

MODUL II

STACK AND QUEUE

2.1 PERMASALAHAN

2.1.1 Medium Level

Dihari berikutnya Yanto kembali menjadi *volunteer* penjaga perpustakaan sendirian dikarenakan yanti tidak dapat hadir, tidak lama setelah Yanto memulai shift-nya beberapa orang mulai muncul untuk meminjam buku. Sesudah orang-orang itu memilih buku yang ingin mereka pinjam, mereka pun berbaris di depan meja Yanto dengan barisan seperti gambar disamping

DAFTAR ANTRIAN	
Nama : Kazuma	
Antrian ke : 1	
Jumlah Buku : 2	
Kartu Spesial : Tidak ada	
Nama : Hu Tao	
Antrian ke : 2	
Jumlah Buku : 3	
Kartu Spesial : Ada	
Nama : Kafka	
Antrian ke : 3	
Jumlah Buku : 3	
Kartu Spesial : Tidak ada	
Nama : Xiangling	
Antrian ke : 4	
Jumlah Buku : 1	
Kartu Spesial : Tidak ada	

Kazuma maju dan menunjukkan buku-buku yang ingin dia pinjam kepada Yanto, Kazuma meminjam 2 buku yang berjudul “Belajar Java” dan “Cara Menjadi Orang Kaya”, atribut lengkap bukunya dapat dilihat pada gambar disamping. Setelah Kazuma menyelesaikan peminjamannya, ia pun pergi. Sekarang tinggal tersisa 3 orang dalam antrian, dan antrian setelah Kazuma menjadi antrian pertama.

BUKU KAZUMA	
Judul Buku : Cara Menjadi Orang Kaya	
Pengarang : Teguh	
Genre : Fantasi	
Status Buku : Buku Biasa	
Judul Buku : Belajar Java	
Pengarang : Raysen	
Genre : Edukasi	
Status Buku : Buku Biasa	

```

=====
=          DAFTAR ANTRIAN          =
=====
Nama : Hu Tao
Antrian ke : 1
Jumlah Buku : 3
Kartu Spesial : Ada
=====
Nama : Kafka
Antrian ke : 2
Jumlah Buku : 3
Kartu Spesial : Tidak ada
=====
Nama : Xiangling
Antrian ke : 3
Jumlah Buku : 1
Kartu Spesial : Tidak ada
=====

```

Sekarang giliran Hu Tao untuk menunjukkan buku yang ingin dia pinjam kepada Yanto, Hu Tao meminjam 3 buku yang berjudul “Cara Tidur Cepat”, “Belajar C++” dan “Belajar Ilmu Hitam”, Atribut lengkap bukunya dapat dilihat pada gambar disamping.

```

=====
=          BUKU HU TAO          =
=====
Judul Buku : Belajar Ilmu Hitam
Pengarang : Megumin
Genre : Unknown
Status Buku : Cursed
=====
Judul Buku : Belajar C++
Pengarang : Rayzen
Genre : Edukasi
Status Buku : Buku Biasa
=====
Judul Buku : Cara Tidur Cepat
Pengarang : Teguh
Genre : Edukasi Kayaknya
Status Buku : Cursed
=====

```

Di perpustakaan ini terdapat dua jenis buku, buku pertama adalah buku biasa dan buku kedua adalah buku terkutuk. Buku terkutuk adalah buku yang memiliki informasi tentang pengetahuan pengetahuan terlarang jadi untuk meminjam buku terkutuk, peminjam harus memiliki kartu spesial. Karena Hu Tao memiliki kartu spesial, dia dapat meminjam buku terkutuk dan setelah menyelesaikan peminjamannya, Hu Tao langsung pergi. Di antrian sekarang tertinggal 2 orang.

```

=====
=          DAFTAR ANTRIAN          =
=====
Nama : Kafka
Antrian ke : 1
Jumlah Buku : 3
Kartu Spesial : Tidak ada
=====
Nama : Xiangling
Antrian ke : 2
Jumlah Buku : 1
Kartu Spesial : Tidak ada
=====

