STUDI KASUS
CLASS, ENKAPSULASI, INHERITANCE, DAN POLIFORFISME
MATA KULIAH PEMROGRAMAN BERORIENTASI OBJEK
KELAS TIF B

Dosen Pengampu

Dr. Eng. Budi Darma Setiawan, S.Kom., M.Cs.

Oleh:

I Gusti Ngurah Ryo Adi Tarta/225150200111011

Muhammad Husain Fadhlillah/225150207111027

Kartika Madania/225150207111025

PROGRAM STUDI TEKNIK INFORMATIKA

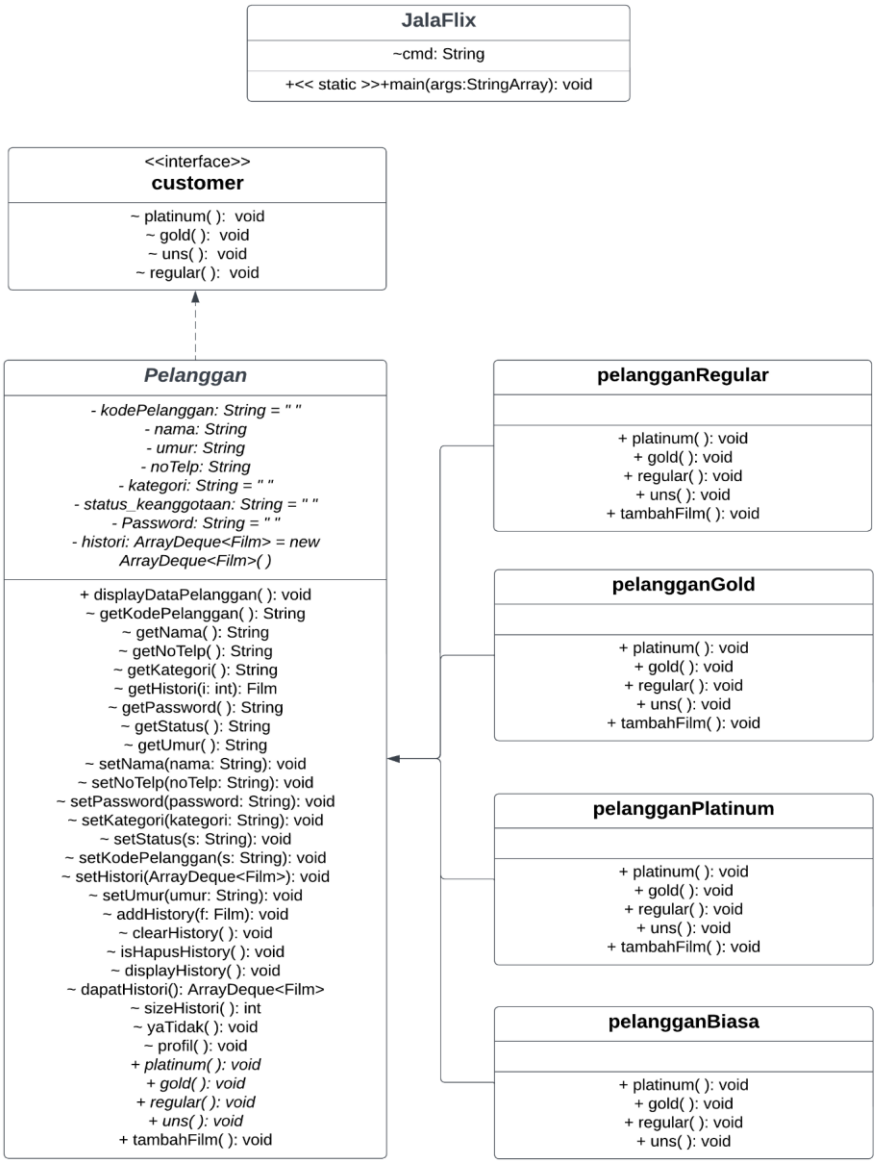
FAKULTAS ILMU KOMPUTER

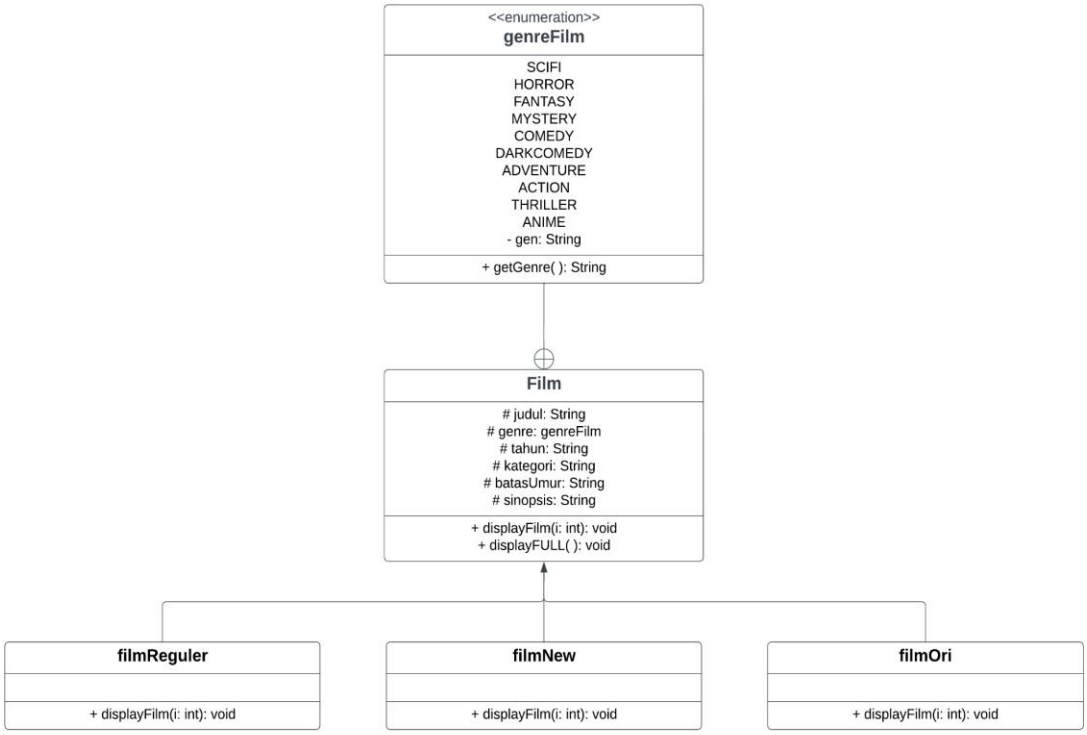
MARET 2023

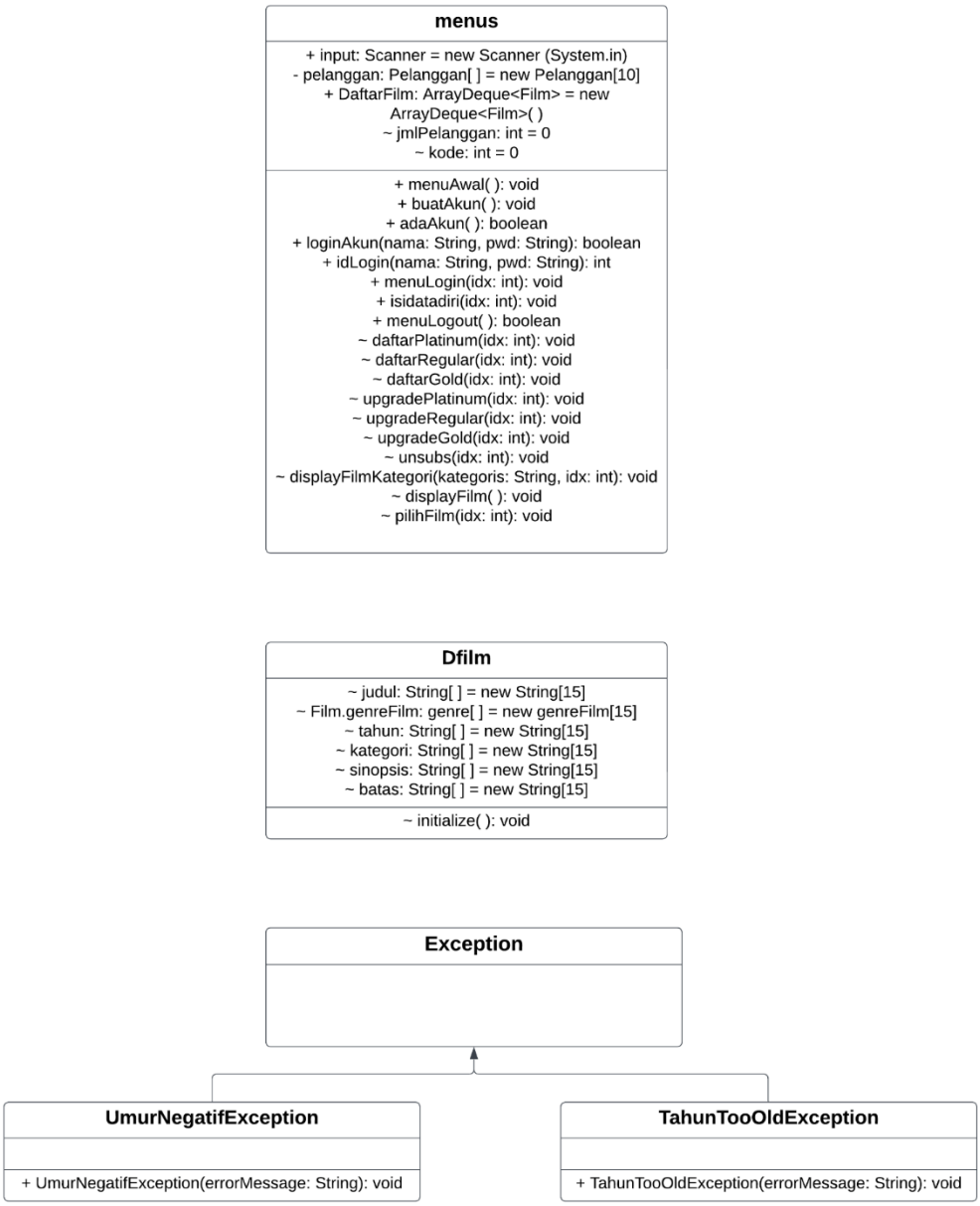
Link Video dan PPT:

https://drive.google.com/drive/folders/1X7uPjphEA4SjvsoFQHItvPxNBSAK3934?usp=share_link

1. Class Diagram




<<enumeration>>
genreFilm
SCIFI
HORROR
FANTASY
MYSTERY
COMEDY
DARKCOMEDY
ADVENTURE
ACTION
THRILLER
ANIME
- gen: String
+ getGenre( ): String
Film
judul: String
genre: genreFilm
tahun: String
kategori: String
batasUmur: String
sinopsis: String
+ displayFilm(i: int): void
+ displayFULL( ): void
filmReguler
+ displayFilm(i: int): void
filmNew
+ displayFilm(i: int): void
filmOri
+ displayFilm(i: int): void


menus
+ input: Scanner = new Scanner (System.in)
- pelanggan: Pelanggan[ ] = new Pelanggan[10]
+ DaftarFilm: ArrayDeque<Film> = new ArrayDeque<Film>( )
~ jmlPelanggan: int = 0
~ kode: int = 0
+ menuAwal( ): void
+ buatAkun( ): void
+ adaAkun( ): boolean
+ loginAkun(nama: String, pwd: String): boolean
+ idLogin(nama: String, pwd: String): int
+ menuLogin(idx: int): void
+ isidatadiri(idx: int): void
+ menuLogout( ): boolean
~ daftarPlatinum(idx: int): void
~ daftarRegular(idx: int): void
~ daftarGold(idx: int): void
~ upgradePlatinum(idx: int): void
~ upgradeRegular(idx: int): void
~ upgradeGold(idx: int): void
~ unsubs(idx: int): void
~ displayFilmKategori(kategoris: String, idx: int): void
~ displayFilm( ): void
~ pilihFilm(idx: int): void
Dfilm
~ judul: String[ ] = new String[15]
~ Film.genreFilm: genre[ ] = new genreFilm[15]
~ tahun: String[ ] = new String[15]
~ kategori: String[ ] = new String[15]
~ sinopsis: String[ ] = new String[15]
~ batas: String[ ] = new String[15]
~ initialize( ): void
Exception
UmurNegatifException
+ UmurNegatifException(errorMessage: String): void
TahunTooOldException
+ TahunTooOldException(errorMessage: String): void

Program JalaFlix yang dibuat terdiri dari 6 kelas, yaitu JalaFlix, Pelanggan, Film, menus, Dfilm, dan ExceptionHandling. Pelanggan dibagi menjadi 4 kelas turunan, yaitu pelanggan Reguler, Gold, Platinum dan Biasa. Kelas Film juga dibagi menjadi 3 sub class bernama Film Regular, New dan Ori.

1.1. JalaFlix

