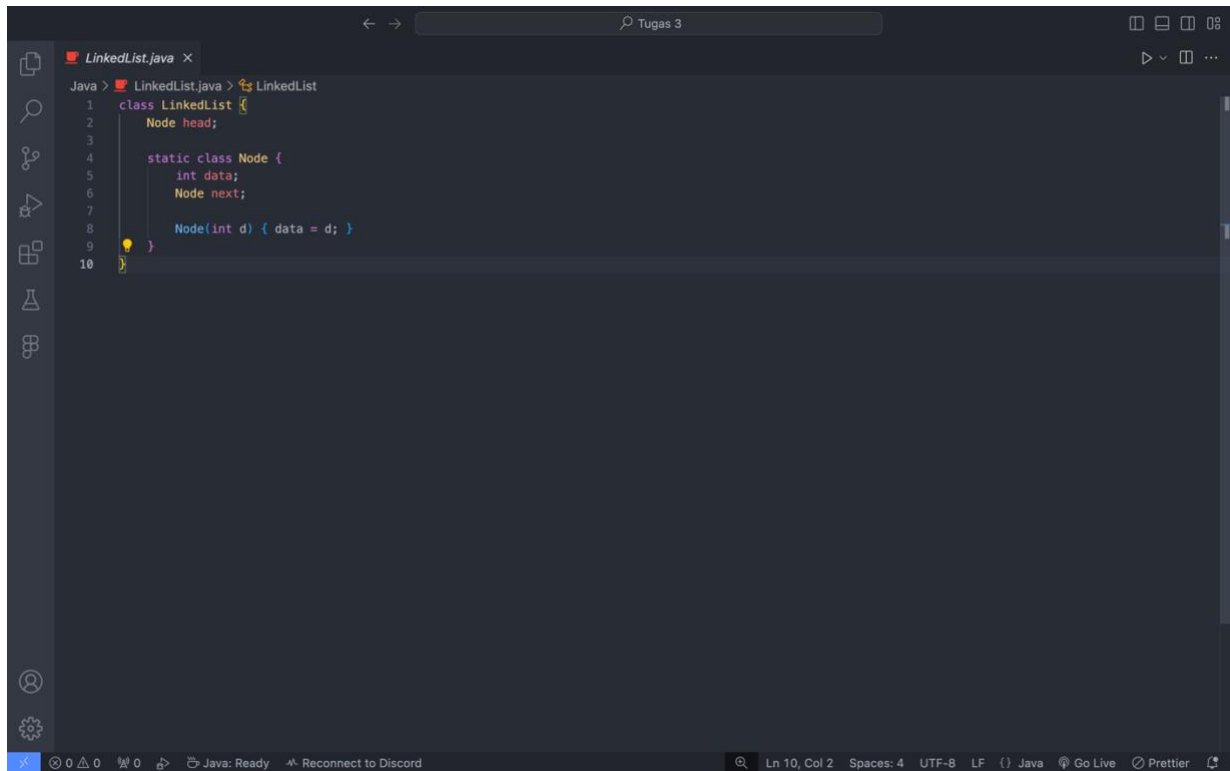


Linked List dalam Bahasa Pemrograman Java

Perancangan Struktur Data

Nama : Faturrohman Fairuz Zaki
NIM : 22106050012
Kelas : Informatika C



The screenshot shows a code editor with a dark theme. The file name is 'LinkedList.java'. The code defines a 'LinkedList' class with a 'Node' class as a static inner class. The 'Node' class has an 'int data' field and a 'Node next' field. The 'LinkedList' class has a 'Node head' field. A constructor for 'Node' is provided, taking an integer 'd' and setting 'data = d;'. The code is as follows:

```
1 class LinkedList {  
2     Node head;  
3  
4     static class Node {  
5         int data;  
6         Node next;  
7  
8         Node(int d) { data = d; }  
9     }  
10 }
```

The status bar at the bottom shows 'Ln 10, Col 2', 'Spaces: 4', 'UTF-8', 'LF', 'Java', 'Go Live', and 'Prettier'.

