

MODUL 2

STACK AND QUEUE

1.1 PERMASALAHAN

1.3.1 List Buku

Suatu hari, seorang programmer bernama DIvo memiliki sebuah misi untuk mengelola antrian buku dengan efisien. Ia ingin membuat sistem antrian untuk beberapa karakter yang memiliki preferensi berbeda dalam membaca buku. Untuk itu, DIvo mendefinisikan kelas Antrian, yang menyimpan nama karakter, posisi antrian, dan koleksi buku yang mereka miliki dalam bentuk tumpukan menggunakan struktur data Stack. DIvo menyadari bahwa setiap karakter dapat memiliki kartu spesial yang memberikan mereka keunggulan dalam antrian. Dalam kelas Antrian, DIvo membuat metode untuk menambahkan buku ke tumpukan, menampilkan buku yang dimiliki, dan menghapus buku di tengah tumpukan jika diperlukan. Dia juga menambahkan metode untuk menghitung jumlah buku yang dimiliki setiap karakter. Setelah itu, DIvo mendefinisikan kelas Buku, yang menyimpan informasi tentang judul, pengarang, genre, dan status buku. Kelas ini juga dilengkapi dengan metode untuk menampilkan informasi buku dengan format yang rapi. Untuk menguji sistem yang dibuatnya, DIvo membuat kelas Main. Dalam kelas ini, ia membuat antrian karakter seperti Kazuma, Hu Tao, dan Kafka, yang masing-masing memiliki koleksi buku yang berbeda. DIvo kemudian menambahkan karakter baru, Sucrose, ke dalam antrian dan melakukan beberapa operasi seperti menampilkan antrian, menghapus karakter dari antrian, serta menukar posisi buku dalam tumpukan. DIvo merasa puas ketika melihat sistemnya berfungsi dengan baik, menampilkan daftar antrian yang jelas dan teratur, serta memastikan bahwa semua karakter dapat mengelola buku mereka dengan efektif.

1.2 HASIL PERCOBAAN

1.4.1 Program List Buku

1. Algoritma

- a. Menambahkan Antrian.
- b. Menggabungkan Daftar Antrian.
- c. Menambahkan Buku ke Antrian
- d. Menghapus Buku dari Antrian.
- e. Menambahkan dan Menghapus Karakter.
- f. Menukar Buku.
- g. Menampilkan Daftar Antrian.

2. Source Code

```
package Hard;
import java.util.Stack;

public class Antrian {
    String nama;
    int antrianKe;
    Stack<Buku> tumpukanBuku;
    boolean kartuSpesial;

    public Antrian(String nama, boolean kartuSpesial) {
        this.nama = nama;
        this.tumpukanBuku = new Stack<>();
        this.kartuSpesial = kartuSpesial;
    }

    public void setAntrianKe(int antrianKe) {
        this.antrianKe = antrianKe;
    }

    public int getJumlahBuku() {
        return tumpukanBuku.size();
    }

    public void tambahBuku(Buku buku) {
        tumpukanBuku.push(buku);
    }

    public void tampilkanBuku() {
        System.out.println("= BUKU " + nama.toUpperCase() + " =");
        for (Buku buku : tumpukanBuku) {
            System.out.println(buku);
        }
        System.out.println("=====");
    }

    public void hapusBukuTengah() {
        if (tumpukanBuku.isEmpty()) {
            System.out.println("Tumpukan buku kosong.");
            return;
        }
    }
}
```

```

        Stack<Buku> tempStack = new Stack<>();
        int middleIndex = tumpukanBuku.size() / 2;

        for (int i = 0; i < middleIndex; i++) {
            tempStack.push(tumpukanBuku.pop());
        }

        tumpukanBuku.pop();

        while (!tempStack.isEmpty()) {
            tumpukanBuku.push(tempStack.pop());
        }
    }

    @Override
    public String toString() {
        return "Nama : " + nama + "\nAntrian ke : " + antrianKe +
            "\nJumlah Buku : " + getJumlahBuku() + "\nKartu Spesial : " +
            (kartuSpesial ? "Ada" : "Tidak ada") + "\n";
    }
}

package Hard;
public class Buku {
    String judul;
    String pengarang;
    String genre;
    String status;

    public Buku(String judul, String pengarang, String genre, String
status) {
        this.judul = judul;
        this.pengarang = pengarang;
        this.genre = genre;
        this.status = status;
    }

    @Override
    public String toString() {
        return "Judul Buku : " + judul + "\nPengarang : " + pengarang
+ "\nGenre : " + genre + "\nStatus Buku : " + status;
    }
}

package Hard;
import java.util.LinkedList;
import java.util.Queue;

public class Main {
    public static void main(String[] args) {
        Queue<Antrian> queue = new LinkedList<>();

        Antrian kazuma = new Antrian("Kazuma", false);
        kazuma.tambahBuku(new Buku("Cara Menjadi Orang Kaya", "Teguh",
"Fantasi", "Buku Biasa"));
        kazuma.tambahBuku(new Buku("Belajar Java", "Raysen",
"Edukasi", "Buku Biasa"));

        Antrian huTao = new Antrian("Hu Tao", true);