JalaFlix adalah kelas yang berisi method main dari program. Program akan dijalankan dari kelas main terlebih dahulu. Kelas ini tidak memiliki method lain selain main, dan hanya terdiri dari instansiasi beberapa object seperti input dan menu. Di dalam kelas main terdapat loop while yang selalu melakukan iterasi selama user tidak memilih untuk Keluar dari JalaFlix.

1.2. Pelanggan

Pelanggan adalah sebuah kelas yang mendeskripsikan setiap pelanggan yang ada. Kelas ini dideklarasikan sebagai sebuah ***abstract*** *class* dengan melakukan implementasi ***interface*** customer. Pelanggan memiliki beberapa atribut, seperti kodePelanggan, nama, umur, noTelp, kategori, status_keanggotaan, Password yang bertipe String dan histori yang bertipe ArrayDeque of Film dengan access modifier private.

Pelanggan menggunakan atribut dengan access modifier private. masing masing atribut memiliki method getter dan setter. Beberapa method lainnya yang terdapat pada kelas ini, antara lain:

1. Film getHistori(int i)

   Method getHistori menerima parameter i. Berbeda dengan method getter dan setter lainnya, method getHistori ini akan memberikan kembalian berupa sebuah Film yang merupakan histori ke-i pada list.
2. Void displayDataPelanggan()

   Method ini adalah sebuah method yang berfungsi untuk mencetak informasi pelanggan pada console. Informasi berupa nama, kodePelanggan, noTelp, kategori, dan status_keanggotaan.
3. Void addHistory(Film f)

Method addHistory(Film f) berfungsi menambahkan suatu film f pada history seorang pelanggan.

4. Void clearHistory()

   Method clearHistory() berfungsi untuk menghapus seluruh history yang dimiliki oleh suatu pelanggan.

5. Void displayHistory()

   Method displayHistory() berfungsi untuk menampilkan informasi mengenai film-film yang tersimpan di dalam history.

6. Int sizeHistori()

   Method sizeHistori() adalah method yang bertugas mengembalikan size dari ArrayDeque histori.

7. Void platinum()

   Memberikan output pada console untuk konfirmasi ketika Pelanggan ingin mengubah kategori ke platinum.

8. Void gold()

   Memberikan output pada console untuk konfirmasi ketika Pelanggan ingin mengubah kategori ke gold.

9. Void regular()

   Memberikan output pada console untuk konfirmasi ketika Pelanggan ingin mengubah kategori ke regular.

10. Void uns()

    Memberikan output pada console untuk konfirmasi ketika Pelanggan ingin mengubah kategori ke User Biasa (Unsubscribe).

11. Void yaTidak()

    Memberikan pilihan antara ya dan tidak ke user.

12. Void profil()

    Menampilkan profil dari pelanggan saat ini.

13. Void tambahFilm()

    Menambahkan film pada daftar film yang tersedia.

