Nama : Roy Martin Silaban
Npm : 233040130
Kelas : IF- A

Latihan

```java
package Latihan;

public class Matakuliah {

    private String kode;
    private String nama;
    private int sks;

    public Matakuliah(String kode, String nama, int sks) {
        this.kode = kode;
        this.nama = nama;
        this.sks = sks;
    }

    // Setter & Getter (mengikuti konvensi Java)
    public void setKode(String kode) {
        this.kode = kode;
    }

    public String getKode() {
        return kode;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getNama() {
        return nama;
    }

    public void setSks(int sks) {
        this.sks = sks;
    }

    public int getSks() {
        return sks;
    }

    // ============================
    // Node Linked List & Operasi List
    // ============================

    private static Node HEAD = null;

    // Static inner class Node agar bisa dipakai di method static
    public static class Node {
        private Matakuliah data;
        private Node next;

```

```java

    // ==========================================
    // Node Linked List & Operasi List
    // ==========================================

    private static Node HEAD = null;

    // Static inner class Node agar bisa dipakai di method static
    public static class Node {
        private Matakuliah data;
        private Node next;

        public Node(Matakuliah data) {
            this.data = data;
        }

        public Matakuliah getData() {
            return data;
        }

        public void setData(Matakuliah data) {
            this.data = data;
        }

        public Node getNext() {
            return next;
        }

        public void setNext(Node next) {
            this.next = next;
        }
    }

    // Cek apakah list kosong
    public static boolean isEmpty() {
        return HEAD == null;
    }

    // Latihan 2: Tambah node di depan
    public static void addHead(Matakuliah data) {
        Node newNode = new Node(data);
        if (isEmpty()) {
            HEAD = newNode;
        } else {
            newNode.setNext(HEAD);
            HEAD = newNode;
        }
    }

```

```java

        public void setNext(Node next) {
            this.next = next;
        }
    }

    // Cek apakah list kosong
    public static boolean isEmpty() {
        return HEAD == null;
    }

    // Latihan 2: Tambah node di depan
    public static void addHead(Matakuliah data) {
        Node newNode = new Node(data);
        if (isEmpty()) {
            HEAD = newNode;
        } else {
            newNode.setNext(HEAD);
            HEAD = newNode;
        }
    }

    // Latihan 3: Tampilkan semua elemen
    public static void displayElement() {
        if (isEmpty()) {
            System.out.println("List Kosong");
        } else {
            Node curNode = HEAD;
            while (curNode != null) {
                Matakuliah mk = curNode.getData();
                System.out.println(mk.getKode() + " - " + mk.getNama() + " - " + mk.getSks() + "
                curNode = curNode.getNext();
            }
        }
    }

    // Untuk uji coba
    public static void main(String[] args) {
        Matakuliah m1 = new Matakuliah("IF101", "Dasar Pemrograman", 3);
        Matakuliah m2 = new Matakuliah("IF102", "Struktur Data", 3);

        addHead(m1);
        addHead(m2);

        displayElement();
    }
}


```

Hasil Outputnya

```
66
67          public void setNext(Node next) {
68              this.next = next;
69          }
70      }
71
72      // Cek apakah list kosong
73      public static boolean isEmpty() {
74          return HEAD == null;
75      }
76
77      // Latihan 2: Tambah node di depan
78      public static void addHead(Matakuliah data) {
79          Node newNode = new Node(data);
80          if (isEmpty()) {
81              HEAD = newNode;
82          } else {
83              newNode.setNext(HEAD);
84              HEAD = newNode;
85          }
86      }
87
88      // Latihan 3: Tampilkan semua elemen
89      public static void displayElement() {
90          if (isEmpty()) {
91              System.out.println("List Kosong");
92          } else {
93              Node curNode = HEAD;
94              while (curNode != null) {
95                  Matakuliah mk = curNode.getData();
96                  System.out.println(mk.getKode() + " - " + mk.getNama() + " - " + mk.getSks() + " SK
97                  curNode = curNode.getNext();
```

```
Console
<terminated> Matakuliah [Java Application] /Users/dhiaulhaqlaturua/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.11.v20240426-1830/jre
IF102 - Struktur Data - 3 SKS
IF101 - Dasar Pemrograman - 3 SKS
```