Membuat Class Linked List dalam Bahasa Java

```
ImplementasiLinkedList.java x
Java > ImplementasiLinkedList.java > ImplementasiLinkedList > Node
1 public class ImplementasiLinkedList {
2     Node head; // head of list
3
4     // Linked list Node.
5     // This inner class is made static
6     // so that main() can access it
7     static class Node {
8
9         int data;
10        Node next;
11
12        // Constructor
13        Node(int d)
14        {
15            data = d;
16            next = null;
17        }
18    }
19
20    // Method to insert a new node
21    public static ImplementasiLinkedList insert(ImplementasiLinkedList list, int data)
22    {
23        // Create a new node with given data
24        Node new_node = new Node(data);
25
26
27        // If the Linked List is empty,
28        // then make the new node as head
29        if (list.head == null) {
30            list.head = new_node;
31        }
32        else {
33            // Else traverse till the last node
34            // and insert the new node there
35            Node last = list.head;
36            while (last.next != null) {
37                last = last.next;
38            }
39        }
40    }
41}
```

Fatur@Faturrohman-MacBook-Pro Tugas 3 % /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-21.0.2.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/Fatur/Library/Application\ Support/Code/User/workspaceStorage/28ffe2793e022eb40e62a4328bfa3237/redhat.java/jdt_ws/Tugas\ 3_c61cc32b/bin Imp
ImplementasiLinkedList: 1 2 3 4 5 6 7 8

Fatur@Faturrohman-MacBook-Pro Tugas 3 %

Contoh Implementasi Linked List

```
TraversalLinkedList.java x
Java > TraversalLinkedList.java > TraversalLinkedList > Node
1 public class TraversalLinkedList {
2     Node head; // head of list
3
4     // Linked list Node.
5     // Node is a static nested class
6     // so main() can access it
7     static class Node {
8
9         int data;
10        Node next;
11
12        // Constructor
13        Node(int d)
14        {
15            data = d;
16            next = null;
17        }
18    }
19
20    // Method to insert a new node
21    public static TraversalLinkedList insert(TraversalLinkedList list,
22        int data)
23    {
24        // Create a new node with given data
25        Node new_node = new Node(data);
26        new_node.next = null;
27
28        // If the Linked List is empty,
29        // then make the new node as head
30        if (list.head == null) {
31            list.head = new_node;
32        }
33        else {
34            // Else traverse till the last node
35            // and insert the new node there
36            Node last = list.head;
37            while (last.next != null) {
38                last = last.next;
39            }
40        }
41    }
42}
```

Fatur@Faturrohman-MacBook-Pro Tugas 3 % /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-21.0.2.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/Fatur/Library/Application\ Support/Code/User/workspaceStorage/28ffe2793e022eb40e62a4328bfa3237/redhat.java/jdt_ws/Tugas\ 3_c61cc32b/bin TraversalLinkedList
LinkedList: 1 2 3 4 5 6 7 8

Fatur@Faturrohman-MacBook-Pro Tugas 3 %

Linked List Traversal

```
Java > DeleteByKeyLinkedList.java > DeleteByKeyLinkedList > Node
1 public class DeleteByKeyLinkedList {
2     Node head; // head of list
3
4     // Linked List Node.
5     // Node is a static nested class
6     // so main() can access it
7     static class Node {
8
9         int data;
10        Node next;
11
12        // Constructor
13        Node(int d) {
14            data = d;
15            next = null;
16        }
17    }
18
19    // Method to insert a new node
20    public static DeleteByKeyLinkedList insert(DeleteByKeyLinkedList list,
21        int data) {
22        // Create a new node with given data
23        Node new_node = new Node(data);
24        new_node.next = null;
25
26        // If the Linked List is empty,
27        // then make the new node as head
28        if (list.head == null) {
29            list.head = new_node;
30        } else {
31            // Else traverse till the last node
32            // and insert the new_node there
33            Node last = list.head;
34            while (last.next != null) {
35                last = last.next;
36            }
37        }
38    }
39}
```

```
Fatur@Faturrohman-MacBook-Pro Tugas 3 % /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-21.0.2.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/Fatur/Library/Application\ Support/Code/User/workspaceStorage/28ffe2793e022eb40e62a4328bfa3237/redhat.java/jdt_ws/Tugas\ 3_c61cc32b/bin DeleteByKeyLinkedList
LinkedList: 1 2 3 4 5 6 7 8
1 found and deleted
LinkedList: 2 3 4 5 6 7 8
4 found and deleted
LinkedList: 2 3 5 6 7 8
10 not found
LinkedList: 2 3 5 6 7 8
Fatur@Faturrohman-MacBook-Pro Tugas 3 %
```