Kelas abstract ini memiliki abstract method platinum(), gold(), regular() dan uns() yang menjadi penanda bahwa seluruh kelas turunannya

harus memiliki keempat method tersebut. Setelah pelanggan melakukan login, pelanggan dapat melakukan modifikasi data. Salah satu data yang dapat dimodifikasi adalah umur. Jika umur yang diinput saat instansiasi objek pelanggan bernilai negatif, program akan melemparkan **UmurNegatifException.** Selain itu, pada penambahan film menggunakan method tambahFilm(), pelanggan mungkin saja memasukkan film dengan tahun rilis yang terlalu tua (<1800 tahun). Jika pada saat instasiasi objek film dan tahun rilisnya < 1800 tahun, maka program akan melemparkan **TahunTooOldException.**

1.3. Film

Film adalah kelas yang mendeskripsikan setiap film. Film memiliki beberapa atribut, seperti judul, genre, tahun, kategori, sinopsis dan batas umur yang diinisialisasi sebagai String dengan access modifier protected. Pada atribut genre, program mengimplementasikan tipe data **enum** yang berisi string konstanta yang akan menjadi genre dari suatu film. Enum digunakan untuk lebih mengspesifikasikan nilai yang boleh menjadi genre dari suatu film.

Method yang terdapat di dalam kelas Film antara lain:

1. Void displayFilm (int i)

   Method ini berfungsi untuk menghandle behavior dari dari Film ke-i ketika ingin di tampilkan ke console. Ini adalah method versi lebih ringkas daripada displayFULL().

2. Void displayFULL()

   Method ini berfungsi untuk menampilkan seluruh data dari suatu Film pada console.

Kelas Film memiliki 3 buah subclass yang merupakan inheritance dari kelas Film. Subclass dibagi menjadi 3 berdasarkan jenis Film yang tersedia, yaitu filmReguler, filmNew dan filmOri.

Pada masing masing subclass ini terdapat sebuah konstruktor yang menerima 5 parameter (judul, genre, tahun, kategori dan batas umur).

Konstruktor ini berfungsi untuk menyimpan nilai ke setiap atribut yang ada di parent class.

Selain itu, terdapat sebuah method bernama displayFilm(int i) yang menerima parameter i. Fungsi ini berperan dalam menampilkan beberapa informasi dari Film ke-i pada console.

1.4. menus

Kelas ini memiliki beberapa atribut yang terdiri dari Array of Object Pelanggan, ArrayDeque of Film Daftar Film, jmlPelanggan, dan kode. Dalam kelas menus terdapat beberapa method:

1. Void menuAwal()

   Method mengeluarkan output berupa pilihan kepada user untuk melakukan login, membuat akun atau keluar dari JalaFlix.

2. Void buatAkun()

   Method buatAkun() adalah method yang berfungsi untuk menerima input berupa nama dan password dari pengguna. Nantinya data nama dan password ini akan disimpan ke array pelanggan dengan menggunakan method setNama dan setPassword.

3. Boolean adaAkun()

   Method ini akan mengembalikan nilai kembalian berupa boolean mengenai ada tidaknya akun yang terdaftar pada JalaFlix.

4. Boolean loginAkun(String nama, String pwd)

   Method ini berfungsi untuk memeriksa apakah nama dan password yang diinput oleh user ketika login terdapat di dalam array pelanggan.

5. Int idLogin(String nama, String pwd)

   Method akan mengembalikan integer berupa id pelanggan yang sedang login.

6. Void menuLogin(int idx)

   Ketika user berhasil melakukan login, method menuLogin() akan dipanggil. Pada method ini terdapat menu utama yang memberikan output pada console berupa 7 pilihan menu. Menu tersebut terdiri dari:

- Daftar Kategori

  Daftar Kategori adalah menu yang mengarahkan pengguna untuk mendaftarkan kategori. Ketika pengguna sudah terdaftar sebagai salah satu kategori, pengguna tidak akan dapat memilih menu ini.
- Nonton Film

  Nonton Film adalah menu yang mengarahkan pengguna untuk menonton Film. Apabila pengguna adalah user biasa, program hanya akan menampilkan list dari film yang tersedia tanpa memberikan opsi untuk menonton.

  Ketika user merupakan salah satu dari kategori regular, gold atau platinum, list dari Film akan ditampilkan. Pengguna akan diberikan pilihan untuk menonton film ke berapa. Film yang dapat ditonton oleh pengguna adalah film yang memiliki warna. Jika pengguna mencoba untuk menonton film yang tidak dapat diakses, maka akan memberikan keterangan gagal menonton film pada console.
- Mengganti Kategori

  Mengganti kategori adalah menu yang mengarahkan pengguna untuk mengganti kategori. Ketika pengguna sudah terdaftar dalam suatu kategori, pengguna dapat mengganti ke kategori lainnya. Pengguna yang belum mendaftarkan kategori tidak dapat membuka menu ini.
- Lihat Daftar Film

  Menu ini akan memanggil method displayFilm yang akan menampilkan list film yang terdapat di JalaFlix.
- Logout

  Menu ini adalah menu untuk melakukan logout akun dan keluar dari JalaFlix atau beralih ke akun lainnya.
- Profil Akun

  Menu ini akan menampilkan informasi dari pengguna yang sedang login ke dalam console.

- History

  Menu ini berfungsi untuk menampilkan histori dari pelanggan ke dalam console.
- Menambahkan Film

  Suatu pengguna dapat menambahkan film pada list daftar Film yang tersedia. Pelanggan reguler mampu menambahkan 5 film, pelanggan gold 10 film dan tidak ada batasan untuk pelanggan platinum.

7. Void isidatadiri(int idx)

   Method ini mengisi program interaktif untuk menerima data diri dari pelanggan.
8. Boolean menuLogout()

   Method ini berfungsi mengonfirmasi kepada pelanggan apakah pelanggan ingin logout. Jika iya, method akan mengembalikan nilai true. Jika tidak, pelanggan akan mengembalikan nilai false.
9. Void daftarPlatinum(int idx)

   Method ini berfungsi untuk mendaftarkan pelanggan ke-idx ke kategori platinum.
10. Void daftarRegular(int idx)

    Method ini berfungsi untuk mendaftarkan pelanggan ke-idx ke kategori Regular.
11. Void daftarGold(int idx)

    Method ini berfungsi untuk mendaftarkan pelanggan ke-idx ke kategori Gold.
12. Void upgradePlatinum(int idx)

    Method ini berfungsi untuk mengubah kategori pelanggan ke-idx menjadi Platinum.
13. Void upgradeRegular(int idx)

    Method ini berfungsi untuk mengubah kategori pelanggan ke-idx menjadi Regular.
14. Void upgradeGold(int idx)

Method ini berfungsi untuk mengubah kategori pelanggan ke-idx menjadi Gold.

15. Void unsubs(int idx)

    Method ini berfungsi untuk mengubah kategori pelanggan ke-idx menjadi Unsubscribe atau User Biasa.

16. Void displayFilmKategori(String kategoris, int idx)

    Method ini menerima parameter berupa kategori dan idx. Method ini bertugas untuk menampilkan film yang dapat ditonton oleh suatu kategori tertentu yang dimiliki oleh pelanggan ke-idx.

17. Void displayFilm()

    Method ini berfungsi untuk mencetak output seluruh film yang terdapat di dalam daftar film ke dalam console.

18. Void pilihFilm(int idx)

    Method ini berfungsi untuk memilih film yang ingin ditonton.

1.5. DFilm

DFilm adalah kelas yang berisi data film-film yang terdapat di dalam JalaFlix. Data terdiri dari beberapa attribute, seperti judul, genre, tahun, kategori, sinopsis dan batasan umur. Data ini disimpan dalam struktur data array dan memiliki access modifier default.

1.6 ExceptionHandling (Class UmurNegatif Exception dan TahunTooOldException)

ExceptionHandling adalah sebuah file yang berisi pembuatan kelas turunan exception untuk menangani beberapa jenis exception seperti **UmurNegatifException** dan **TahunTooOldException**.

