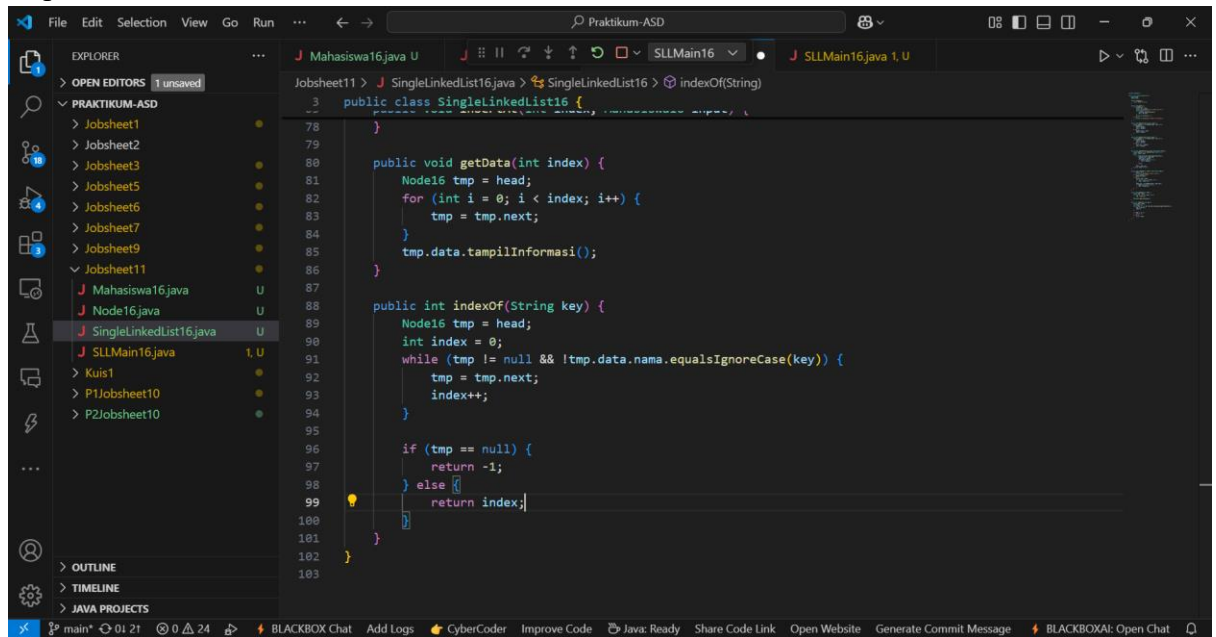
Modifikasi Elemen pada Single Linked List

1. Implementasikan method untuk mengakses data dan indeks pada linked list
2. Tambahkan method untuk mendapatkan data pada indeks tertentu pada class Single Linked List



```
3 public class SingleLinkedList16 {
4     public void insertAt(int index, Mahasiswa16 input) {
5         // ... (code for insertAt) ...
6     }
7
8     public void getData(int index) {
9         Node16 tmp = head;
10        for (int i = 0; i < index; i++) {
11            tmp = tmp.next;
12        }
13        tmp.data.tampilInformasi();
14    }
15 }
```