```

Setelah Hu Tao keluar, seseorang datang dengan terburu buru masuk ke dalam antrian. Orang itu adalah Sucrose, ia tampaknya ingin meminjam buku juga. di tangannya terdapat 3 tumpukan buku.

```

=====
=          DAFTAR ANTRIAN          =
=====
Nama : Kafka
Antrian ke : 1
Jumlah Buku : 3
Kartu Spesial : Tidak ada
=====
Nama : Xiangling
Antrian ke : 2
Jumlah Buku : 1
Kartu Spesial : Tidak ada
=====
Nama : Sucrose
Antrian ke : 3
Jumlah Buku : 3
Kartu Spesial : Ada
=====

```

Xiangling tiba tiba mendapat panggilan dari Zhongli, Yanto bisa mendengar samar samar percakapan xiangling di telefonnya, Yanto mendengar bahwa Gouba sedang mengamuk di Liyue dan butuh bantuan xiangling secepat mungkin. Karena urusan mendesak itu xiangling menyimpan Kembali bukunya dan langsung pergi. Tinggal tersisa dua orang di antrian.

```

=====
=          DAFTAR ANTRIAN          =
=====
Nama : Kafka
Antrian ke : 1
Jumlah Buku : 3
Kartu Spesial : Tidak ada
=====
Nama : Sucrose
Antrian ke : 2
Jumlah Buku : 3
Kartu Spesial : Ada
=====

```

Sucrose juga nampaknya sedang terburu buru, ia pun menanyakan kepada kafka apakah dia boleh bertukar tempat dengan kafka. Kafka setuju, Sucrose dan Kafka pun bertukar tempat.

```

=====
=          DAFTAR ANTRIAN          =
=====
Nama : Sucrose
Antrian ke : 1
Jumlah Buku : 3
Kartu Spesial : Ada
=====
Nama : Kafka
Antrian ke : 2
Jumlah Buku : 3
Kartu Spesial : Tidak ada
=====

```

Sucrose menunjukkan buku-buku yang ingin ia pinjam ke Yanto, Sucrose meminjam 3 buku dengan judul "Resurrection", "Alhcemy" dan "Durin The Forgotten Dragon", Atribut lengkap bukunya dapat dilihat pada gambar disamping.

```
=====
=          BUKU SUCROSE          =
=====
Judul Buku : Durin The Forgotten Dragon
Pengarang : Gold
Genre : Misteri
Status Buku : Buku Biasa
=====
Judul Buku : Alchemy
Pengarang : Albado
Genre : Science
Status Buku : Cursed
=====
Judul Buku : Resurrection
Pengarang : Unknown
Genre : Unknown
Status Buku : Cursed
=====
```

Setelah Sucrose menyelesaikan meminjamannya, ia pun pergi dengan terburu-buru. Sekarang tinggal tersisa 1 orang dalam antrian yaitu Kafka.

```
=====
=          DAFTAR ANTRIAN          =
=====
Nama : Kafka
Antrian ke : 1
Jumlah Buku : 3
Kartu Spesial : Tidak ada
=====
```