Menghapus node berdasarkan Key pada Linked List

```
Java > DeletionAtPositionLinkedList.java > DeletionAtPositionLinkedList > Node
1 public class DeletionAtPositionLinkedList {
2     Node head; // head of list
3
4     // Linked List Node.
5     // Node is a static nested class
6     // so main() can access it
7     static class Node {
8
9         int data;
10        Node next;
11
12        // Constructor
13        Node(int d) {
14            data = d;
15            next = null;
16        }
17    }
18
19    // Method to insert a new node
20    public static DeletionAtPositionLinkedList insert(DeletionAtPositionLinkedList list,
21        int data) {
22    {
23        // Create a new node with given data
24        Node new_node = new Node(data);
25        new_node.next = null;
26
27        // If the Linked List is empty,
28        // then make the new node as head
29        if (list.head == null) {
30            list.head = new_node;
31        } else {
32            // Else traverse till the last node
33            // and insert the new_node there
34            Node last = list.head;
35            while (last.next != null) {
36                last = last.next;
37            }
38            last.next = new_node;
39        }
40    }
41}
```

```
Fatur@Faturrohman-MacBook-Pro Tugas 3 % /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-21.0.2.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/Fatur/Library/Application\ Support/Code/User/workspaceStorage/28ffe2793e022eb40e62a4328bfa3237/redhat.java/jdt_ws/Tugas\ 3_c61cc32b/bin DeletionAtPositionLinkedList
LinkedList: 1 2 3 4 5 6 7 8
0 position element deleted
LinkedList: 2 3 4 5 6 7 8
2 position element deleted
LinkedList: 2 3 5 6 7 8
10 position element not found
LinkedList: 2 3 5 6 7 8
Fatur@Faturrohman-MacBook-Pro Tugas 3 %
```

Menghapus node berdasarkan Posisi pada Linked List

```
Java > FinalLinkedList.java > FinalLinkedList > Node > next
1 public class FinalLinkedList {
2     Node head; // head of list
3
4     // Linked List Node.
5     // Node is a static nested class
6     // so main() can access it
7     static class Node {
8
9         int data;
10        Node next;
11
12        // Constructor
13        Node(int d)
14        {
15            data = d;
16            next = null;
17        }
18    }
19
20    // *****INSERTION*****
21
22    // Method to insert a new node
23    public static FinalLinkedList insert(FinalLinkedList list,
24                                         int data)
25    {
26        // Create a new node with given data
27        Node new_node = new Node(data);
28        new_node.next = null;
29
30        // If the Linked List is empty,
31        // then make the new node as head
32        if (list.head == null) {
33            list.head = new_node;
34        }
35        else {
36            // Else traverse till the last node
37            // and insert the new_node there
```

```
Fatur@Faturrohman-MacBook-Pro Tugas 3 % /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-21.0.2.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/Fatur/Library/Application\ Support/Code/User/workspaceStorage/28ffe2793e022eb40e62a4328bfa3237/redhat.java/jdt_ws/Tugas\ 3_c61cc32b/bin FinalLinkedList
FinalLinkedList: 1 2 3 4 5 6 7 8
1 found and deleted
FinalLinkedList: 2 3 4 5 6 7 8
4 found and deleted
FinalLinkedList: 2 3 5 6 7 8
10 not found
FinalLinkedList: 2 3 5 6 7 8
0 position element deleted
FinalLinkedList: 3 5 6 7 8
2 position element deleted
FinalLinkedList: 3 5 7 8
10 position element not found
FinalLinkedList: 3 5 7 8
Fatur@Faturrohman-MacBook-Pro Tugas 3 %
```

Implementasi pembuatan, penambahan, dan penghapusan pada Linked List