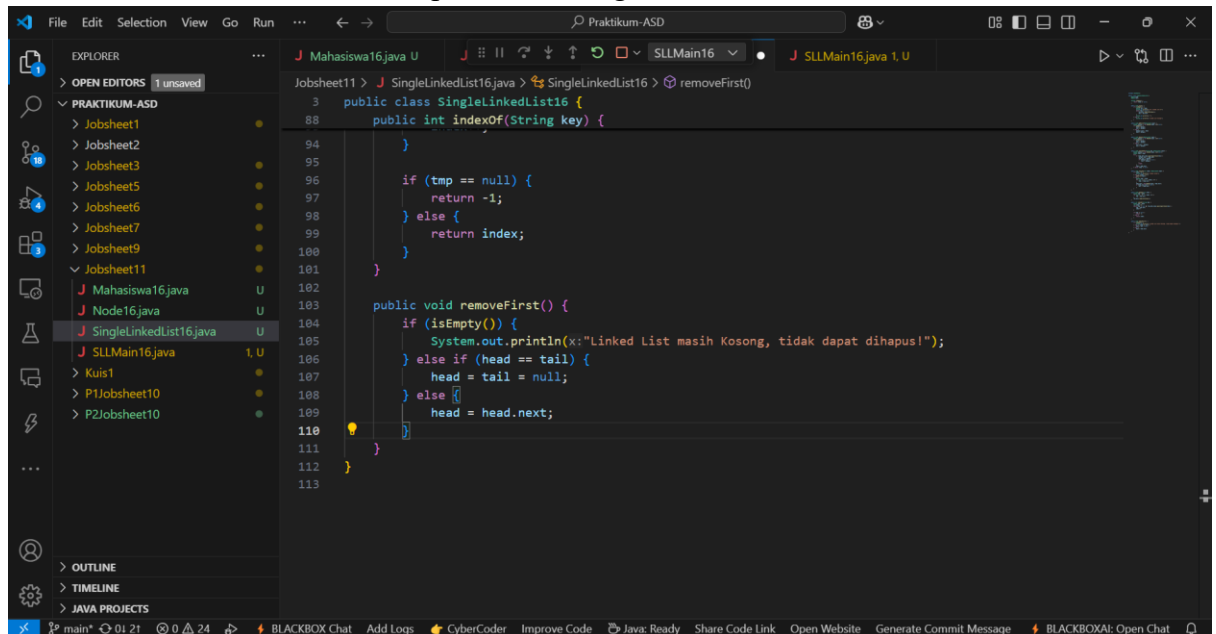
3. Implementasikan method indexOf.



The screenshot shows the VS Code editor with the file `SingleLinkedList16.java` open. The `indexOf` method is implemented as follows:

```
public class SingleLinkedList16 {  
    // ...  
    public void getData(int index) {  
        Node16 tmp = head;  
        for (int i = 0; i < index; i++) {  
            tmp = tmp.next;  
        }  
        tmp.data.tampilInformasi();  
    }  
  
    public int indexOf(String key) {  
        Node16 tmp = head;  
        int index = 0;  
        while (tmp != null && !tmp.data.nama.equalsIgnoreCase(key)) {  
            tmp = tmp.next;  
            index++;  
        }  
  
        if (tmp == null) {  
            return -1;  
        } else {  
            return index;  
        }  
    }  
}
```

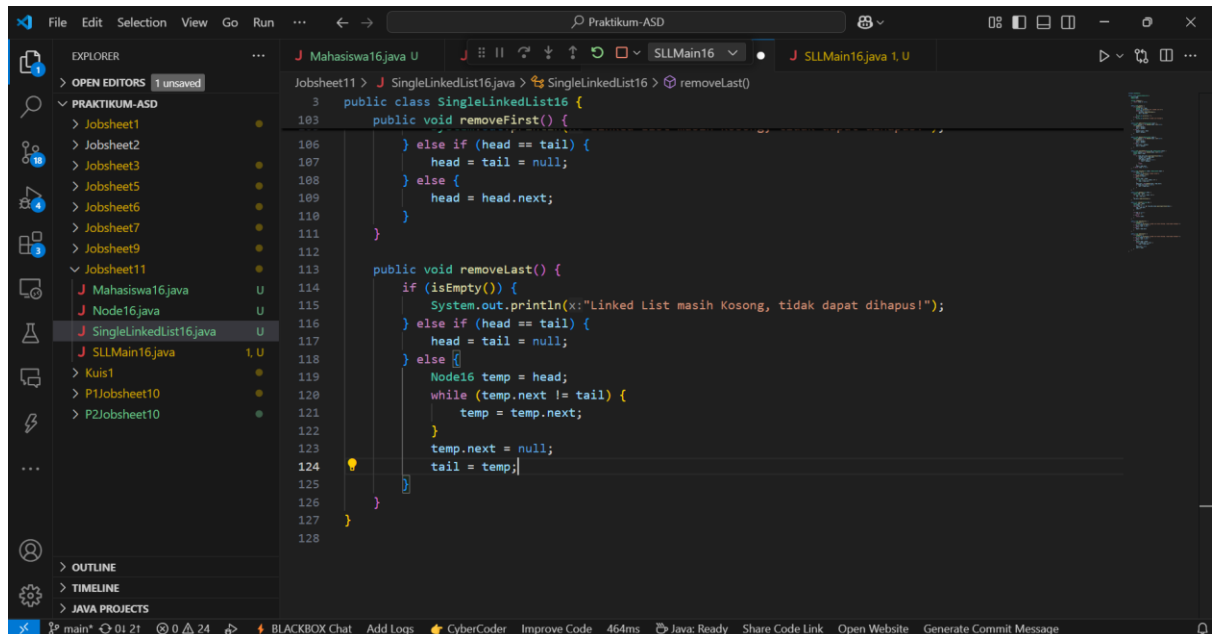
4. Tambahkan method removeFirst pada class SingleLinkedList