Source Code

1. Main JalaFlix

JalaFlix.java	
1.	`package komponen;`
2.	`import java.util.*;`
3.	
4.	`public class JalaFlix {`
5.	
6.	`    static Scanner input = new Scanner(System.in);`
7.	`    public static void main(String[] args) {`
8.	
9.	`        menus mn = new menus();`
10.	`        while(true){`
11.	`            mn.menuAwal();`
12.	`            String cmd = input.nextLine();`
13.	`            switch(cmd){`
14.	`                case "1":`
15.	`                    if(!mn.adaAkun()){`
16.	`System.out.println("\nTidak ada akun yang terdaftar");`
17.	`                        break;`
18.	`                    }`
19.	`                    System.out.println("\nLogin ke Akun Yang Sudah ada!");`
20.	`                    System.out.println("Nama: ");`
21.	`                    String nama = input.nextLine();`
22.	`System.out.println("Password: ");`

```
23.                         String pwd = input.nextLine();
24.                         boolean test = mn.loginAkun(nama, pwd);
25.                         if(test){
26.                             int xi = mn.idLogin(nama, pwd);
27.                             if(xi!=-1){
28.                                 mn.menuLogin(xi);
29.                             }else{
30.                                 System.out.println("Gagal Masuk!\n2");
31.                             }
32.                         }
33.                     break;
34.                     case "2":
35.                         mn.buatAkun();
36.                     break;
37.                     case"3":
38.                     System.out.println("\nKeluar JalaFlix. Terima Kasih");
39.                         return;
40.                     default:
41.                     System.out.println("Input salah!");
42.                 }
43.             }
44.         }
45.     }
46.
```

2. Kelas Pelanggan

Pelanggan.java	
1.	package komponen;
2.	
3.	
4.	import java.util.*;
5.	
6.	interface customer{
7.	void platinum();
8.	void gold();
9.	void uns();
10.	void regular();
11.	}
12.	
13.	public abstract class Pelanggan implements customer{
14.	private String kodePelanggan="";
15.	private String nama;
16.	private String umur;
17.	private String noTelp;
18.	private String kategori = "";
19.	private String status_keanggotaan= "";
20.	private String Password="";
21.	private ArrayDeque<Film> histori = new ArrayDeque<Film>();
22.	
23.	Pelanggan(String kodePelanggan, String nama, String noTelp, String kategori, String status_keanggotaan, String umur) throws UmurNegatifException{
24.	this.kodePelanggan = kodePelanggan;

```
25.       this.nama = nama;
26.       this.noTelp = noTelp;
27.       this.kategori = kategori;
28.       this.status_keanggotaan = status_keanggotaan;
29.       if(umur.length()==0){
30.             this.umur = umur;
31.       }else if(Character.compare(umur.charAt(0), '-')==0){
32.             throw new UmurNegatifException("\n\u001B[31mUmur harus positif! Gagal memperbarui umur!\u001B[0m\n");
33.       }else{
34.             try{
35.                   Integer.parseInt(umur);
36.                   this.umur = umur;
37.                   System.out.println("Berhasil memperbaharui data!\n");
38.             }catch(Exception e){
39.                   System.out.println("\n\u001B[31mMasukkan umur yang benar! gagal memperbarui umur!\u001B[0m\n");
40.             }
41.
42.       }
43.    }
44.
45.    Pelanggan(String kodePelanggan, String nama, String noTelp, String kategori, String status_keanggotaan, String Password, ArrayDeque<Film> histori, String umur){
```

```
46.       this.kodePelanggan = kodePelanggan;
47.       this.nama = nama;
48.       this.noTelp = noTelp;
49.       this.kategori = kategori;
50.       this.status_keanggotaan = status_keanggotaan;
51.       this.Password = Password;
52.       this.histori = histori;
53.       this.umur = umur;
54.    }
55.
56.    Pelanggan(){
57.
58.    }
59.
60.    public void displayDataPelanggan(){
61.          System.out.println("\nData Akun:");
62.          System.out.printf("%-20s: %s\n", "Nama", nama);
63.          System.out.printf("%-20s: %s\n", "Kode Pelanggan", kodePelanggan);
64.          System.out.printf("%-20s: %s\n", "Umur", umur);
65.          System.out.printf("%-20s: %s\n", "No. Telepon", noTelp);
66.          System.out.printf("%-20s: %s\n", "Kategori", kategori);
67.          System.out.printf("%-20s: %s\n", "Status Keanggotaan", status_keanggotaan);
68.    }
69.
70.    String getKodePelanggan(){
```

```
71.         return this.kodePelanggan;
72.     }
73.
74.     String getNama(){
75.         return this.nama;
76.     }
77.
78.     String getNoTelp(){
79.         return this.noTelp;
80.     }
81.
82.     String getKategori(){
83.         return kategori;
84.     }
85.
86.     Film getHistori(int i){
87.       int id = 1;
88.       Film a = new Film();
89.          for(Film x: histori){
90.            if(id == i){
91.                 try{
92.                     a = new Film(x.judul, x.genre, x.tahun, x.kategori, x.sinopsis, x.batasUmur);
93.                 }catch(TahunTooOldException ex){
94.
System.out.println(ex.getMessage());
95.                 }
96.            }
97.          }
98.       return a;
```

```
99.     }
100.
101.    String getPassword(){
102.          return Password;
103.    }
104.
105.    String getStatus(){
106.       return status_keanggotaan;
107.    }
108.    String getUmur(){
109.      return umur;
110.    }
111.
112.    void setNama(String nama){
113.          this.nama = nama;
114.    }
115.
116.    void setNoTelp(String noTelp){
117.          this.noTelp = noTelp;
118.    }
119.
120.    void setPassword(String password){
121.        this.Password = password;
122.    }
123.
124.    void setKategori(String kategori){
125.          this.kategori = kategori;
126.    }
127.
128.    void setStatus(String s){
129.      status_keanggotaan = s;
```

```
130.    }
131.
132.    void setKodePelanggan(String s){
133.      kodePelanggan = s;
134.    }
135.
136.    void setHistori(ArrayDeque<Film> histori){
137.      this.histori = histori;
138.    }
139.
140.    void setUmur(String umur){
141.      this.umur = umur;
142.    }
143.
144.
145.    void addHistory(Film f){
146.          if(histori.size()<10){
147.              histori.addFirst(f);
148.          }else{
149.              histori.addFirst(f);
150.              histori.removeLast();
151.          }
152.    }
153.
154.    void clearHistory(){
155.          histori.clear();
156.    }
157.
158.    void isHapusHistory(){
159.      this.displayHistory();
```

```
160.      System.out.println("\nIngin Menghapus Histori?");
161.      yaTidak();
162.    }
163.
164.    void displayHistory(){
165.          int id = 1;
166.          System.out.println("\nHistory tontonan: ");
167.          System.out.printf("%-4s%-40s|%-25s|%-8s|%-10s|\n", "","Judul", "Genre", "Tahun", "Kategori");
168.          for(Film x: histori){
169.              System.out.printf("%d.    |%-40s|%-25s|%-8s|%-10s|\n", id, x.judul, x.genre, x.tahun, x.kategori);
170.              id++;
171.          }
172.    }
173.
174.    ArrayDeque<Film> dapatHistori(){
175.      return histori;
176.    }
177.
178.    int sizeHistori(){
179.      return histori.size();
180.    }
181.
182.    void yaTidak(){
183.      System.out.println("1. YA");
184.      System.out.println("2. TIDAK");
```

```
185.    }
186.
187.    void profil(){
188.        System.out.println("Informasi Akun Anda: ");
189.        this.displayDataPelanggan();
190.        System.out.println("\nIngin mengganti/melengkapi Informasi Akun?");
191.        this.yaTidak();
192.    }
193.
194.    public abstract void platinum();
195.    public abstract void gold();
196.    public abstract void regular();
197.    public abstract void uns();
198.
199.    public void tambahFilm(){
200.        Scanner input = new Scanner(System.in);
201.        if(this.kategori.equals("User Biasa")){
202.            System.out.println("Anda harus daftar kategori terlebih dahulu!");
203.        }else{
204.            System.out.println("\nMasukkan data Film!");
205.            String judul,genre, tahun, kategori, umur, sinopsis;
206.            Film.genreFilm gen = null;
207.            System.out.print("Judul: ");
208.            judul = input.nextLine();
209.            while(gen == null){
210.                try{
```

```
211.                        System.out.print("Genre
(SCIFI, ACTION, FANTASY, COMEDY): ");
212.                        genre = input.nextLine();
213.                        gen                          =
Film.genreFilm.valueOf(genre);
214.                }catch(Exception e){
215.                        System.out.println("Genre
Salah, Masukkan Ulang! ");
216.                }
217.            }
218.            System.out.print("Tahun: ");
219.            tahun = input.nextLine();
220.            System.out.print("Kategori: ");
221.            kategori = input.nextLine();
222.            System.out.print("Batas Umur: ");
223.            umur = input.nextLine();
224.            System.out.print("Sinopsis: ");
225.            sinopsis = input.nextLine();
226.            Film a = new Film();
227.
228.
229.            try{
230.                a = new Film(judul, gen, tahun,
kategori, sinopsis, umur);
231.                menus.DaftarFilm.addLast(a);
232.            }catch(TahunTooOldException ex){
233.
System.out.println(ex.getMessage());
234.            }
235.
236.
```

```
237.        }
238.
239.    }
240.
241.
242. }
243.
244. class pelangganRegular extends Pelanggan{
245.         int jmlFilmTambah = 0;
246.         pelangganRegular(String kodePelanggan, String nama, String noTelp, String kategori, String status_keanggotaan, String Password, ArrayDeque<Film> histori, String umur){
247.             super(kodePelanggan, nama, noTelp, kategori, status_keanggotaan, Password, histori, umur);
248.         }
249.
250.         public void platinum(){
251.             System.out.println("Ingin Upgrade ke Platinum? - Harga Rp.75000");
252.             yaTidak();
253.         }
254.
255.         public void gold(){
256.             System.out.println("Ingin Upgrade ke Gold? - Harga Rp.35000");
257.             yaTidak();
258.         }
259.
260.         public void regular(){
```

```
261.            System.out.println("Anda tidak akan upgrade - Harga Rp.0");
262.            yaTidak();
263.        }
264.
265.        public void uns(){
266.            System.out.println("Yakin ingin unsubscribe?");
267.            yaTidak();
268.        }
269.
270.        public void tambahFilm(){
271.            if(jmlFilmTambah<5){
272.                super.tambahFilm();
273.                jmlFilmTambah++;
274.            }else{
275.                System.out.println("\n Tidak bisa menambah Film Lagi! \n");
276.            }
277.        }
278.    }
279.
280.    class pelangganGold extends Pelanggan{
281.        int jmlFilmTambah = 0;
282.        pelangganGold(String kodePelanggan, String nama, String noTelp, String kategori, String status_keanggotaan, String Password, ArrayDeque<Film> histori, String umur){
283.            super(kodePelanggan, nama, noTelp, kategori, status_keanggotaan, Password, histori, umur);
284.        }
```

```
285.
286.      public void platinum(){
287.            System.out.println("Ingin Upgrade ke
          Platinum? - Harga Rp.50000");
288.            yaTidak();
289.      }
290.
291.      public void gold(){
292.            System.out.println("Anda tidak akan
          upgrade - Harga Rp.0");
293.            yaTidak();
294.      }
295.
296.      public void regular(){
297.            System.out.println("Ingin downgrade
          ke Regular? - Harga Rp.0");
298.            yaTidak();
299.      }
300.
301.      public void uns(){
302.            System.out.println("Yakin ingin
          unsubscribe?");
303.            yaTidak();
304.      }
305.
306.      public void tambahFilm(){
307.            if(jmlFilmTambah<10){
308.                  super.tambahFilm();
309.                  jmlFilmTambah++;
310.            }else{
```

```
311.                System.out.println("\n      Tidak bisa menambah Film Lagi! \n");
312.            }
313.        }
314.
315.    }
316.
317.    class pelangganPlatinum extends Pelanggan{
318.        int jmlFilmTambah = 0;
319.        pelangganPlatinum(String kodePelanggan, String nama, String noTelp, String kategori, String status_keanggotaan, String Password, ArrayDeque<Film> histori, String umur){
320.            super(kodePelanggan, nama, noTelp, kategori, status_keanggotaan, Password, histori, umur);
321.        }
322.
323.        public void platinum(){
324.            System.out.println("Anda tidak akan upgrade - Harga Rp.0");
325.            yaTidak();
326.        }
327.
328.        public void gold(){
329.            System.out.println("Ingin downgrade ke Gold? - Harga Rp.0");
330.            yaTidak();
331.        }
332.
333.        public void regular(){
```

```
334.            System.out.println("Ingin downgrade ke Regular? - Harga Rp.0");
335.            yaTidak();
336.        }
337.
338.        public void uns(){
339.            System.out.println("Yakin ingin unsubscribe?");
340.            yaTidak();
341.        }
342.
343.        public void tambahFilm(){
344.            super.tambahFilm();
345.            jmlFilmTambah++;
346.        }
347.    }
348.
349.    class pelangganBiasa extends Pelanggan{
350.        pelangganBiasa(String kodePelanggan, String nama, String noTelp, String kategori, String status_keanggotaan, String Password, ArrayDeque<Film> histori, String umur){
351.            super(kodePelanggan, nama, noTelp, kategori, status_keanggotaan, Password, histori, umur);
352.        }
353.        pelangganBiasa(String kodePelanggan, String nama, String noTelp, String kategori, String status_keanggotaan, String umur)throws UmurNegatifException{
354.            super(kodePelanggan, nama, noTelp, kategori, status_keanggotaan, umur);
```

```
355.        }
356.        public void platinum(){
357.                System.out.println("Ingin membeli Platinum - Harga Rp.100000");
358.                yaTidak();
359.        }
360.        public void gold(){
361.                System.out.println("Ingin membeli Gold - Harga Rp.50000");
362.                yaTidak();
363.        }
364.
365.        public void regular(){
366.                System.out.println("Ingin membeli Regular - Harga Rp.15000");
367.                yaTidak();
368.        }
369.
370.        public void uns(){
371.                System.out.println("Anda Pelanggan biasa - Unsubscribe berarti tidak ada perubahan");
372.        }
373.
374.
375.    }
376.
```

