

MODUL II

STACK AND QUEUE

2.1 PERMASALAHAN

2.1.1 Antrian Pemusing Kepala

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 ini.



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 di bawah. Setelah Kazuma menyelesaikan peminjamannya, ia pun pergi. Sekarang tinggal tersisa 3 orang dalam antrian, dan antrian setelah Kazuma menjadi antrian pertama. (Buat antrian menggunakan Queue dan setiap orang dalam antrian memiliki tumpukan buku (Stack). “Antrian ke” dan “Jumlah Buku” harus dinamis).



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 di bawah

```
=====
                        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 special. Karena Hu Tao memiliki kartu special, dia dapat meminjam buku terkutuk dan setelah menyelesaikan peminjamannya, Hu Tao langsung pergi. Di antrian sekarang tinggal 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 teleponnya, 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 (Gunakan temp stack untuk menghapus node ditengah stack. Gunakan method swap untuk menukar posisi node dalam queue).

```
=====
=          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 denga njudul “Resurrection”, “Alchemy” dan “Durin The Forgotten Dragon”, atribut lengkap bukunya dapat dilihat pada gambar di bawah.

```
=====
=          BUKU SUCROSE          =
=====
Judul Buku : Durin The Forgotten Dragon
Pengarang : Gold
Genre : Misteri
Status Buku : Buku Biasa
=====
Judul Buku : Alhcemy
Pengarang : Albedo
Genre : Sience
Status Buku : Cursed
=====
Judul Buku : Resurrection
Pengarang : Unknown
Genre : Unknown
Status Buku : Cursed
=====
```

Setelah Sucrose menyelesaikan peminjamannya, ia pun pergi denga 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 punmemberi tahu Kafka bahwa Kafka memerlukan kartu special untuk meminjam buku terkutuk. Yanto pun memindahkan buku yang terkutuk ke bagian paling atas tumpukan dan meminta Kafka untuk mengembalikan buku tersebut. Setelah Kafka mengembalikan buku yang terkutuk, ia menyelesaikan proses peminjaman. Sudah tidak ada orang dalam antrian (Gunakan method swap untuk menukar posisi buku dalam stack).

```
=====
=          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
=====
```

2.2 HASIL PERCOBAAN

2.2.1 Antrian Pemusing Kepala

1. Algoritma

- a. Buat data yang akan digunakan.
- b. Buat *stack* dengan *method push*, *pop*, *printstack* dan *arrangedcursed* yaitu menambah, menghapus, menampilkan buku dan memprioritaskan buku dengan status *cursed*.
- c. Buat *queue* dengan *method enqueue* untuk menambah antrian, *dequeue* untuk mengurangi antrian, *delete* untuk menghapus karakter, *printqueue* untuk menampilkan antrian dan *swap* untuk menukar antrian karakter.
- d. Tambah beberapa karakter dengan *enqueue* dan buku yang dibawa dengan *push*.
- e. Hapus karakter dengan *dequeue* dan tampilkan buku yang dibawa.
- f. Tambah karakter lagi.
- g. Hapus karakter dengan *delete*.
- h. Tukar posisi karakter dengan *swap* kemudian hapus dengan *dequeue*.
- i. Terakhir hapus lagi dengan *dequeue* namun karena karakter tidak memiliki kartu spesial serta membawa buku terlarang, hapus buku terlarang terlebih dahulu dengan diatur (*arrangedcursed*) lalu *pop*