```

```

        huTao.tambahBuku(new Buku("Belajar Ilmu Hitam", "Megumin",
"Unknown", "Cursed"));
        huTao.tambahBuku(new Buku("Belajar C++", "Raysen", "Edukasi",
"Buku Biasa"));
        huTao.tambahBuku(new Buku("Cara Tidur Cepat", "Teguh",
"Edukasi Kayaknya", "Cursed"));

        Antrian kafka = new Antrian("Kafka", false);
        kafka.tambahBuku(new Buku("Cara Menjadi Milioner Dalam 1
Jam", "Master Oogway", "Edukasi", "Buku Biasa"));
        kafka.tambahBuku(new Buku("Misteri Menghilangnya Nasi
Puyung", "Optimus", "Misteri", "Buku Biasa"));
        kafka.tambahBuku(new Buku("Raysen the Forgotten One",
"Unknown", "Sejarah", "Cursed"));

        Antrian xiangling = new Antrian("Xiangling", false);
        xiangling.tambahBuku(new Buku("Cara Memasak", "Chef Ling",
"Masakan", "Buku Biasa"));

        queue.add(kazuma);
        queue.add(huTao);
        queue.add(kafka);
        queue.add(xiangling);

        tampilkanDaftarAntrian(queue);

        kazuma.tampilkanBuku();
        queue.poll();

        tampilkanDaftarAntrian(queue);

        huTao.tampilkanBuku();
        queue.poll();

        tampilkanDaftarAntrian(queue);

        Antrian sucrose = new Antrian("Sucrose", true);
        sucrose.tambahBuku(new Buku("Durin The Forgotten Dragon",
"Gold", "Misteri", "Buku Biasa"));
        sucrose.tambahBuku(new Buku("Alchemy", "Albedo", "Science",
"Cursed"));
        sucrose.tambahBuku(new Buku("Resurrection", "Unknown",
"Unknown", "Cursed"));
        queue.add(sucrose);

        tampilkanDaftarAntrian(queue);

        sucrose.tampilkanBuku();
        queue.poll();

        tampilkanDaftarAntrian(queue);

        kafka.tampilkanBuku();

        swapBukuStack(kafka, 1, 3);
        kafka.tampilkanBuku();

        kafka.tumpukanBuku.removeIf(buku ->
buku.status.equals("Cursed"));
        kafka.tampilkanBuku();