3. Film

Film.java

```
1.  package komponen;
```

```
2.  public class Film {
3.      protected String judul;
4.      protected genreFilm genre;
5.      protected String tahun;
6.      protected String kategori;
7.      protected String batasUmur;
8.      protected String sinopsis;
9.      Film(){
10.
11.     }
12.
13.     public enum genreFilm{
14.         SCIFI("Science Fiction"),
15.         HORROR("Horror"),
16.         FANTASY("Fantasy"),
17.         MYSTERY("Mystery"),
18.         COMEDY("Comdedy"),
19.         DARKCOMEDY("Dark Comedy"),
20.         ADVENTURE("Adventure"),
21.         ACTION("Action"),
22.         THRILLER("Thriller"),
23.         ANIME("Anime");
24.
25.
26.         private String gen;
27.
28.         private genreFilm(String gen){
29.             this.gen = gen;
30.         }
31.
32.         public String getGenre(){
```

```
33.             return gen;
34.         }
35.
36.     }
37.
38.     Film(String judul, genreFilm genre, String tahun, String kategori, String sinopsis, String batasUmur) throws TahunTooOldException{
39.         int thn = 0;
40.         if(!tahun.equals("")){
41.             try{
42.                 thn = Integer.parseInt(tahun);
43.             }catch(Exception e){
44.                 System.out.println("\n Masukkan tahun yang benar!");
45.                 return;
46.             }
47.         }else if(tahun.equals("")){
48.
49.         }
50.
51.         if(!tahun.equals("")){
52.             if(thn<1800){
53.                 throw new TahunTooOldException("\n\u001B[31mTahun film terlalu tua! gagal menambahkan film.\u001B[0m\n");
54.             }
55.         }
56.
57.         this.judul = judul;
```

```
58.         this.genre = genre;
59.         this.tahun = tahun;
60.         this.kategori = kategori;
61.         this.sinopsis = sinopsis;
62.         this.batasUmur = batasUmur;
63.     }
64.
65.     public void displayFilm(int i){
66.
67.         System.out.printf("%3s %-40s|%-25s|%-8s|%-10s|%-5s|\n", (i+1)+".", judul, genre, tahun, kategori, batasUmur);
68.     }
69.
70.     public void displayFULL(){
71.         System.out.println("\nJudul: "+judul);
72.         System.out.println("Genre: "+genre);
73.         System.out.println("Tahun: "+tahun);
74.         System.out.println("Kategori: "+kategori);
75.         System.out.println("Sinopsis: "+sinopsis);
76.         System.out.println("Batas: "+batasUmur);
77.     }
78.
79. }
80.
81. class filmReguler extends Film{
82.     filmReguler(String judul, genreFilm genre, String tahun, String kategori, String batasUmur)throws TahunTooOldException{
```

