

Bab 4 Double Link List Kelas Object

Identitas

Kajian

Double Link List

Topik

1. Double Linked List
2. Double Linked List dalam pengelolaan Kelas Object

Referensi

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms Third Edition. Cambridge: The MIT Press.
2. Sedgewick, R. (2002). Algorithm in Java: parts 1-4 Third Edition. Boston: Pearson Education, Inc.
3. P. Deitel and H. Deitel, Java How To Program 9th Edition, Prentice Hall, 2011.
4. Poo, Danny, Derek Kiong, and Swarnalatha Ashok. Object-Oriented Programming and Java, 2nd Edition. Springer, 2010.
5. Wu, C. Thomas. An Introduction to Object-Oriented Programming with Java. McGraw-Hill, 2009.

Kompetensi Utama

1. Mahasiswa mampu menggunakan bahasa java untuk membangun sebuah aplikasi sederhana yang digunakan untuk mengelola data dalam struktur Double Link list dengan menggunakan kelas object

Lama Kegiatan Kajian

1. Pertemuan Terbimbing : 2 x 500 menit
2. Kegiatan Mandiri : 1 x 70 menit

Parameter Penilaian

1. Jurnal 60%
2. Tugas Akhir 40%

Pengumpulan

Kumpulkan jawaban dari Jurnal tembimbing dan jurnal mandiri pada LMS yang telah disediakan



Jurnal Terbimbing

Lengkapi kode program berikut untuk menyusun sebuah program untuk mengelola data dengan menggunakan Double linked list

```
import java.io.*;
import java.util.Scanner;

class Node {
    int data;
    Node next;
    Node prev;

    Node(int d) {
        data = d;
        next = null;
        prev = null;
    }
}

public class DoubleLinkedList {
    Node head; // head of list
    Node tail; // tail of list
    Scanner inp = new Scanner(System.in);

    /*      Insert last      */
    public void insertEnd(int data) {
        Node nn = new Node(data);
        if (tail == null) {
            head = tail = nn;
        }
        else {
            tail.next = _____;
            nn.prev = _____;
            tail = _____;
        }
        System.out.println("Node baru "+data+" diposisi belakang");
    }

    /*      Insert First      */
    public void insertFirst(int data) {
        Node nn = new Node(data);
        if (head == null) {
            head = tail = nn;
        }
        else {
            head.prev = _____;
            nn.next = _____;
            head = _____;
        }
        System.out.println("Node baru "+data+" diposisi depan");
    }
}
```



```

/*      Insert sorted*/
public void insertSorted(int data) {
    Node newNode = new Node(data);

    // Kasus 1. Jika List empty
    if (head == null) {
        head = tail = _____;
        System.out.println("Node baru "+data+" paling depan");
        return;
    }

    // Kasus 2. Jika new node adalah angka paling kecil
    if (data <= head.data) {
        newNode.next = _____;
        head.prev = _____;
        head = _____;
        System.out.println("Node baru "+data+" paling depan");
        return;
    }

    // Kasus 3: jika new node angka paling besar
    if (data >= tail.data) {
        newNode.prev = tail;
        tail.next = newNode;
        tail = newNode;
        System.out.println("Node baru "+data+" paling belakang");
        return;
    }

    // Kasus 4: diantara head dan tail
    Node current = head;
    while (current != null && current.data < data) {
        current = current.next;
    }

    // Insert the node before the current node
    newNode.next = _____;
    newNode.prev = _____;

    if (current.prev != null) {
        current.prev.next = newNode;
    }
    current.prev = _____;
    System.out.println("Node baru "+data+" di tengah");
}

```



```

// Delete node dengan data tertentu
public void delete(int data) {
    if (head == null) {
        System.out.println("List is empty.");
        return;
    }

    Node current = head;

    // Case 1: Delete the head node
    if (head.data == data) {
        head = head.next;
        if (head != null) {
            head.prev = null;
        } else {
            tail = null; // If the list becomes empty
        }
        return;
    }

    // Traverse to find the node to delete
    while (current != null && current.data != data) {
        current = _____;
    }

    // Case 2: Node not found
    if (current == null) {
        System.out.println("Node " + data + " tidak ditemukan.");
        return;
    }

    // Case 3: Delete a middle node
    if (current.next != null) {
        current.next.prev = current.prev;
    } else {
        tail = current.prev; // jika node yang dihapus adalah tail
    }

    if (current.prev != null) {
        current.prev.next = current.next;
    }
}

public void printListBackward() {
    Node current = head;
    System.out.print("Double LinkedList Backward: ");
    while (current != _____) {
        System.out.print(_____ + " ");
        current = current.next;
    }
    System.out.println("");
}

```



```

public void printListForward() {
    Node current = tail;
    System.out.print("Double LinkedList Forward: ");
    while (_____ != null) {
        System.out.print(_____ + " ");
        current = current.prev;
    }
    System.out.println("");
}

public static void main(String[] args) {
    DoubleLinkedList dll = new DoubleLinkedList();
    dll.runThis();
}

void runThis() {
    //INSERT BIASA
    /*
    insertFirst(1);
    insertEnd(5);
    insertEnd(3);
    insertFirst(2);
    insertFirst(7);
    insertEnd(6);
    insertFirst(9);
    insertEnd(8);
    printListBackward();
    printListForward();

    //INSERT SORTED
    insertSorted(1);
    insertSorted(5);
    insertSorted(3);
    insertSorted(2);
    insertSorted(7);
    insertSorted(6);
    insertSorted(9);
    insertSorted(8);
    printListBackward();
    printListForward();
    */

    delete(6);
    printList();
    delete(1);
    printList();
    delete(9);
}
}

```



Jurnal Mandiri

Modifikasi program diatas, sehingga data yang dikelola bukan dalam bentuk data Integer saja, namun untuk mengelola kelas berikut ini

```
class Barang{
    private int id;
    private String nama;
    private int stok;

    public Barang(int id, String nama, int stok){
        this.id=id;
        this.nama=nama;
        this.stok=stok;
    }

    public int getId(){
        return id;
    }

    public String info(){
        return id+" "+nama+" " +stok;
    }
}
```

