

Bab 5 Stack

Identitas

Kajian

Stack

Topik

1. Stack

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 terbimbing dan jurnal mandiri pada LMS yang telah disediakan



Jurnal Terbimbing

Lengkapi kode program berikut untuk menyusun sebuah program untuk mengelola data dengan menggunakan Stack

```
import java.io.*;
class Node {
    String data;
    Node next;

    Node(String d) {
        data = d;
        next = null;
    }
}

public class StackList {
    Node top; // top of stack
    Scanner inp = new Scanner(System.in);

    public void push(String data) {
        Node newNode = new Node(data);

        if (_____) {
            // Kasus 1. Jika stack empty
            // maka top diisi dengan node baru
            top = _____;
        } else {
            // Kasus 2. jika stack ada node
            // New node diletakkan paling, dan top lama menjadi next
            _____ = top;
            _____ = newNode;
        }
        System.out.println("PUSH "+data);
    }

    public void pop() {
        Node node = null;

        // Kasus 1. Jika stack empty
        if (top == _____) {
            //Jika stack kosong, berikan informasi stack kosong
            _____;
            return;
        } else{
            //Jika stack ada, node baru akan diisi dengan top
            //next dari top akan menjadi top baru
            //untuk membersihkan node baru, set next dengan null
            _____ = _____;
            _____ = _____;
            _____ = null;
        }
        System.out.println("POP "+node.data);
    }
}
```



```

public void delete(String data) {
    if (_____ == null) {
        //jika stack, tampilkan informasi stack kosong
        _____;
        return;
    }

    // Case 1: Delete the top node
    if (_____) {
        //jika data yang dicari sama dengan data top
        if (_____ == null) {
            //jika node selanjutnya dari top adalah null
            top = _____;
        } else {
            top = _____;
        }
        System.out.println("DELETE : " + data);
        return;
    }
    Node current = top;        //data iterasi yang akan dihapus
    Node prev = top;          //node sebelumnya
    // Traverse untuk menemukan node
    while (current != null && current.data != data) {
        prev = _____;
        _____ = current.next;
    }

    // jika node tidak ditemukan, posisi berada paling akhir
    if (current == _____) {
        System.out.println("DELETE : " + data + " NOT FOUND");
        return;
    }

    // jika node ditemukan di pertengahan atau akhir
    if (current.next != _____) {
        //jika node bukan bagian paling akhir
        //gunakan prev untuk menghubungkan node prev dan next node
        prev._____ = _____;
    } else {
        //Jika node paling akhir, set next dari prev dengan null
        _____ = null;
    }
    System.out.println("DELETE : " + data);
    current = null;
}

```



```

public void print() {
    if ( _____ == _____ ) {
        //jika stack kosong, tampilkan stack kosong
        _____;
        return;
    } else {
        Node current = _____;
        System.out.print("Stack : ");
        //Ulang hingga node berada paling akhir
        while (current != _____) {
            System.out.print(current.data + " ");
            current = _____;
        }
        System.out.println("");
    }
}

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

void runThis(){
    push("hari");
    push("ini");
    push("kita");
    push("mempelajari");
    push("materi");
    push("struktur");
    push("data");
    push("tentang");
    push("stack");
    print();
    pop();
    print();
    pop();
    print();
    delete("struktur");
    print();
    delete("kita");
    print();
    delete("stack");
    print();
}
}

```



Jurnal Mandiri

Modifikasi program diatas, sehingga data yang dikelola bukan dalam bentuk data String 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;
    }
}
```