83.	super(judul, genre, tahun, kategori, kategori, batasUmur);
84.	}
85.	public void displayFilm(int i){
86.	System.out.printf("\u001B[33m%3s %-40s|%-25s|%-8s|%-10s|%-5s|\u001B[0m\n", (i+1)+".", super.judul, super.genre, super.tahun, super.kategori, super.batasUmur);
87.	}
88.	}
89.	
90.	class filmNew extends Film{
91.	filmNew(String judul, genreFilm genre, String tahun, String kategori, String batasUmur)throws TahunTooOldException{
92.	super(judul, genre, tahun, kategori, kategori, batasUmur);
93.	}
94.	public void displayFilm(int i){
95.	System.out.printf("\u001B[32m%3s %-40s|%-25s|%-8s|%-10s|%-5s|\u001B[0m\n", (i+1)+".", super.judul, super.genre, super.tahun, super.kategori, super.batasUmur);
96.	}
97.	}
98.	
99.	class filmOri extends Film{
100.	filmOri(String judul, genreFilm genre, String tahun, String kategori, String batasUmur)throws TahunTooOldException{
101.	super(judul, genre, tahun, kategori, kategori, batasUmur);

```
102.     }
103.     public void displayFilm(int i){
104.         System.out.printf("\u001B[34m%3s %-40s|%-25s|%-8s|%-10s|%-5s|\u001B[0m\n", (i+1)+".", super.judul, super.genre, super.tahun, super.kategori, super.batasUmur);
105.     }
106. }
107.
```

4. Kelas menus

menus.java

```
1.  package komponen;
2.  import java.util.*;
3.
4.  class menus {
5.      public Scanner input = new Scanner(System.in);
6.      private Pelanggan[] pelanggan = new Pelanggan[10];
7.      public static ArrayDeque<Film> DaftarFilm = new ArrayDeque<Film>();
8.      int jmlPelanggan = 0;
9.      int kode = 0;
10.
11.     menus(){
12.         Dfilm fil = new Dfilm();
13.         for(int i=0; i<15; i++){
14.             Film temp = new Film();
15.             try{
```

16.	temp = new Film(fil.judul[i], fil.genre[i], fil.tahun[i], fil.kategori[i], fil.sinopsis[i], fil.batas[i]);
17.	DaftarFilm.addLast(temp);
18.	}catch(TahunTooOldException ex){
19.	System.out.println(ex.getMessage());
20.	}
21.	
22.	}
23.	for(int i=0; i<10; i++){
24.	try{
25.	pelanggan[i] = new pelangganBiasa("","","","User Biasa","", "");
26.	}catch(UmurNegatifException ex){
27.	System.out.println(ex.getMessage());
28.	}
29.	}
30.	}
31.	
32.	public void menuAwal(){
33.	System.out.println("Selamat datang di JalaFlix! (Angka)");
34.	System.out.println("1. Login");
35.	System.out.println("2. Buat Akun");
36.	System.out.println("3. Keluar dari JalaFlix");
37.	}
38.	
39.	public void buatAkun(){
40.	System.out.println("Nama: ");

41.	pelanggan[jmlPelanggan].setNama(input.nextLine()) ;
42.	System.out.println("Password: ");
43.	pelanggan[jmlPelanggan].setPassword(input.nextLine());
44.	System.out.println("Akun Berhasil dibuat!");
45.	pelanggan[jmlPelanggan].setStatus("Aktif");
46.	pelanggan[jmlPelanggan].setKodePelanggan("User-0"+kode);
47.	kode++;
48.	jmlPelanggan++;
49.	}
50.	
51.	public boolean adaAkun(){
52.	if(jmlPelanggan==0){
53.	return false;
54.	}
55.	return true;
56.	}
57.	
58.	public boolean loginAkun(String nama, String pwd){
59.	for(int i=0; i<jmlPelanggan; i++){
60.	if(pelanggan[i].getNama().equals(nama)){
61.	if(pelanggan[i].getPassword().equals(pwd)){

```
62.                         return true;
63.                     }
64.                 }
65.             }
66.             System.out.println("Nama/Password
    Salah atau tidak ditemukan!");
67.             return false;
68.         }
69.
70.         public int idLogin(String nama, String
    pwd){
71.             for(int i=0; i<jmlPelanggan; i++){
72.
    if(pelanggan[i].getNama().equals(nama)){
73.
    if(pelanggan[i].getPassword().equals(pwd)){
74.
    System.out.println("Berhasil Login!\n");
75.                         return i;
76.                     }
77.                 }
78.             }
79.             return -1;
80.         }
81.
82.         public void menuLogin(int idx){
83.             while(true){
84.                 System.out.println("Selamat datang
    di JalaFlix! Silahkan pilih menu! (Angka)");
85.                 System.out.println("1.       Daftar
    Kategori");
```

86.	System.out.println("2. Nonton Film");
87.	System.out.println("3. Mengganti Kategori");
88.	System.out.println("4. Lihat Daftar Film");
89.	System.out.println("5. Logout");
90.	System.out.println("6. Profil Akun dan Lengkapi Profil");
91.	System.out.println("7. History");
92.	System.out.println("8. Menambahkan Film");
93.	String id = input.nextLine();
94.	switch(id){
95.	case"1":
96.	if(pelanggan[idx].getKategori().equals("User Biasa")){
97.	System.out.println("Pilih Kategori yang diinginkan! (angka)");
98.	System.out.println("1. Regular");
99.	System.out.println("2. Gold");
100.	System.out.println("3. Platinum");
101.	String angka = input.nextLine();
102.	switch(angka){
103.	case "1":

104.	this.daftarRegular(idx);
105.	break;
106.	case "2":
107.	this.daftarGold(idx);
108.	break;
109.	case "3":
110.	this.daftarPlatinum(idx);
111.	break;
112.	default:
113.	System.out.println("Input Salah!!");
114.	}
115.	}else{
116.	System.out.println("Anda sudah terdaftar sebagai pelanggan "+pelanggan[idx].getKategori());
117.	}
118.	break;
119.	case"2":
120.	String anggota = pelanggan[idx].getKategori();
121.	displayFilmKategori(anggota, idx);
122.	break;
123.	case"3":
124.	if(!pelanggan[idx].getKategori().equals("")){

```
125.  System.out.println("Anda sedang terdaftar sebagai pelanggan "+pelanggan[idx].getKategori());
126.  System.out.println("Ingin mengganti kategori?");
127.                        System.out.println("1. Regular");
128.                        System.out.println("2. Gold");
129.                        System.out.println("3. Platinum");
130.                        System.out.println("4. Unsubscribe");
131.                        String cmd = input.nextLine();
132.                        switch(cmd){
133.                            case "1":
134.  this.upgradeRegular(idx);
135.                            break;
136.                            case "2":
137.  this.upgradeGold(idx);
138.                            break;
139.                            case "3":
140.  this.upgradePlatinum(idx);
141.                            break;
142.                            case"4":
143.                            this.unsubs(idx);
```

```
144.                                    break;
145.                                    default:
146.    System.out.println("Gagal Mengganti Kategori");
147.                                }
148.                            }else{
149.    System.out.println("Anda belum terdaftar dalam kategori manapun! Daftar terlebih dahulu");
150.                            }
151.                                break;
152.                        case"4":
153.                            System.out.println("Daftar Film JafaFlix: \n");
154.                            displayFilm();
155.                        break;
156.                        case"5":
157.                            if(menuLogout()) return;
158.                        break;
159.                        case"6":
160.                            pelanggan[idx].profil();
161.                            String cmd = input.nextLine();
162.                            switch(cmd){
163.                                case"1":
164.                                    isidatadiri(idx);
165.                                break;
166.                                default:
167.    System.out.println("Gagal mengganti data akun\n");
168.                            }
```

```
169.                    break;
170.                    case "7":
171.    if(pelanggan[idx].dapatHistori().size()==0){
172.                        System.out.println("Tidak
        ada Histori\n");
173.                    }else{
174.    pelanggan[idx].displayHistory();
175.    System.out.println("\nApakah ingin menghapus
        History?");
176.    pelanggan[idx].yaTidak();
177.                        String cmd2 =
        input.nextLine();
178.                        switch(cmd2){
179.                            case"1":
180.    pelanggan[idx].clearHistory();
181.    System.out.println("\nBerhasil menghapus
        History!");
182.                            break;
183.                            default:
184.    System.out.println("Gagal Menghapus!");
185.                        }
186.                    }
187.                    break;
188.                    case "8":
```

```
189.  pelanggan[idx].tambahFilm();
190.                  break;
191.                  default:
192.                  System.out.println("Input Salah!");
193.              }
194.          }
195.
196.      }
197.
198.      public void isidatadiri(int idx){
199.          String nama ="";
200.          String noTelp="";
201.          String umur = "";
202.          System.out.println("Mengganti data akun!");
203.          System.out.println("Nama");
204.          nama = input.nextLine();
205.          System.out.println("No.Telepon:");
206.          noTelp = input.nextLine();
207.          System.out.println("Umur: ");
208.          umur = input.nextLine();
209.
210.          try{
211.              ArrayDeque<Film> his = pelanggan[idx].dapatHistori();
212.              String pass = pelanggan[idx].getPassword();
213.              pelanggan[idx] = new pelangganBiasa(pelanggan[idx].getKodePelanggan
```