2. Source Code

```
public class Buku {
    String JudulBuku, Pengarang, Genre, StatusBuku;
    Buku next;

    Buku(String inputJudulBuku, String inputPengarang, String
inputGenre, String inputStatusBuku){
        this.JudulBuku = inputJudulBuku;
        this.Pengarang = inputPengarang;
        this.Genre = inputGenre;
        this.StatusBuku = inputStatusBuku;
        this.next = null;
    }
}

public class Karakter {
    String Nama, Kartu;
    int JumlahBuku;
    Karakter next;
    Buku top;
```

```

        Karakter(String inputNama, int inputJumlahBuku, String
inputKartu){
            this>Nama = inputNama;
            this.JumlahBuku = inputJumlahBuku;
            this.Kartu = inputKartu;
            this.next = null;
        }

        void push(String judul, String Pengarang, String Genre, String
StatusBuku){
            Buku NewBuku = new Buku(judul, Pengarang, Genre,
StatusBuku);
            NewBuku.next = top;
            top = NewBuku;
            JumlahBuku++;
        }

        void pop(){
            if (top == null){
                System.out.println("Tidak ada buku untuk dihapus");
            } else {
                top = top.next;
                JumlahBuku--;
            }
        }
    }

public class Queue { //QUEUE AND STACK
    Karakter front, rear;

    void enqueue(String Nama, int JumlahBuku, String Kartu){
        Karakter NewKarakter = new Karakter(Nama, JumlahBuku,
Kartu);
        if(rear == null){
            front = NewKarakter;
            rear = NewKarakter;
        }else{
            rear.next = NewKarakter;
            rear = NewKarakter;
            rear.next = null;
        }
    } //-----

    void dequeue(){
        if(rear == null){
            System.out.println("Tidak ada karakter yang
mengantri");
        }else{
            Karakter current = front;
            Buku IterasiBuku = current.top;
            int cursedCount = 0;

            while (IterasiBuku != null) {
                if (IterasiBuku.StatusBuku == "Cursed") {
                    cursedCount++;
                }
                IterasiBuku = IterasiBuku.next;
            }

            if (cursedCount > 0 && front.Kartu == "Tidak ada") {
                arrangeCursed(front>Nama);
            }
        }
    }
}

```

```

        printstack(front>Nama);
        for (int i = 0; i < cursedCount; i++){
            front.pop();
        }
        System.out.println(front>Nama + " memiliki Buku
terkutuk dan tidak memiliki kartu spesial");
        return;
    }

    printstack(front>Nama);
    front = front.next;
    if(front == null){
        rear = null;
    }
}
}

//-----

void delete(String nama) {
    if (front == null) {
        System.out.println("Tidak ada yang mengantri");
        return;
    }

    if (front>Nama == nama){
        front = front.next;
        System.out.println(nama + " telah keluar dari
antrian");
        return;
    }

    Karakter temp = front;
    Karakter prev = null;

    while (temp != null && temp>Nama != nama){
        prev = temp;
        temp = temp.next;
    }
    if (temp == null) {
        System.out.println("Nama tidak ditemukan");
        return;
    }
    prev.next = temp.next;
    System.out.println(nama + " telah pergi");
}

//-----

void arrangeCursed(String nama) {
    if (front == null) {
        System.out.println("List is empty.");
        return;
    }
    Karakter currentDaftar = front;
    while (currentDaftar != null && currentDaftar>Nama != nama)
    {
        currentDaftar = currentDaftar.next;
    }

    if (currentDaftar == null) {
        System.out.println(nama + " tidak ditemukan.");
        return;
    }
}

```

```

        Buku current = currentDaftar.top;
        Buku cursedHead = null;
        Buku cursedTail = null;
        Buku nonCursedHead = null;
        Buku nonCursedTail = null;

        while (current != null) {
            Buku nextBook = current.next;

            if (current.StatusBuku == "Cursed") {
                if (cursedHead == null) {
                    cursedHead = current;
                    cursedTail = current;
                } else {
                    cursedTail.next = current;
                    cursedTail = current;
                }
            } else {
                if (nonCursedHead == null) {
                    nonCursedHead = current;
                    nonCursedTail = current;
                } else {
                    nonCursedTail.next = current;
                    nonCursedTail = current;
                }
            }
            current.next = null;
            current = nextBook;
        }
        if (cursedTail != null) {
            cursedTail.next = nonCursedHead;
            currentDaftar.top = cursedHead;
        } else {
            currentDaftar.top = nonCursedHead;
        }
        System.out.println("Buku Cursed Sudah diatur ke atas untuk
" + nama);
    } //------

    void printqueue(){
        Karakter current = front; int i = 1;

        System.out.println("=====
=====");
        System.out.println("=
                                DAFTAR ANTRIAN
=");
        while(current != null){

        System.out.println("=====
=====");
            System.out.println("Nama : " + current.Nama
                                + "\nAntrian ke : " + (i++)
                                + "\nJumlah Buku : " + current.JumlahBuku
                                + "\nKartu Spesial : " + current.Kartu
                                );
            System.out.println("=====
=====");
            current = current.next;
        }
    } //------