Yanto memeriksa satu per-satu buku yang ingin di pinjam kafka, dan salah satu buku yang ingin kafka pinjam adalah buku terkutuk. Yanto menanyakan kepada Kafka apakah dia memiliki kartu special, dan kafka menjawab bahwa dia tidak memiliki kartu special. Yanto pun memberi tahu Kafka bahwa Kafka memerlukan kartu special untuk meminjam buku terkutuk. Yanto pun memindahkan buku yang terkutuk ke bagian paling atas tumpukkan dan meminta kafka untuk mengembalikan buku tersebut. Setelah kafka mengembalikan buku yang terkutuk, ia menyelesaikan proses peminjaman. Sudah tidak ada orang dalam antrian.

```
=====
=          BUKU KAFKA          =
=====
Judul Buku : Cara Menjadi Milioner Dalam 1 Jam
Pengarang : Master Oogway
Genre : Edukasi
Status Buku : Buku Biasa
=====
Judul Buku : Misteri Menghilangnya Nasi Puyung
Pengarang : Optimus
Genre : Misteri
Status Buku : Buku Biasa
=====
Judul Buku : Raysen the Forgotten One
Pengarang : Unknown
Genre : Sejarah
Status Buku : Cursed
=====
Tidak Bisa Meminjam Buku Terkutuk Bila Tidak Memiliki Kartu Spesial
```

```
=====
=          BUKU KAFKA          =
=====
Judul Buku : Raysen the Forgotten One
Pengarang : Unknown
Genre : Sejarah
Status Buku : Cursed
=====
Judul Buku : Misteri Menghilangnya Nasi Puyung
Pengarang : Optimus
Genre : Misteri
Status Buku : Buku Biasa
=====
Judul Buku : Cara Menjadi Milioner Dalam 1 Jam
Pengarang : Master Oogway
Genre : Edukasi
Status Buku : Buku Biasa
=====
```

```
=====
=          DAFTAR ANTRIAN          =
=====
Antrian Kosong
=====
```

2.2 HASIL PERCOBAAN

2.2.1 Hard Level

1. Algoritma

- a. Inisialisasi program.
- b. Membuat antrian.
- c. Menambahkan antrian ke *queue*.
- d. Menampilkan antrian.
- e. Menampilkan buku dan menghapus dari antrian.
- f. Menampilkan buku dan menghapus dari antrian.
- g. Menambahkan antrian baru.
- h. Menangani antrian.
- i. Menampilkan buku setelah penghapusan.

2. Source Code

```
class Stack {
    private Buku top;

    public Stack() {
        top = null;
    }

    public boolean isEmpty() {
        return top == null;
    }

    public void push(Buku newBook) {
        newBook.setNext(top);
        top = newBook;
    }

    public Buku pop() {
        if (isEmpty()) return null;
        Buku item = top;
        top = top.getNext();
        return item;
    }

    public boolean isCursed(Buku buku, Antrian orang) {
        return buku != null && orang != null &&
        buku.status_buku && !orang.spec_card;
    }

    public void swapStack(Buku cursedNode) {
        if (cursedNode == null || cursedNode == top)
            return;

        Buku current = top;
        Buku previous = null;

        while (current != null && current != cursedNode) {
            previous = current;
```

```

        current = current.getNext();
    }

    if (previous != null) {
        previous.setNext(cursedNode.getNext());
    }
    cursedNode.setNext(top);
    top = cursedNode;
}

public void displayStack(Antrian orang) {
    Buku current = top;
    boolean hasCursed = false;

    while (current != null) {

System.out.println("=====
=====");
        System.out.println("Judul Buku: " +
current.judul);
        System.out.println("Pengarang: " +
current.pengarang);
        System.out.println("Genre: " + current.genre);
        System.out.println("Status Buku: " +
(current.status_buku ? "Cursed" : "Buku biasa"));

System.out.println("=====
=====");

        if (isCursed(current, orang)) {
            hasCursed = true;
        }
        current = current.getNext();
    }

    if (hasCursed) {
        current = top;

        while (current != null) {
            if (isCursed(current, orang)) {
                swapStack(current);
                break;
            }
            current = current.getNext();
        }

        current = top;
        while (current != null) {

System.out.println("=====
=====");
            System.out.println("Judul Buku: " +
current.judul);
            System.out.println("Pengarang: " +
current.pengarang);
            System.out.println("Genre: " +
current.genre);
            System.out.println("Status Buku: " +
(current.status_buku ? "Cursed" : "Buku biasa"));