	(), nama, noTelp, pelanggan[idx].getKategori(), pelanggan[idx].getStatus(), umur);
214.	pelanggan[idx].setHistori(his);
215.	pelanggan[idx].setPassword(pass);
216.	}catch(UmurNegatifException ex){
217.	System.out.println(ex.getMessage());
218.	}
219.	
220.	}
221.	
222.	public boolean menuLogout(){
223.	while(true){
224.	System.out.println("Anda Yakin Ingin Logout?");
225.	System.out.println("1.YA");
226.	System.out.println("2.TIDAK");
227.	String id = input.nextLine();
228.	switch(id){
229.	case"1":
230.	return true;
231.	case"2":
232.	return false;
233.	default:
234.	System.out.println("Input Salah!");
235.	}
236.	}
237.	}
238.	
239.	void daftarPlatinum(int idx){

```
240.            pelanggan[idx].platinum();
241.            String angka3 = input.nextLine();
242.            switch(angka3){
243.                case"1":
244.                    System.out.println("Berhasil
Mendaftar ke kategori Platinum");
245.
pelanggan[idx].setKategori("Platinum");
246.                    String kat =
pelanggan[idx].getKodePelanggan();
247.                    if(kat.contains("User")){
248.
pelanggan[idx].setKodePelanggan(kat.replace("U
ser", "PLT"));
249.                    }else if(kat.contains("REG")){
250.
pelanggan[idx].setKodePelanggan(kat.replace("R
EG", "PLT"));
251.                    }else
if(kat.contains("GOLD")){
252.
pelanggan[idx].setKodePelanggan(kat.replace("G
OLD", "PLT"));
253.                    }
254.                    pelanggan[idx] = new
pelangganPlatinum(pelanggan[idx].getKodePelang
gan(),                  pelanggan[idx].getNama(),
pelanggan[idx].getNoTelp(),
pelanggan[idx].getKategori(),
pelanggan[idx].getStatus(),
pelanggan[idx].getPassword(),
```

	pelanggan[idx].dapatHistori(), pelanggan[idx].getUmur());
255.	break;
256.	default:
257.	System.out.println("Gagal Mendaftar kategori");
258.	}
259.	}
260.	
261.	void daftarRegular(int idx){
262.	pelanggan[idx].regular();
263.	String angka3 = input.nextLine();
264.	switch(angka3){
265.	case"1":
266.	System.out.println("Berhasil Mendaftar ke kategori Regular");
267.	pelanggan[idx].setKategori("Regular");
268.	String kat = pelanggan[idx].getKodePelanggan();
269.	if(kat.contains("User")){
270.	pelanggan[idx].setKodePelanggan(kat.replace("User", "REG"));
271.	}else if(kat.contains("PLT")){
272.	pelanggan[idx].setKodePelanggan(kat.replace("PLT", "REG"));
273.	}else if(kat.contains("GOLD")){

```
274.  pelanggan[idx].setKodePelanggan(kat.replace("GOLD", "REG"));
275.                  }
276.                  pelanggan[idx] = new pelangganRegular(pelanggan[idx].getKodePelanggan(), pelanggan[idx].getNama(), pelanggan[idx].getNoTelp(), pelanggan[idx].getKategori(), pelanggan[idx].getStatus(), pelanggan[idx].getPassword(), pelanggan[idx].dapatHistori(), pelanggan[idx].getUmur());
277.              break;
278.              default:
279.                  System.out.println("Gagal Mendaftar kategori");
280.          }
281.      }
282.
283.      void daftarGold(int idx){
284.          pelanggan[idx].gold();
285.          String angka3 = input.nextLine();
286.          switch(angka3){
287.              case"1":
288.                  System.out.println("Berhasil Mendaftar ke kategori Gold");
289.  pelanggan[idx].setKategori("Gold");
290.                  String kat = pelanggan[idx].getKodePelanggan();
291.                  if(kat.contains("User")){
```

```
292.  pelanggan[idx].setKodePelanggan(kat.replace("User", "GOLD"));
293.                  }else if(kat.contains("REG")){
294.  pelanggan[idx].setKodePelanggan(kat.replace("REG", "GOLD"));
295.                  }else if(kat.contains("PLT")){
296.  pelanggan[idx].setKodePelanggan(kat.replace("PLT", "GOLD"));
297.                  }
298.                  pelanggan[idx] = new pelangganGold(pelanggan[idx].getKodePelanggan(), pelanggan[idx].getNama(), pelanggan[idx].getNoTelp(), pelanggan[idx].getKategori(), pelanggan[idx].getStatus(), pelanggan[idx].getPassword(), pelanggan[idx].dapatHistori(), pelanggan[idx].getUmur());
299.              break;
300.              default:
301.                  System.out.println("Gagal Mendaftar kategori");
302.          }
303.      }
304.
305.      void upgradePlatinum(int idx){
306.          pelanggan[idx].platinum();
307.          String angka3 = input.nextLine();
308.          switch(angka3){
```

```
309.            case"1":
310.                System.out.println("Berhasil
    Mendaftar ke kategori Platinum");
311.
    pelanggan[idx].setKategori("Platinum");
312.                String kat =
    pelanggan[idx].getKodePelanggan();
313.                if(kat.contains("User")){
314.
    pelanggan[idx].setKodePelanggan(kat.replace("U
    ser", "PLT"));
315.                }else if(kat.contains("REG")){
316.
    pelanggan[idx].setKodePelanggan(kat.replace("R
    EG", "PLT"));
317.                }else
    if(kat.contains("GOLD")){
318.
    pelanggan[idx].setKodePelanggan(kat.replace("G
    OLD", "PLT"));
319.                }
320.                pelanggan[idx] = new
    pelangganPlatinum(pelanggan[idx].getKodePelang
    gan(), pelanggan[idx].getNama(),
    pelanggan[idx].getNoTelp(),
    pelanggan[idx].getKategori(),
    pelanggan[idx].getStatus(),
    pelanggan[idx].getPassword(),
    pelanggan[idx].dapatHistori(),
    pelanggan[idx].getUmur());
321.            break;
322.            default:
```

```
323.                    System.out.println("Gagal
    Mendaftar kategori");
324.            }
325.        }
326.
327.        void upgradeRegular(int idx){
328.            pelanggan[idx].regular();
329.            String angka3 = input.nextLine();
330.            switch(angka3){
331.                case"1":
332.                    System.out.println("Berhasil
    Mendaftar ke kategori Regular");
333.
    pelanggan[idx].setKategori("Regular");
334.                    String          kat          =
    pelanggan[idx].getKodePelanggan();
335.                    if(kat.contains("User")){
336.
    pelanggan[idx].setKodePelanggan(kat.replace("U
    ser", "REG"));
337.                    }else if(kat.contains("PLT")){
338.
    pelanggan[idx].setKodePelanggan(kat.replace("P
    LT", "REG"));
339.                    }else
    if(kat.contains("GOLD")){
340.
    pelanggan[idx].setKodePelanggan(kat.replace("G
    OLD", "REG"));
341.                    }
342.                    pelanggan[idx]      =      new
    pelangganRegular(pelanggan[idx].getKodePelangg
```