```



```

void swap(String Nama, String Nama2) {
    if (Nama == Nama2) {
        System.out.println("Tidak bisa menukar nama yang
sama");
        return;
    }
    Karakter prevone = null, currone = front;
    Karakter prevtwo = null, currtwo = front;

    while (currone != null && currone>Nama != Nama) {
        prevone = currone;
        currone = currone.next;
    }
    while (currtwo != null && currtwo>Nama != Nama2) {
        prevtwo = currtwo;
        currtwo = currtwo.next;
    }
    if (currone == null || currtwo == null) {
        System.out.println("Nama tidak ada di antrian");
        return;
    }
    if (prevone != null) {
        prevone.next = currtwo;
    } else {
        front = currtwo;
    }
    if (prevtwo != null) {
        prevtwo.next = currone;
    } else {
        front = currone;
    }
    Karakter temp = currone.next;
    currone.next = currtwo.next;
    currtwo.next = temp;

    System.out.println>Nama + " dan " + Nama2 + " telah
bertukar tempat ");
    printqueue();
} //-----

void push(String>Nama, String judul, String pengarang, String
genre, String status){
    Karakter current = front;
    while(current != null){
        if(current>Nama.equals>Nama){
            current.push(judul, pengarang, genre,status);
            return;
        }
        current = current.next;
    }
    System.out.println("Nama tidak ditemukan");
}

void printstack(String>Nama) {
    Karakter current = front;
    if (current>Nama ==>Nama) {
        Buku IterasiBuku = current.top;

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

```

```

        System.out.println("                Buku " +
Nama);
        while (IterasiBuku != null) {

System.out.println("=====
=====");
                System.out.println("Judul Buku : " +
IterasiBuku.JudulBuku
                        + "\nPengarang : " + IterasiBuku.Pengarang
                        + "\nGenre : " + IterasiBuku.Genre
                        + "\nStatus Buku : " +
IterasiBuku.StatusBuku
                );

System.out.println("=====
=====");
                IterasiBuku = IterasiBuku.next;
        }
    }
} //-----
}

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

        // Kazuma
        DaftarKarakter.enqueue("Kazuma", 0, "Tidak Ada");
        DaftarKarakter.push("Kazuma", "Belajar Java", "Raysen",
"Edukasi", "Buku Biasa");
        DaftarKarakter.push("Kazuma", "Cara Menjadi Orang Kaya",
"Teguh", "Fantasi", "Buku Biasa");
        //Hu Tao
        DaftarKarakter.enqueue("Hu Tao",0, "Ada");
        DaftarKarakter.push("Hu Tao", "Cara Tidur Cepat", "Teguh",
"Edukasi Kayaknya", "Cursed");
        DaftarKarakter.push("Hu Tao", "Belajar C++", "Raysen",
"Edukasi", "Buku Biasa");
        DaftarKarakter.push("Hu Tao", "Belajar Ilmu Hitam",
"Megumin", "Unknown", "Cursed");
        // Kafka
        DaftarKarakter.enqueue("Kafka", 0, "Tidak ada");
        DaftarKarakter.push("Kafka", "Raysen the Forgotten One",
"Unknown", "Sejarah", "Cursed");
        DaftarKarakter.push("Kafka", "Misteri Menghilangnya Nasi
Puyung", "Optimus", "Misteri", "Buku Biasa");
        DaftarKarakter.push("Kafka", "Cara Menjadi Milioner Dalam 1
Jam", "Master Oogway", "Edukasi", "Buku Biasa");
        // Xiangling
        DaftarKarakter.enqueue("Xiangling", 0, "Tidak ada");
        DaftarKarakter.push("Xiangling", "Crime and Punishment",
"Fyodor Dvovtoesky", "Psychology", "Buku Biasa");

        DaftarKarakter.printqueue();
        System.out.println("                DEQUEUE");
        DaftarKarakter.dequeue();
        System.out.println("                DEQUEUE 2");
        DaftarKarakter.dequeue();
        System.out.println("                ANTRIAN SEKARANG");
        DaftarKarakter.printqueue();
    }
}

```

```
//SUCROSE
DaftarKarakter.enqueue("Sucrose", 0, "Ada");
DaftarKarakter.push("Sucrose","Durin the Forgotten Dragon",
"Gold", "Misteri", "Buku Biasa");
DaftarKarakter.push("Sucrose","Alhcemy", "Albedo",
"Science", "Cursed");
DaftarKarakter.push("Sucrose","Resurrection", "Unknown",
"Unknown", "Cursed");

DaftarKarakter.printqueue();
DaftarKarakter.delete("Xiangling");
DaftarKarakter.printqueue();
DaftarKarakter.swap("Kafka", "Sucrose");
System.out.println("                DEQUEUE 3");
DaftarKarakter.dequeue();
DaftarKarakter.printqueue();
System.out.println("                DEQUEUE 3");
DaftarKarakter.dequeue();
DaftarKarakter.printqueue();
System.out.println("                DEQUEUE 4");
DaftarKarakter.dequeue();
DaftarKarakter.printqueue();
    }
}
```