```

```

System.out.println("=====
=====");
        current = current.getNext();
    }

    pop();
    System.out.println();
    current = top;
    while (current != null) {

System.out.println("=====
=====");
        System.out.println("Judul Buku: " +
current.judul);
        System.out.println("Pengarang: " +
current.pengarang);
        System.out.println("Genre: " +
current.genre);
        System.out.println("Status Buku: " +
(current.status_buku ? "Cursed" : "Buku biasa"));

System.out.println("=====
=====");
        current = current.getNext();
    }
}

public int getLength() {
    Buku counter = top;
    int size = 0;
    while (counter != null) {
        size++;
        counter = counter.getNext();
    }
    return size;
}

class Buku {
    String judul, pengarang, genre;
    Buku next = null;
    boolean status_buku;

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

    public Buku getNext() {
        return this.next;
    }

    public void setNext(Buku newNode) {
        this.next = newNode;
    }
}

```

```

class Antrian {
    Antrian next = null;
    String name;
    int jumlah_buku;
    boolean spec_card;
    Stack stack;

    public Antrian(String name, int jumlah_buku, boolean
spec_card) {
        this.name = name;
        this.jumlah_buku = jumlah_buku;
        this.spec_card = spec_card;
        this.stack = new Stack();
    }

    public void displayBuku() {

System.out.println("=====
=====");
        System.out.println("\t\tBuku " + this.name +
"\t\t\t=");
        this.stack.displayStack(this);
    }

    public Antrian getNext() {
        return this.next;
    }

    public void setNext(Antrian newNode) {
        this.next = newNode;
    }
}

class Queue {
    Antrian front, rear;

    public Queue() {
        front = null;
        rear = null;
    }

    public void printQueue() {

System.out.println("=====
=====");
        System.out.println("\t\tDAFTAR ANTRIAN\t\t\t=");

System.out.println("=====
=====");
        Antrian current = front;
        int urutAntrian = 1;

        while (current != null) {
            System.out.println("Nama: " + current.name);
            System.out.println("Antrian ke: " +
urutAntrian);
            System.out.println("Jumlah Buku: " +
current.stack.getLength());
            System.out.println("Kartu Spesial: " +
(current.spec_card ? "Ada" : "Tidak ada"));

```



```

System.out.println("=====
=====");
        current = current.getNext();
        urutAntrian++;
    }
    System.out.println();
}

public boolean isEmpty() {
    return front == null;
}

public void enqueue(Antrian newQueue) {
    if (rear != null) {
        rear.setNext(newQueue);
    }
    rear = newQueue;
    if (front == null) {
        front = rear;
    }
}

public void swapQueue(Antrian orang1, Antrian orang2) {
    if (orang1 == null || orang2 == null) {
        return;
    }
    Antrian current = front;
    Antrian previous = null;

    while (current != null && current != orang2) {
        previous = current;
        current = current.getNext();
    }
    if (previous != null) {
        previous.setNext(orang2.getNext());
    }
    orang2.setNext(front);
    front = orang2;
}

public Antrian dequeue() {
    if (isEmpty()) return null;
    Antrian removedQueue = front;
    front = front.getNext();
    if (front == null) {
        rear = null;
    }
    return removedQueue;
}

public void tempStack(Antrian antrian) {
    if (antrian == front) {
        front = front.getNext();
    } else {
        Antrian counter = front;
        while (counter != null) {
            if (counter.getNext() == antrian) {
                counter.setNext(antrian.getNext());
            }
            counter = counter.getNext();
        }
    }
}