The screenshot shows the VS Code editor with the file `SingleLinkedList16.java` open. The `removeFirst` method is implemented as follows:

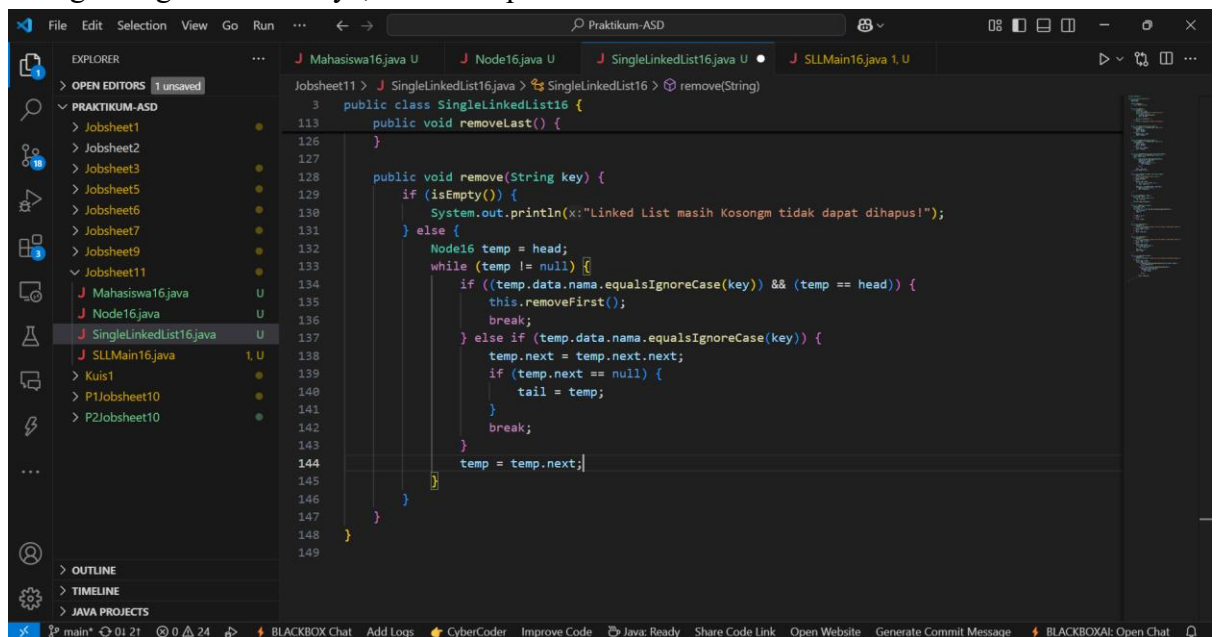
```
public class SingleLinkedList16 {  
    // ...  
    public int indexOf(String key) {  
        // ...  
    }  
  
    public void removeFirst() {  
        if (isEmpty()) {  
            System.out.println("Linked List masih Kosong, tidak dapat dihapus!");  
        } else if (head == tail) {  
            head = tail = null;  
        } else {  
            head = head.next;  
        }  
    }  
}
```

