

Nama : Muhammad Dhiaulhaq Triyudhistira Laturua

Npm : 233040171

Kelas : IF- A

Latihan

```
1 package Latihan;
2
3 public class Matakuliah {
4
5     private String kode;
6     private String nama;
7     private int sks;
8
9     public Matakuliah(String kode, String nama, int sks) {
10         this.kode = kode;
11         this.nama = nama;
12         this.sks = sks;
13     }
14
15     // Setter & Getter (menaikuti konvensi Java)
16     public void setKode(String kode) {
17         this.kode = kode;
18     }
19
20     public String getKode() {
21         return kode;
22     }
23
24     public void setNama(String nama) {
25         this.nama = nama;
26     }
27
28     public String getNama() {
29         return nama;
30     }
31
32     public void setSks(int sks) {
33         this.sks = sks;
34     }
35
36     public int getSks() {
37         return sks;
38     }
39
40     // =====
41     // Node Linked List & Operasi List
42     // =====
43
44     private static Node HEAD = null;
45
46     // Static inner class Node agar bisa dipakai di method static
47     public static class Node {
48         private Matakuliah data;
49         private Node next;
50     }
```

```
39 // =====
40 // Node Linked List & Operasi List
41 // =====
42
43 private static Node HEAD = null;
44
45 // Static inner class Node agar bisa dipakai di method static
46 public static class Node {
47     private Matakuliah data;
48     private Node next;
49
50     public Node(Matakuliah data) {
51         this.data = data;
52     }
53
54     public Matakuliah getData() {
55         return data;
56     }
57
58     public void setData(Matakuliah data) {
59         this.data = data;
60     }
61
62     public Node getNext() {
63         return next;
64     }
65
66     public void setNext(Node next) {
67         this.next = next;
68     }
69 }
70
71 // Cek apakah list kosong
72 public static boolean isEmpty() {
73     return HEAD == null;
74 }
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
```

```
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());
97             curNode = curNode.getNext();
98         }
99     }
100 }
101
102 // Untuk uji coba
103 public static void main(String[] args) {
104     Matakuliah m1 = new Matakuliah("IF101", "Dasar Pemrograman", 3);
105     Matakuliah m2 = new Matakuliah("IF102", "Struktur Data", 3);
106
107     addHead(m1);
108     addHead(m2);
109
110     displayElement();
111 }
112 }
113
114 }
```

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();
98         }
99     }
100 }
```

Console X

```
<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

```
1 package TUGAS;
2
3 public class Matakuliah {
4
5     private String kode;
6     private String nama;
7     private int sks;
8
9     public Matakuliah(String kode, String nama, int sks) {
10         this.kode = kode;
11         this.nama = nama;
12         this.sks = sks;
13     }
14
15     public void setKode(String kode) {
16         this.kode = kode;
17     }
18
19     public String getKode() {
20         return kode;
21     }
22
23     public void setNama(String nama) {
24         this.nama = nama;
25     }
26
27     public String getNama() {
28         return nama;
29     }
30
31     public void setSks(int sks) {
32         this.sks = sks;
33     }
34
35     public int getSks() {
36         return sks;
37     }
38
39     // =====
40     // Inner class Node dan LinkedList
41     // =====
42
43     private static Node HEAD = null;
44
45     public static class Node {
46         private Matakuliah data;
47         private Node next;
48
49         public Node(Matakuliah data) {
50             this.data = data;
51         }
52
53         public Matakuliah getData() {
54             return data;
55         }
56
57         public void setData(Matakuliah data) {
58             this.data = data;
59         }
60
61         public Node getNext() {
62             return next;
63         }
64
65         public void setNext(Node next) {
66             this.next = next;
67         }
68     }
69
70     public static boolean isEmpty() {
71         return HEAD == null;
72     }
73
74     // Tes-1 & Tes-2: Tambah di depan (addHead)
75     public static void addHead(Matakuliah data) {
76         Node newNode = new Node(data);
77         if (isEmpty()) {
78             HEAD = newNode;
79         } else {
80             newNode.setNext(HEAD);
81             HEAD = newNode;
82         }
83     }
84 }
```

```

64      // Tes-2: Tambah di akhir (addTail)
65      public static void addTail(Matakuliah data) {
66          Node newNode = new Node(data);
67          if (isEmpty()) {
68              HEAD = newNode;
69          } else {
70              Node cur = HEAD;
71              while (cur.getNext() != null) {
72                  cur = cur.getNext();
73              }
74              cur.setNext(newNode);
75          }
76      }
77
78
79      // Tes-3: Tambah di tengah setelah kode tertentu (misalnya setelah IF002)
80      public static void addMid(Matakuliah data, String afterKode) {
81          Node newNode = new Node(data);
82          if (isEmpty()) {
83              HEAD = newNode;
84          } else {
85              Node cur = HEAD;
86              while (cur != null && !cur.getData().getKode().equals(afterKode)) {
87                  cur = cur.getNext();
88              }
89              if (cur != null) {
90                  newNode.setNext(cur.getNext());
91                  cur.setNext(newNode);
92              } else {
93                  System.out.println("Node dengan kode " + afterKode + " tidak ditemukan.");
94              }
95          }
96      }
97
98
99      // Tes-1: Menampilkan semua data
100     public static void displayElement() {
101         if (isEmpty()) {
102             System.out.println("List Kosong");
103         } else {
104             Node curNode = HEAD;
105             while (curNode != null) {
106                 Matakuliah mk = curNode.getData();
107                 System.out.println("Matakuliah: " + mk.getKode() + ", " + mk.getNama() + ", " + mk.getSkS());
108                 curNode = curNode.getNext();
109             }
110         }
111     }
112
113
114     public static void addMid(Matakuliah data, String afterKode) {
115         Node newNode = new Node(data);
116         if (isEmpty()) {
117             HEAD = newNode;
118         } else {
119             Node cur = HEAD;
120             while (cur != null && !cur.getData().getKode().equals(afterKode)) {
121                 cur = cur.getNext();
122             }
123             if (cur != null) {
124                 newNode.setNext(cur.getNext());
125                 cur.setNext(newNode);
126             } else {
127                 System.out.println("Node dengan kode " + afterKode + " tidak ditemukan.");
128             }
129         }
130     }
131
132
133     // Tes-4: Uji coba addHead, addTail, addMid
134     public static void main(String[] args) {
135         Matakuliah mk1 = new Matakuliah("IF001", "Dasar Pemrograman", 4);
136         Matakuliah mk2 = new Matakuliah("IF002", "Pemrograman Web", 3);
137         Matakuliah mk3 = new Matakuliah("IF003", "Struktur Diskrit", 3);
138         Matakuliah mk4 = new Matakuliah("IF004", "Konstruksi PL Berorientasi Objek", 3);
139
140         addHead(mk1);           // IF001
141         addTail(mk2);          // IF002
142         addTail(mk3);          // IF003
143         addHead(mk4);          // IF004 (paling atas)
144
145         displayElement();       // Output sesuai Tes-1
146     }

```

Console X

```

<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

```