```

```
        System.out.println("= DAFTAR ANTRIAN =\nAntrian Kosong");
    }

    public static void tampilkanDaftarAntrian(Queue<Antrian> queue)
    {
        System.out.println("= DAFTAR ANTRIAN =");
        int antrianKe = 1;
        for (Antrian antrian : queue) {
            antrian.setAntrianKe(antrianKe++);
            System.out.println(antrian);
        }
        System.out.println("=====");
    }

    public static void swapBukuStack(Antrian antrian, int pos1, int
pos2) {
        if (pos1 < 1 || pos2 < 1 || pos1 > antrian.tumpukanBuku.size()
|| pos2 > antrian.tumpukanBuku.size()) {
            System.out.println("Posisi buku tidak valid.");
            return;
        }

        Buku[] bukuArray = antrian.tumpukanBuku.toArray(new Buku[0]);
        Buku temp = bukuArray[pos1 - 1];
        bukuArray[pos1 - 1] = bukuArray[pos2 - 1];
        bukuArray[pos2 - 1] = temp;

        antrian.tumpukanBuku.clear();
        for (int i = bukuArray.length - 1; i >= 0; i--) {
            antrian.tumpukanBuku.push(bukuArray[i]);
        }
    }
}
```

1.3 ANALISIS DATA

1.5.1 Antrian

```
package Hard;
import java.util.Stack;
public class Antrian {
    String nama;
    int antrianKe;
    Stack<Buku> tumpukanBuku;
    boolean kartuSpesial;
    public Antrian(String nama, boolean kartuSpesial) {
        THIS.NAMA = NAMA;
        THIS.TUMPUKANBUKU = NEW STACK<>();
        THIS.KARTUSPESIAL = KARTUSPESIAL;
    }
    PUBLIC VOID SETANTRIANKE(INT ANTRIANKE) {
        THIS.ANTRIANKE = ANTRIANKE;
    }
    PUBLIC INT GETJUMLAHBUKU() {
        RETURN TUMPUKANBUKU.SIZE();
    }
    PUBLIC VOID TAMBAHBUKU(BUKU BUKU) {
        TUMPUKANBUKU.PUSH(BUKU);
    }
    PUBLIC VOID TAMPILKANBUKU() {
        SYSTEM.OUT.PRINTLN("= BUKU " + NAMA.TOUPPERCASE() + " =");
        FOR (BUKU BUKU : TUMPUKANBUKU) {
            SYSTEM.OUT.PRINTLN(BUKU);
            SYSTEM.OUT.PRINTLN("=====");
        }
    }
    PUBLIC VOID HAPUSBUKUTENGAH() {
        IF (TUMPUKANBUKU.ISEMPTY()) {
            SYSTEM.OUT.PRINTLN("TUMPUKAN BUKU KOSONG.");
            RETURN;
        }

        STACK<BUKU> TEMPSTACK = NEW STACK<>();
        INT MIDDLEINDEX = TUMPUKANBUKU.SIZE() / 2;

        FOR (INT I = 0; I < MIDDLEINDEX; I++) {
            TEMPSTACK.PUSH(TUMPUKANBUKU.POP());
        }
        TUMPUKANBUKU.POP();

        WHILE (!TEMPSTACK.ISEMPTY()) {
            TUMPUKANBUKU.PUSH(TEMPSTACK.POP());
        }
    }
    @Override
    PUBLIC STRING TOSTRING() {
        RETURN "NAMA : " + NAMA + "\nANTRIAN KE : " + ANTRIANKE +
            "\nJUMLAH BUKU : " + GETJUMLAHBUKU() + "\nKARTU SPESIAL : " +
            (KARTUSPESIAL ? "ADA" : "TIDAK ADA") + "\n";
    }
}
```

Script “public class Antrian {” adalah definisi dari sebuah kelas dalam bahasa pemrograman Java yang digunakan untuk merepresentasikan antrian dari buku-buku. Kelas