```

```

    }
    }
}

public class Main {
    public static void main(String[] args) {
        Queue listAntrian = new Queue();

        Antrian orang1 = new Antrian("Kazuma", 2, false);
        orang1.stack.push(new Buku("Belajar Java",
"Raysen", "Edukasi", false));
        orang1.stack.push(new Buku("Cara Menjadi Orang
Kaya", "Teguh", "Fantasi", false));
        listAntrian.enqueue(orang1);

        Antrian orang2 = new Antrian("Hu Tao", 3, true);
        orang2.stack.push(new Buku("Cara Tidur Cepat",
"Teguh", "Edukasi Kayaknya", true));
        orang2.stack.push(new Buku("Belajar C++", "Raysen",
"Edukasi", false));
        orang2.stack.push(new Buku("Belajar Ilmu Hitam",
"Megumin", "Unknwon", true));
        listAntrian.enqueue(orang2);

        Antrian orang3 = new Antrian("Kafka", 3, false);
        orang3.stack.push(new Buku("Raysen the Forgotten
One", "Unknown", "Sejarah", true));
        orang3.stack.push(new Buku("Misteri Menghilangnya
Nasi Puyung", "Optimus", "Misteri", false));
        orang3.stack.push(new Buku("Cara Menjadi Milioner
Dalam 1 Jam", "Unknown", "Sejarah", true));
        listAntrian.enqueue(orang3);

        Antrian orang4 = new Antrian("Xiangling", 1,
false);
        orang4.stack.push(new Buku(null, null, null,
false));
        listAntrian.enqueue(orang4);
        listAntrian.printQueue();

        System.out.println();
        orang1.displayBuku();
        listAntrian.dequeue();
        listAntrian.printQueue();

        orang2.displayBuku();
        listAntrian.dequeue();
        listAntrian.printQueue();

        Antrian orang5 = new Antrian("Sucrose", 3, true);
        orang5.stack.push(new Buku("Resurrection",
"Unknown", "Unknown", true));
        orang5.stack.push(new Buku("Alchemy", "Albedo",
"Science", true));
        orang5.stack.push(new Buku("Durin the Forgotten
Dragon", "Gold", "Misteri", false));

        listAntrian.enqueue(orang5);

        listAntrian.tempStack(orang4);
    }
}

```

```
listAntrian.printQueue();  
System.out.println();  
listAntrian.swapQueue(orang3, orang5);  
listAntrian.printQueue();  
  
orang5.displayBuku();  
listAntrian.dequeue();  
listAntrian.printQueue();  
orang3.displayBuku();  
    }  
}
```

2.3 ANALISIS DATA

2.3.1 *Hard Level*

```
class Stack {
    private Buku top;

    public Stack() {
        top = null;
    }

    public boolean isEmpty() {
        return top == null;
    }

    public void push(Buku newBook) {
        newBook.setNext(top);
        top = newBook;
    }

    public Buku pop() {
        if (isEmpty()) return null;
        Buku item = top;
        top = top.getNext();
        return item;
    }

    public boolean isCursed(Buku buku, Antrian orang) {
        return buku != null && orang != null && buku.status_buku &&
!orang.spec_card;
    }

    public void swapStack(Buku cursedNode) {
        if (cursedNode == null || cursedNode == top) return;

        Buku current = top;
        Buku previous = null;

        while (current != null && current != cursedNode) {
            previous = current;
            current = current.getNext();
        }

        if (previous != null) {
            previous.setNext(cursedNode.getNext());
        }
        cursedNode.setNext(top);
        top = cursedNode;
    }
}
```

```

public void displayStack(Antrian orang) {
    Buku current = top;
    boolean hasCursed = false;

    while (current != null) {

System.out.println("=====
=");

        System.out.println("Judul Buku: " + current.judul);
        System.out.println("Pengarang: " + current.pengarang);
        System.out.println("Genre: " + current.genre);
        System.out.println("Status Buku: " +
(current.status_buku ? "Cursed" : "Buku biasa"));

System.out.println("=====
=");

        if (isCursed(current, orang)) {
            hasCursed = true;
        }
        current = current.getNext();
    }

    if (hasCursed) {
        current = top;

        while (current != null) {
            if (isCursed(current, orang)) {
                swapStack(current);
                break;
            }
            current = current.getNext();
        }

        current = top;
        while (current != null) {

System.out.println("=====
=");

            System.out.println("Judul Buku: " + current.judul);
            System.out.println("Pengarang: " +
current.pengarang);
            System.out.println("Genre: " + current.genre);
            System.out.println("Status Buku: " +
(current.status_buku ? "Cursed" : "Buku biasa"));

System.out.println("=====
=");

            current = current.getNext();