2.3 ANALISIS DATA

2.3.1 Antrian Pemusing Kepala

```
public class Buku {
    String JudulBuku, Pengarang, Genre, StatusBuku;
    Buku next;

    Buku(String inputJudulBuku, String inputPengarang, String
inputGenre, String inputStatusBuku){
        this.JudulBuku = inputJudulBuku;
        this.Pengarang = inputPengarang;
        this.Genre = inputGenre;
        this.StatusBuku = inputStatusBuku;
        this.next = null;
    }
}
```

Script “public class Buku” adalah *script* yang menyimpan data, yang akan digunakan untuk *stack* dan mengatur parameter “Buku” dengan “Buku (JudulBuku, Pengarang, Genre, StatusBuku)”.

```
public class Karakter {
    String Nama, Kartu;
    int JumlahBuku;
    Karakter next;
    Buku top;

    Karakter(String inputNama, int inputJumlahBuku, String inputKartu){
        this.Nama = inputNama;
        this.JumlahBuku = inputJumlahBuku;
        this.Kartu = inputKartu;
        this.next = null;
    }
}
```

Script “public class Karakter” adalah *script* yang menyimpan data, yang akan digunakan untuk *queue* dan mengatur parameter “Karakter” dengan “Karakter (Nama, JumlahBuku, Kartu)”.

```
void push(String judul, String Pengarang, String Genre, String
StatusBuku){
    Buku NewBuku = new Buku(judul, Pengarang, Genre, StatusBuku);
    NewBuku.next = top;
    top = NewBuku;
    JumlahBuku++;
}
```

Script “void push (String judul, String Pengarang, String Genre, String StatusBuku)” adalah *method* untuk menambahkan data pada *stack* dengan menggunakan *pointer* “top”. Selain itu *method* ini juga akan menambah jumlah buku yang dibawa karakter.

```
void pop(){
    if (top == null){
        System.out.println("Tidak ada buku untuk dihapus");
    } else {
        top = top.next;
    }
}
```

```

        JumlahBuku--;
    }
}

```

Script “void pop ()” adalah *method* untuk mengurangi data pada *stack* dengan menggunakan *pointer* “top”. Selain itu *method* ini juga akan mengurangi jumlah buku yang dibawa karakter.

```

public class Queue {
    Karakter front, rear;

    void enqueue(String Nama, int JumlahBuku, String Kartu){
        Karakter NewKarakter = new Karakter(Nama, JumlahBuku, Kartu);
        if(rear == null){
            front = NewKarakter;
            rear = NewKarakter;
        }else{
            rear.next = NewKarakter;
            rear = NewKarakter;
            rear.next = null;
        }
    }
} //-----

```

Script “public class Queue ()” adalah *class* yang mengatur struktur data sesuai dengan *queue* yang memiliki beberapa *method* salah satunya *method* “void enqueue (String Nama, int JumlahBuku, String Kartu)” untuk menambah data pada *queue*.

```

void dequeue(){
    if(rear == null){
        System.out.println("Tidak ada karakter yang mengantri");
    }else{
        Karakter current = front;
        Buku IterasiBuku = current.top;
        int cursedCount = 0;

        while (IterasiBuku != null) {
            if (IterasiBuku.StatusBuku == "Cursed") {
                cursedCount++;
            }
            IterasiBuku = IterasiBuku.next;
        }

        if (cursedCount > 0 && front.Kartu == "Tidak ada") {
            arrangeCursed(front.Nama);
            printstack(front.Nama);
            for (int i = 0; i < cursedCount; i++){
                front.pop();
            }
            System.out.println(front.Nama + " memiliki Buku terkutuk dan tidak memiliki kartu spesial");
            return;
        }

        printstack(front.Nama);
        front = front.next;
        if(front == null){
            rear = null;
        }
    }
}

```

```

    }
}
} //-----

```

Script “void dequeue ()” adalah *method* untuk menghapus data pertama, di *method* ini juga akan menampilkan semua buku karakter yang akan dihapus dengan “dequeue ()”, selain itu *method* ini juga akan melakukan pengecekan apakah karakter memiliki buku *cursed*, bila iya buku akan dihapus apabila karakter tidak memiliki kartu spesial.

```

void delete(String nama) {
    if (front == null) {
        System.out.println("Tidak ada yang mengantri");
        return;
    }

    if (front>Nama == nama){
        front = front.next;
        System.out.println(nama + " telah keluar dari antrian");
        return;
    }

    Karakter temp = front;
    Karakter prev = null;

    while (temp != null && temp>Nama != nama){
        prev = temp;
        temp = temp.next;
    }

    if (temp == null) {
        System.out.println("Nama tidak ditemukan");
        return;
    }

    prev.next = temp.next;
    System.out.println(nama + " Telah Berpergi Berperang");
} //-----