ini memiliki beberapa atribut penting, yaitu nama yang menyimpan nama pemilik antrian, antrianKe yang menunjukkan posisi antrian, tumpukanBuku yang merupakan stack untuk menyimpan objek-objek buku, dan kartuSpesial yang menunjukkan apakah pemilik antrian memiliki kartu spesial. Kelas ini juga memiliki konstruktor Antrian(String nama, boolean kartuSpesial) yang digunakan untuk menginisialisasi atribut nama dan kartuSpesial, serta membuat stack buku baru. Metode setAntrianKe(int antrianKe) digunakan untuk menetapkan posisi antrian, sedangkan getJumlahBuku() mengembalikan jumlah buku dalam stack. Untuk menambah buku ke dalam antrian, kelas ini menyediakan metode tambahBuku(Buku buku) yang menggunakan fungsi push() dari stack. Metode tampilkanBuku() menampilkan informasi mengenai semua buku yang ada di dalam antrian. Selain itu, metode hapusBukuTengah() berfungsi untuk menghapus buku yang berada di posisi tengah stack. Metode ini memindahkan buku-buku sebelum posisi tengah ke stack sementara, menghapus buku di posisi tengah, dan kemudian memindahkan kembali buku-buku dari stack sementara ke stack utama. Terakhir, metode toString() memberikan representasi string dari objek Antrian, menampilkan nama, posisi antrian, jumlah buku, dan status kartu spesial..

1.5.2 Buku

```
package Hard;
public class Buku {
    String judul;
    String pengarang;
    String genre;
    String status;

    public Buku(String judul, String pengarang, String genre, String
status) {
        this.judul = judul;
        this.pengarang = pengarang;
        this.genre = genre;
        this.status = status;
    }

    @Override
    public String toString() {
        return "Judul Buku : " + judul + "\nPengarang : " + pengarang +
"\nGenre : " + genre + "\nStatus Buku : " + status;
    }
}
```

Script “public class Buku{” adalah definisi dari sebuah kelas dalam bahasa pemrograman Java yang digunakan untuk merepresentasikan informasi mengenai buku. Kelas ini memiliki beberapa atribut penting, yaitu judul, yang menyimpan judul buku; pengarang, yang menyimpan nama pengarang buku; genre, yang menunjukkan jenis atau kategori buku; dan status, yang menyatakan kondisi buku tersebut, seperti apakah buku tersebut tersedia, dipinjam, atau status lainnya. Kelas ini dilengkapi dengan konstruktor

Buku(String judul, String pengarang, String genre, String status) yang digunakan untuk menginisialisasi semua atribut ketika objek buku dibuat. Metode toString() yang di-override memberikan representasi string dari objek Buku, menampilkan informasi mengenai judul, pengarang, genre, dan status buku secara terformat. Dengan demikian, kelas ini memudahkan pengelolaan dan tampilan informasi buku.