5. Tambahkan method untuk menghapus data pada bagian belakang pada class SingleLinkedList



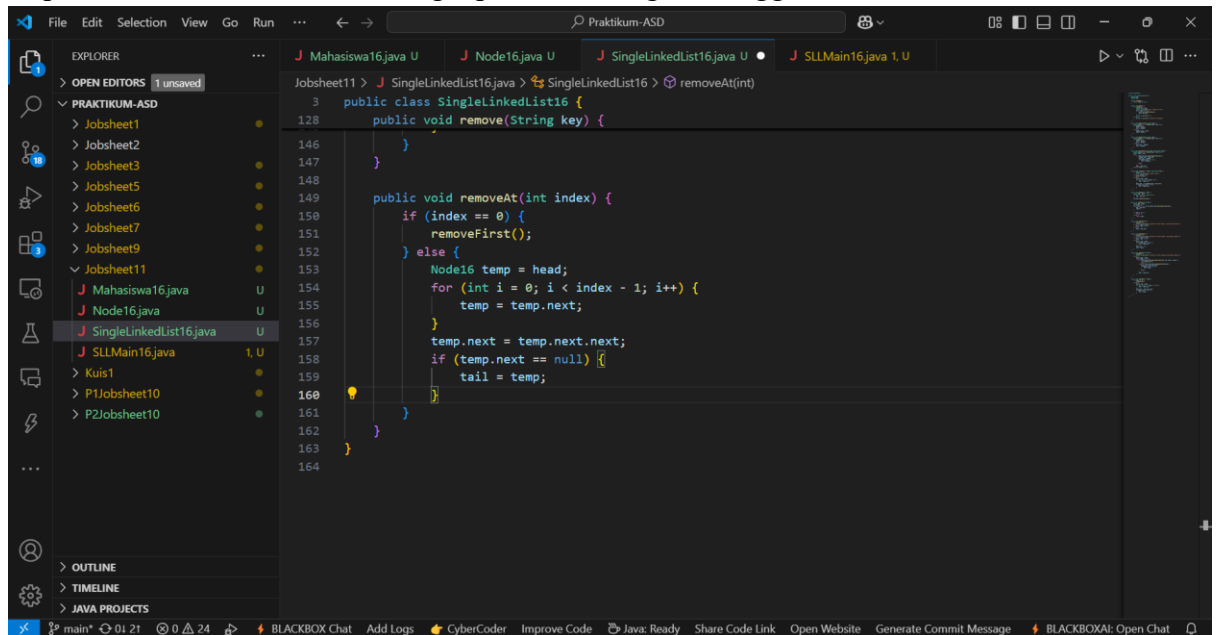
```
public class SingleLinkedList16 {
    public void removeFirst() {
        // ...
    }
    public void removeLast() {
        if (isEmpty()) {
            System.out.println("Linked List masih Kosong, tidak dapat dihapus!");
        } else if (head == tail) {
            head = tail = null;
        } else {
            Node16 temp = head;
            while (temp.next != tail) {
                temp = temp.next;
            }
            temp.next = null;
            tail = temp;
        }
    }
}
```

6. Sebagai langkah berikutnya, akan diimplementasikan method remove



```
public class SingleLinkedList16 {
    public void removeLast() {
        // ...
    }
    public void remove(String key) {
        if (isEmpty()) {
            System.out.println("Linked List masih Kosong, tidak dapat dihapus!");
        } else {
            Node16 temp = head;
            while (temp != null) {
                if ((temp.data.nama.equalsIgnoreCase(key)) && (temp == head)) {
                    this.removeFirst();
                    break;
                } else if (temp.data.nama.equalsIgnoreCase(key)) {
                    temp.next = temp.next.next;
                    if (temp.next == null) {
                        tail = temp;
                    }
                    break;
                }
                temp = temp.next;
            }
        }
    }
}
```