```

```

    }

    pop();
    System.out.println();
    current = top;
    while (current != null) {

System.out.println("=====
=");

        System.out.println("Judul Buku: " + current.judul);
        System.out.println("Pengarang: " +
current.pengarang);
        System.out.println("Genre: " + current.genre);
        System.out.println("Status Buku: " +
(current.status_buku ? "Cursed" : "Buku biasa"));

System.out.println("=====
=");

        current = current.getNext();
    }
}

public int getLength() {
    Buku counter = top;
    int size = 0;
    while (counter != null) {
        size++;
        counter = counter.getNext();
    }
    return size;
}
}

```

Kelas “Stack” adalah sebuah struktur data yang digunakan untuk mengelola tumpukan buku dalam bentuk objek “Buku”. Di dalam kelas ini, terdapat atribut bernama “top” yang menyimpan buku yang berada di posisi teratas dalam tumpukan. Ketika kelas ini diinisialisasi, “top” akan bernilai “null”, menandakan bahwa tumpukan masih kosong.

Kelas ini ada beberapa metode. Metode “isEmpty” digunakan untuk memeriksa apakah tumpukan kosong, dengan mengembalikan nilai *true* jika tidak ada buku di dalamnya. Metode “push” digunakan untuk menambahkan buku baru ke atas tumpukan, di mana buku tersebut akan menjadi “top” yang baru setelah ditambahkan. Sebaliknya, metode “pop” akan mengambil buku teratas dan menghapusnya dari tumpukan, mengembalikannya sebagai hasil.

Metode “isCursed”, yang memeriksa apakah sebuah buku memiliki status “Cursed”. Buku dianggap terkutuk jika memenuhi beberapa kondisi tertentu, seperti status buku yang aktif dan orang yang meminjamnya tidak memiliki kartu spesial.

Metode “swapStack” memungkinkan kita untuk menukar posisi buku terkutuk dengan buku teratas, jika buku terkutuk tidak berada di posisi teratas. Metode “displayStack” menampilkan semua buku dalam tumpukan, termasuk judul, pengarang, genre, dan statusnya, sehingga pengguna bisa melihat informasi tersebut dengan jelas. Terakhir, metode “getLength” menghitung jumlah buku yang ada dalam tumpukan dengan cara mengiterasi dari buku teratas hingga tidak ada lagi buku.

```
class Buku {
    String judul, pengarang, genre;
    Buku next = null;
    boolean status_buku;

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

    public Buku getNext() {
        return this.next;
    }

    public void setNext(Buku newNode) {
        this.next = newNode;
    }
}
```

Kelas “Buku” adalah representasi dari objek buku dalam sistem. Kelas ini memiliki beberapa atribut yang menyimpan informasi penting tentang setiap buku. Terdapat empat atribut utama: judul, pengarang, genre, dan status_buku. Kelas ini juga memiliki atribut *next*, yang bertipe “Buku” dan berfungsi untuk menghubungkan buku satu dengan buku lainnya dalam struktur data yang lebih besar, seperti *linked list*. Atribut ini akan menyimpan referensi ke buku berikutnya dalam daftar.

Kelas “Buku” dilengkapi dengan metode khusus yang digunakan untuk menginisialisasi semua atribut saat objek buku baru dibuat. Metode “getNext” digunakan untuk mengakses

buku berikutnya dalam daftar, sementara metode “`setNext`” digunakan untuk menetapkan buku yang akan menjadi buku berikutnya.

```
class Antrian {
    Antrian next = null;
    String name;
    int jumlah_buku;
    boolean spec_card;
    Stack stack;

    public Antrian(String name, int jumlah_buku, boolean spec_card)
    {
        this.name = name;
        this.jumlah_buku = jumlah_buku;
        this.spec_card = spec_card;
        this.stack = new Stack();
    }

    public void displayBuku() {

        System.out.println("=====
        =");
        System.out.println("\t\tBuku " + this.name + "\t\t\t=");
        this.stack.displayStack(this);
    }

    public Antrian getNext() {
        return this.next;
    }

    public void setNext(Antrian newNode) {
        this.next = newNode;
    }
}
```