```

Script “void delete (String nama)” adalah *method* untuk menghapus karakter berdasarkan nama karakter untuk itu diperlukan pencarian nama karakter dengan *looping*, bila nama karakter ditemukan karakter akan dihapus.

```

void arrangeCursed(String nama) {
    if (front == null) {
        System.out.println("List is empty.");
        return;
    }

    Karakter currentDaftar = front;
    while (currentDaftar != null && currentDaftar>Nama != nama) {
        currentDaftar = currentDaftar.next;
    }

    if (currentDaftar == null) {
        System.out.println(nama + " tidak ditemukan.");
        return;
    }

    Buku current = currentDaftar.top;
    Buku cursedHead = null;

```

```

Buku cursedTail = null;
Buku nonCursedHead = null;
Buku nonCursedTail = null;

while (current != null) {
    Buku nextBook = current.next;

    if (current.StatusBuku == "Cursed") {
        if (cursedHead == null) {
            cursedHead = current;
            cursedTail = current;
        } else {
            cursedTail.next = current;
            cursedTail = current;
        }
    } else {
        if (nonCursedHead == null) {
            nonCursedHead = current;
            nonCursedTail = current;
        } else {
            nonCursedTail.next = current;
            nonCursedTail = current;
        }
    }
    current.next = null;
    current = nextBook;
}
if (cursedTail != null) {
    cursedTail.next = nonCursedHead;
    currentDaftar.top = cursedHead;
} else {
    currentDaftar.top = nonCursedHead;
}
System.out.println("Buku Cursed Sudah diatur ke atas untuk " +
nama);
} //------

```

Script “void arrangeCursed (String nama)” adalah *method* untuk memprioritaskan buku dengan status *cursed* terlebih dahulu dibandingkan dengan buku dengan status buku biasa.

```

void printqueue(){
    Karakter Current = front; int i = 1;

    System.out.println("=====
=====");
    System.out.println("=
                                DAFTAR ANTRIAN
=");
    while(Current != null){

    System.out.println("=====
=====");
        System.out.println("Nama : " + Current.Nama
            + "\nAntrian ke : " + (i++)
            + "\nJumlah Buku : " + Current.JumlahBuku
            + "\nKartu Spesial : " + Current.Kartu
        );

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

```

```

        Current = Current.next;
    }
} //------

```

Script “void printqueue()” adalah *method* untuk menampilkan seluruh karakter yang mengantri dengan cara menggunakan *looping* menggunakan pointer “Current = Current.next” sehingga seluruh karakter akan ditampilkan.

```

void swap(String Nama, String Nama2) {
    if (Nama == Nama2) {
        System.out.println("Tidak bisa menukar nama yang sama");
        return;
    }
    Karakter prevone = null, currone = front;
    Karakter prevtwo = null, currtwo = front;

    while (currone != null && currone>Nama != Nama) {
        prevone = currone;
        currone = currone.next;
    }
    while (currtwo != null && currtwo>Nama != Nama2) {
        prevtwo = currtwo;
        currtwo = currtwo.next;
    }
    if (currone == null || currtwo == null) {
        System.out.println("Nama tidak ada di antrian");
        return;
    }
    if (prevone != null) {
        prevone.next = currtwo;
    } else {
        front = currtwo;
    }
    if (prevtwo != null) {
        prevtwo.next = currone;
    } else {
        front = currone;
    }
    Karakter temp = currone.next;
    currone.next = currtwo.next;
    currtwo.next = temp;

    System.out.println(Nama + " dan " + Nama2 + " telah bertukar tempat ");
    printqueue();
} //------

```

Script “void swap (String Nama, String Nama2)” adalah *method* untuk menukar antrian karakter 1 dan karakter 2, untuk menukar 2 karakter diperlukan 4 *pointer* yaitu “prevone”, “prevtwo”, “curreone” dan “currtwo”.

```

void push(String Nama, String judul, String pengarang, String genre,
String status){
    Karakter current = front;
    while(current != null){
        if(current>Nama.equals>Nama){
            current.push(judul, pengarang, genre,status);
            return;
        }
    }
}