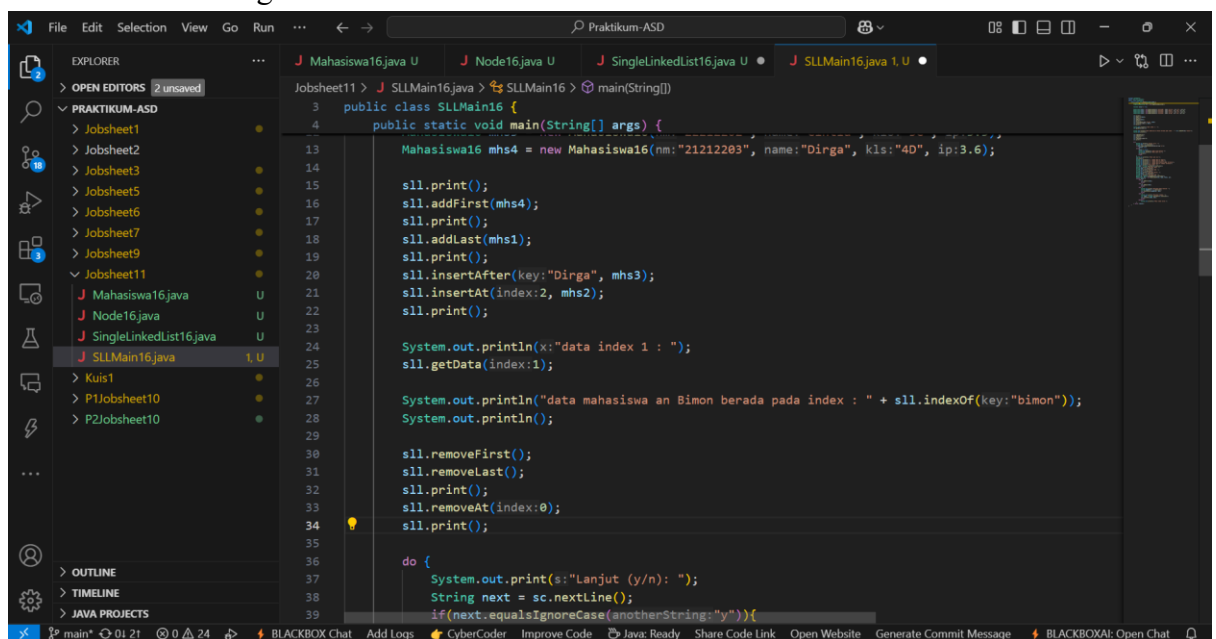
7. Implementasi method untuk menghapus node dengan menggunakan index.



```
File Edit Selection View Go Run ... < -> Praktikum-ASD
EXPLORER
  OPEN EDITORS 1 unsaved
  PRAKTIKUM-ASD
    Jobsheet1
    Jobsheet2
    Jobsheet3
    Jobsheet5
    Jobsheet6
    Jobsheet7
    Jobsheet9
    Jobsheet11
    Mahasiswa16.java
    Node16.java
    SingleLinkedList16.java
    SLLMain16.java
    Kuis1
    P1Jobsheet10
    P2Jobsheet10
  OUTLINE
  TIMELINE
  JAVA PROJECTS

Jobsheet11 > J SingleLinkedList16.java > SingleLinkedList16 > removeAt(int)
3 public class SingleLinkedList16 {
128     public void remove(String key) {
146     }
147 }
148
149     public void removeAt(int index) {
150         if (index == 0) {
151             removeFirst();
152         } else {
153             Node16 temp = head;
154             for (int i = 0; i < index - 1; i++) {
155                 temp = temp.next;
156             }
157             temp.next = temp.next.next;
158             if (temp.next == null) {
159                 tail = temp;
160             }
161         }
162     }
163 }
164
```