Kelas “`Antrian`” digunakan untuk merepresentasikan setiap orang dalam antrian yang ingin meminjam buku. Kelas ini memiliki beberapa atribut penting. Pertama, `name` menyimpan nama orang yang meminjam, sementara `jumlah_buku` mencatat jumlah buku yang mereka miliki. Atribut “`spec_card`” adalah “`boolean`” yang menunjukkan apakah orang tersebut memiliki kartu spesial atau tidak. Selain itu, ada atribut `stack` yang merupakan objek dari kelas “`Stack`”, digunakan untuk menyimpan koleksi buku yang dimiliki oleh orang tersebut.

Metode penting dalam kelas ini adalah “`displayBuku`”, yang digunakan untuk menampilkan informasi tentang buku-buku yang dimiliki oleh orang dalam antrian. Metode

ini akan memanggil metode “displayBuku” dari kelas “Stack”, sehingga semua buku yang dimiliki akan ditampilkan dengan rapi.

```
class Queue {
    Antrian front, rear;

    public Queue() {
        front = null;
        rear = null;
    }

    public void printQueue() {
        System.out.println("=====
=====");
        System.out.println("\t\tDAFTAR ANTRIAN\t\t\t");
        System.out.println("=====
=====");
        Antrian current = front;
        int urutanAntrian = 1;

        while (current != null) {
            System.out.println("Nama: " + current.name);
            System.out.println("Antrian ke: " + urutanAntrian);
            System.out.println("Jumlah Buku: " +
current.stack.getLength());
            System.out.println("Kartu Spesial: " +
(current.spec_card ? "Ada" : "Tidak ada"));
            System.out.println("=====
=====");
            current = current.getNext();
            urutanAntrian++;
        }
        System.out.println();
    }

    public boolean isEmpty() {
        return front == null;
    }

    public void enqueue(Antrian newQueue) {
        if (rear != null) {
            rear.setNext(newQueue);
        }
        rear = newQueue;
        if (front == null) {
            front = rear;
        }
    }
}
```

```

public void swapQueue(Antrian orang1, Antrian orang2) {
    if (orang1 == null || orang2 == null) {
        return;
    }
    Antrian current = front;
    Antrian previous = null;

    while (current != null && current != orang2) {
        previous = current;
        current = current.getNext();
    }
    if (previous != null) {
        previous.setNext(orang2.getNext());
    }
    orang2.setNext(front);
    front = orang2;
}

public Antrian dequeue() {
    if (isEmpty()) return null;
    Antrian removedQueue = front;
    front = front.getNext();
    if (front == null) {
        rear = null;
    }
    return removedQueue;
}

public void tempStack(Antrian antrian) {
    if (antrian == front) {
        front = front.getNext();
    } else {
        Antrian counter = front;
        while (counter != null) {
            if (counter.getNext() == antrian) {
                counter.setNext(antrian.getNext());
            }
            counter = counter.getNext();
        }
    }
}
}

```

Kelas “Queue” adalah implementasi dari antrian yang digunakan untuk mengelola objek “Antrian”. Kelas ini memiliki dua atribut utama, yaitu “front” dan “rear”, yang masing-masing menyimpan referensi ke elemen pertama dan terakhir dalam antrian. Ketika objek “Queue” diinisialisasi, kedua atribut ini akan bernilai “null”, menandakan bahwa antrian

masih kosong. Metode “printQueue” digunakan untuk menampilkan daftar orang dalam antrian beserta informasi mereka seperti nama, posisi dalam antrian, jumlah buku yang dimiliki, dan apakah mereka memiliki kartu spesial.

Metode “isEmpty” berfungsi untuk memeriksa apakah antrian kosong dengan memeriksa apakah “front” bernilai “null”. Jika kosong, metode ini akan mengembalikan “true” jika tidak maka “false”. Metode “enqueue” digunakan untuk menambahkan objek “Antrian” baru ke belakang antrian. Jika “rear” tidak “null”, objek baru akan ditambahkan di belakang “rear”, dan “rear” akan diperbarui untuk menunjuk ke objek baru.