Tugas

```java
package TUGAS;

public class Matakuliah {

      private String kode;
        private String nama;
        private int sks;

        public Matakuliah(String kode, String nama, int sks) {
            this.kode = kode;
            this.nama = nama;
            this.sks = sks;
        }

        public void setKode(String kode) {
            this.kode = kode;
        }

        public String getKode() {
            return kode;
        }

        public void setNama(String nama) {
            this.nama = nama;
        }

        public String getNama() {
            return nama;
        }

        public void setSks(int sks) {
            this.sks = sks;
        }

        public int getSks() {
            return sks;
        }

        // ================================
        // Inner class Node dan LinkedList
        // ================================

        private static Node HEAD = null;

        public static class Node {
            private Matakuliah data;
            private Node next;

            public Node(Matakuliah data) {
                this.data = data;
            }

            public Matakuliah getData() {
                return data;
            }

            public void setData(Matakuliah data) {
                this.data = data;
            }

            public Node getNext() {
                return next;
            }

            public void setNext(Node next) {
                this.next = next;
            }
        }

        public static boolean isEmpty() {
            return HEAD == null;
        }

        // Tes-1 & Tes-2: Tambah di depan (addHead)
        public static void addHead(Matakuliah data) {
            Node newNode = new Node(data);
            if (isEmpty()) {
                HEAD = newNode;
            } else {
                newNode.setNext(HEAD);
                HEAD = newNode;
            }
        }
```

```java
        // Tes-2: Tambah di akhir (addTail)
        public static void addTail(Matakuliah data) {
            Node newNode = new Node(data);
            if (isEmpty()) {
                HEAD = newNode;
            } else {
                Node cur = HEAD;
                while (cur.getNext() != null) {
                    cur = cur.getNext();
                }
                cur.setNext(newNode);
            }
        }

        // Tes-3: Tambah di tengah setelah kode tertentu (misalnya setelah IF002)
        public static void addMid(Matakuliah data, String afterKode) {
            Node newNode = new Node(data);
            if (isEmpty()) {
                HEAD = newNode;
            } else {
                Node cur = HEAD;
                while (cur != null && !cur.getData().getKode().equals(afterKode)) {
                    cur = cur.getNext();
                }
                if (cur != null) {
                    newNode.setNext(cur.getNext());
                    cur.setNext(newNode);
                } else {
                    System.out.println("Node dengan kode " + afterKode + " tidak ditemukan.");
                }
            }
        }

        // Tes-1: Menampilkan semua data
        public static void displayElement() {
            if (isEmpty()) {
                System.out.println("List Kosong");
            } else {
                Node curNode = HEAD;
                while (curNode != null) {
                    Matakuliah mk = curNode.getData();
                    System.out.println("Matakuliah: " + mk.getKode() + ", " + mk.getNama() + ", " + mk.getSks());
                    curNode = curNode.getNext();
                }
            }
        }
    public static void addMid(Matakuliah data, String afterKode) {
        Node newNode = new Node(data);
        if (isEmpty()) {
            HEAD = newNode;
        } else {
            Node cur = HEAD;
            while (cur != null && !cur.getData().getKode().equals(afterKode)) {
                cur = cur.getNext();
            }
            if (cur != null) {
                newNode.setNext(cur.getNext());
                cur.setNext(newNode);
            } else {
                System.out.println("Node dengan kode " + afterKode + " tidak ditemukan.");
            }
        }
    }

    // Tes-1: Menampilkan semua data
    public static void displayElement() {
        if (isEmpty()) {
            System.out.println("List Kosong");
        } else {
            Node curNode = HEAD;
            while (curNode != null) {
                Matakuliah mk = curNode.getData();
                System.out.println("Matakuliah: " + mk.getKode() + ", " + mk.getNama() + ", " + mk.getSks());
                curNode = curNode.getNext();
            }
        }
    }

    // Tes-4: Uji coba addHead, addTail, addMid
    public static void main(String[] args) {
        Matakuliah mk1 = new Matakuliah("IF001", "Dasar Pemrograman", 4);
        Matakuliah mk2 = new Matakuliah("IF002", "Pemrograman Web", 3);
        Matakuliah mk3 = new Matakuliah("IF003", "Struktur Diskrit", 3);
        Matakuliah mk4 = new Matakuliah("IF004", "Konstruksi PL Berorientasi Objek", 3);

        addHead(mk1);               // IF001
        addTail(mk2);               // IF002
        addTail(mk3);               // IF003
        addHead(mk4);               // IF004 (paling atas)

        displayElement();           // Output sesuai Tes-1
    }
}
```

Console

```
<terminated> Matakuliah (1) [Java Application] /Users/dhiaulhaqlaturua/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.11.v20240426-1830
Matakuliah: IF004, Konstruksi PL Berorientasi Objek, 3
Matakuliah: IF001, Dasar Pemrograman, 4
Matakuliah: IF002, Pemrograman Web, 3
Matakuliah: IF003, Struktur Diskrit, 3
```