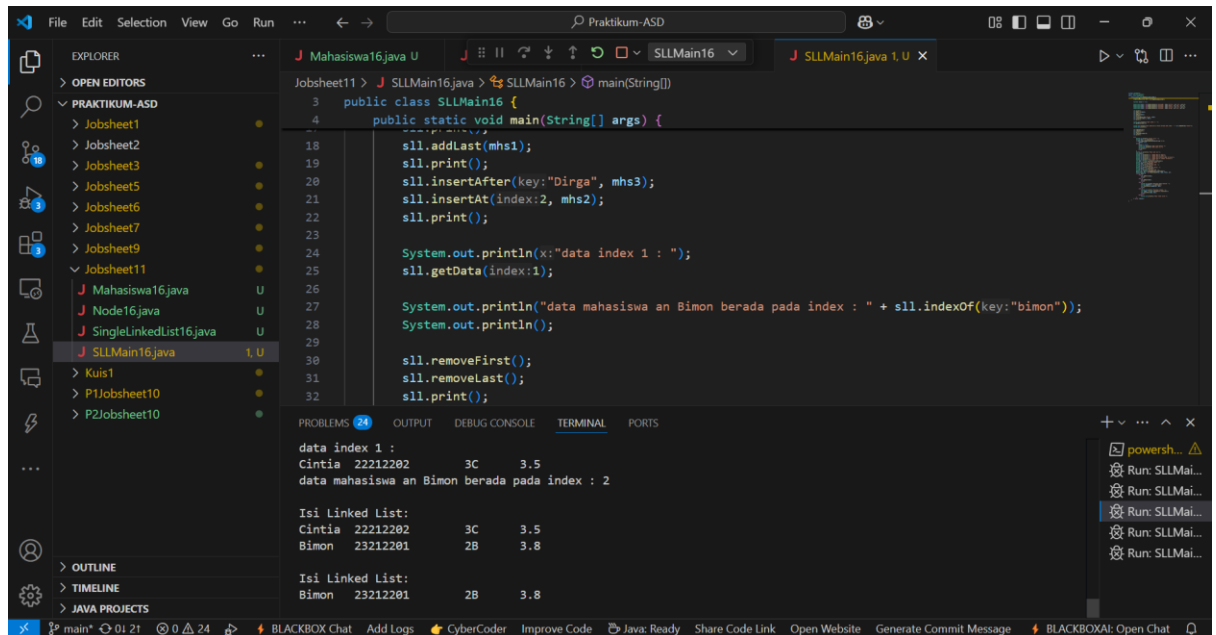
8. Kemudian, coba lakukan pengaksesan dan penghapusan data di method main pada class SLLMain dengan menambahkan kode berikut



```
File Edit Selection View Go Run ... < -> Praktikum-ASD
EXPLORER
  OPEN EDITORS 2 unsaved
  PRAKTIKUM-ASD
    Jobsheet1
    Jobsheet2
    Jobsheet3
    Jobsheet5
    Jobsheet6
    Jobsheet7
    Jobsheet9
    Jobsheet11
    Mahasiswa16.java
    Node16.java
    SingleLinkedList16.java
    SLLMain16.java
    Kuis1
    P1Jobsheet10
    P2Jobsheet10
  OUTLINE
  TIMELINE
  JAVA PROJECTS

Jobsheet11 > J SLLMain16.java > SLLMain16 > main(String[])
3 public class SLLMain16 {
4     public static void main(String[] args) {
13         Mahasiswa16 mhs4 = new Mahasiswa16(nm:"21212203", name:"Dirga", kls:"4D", ip:3.6);
14
15         sll.print();
16         sll.addFirst(mhs4);
17         sll.print();
18         sll.addLast(mhs1);
19         sll.print();
20         sll.insertAfter(key:"Dirga", mhs3);
21         sll.insertAt(index:2, mhs2);
22         sll.print();
23
24         System.out.println(x:"data index 1 : ");
25         sll.getData(index:1);
26
27         System.out.println("data mahasiswa an Bimon berada pada index : " + sll.indexOf(key:"bimon"));
28         System.out.println();
29
30         sll.removeFirst();
31         sll.removeLast();
32         sll.print();
33         sll.removeAt(index:0);
34         sll.print();
35
36         do {
37             System.out.print(s:"Lanjut (y/n) : ");
38             String next = sc.nextLine();
39             if (next.equalsIgnoreCase("y")) {
```

9. Jalankan class SLLMain



Pertanyaan

1. Mengapa digunakan keyword break pada fungsi remove? Jelaskan!
 - break digunakan supaya proses berhenti setelah node yang dicari berhasil dihapus dan supaya tidak menyebabkan penghapusan lebih dari satu node dengan nama yang sama
2. Jelaskan kegunaan kode dibawah pada method remove

```
temp.next = temp.next.next;
if (temp.next == null) {
    tail = temp;
}
```

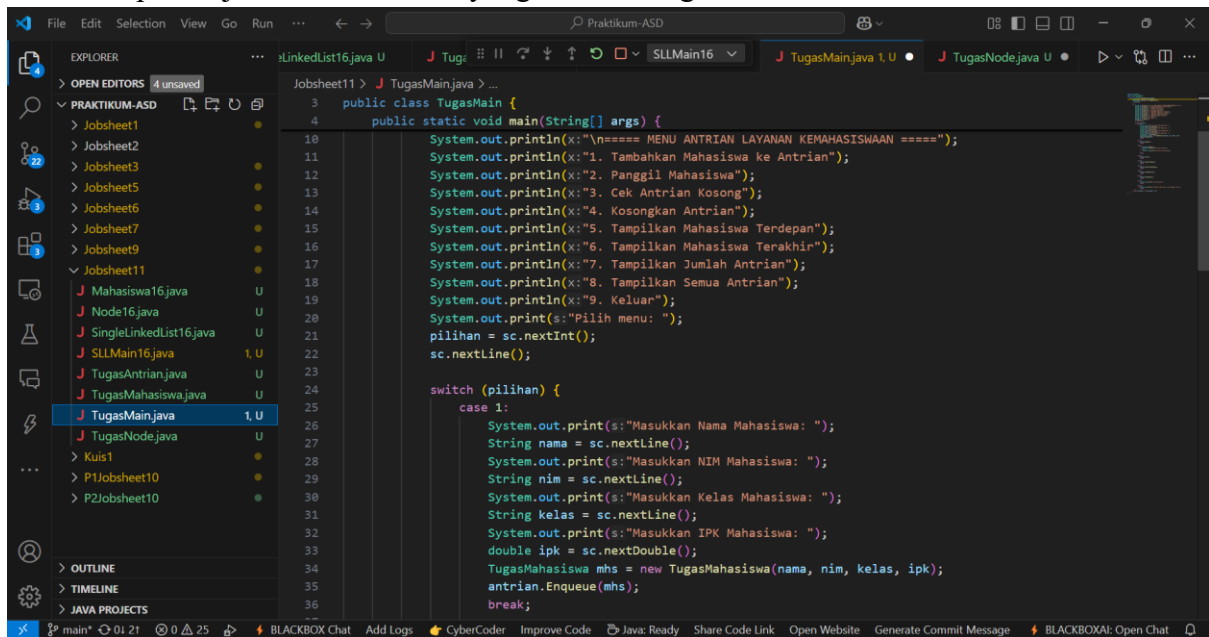
- Kegunaan dari `temp.next = temp.next.next;` sendiri adalah untuk menghapus node setelah temp karena temp.next sendiri dilewati dan di Ganti dengan temp.next.next
- Untuk kondisi `temp.next == null` untuk mengecek apakah temp.next yang telah di Ganti apakah kosong dan jika benar maka akan memperbarui tail