Metode “swapQueue” memungkinkan kita untuk menukar posisi dua orang dalam antrian. Metode “dequeue” berfungsi untuk mengambil dan menghapus orang terdepan dari antrian. Jika antrian kosong, metode ini akan mengembalikan “null”. Terakhir, metode “tempStack” digunakan untuk menghapus seseorang dari antrian tanpa mengembalikannya, berguna untuk pengelolaan yang lebih kompleks.

```
public class Main {
    public static void main(String[] args) {
        Queue listAntrian = new Queue();

        Antrian orang1 = new Antrian("Kazuma", 2, false);
        orang1.stack.push(new Buku("Belajar Java", "Raysen",
        "Edukasi", false));
        orang1.stack.push(new Buku("Cara Menjadi Orang Kaya",
        "Teguh", "Fantasi", false));
        listAntrian.enqueue(orang1);

        Antrian orang2 = new Antrian("Hu Tao", 3, true);
        orang2.stack.push(new Buku("Cara Tidur Cepat", "Teguh",
        "Edukasi Kayaknya", true));
        orang2.stack.push(new Buku("Belajar C++", "Raysen",
        "Edukasi", false));
        orang2.stack.push(new Buku("Belajar Ilmu Hitam", "Megumin",
        "Unknown", true));
        listAntrian.enqueue(orang2);

        Antrian orang3 = new Antrian("Kafka", 3, false);
        orang3.stack.push(new Buku("Raysen the Forgotten One",
        "Unknown", "Sejarah", true));
        orang3.stack.push(new Buku("Misteri Menghilangnya Nasi
        Puyung", "Optimus", "Misteri", false));
        orang3.stack.push(new Buku("Cara Menjadi Milioner Dalam 1
        Jam", "Unknown", "Sejarah", true));
```

```

        listAntrian.enqueue(orang3);

        Antrian orang4 = new Antrian("Xiangling", 1, false);
        orang4.stack.push(new Buku(null, null, null, false));
        listAntrian.enqueue(orang4);
        listAntrian.printQueue();

        System.out.println();
        orang1.displayBuku();
        listAntrian.dequeue();
        listAntrian.printQueue();

        orang2.displayBuku();
        listAntrian.dequeue();
        listAntrian.printQueue();

        Antrian orang5 = new Antrian("Sucrose", 3, true);
        orang5.stack.push(new Buku("Resurrection", "Unknown",
"Unknown", true));
        orang5.stack.push(new Buku("Alchemy", "Albedo", "Sience",
true));
        orang5.stack.push(new Buku("Durin the Forgotten Dragon",
"Gold", "Misteri", false));

        listAntrian.enqueue(orang5);

        listAntrian.tempStack(orang4);
        listAntrian.printQueue();
        System.out.println();
        listAntrian.swapQueue(orang3, orang5);
        listAntrian.printQueue();

        orang5.displayBuku();
        listAntrian.dequeue();
        listAntrian.printQueue();
        orang3.displayBuku();
    }
}

```

Kelas “Main” berfungsi sebagai titik masuk untuk program pengelolaan antrian peminjaman buku. Di dalam metode main, objek “Queue” diinisialisasi untuk menyimpan orang-orang dalam antrian, seperti “Kazuma” “Hu Tao” “Kafka” dan “Xiangling” yang masing-masing memiliki tumpukan buku. Program menampilkan antrian, kemudian menunjukkan dan menghapus orang dari antrian satu persatu, sambil menampilkan buku yang mereka miliki. Selanjutnya, objek baru “Sucrose” ditambahkan, dan manipulasi

antrian dilakukan, termasuk menghapus “Xiangling” dan menukar posisi “Kafka” dan “Sucrose”. Proses ini menggambarkan interaksi antara objek “Antrian” dan “Buku” serta cara efektif mengelola antrian dalam konteks peminjaman buku.