```



```

    }
    current = current.next;
}
System.out.println("Nama tidak ditemukan");
}

```

Script “void push (String Nama, String judul, String pengarang, String genre, String status)” adalah *method* untuk menambahkan data buku di suatu karakter pada *stack* dengan menggunakan parameter karakter dan data buku.

```

void printstack(String Nama) {
    Karakter current = front;
    if (current>Nama == Nama) {
        Buku IterasiBuku = current.top;

System.out.println("=====
=====");
        System.out.println("                        Buku " + Nama);
        while (IterasiBuku != null) {

System.out.println("=====
=====");
        System.out.println("Judul      Buku      :      "      +
IterasiBuku.JudulBuku
                        + "\nPengarang : " + IterasiBuku.Pengarang
                        + "\nGenre : " + IterasiBuku.Genre
                        + "\nStatus Buku : " + IterasiBuku.StatusBuku
                        );

System.out.println("=====
=====");
        IterasiBuku = IterasiBuku.next;
    }
}
} //-----
}

```

Script “void printstack(String Nama)” adalah *method* untuk menampilkan judul buku, pengarang, genre dan status buku yang dimiliki karakter *method* ini akan digunakan di *method* “void dequeue”.

```

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

        // Kazuma
        DaftarKarakter.enqueue("Kazuma", 0, "Tidak Ada");
        DaftarKarakter.push("Kazuma", "Belajar Java", "Raysen",
"Edukasi", "Buku Biasa");
        DaftarKarakter.push("Kazuma", "Cara Menjadi Orang Kaya",
"Teguh", "Fantasi", "Buku Biasa");
        //Hu Tao
        DaftarKarakter.enqueue("Hu Tao",0, "Ada");
        DaftarKarakter.push("Hu Tao", "Cara Tidur Cepat", "Teguh",
"Edukasi Kayaknya", "Cursed");
        DaftarKarakter.push("Hu Tao", "Belajar C++", "Raysen",
"Edukasi", "Buku Biasa");

```

```

        DaftarKarakter.push("Hu Tao", "Belajar Ilmu Hitam", "Megumin",
"Unknown", "Cursed");
        // Kafka
        DaftarKarakter.enqueue("Kafka", 0, "Tidak ada");
        DaftarKarakter.push("Kafka", "Raysen the Forgotten One",
"Unknown", "Sejarah", "Cursed");
        DaftarKarakter.push("Kafka", "Misteri Menghilangnya Nasi
Puyung", "Optimus", "Misteri", "Buku Biasa");
        DaftarKarakter.push("Kafka", "Cara Menjadi Milioner Dalam 1
Jam", "Master Oogway", "Edukasi", "Buku Biasa");
        // Xiangling
        DaftarKarakter.enqueue("Xiangling", 0, "Tidak ada");
        DaftarKarakter.push("Xiangling", "Crime and Punishment", "Fyodor
Dovtoesky", "Psychology", "Buku Biasa");

        DaftarKarakter.printqueue();
        System.out.println("                DEQUEUE");
        DaftarKarakter.dequeue();
        System.out.println("                DEQUEUE 2");
        DaftarKarakter.dequeue();
        System.out.println("                ANTRIAN SEKARANG");
        DaftarKarakter.printqueue();

        //SUCROSE
        DaftarKarakter.enqueue("Sucrose", 0, "Ada");
        DaftarKarakter.push("Sucrose", "Durin the Forgotten Dragon",
"Gold", "Misteri", "Buku Biasa");
        DaftarKarakter.push("Sucrose", "Alhcemy", "Albedo", "Sience",
"Cursed");
        DaftarKarakter.push("Sucrose", "Resurrection", "Unknown",
"Unknown", "Cursed");

        DaftarKarakter.printqueue();
        DaftarKarakter.delete("Xiangling");
        DaftarKarakter.printqueue();
        DaftarKarakter.swap("Kafka", "Sucrose");
        System.out.println("                DEQUEUE 3");
        DaftarKarakter.dequeue();
        DaftarKarakter.printqueue();
        System.out.println("                DEQUEUE 3");
        DaftarKarakter.dequeue();
        DaftarKarakter.printqueue();
        System.out.println("                DEQUEUE 4");
        DaftarKarakter.dequeue();
        DaftarKarakter.printqueue();
    }
}

```

Script “public class Main {” adalah *class* yang akan menjalankan *method-method* pada *class* “public class Queue” dengan mendeklarasikan “Queue DaftarKarakter = new Queue ();” sehingga *method* pada “public class Queue” bisa digunakan.