Tugas

Buatlah implementasi program antrian layanan unit kemahasiswaan sesuai dengan berikut ini :

- a. Implementasi antrian menggunakan Queue berbasis Linked List!
- b. Program merupakan proyek baru bukan modifikasi dari percobaan

- c. Ketika seorang mahasiswa akan mengantri, maka dia harus mendaftarkan datanya
- d. Cek antrian kosong, Cek antrian penuh, Mengosongkan antrian.
- e. Menambahkan antrian
- f. Memanggil antrian
- g. Menampilkan antrian terdepan dan antrian paling akhir
- h. Menampilkan jumlah mahasiswa yang masih mengantre.



The screenshot shows an IDE with a project named 'PRAKTIKUM-ASD'. The Explorer panel on the left lists several files, including 'TugasMain.java' which is currently selected. The main editor displays the code for 'TugasMain.java'. The code defines a 'TugasMain' class with a 'main' method. The 'main' method prints a menu with options 1 through 9. Option 1 is 'Tambahkan Mahasiswa ke Antrian', which is implemented in a 'switch' statement. The 'switch' statement for case 1 prompts the user to enter the student's name, NIM, class, and IPK, then creates a 'TugasMahasiswa' object and enqueues it. The code also includes other menu options like 'Cek Antrian Kosong', 'Kosongkan Antrian', 'Tampilkan Mahasiswa Terdepan', 'Tampilkan Mahasiswa Terakhir', 'Tampilkan Jumlah Antrian', and 'Tampilkan Semua Antrian'.

```
3 public class TugasMain {
4     public static void main(String[] args) {
10        System.out.println(x:"\n==== MENU ANTRIAN LAYANAN KEMAHASISWAAN =====");
11        System.out.println(x:"1. Tambahkan Mahasiswa ke Antrian");
12        System.out.println(x:"2. Panggil Mahasiswa");
13        System.out.println(x:"3. Cek Antrian Kosong");
14        System.out.println(x:"4. Kosongkan Antrian");
15        System.out.println(x:"5. Tampilkan Mahasiswa Terdepan");
16        System.out.println(x:"6. Tampilkan Mahasiswa Terakhir");
17        System.out.println(x:"7. Tampilkan Jumlah Antrian");
18        System.out.println(x:"8. Tampilkan Semua Antrian");
19        System.out.println(x:"9. Keluar");
20        System.out.print(s:"Pilih menu: ");
21        pilihan = sc.nextInt();
22        sc.nextLine();
23
24        switch (pilihan) {
25            case 1:
26                System.out.print(s:"Masukkan Nama Mahasiswa: ");
27                String nama = sc.nextLine();
28                System.out.print(s:"Masukkan NIM Mahasiswa: ");
29                String nim = sc.nextLine();
30                System.out.print(s:"Masukkan Kelas Mahasiswa: ");
31                String kelas = sc.nextLine();
32                System.out.print(s:"Masukkan IPK Mahasiswa: ");
33                double ipk = sc.nextDouble();
34                TugasMahasiswa mhs = new TugasMahasiswa(nama, nim, kelas, ipk);
35                antrian.Enqueue(mhs);
36                break;
--
```