1.5.3 Main

```
package Hard;
import java.util.LinkedList;
import java.util.Queue;

public class Main {
    public static void main(String[] args) {
        Queue<Antrian> queue = new LinkedList<>();

        Antrian kazuma = new Antrian("Kazuma", false);
        kazuma.tambahBuku(new Buku("Cara Menjadi Orang Kaya", "Teguh",
"Fantasi", "Buku Biasa"));
        kazuma.tambahBuku(new Buku("Belajar Java", "Raysen", "Edukasi",
"Buku Biasa"));

        Antrian huTao = new Antrian("Hu Tao", true);
        huTao.tambahBuku(new Buku("Belajar Ilmu Hitam", "Megumin",
"Unknown", "Cursed"));
        huTao.tambahBuku(new Buku("Belajar C++", "Raysen", "Edukasi",
"Buku Biasa"));
        huTao.tambahBuku(new Buku("Cara Tidur Cepat", "Teguh", "Edukasi
Kayaknya", "Cursed"));

        Antrian kafka = new Antrian("Kafka", false);
        kafka.tambahBuku(new Buku("Cara Menjadi Milioner Dalam 1 Jam",
"Master Oogway", "Edukasi", "Buku Biasa"));
        kafka.tambahBuku(new Buku("Misteri Menghilangnya Nasi Puyung",
"Optimus", "Misteri", "Buku Biasa"));
        kafka.tambahBuku(new Buku("Raysen the Forgotten One", "Unknown",
"Sejarah", "Cursed"));

        Antrian xiangling = new Antrian("Xiangling", false);
        xiangling.tambahBuku(new Buku("Cara Memasak", "Chef Ling",
"Masakan", "Buku Biasa"));

        queue.add(kazuma);
        queue.add(huTao);
        queue.add(kafka);
        queue.add(xiangling);

        tampilkanDaftarAntrian(queue);

        kazuma.tampilkanBuku();
        queue.poll();

        tampilkanDaftarAntrian(queue);

        huTao.tampilkanBuku();
        queue.poll();

        tampilkanDaftarAntrian(queue);
    }
}
```



```

        Antrian sucrose = new Antrian("Sucrose", true);
        sucrose.tambahBuku(new Buku("Durin The Forgotten Dragon",
"Gold", "Misteri", "Buku Biasa"));
        sucrose.tambahBuku(new Buku("Alchemy", "Albedo", "Science",
"Cursed"));
        sucrose.tambahBuku(new Buku("Resurrection", "Unknown",
"Unknown", "Cursed"));
        queue.add(sucrose);

        tampilkanDaftarAntrian(queue);

        sucrose.tampilkanBuku();
        queue.poll();

        tampilkanDaftarAntrian(queue);

        kafka.tampilkanBuku();

        swapBukuStack(kafka, 1, 3);
        kafka.tampilkanBuku();

        kafka.tumpukanBuku.removeIf(buku
buku.status.equals("Cursed"));
        kafka.tampilkanBuku();

        System.out.println("= DAFTAR ANTRIAN =\nAntrian Kosong");
    }

    public static void tampilkanDaftarAntrian(Queue<Antrian> queue) {
        System.out.println("= DAFTAR ANTRIAN =");
        int antrianKe = 1;
        for (Antrian antrian : queue) {
            antrian.setAntrianKe(antrianKe++);
            System.out.println(antrian);
        }
        System.out.println("=====");
    }

    public static void swapBukuStack(Antrian antrian, int pos1, int pos2)
    {
        if (pos1 < 1 || pos2 < 1 || pos1 > antrian.tumpukanBuku.size()
|| pos2 > antrian.tumpukanBuku.size()) {
            System.out.println("Posisi buku tidak valid.");
            return;
        }

        Buku[] bukuArray = antrian.tumpukanBuku.toArray(new Buku[0]);
        Buku temp = bukuArray[pos1 - 1];
        bukuArray[pos1 - 1] = bukuArray[pos2 - 1];
        bukuArray[pos2 - 1] = temp;

        antrian.tumpukanBuku.clear();
        for (int i = bukuArray.length - 1; i >= 0; i--) {
            antrian.tumpukanBuku.push(bukuArray[i]);
        }
    }
}

```

Script “public class Main {” adalah definisi dari kelas utama dalam program yang mengelola antrian objek Antrian dan interaksi dengan buku-buku yang terdapat dalam

antrian tersebut. Di dalam metode main, sebuah objek Queue dideklarasikan untuk menyimpan antrian Antrian, yang akan berisi beberapa entri dengan nama karakter dan buku-buku yang mereka miliki. Setiap objek Antrian diinisialisasi dengan nama dan status kartu spesial, diikuti dengan penambahan beberapa objek Buku yang memiliki atribut seperti judul, pengarang, genre, dan status. Setelah menambahkan karakter ke dalam antrian, program menampilkan daftar antrian yang ada dengan memanggil metode tampilkanDaftarAntrian(), yang mencetak semua antrian beserta informasi mereka. Selanjutnya, setiap objek Antrian dalam antrian dapat menampilkan buku-buku mereka melalui metode tampilkanBuku(), dan objek terdepan dalam antrian dihapus dengan queue.poll(). Selain itu, program juga menangani penambahan objek Antrian baru (Sucrose) ke dalam antrian, menampilkan daftar setelah setiap perubahan, dan melakukan operasi tambahan seperti menukar buku di dalam tumpukan menggunakan metode swapBukuStack(), serta menghapus buku dengan status tertentu menggunakan metode removeIf(). Program diakhiri dengan pesan bahwa antrian telah kosong.