	an(), pelanggan[idx].getNama(), pelanggan[idx].getNoTelp(), pelanggan[idx].getKategori(), pelanggan[idx].getStatus(), pelanggan[idx].getPassword(), pelanggan[idx].dapatHistori(), pelanggan[idx].getUmur());
343.	break;
344.	default:
345.	System.out.println("Gagal Mendaftar kategori");
346.	}
347.	}
348.	
349.	void upgradeGold(int idx){
350.	pelanggan[idx].gold();
351.	String angka3 = input.nextLine();
352.	switch(angka3){
353.	case"1":
354.	System.out.println("Berhasil Mendaftar ke kategori Gold");
355.	pelanggan[idx].setKategori("Gold");
356.	String kat = pelanggan[idx].getKodePelanggan();
357.	if(kat.contains("User")){
358.	pelanggan[idx].setKodePelanggan(kat.replace("User", "GOLD"));
359.	}else if(kat.contains("REG")){

```
360.  pelanggan[idx].setKodePelanggan(kat.replace("REG", "GOLD"));
361.                  }else if(kat.contains("PLT")){
362.  pelanggan[idx].setKodePelanggan(kat.replace("PLT", "GOLD"));
363.                  }
364.                  pelanggan[idx] = new pelangganGold(pelanggan[idx].getKodePelanggan(), pelanggan[idx].getNama(), pelanggan[idx].getNoTelp(), pelanggan[idx].getKategori(), pelanggan[idx].getStatus(), pelanggan[idx].getPassword(), pelanggan[idx].dapatHistori(), pelanggan[idx].getUmur());
365.              break;
366.              default:
367.                  System.out.println("Gagal Mendaftar kategori");
368.          }
369.      }
370.
371.      void unsubs(int idx){
372.          pelanggan[idx].uns();
373.          String angka3 = input.nextLine();
374.          switch(angka3){
375.              case"1":
376.                  System.out.println("Berhasil Unsubscribe");
```

377.	pelanggan[idx].setKategori("User Biasa");
378.	String kat = pelanggan[idx].getKodePelanggan();
379.	if(kat.contains("User")){
380.	pelanggan[idx].setKodePelanggan(kat.replace("User", "User"));
381.	}else if(kat.contains("REG")){
382.	pelanggan[idx].setKodePelanggan(kat.replace("REG", "User"));
383.	}else if(kat.contains("GOLD")){
384.	pelanggan[idx].setKodePelanggan(kat.replace("GOLD", "User"));
385.	}else{
386.	pelanggan[idx].setKodePelanggan(kat.replace("PLT", "User"));
387.	}
388.	pelanggan[idx] = new pelangganBiasa(pelanggan[idx].getKodePelanggan(), pelanggan[idx].getNama(), pelanggan[idx].getNoTelp(), pelanggan[idx].getKategori(), pelanggan[idx].getStatus(), pelanggan[idx].getPassword(), pelanggan[idx].dapatHistori(), pelanggan[idx].getUmur());
389.	break;

```
390.             default:
391.                 System.out.println("Gagal
Melakukan Unsubscribe");
392.         }
393.
394.     }
395.
396.     void displayFilmKategori(String kategoris,
int idx){
397.         if(kategoris.equals("User Biasa")){
398.             this.displayFilm();
399.         } else if(kategoris.equals("Regular")){
400.             System.out.println("\nKamu     User
Regular, kamu dapat mengakses film Reguler:
\n");
401.
System.out.println("\033[38;5;94m============
============================================
==================================\033[0m");
402.             System.out.printf("%-4s%-40s|%-
25s|%-8s|%-10s|%-5s|\n",  "","Judul",  "Genre",
"Tahun", "Kategori", "Umur");
403.             int id = 0;
404.             for(Film x: DaftarFilm){
405.
if(x.kategori.equals("Regular")){
406.                     try{
407.                         x        =        new
filmReguler(x.judul,     x.genre,     x.tahun,
x.kategori, x.batasUmur);
408.
}catch(TahunTooOldException ex){
```

409.	System.out.println(ex.getMessage());
410.	}
411.	x.displayFilm(id);
412.	}else{
413.	x.displayFilm(id);
414.	}
415.	id++;
416.	}
417.	System.out.println("\033[38;5;94m========== ==========\033[0m");
418.	System.out.println();
419.	pilihFilm(idx);
420.	} else if(kategoris.equals("Gold")){
421.	System.out.println("\nKamu User Gold, Kamu dapat mengakses film reguler dan new:\n");
422.	System.out.println("\033[38;5;220m========== ==========\033[0m");
423.	System.out.printf("%-4s%-40s|%-25s|%-8s|%-10s|%-5s|\n", "","Judul", "Genre", "Tahun", "Kategori", "Umur");
424.	int id = 0;
425.	for(Film x: DaftarFilm){
426.	if(x.kategori.equals("Regular")){
427.	try{

```
428.                                x = new filmReguler(x.judul, x.genre, x.tahun, x.kategori, x.batasUmur);
429.    }catch(TahunTooOldException ex){
430.    System.out.println(ex.getMessage());
431.                            }
432.
433.                            x.displayFilm(id);
434.                        }else if(x.kategori.equals("New")){
435.                            try{
436.                                x = new filmNew(x.judul, x.genre, x.tahun, x.kategori, x.batasUmur);
437.    }catch(TahunTooOldException ex){
438.    System.out.println(ex.getMessage());
439.                            }
440.                            x.displayFilm(id);
441.                        }else{
442.                            x.displayFilm(id);
443.                        }
444.                        id++;
445.                    }
446.    System.out.println("\033[38;5;220m==========
    ==========\033[0m");
```

447.	System.out.println();
448.	pilihFilm(idx);
449.	} else if(kategoris.equals("Platinum")){
450.	System.out.println("\nKamu User Platinum, kamu dapat mengakses semua Film: ");
451.	System.out.println("\033[35m========== ========== ==========\033[0m");
452.	System.out.printf("%-4s%-40s|%-25s|%-8s|%-10s|%-5s|\n", "","Judul", "Genre", "Tahun", "Kategori", "Umur");
453.	int id = 0;
454.	for(Film x: DaftarFilm){
455.	if(x.kategori.equals("Regular")){
456.	try{
457.	x = new filmReguler(x.judul, x.genre, x.tahun, x.kategori, x.batasUmur);
458.	}catch(TahunTooOldException ex){
459.	System.out.println(ex.getMessage());
460.	}
461.	x.displayFilm(id);
462.	}else if(x.kategori.equals("New")){
463.	try{

```
464.                                x = new filmNew(x.judul, x.genre, x.tahun, x.kategori, x.batasUmur);
465.  }catch(TahunTooOldException ex){
466.  System.out.println(ex.getMessage());
467.                            }
468.                            x.displayFilm(id);
469.                        }else{
470.                            try{
471.                                x = new filmOri(x.judul, x.genre, x.tahun, x.kategori, x.batasUmur);
472.  }catch(TahunTooOldException ex){
473.  System.out.println(ex.getMessage());
474.                            }
475.                            x.displayFilm(id);
476.                        }
477.                        id++;
478.                }
479.  System.out.println("\033[35m==================
      ==============================================
      ==============================\033[0m");
480.                System.out.println();
481.                pilihFilm(idx);
482.            }
483.        }
484.
```

```
485.     void displayFilm(){
486.         System.out.println("\nKamu User biasa, tidak dapat menonton film. Ini list daftar film yang ada: \n");
487.         System.out.println("================================================================================");
488.         System.out.printf("%-4s%-40s|%-25s|%-8s|%-10s|%-5s|\n","",     "Judul",     "Genre",     "Tahun", "Kategori", "Umur");
489.         int id = 0;
490.             for(Film x: DaftarFilm){
491.                 x.displayFilm(id);
492.                 id++;
493.             }
494.         System.out.println("================================================================================");
495.         System.out.println();
496.     }
497.
498.     void pilihFilm(int idx){
499.         int id = 1;
500.         System.out.println("Pilih   Film   yang ingin ditonton: \n");
501.         //String           anggota           = pelanggan[idx].getKategori();
502.         String cmd = input.nextLine();
503.         boolean ada = false ;
504.         try{
```

505.	Integer.parseInt(cmd);
506.	}catch(Exception e){
507.	System.out.println("Input salah!");
508.	return;
509.	}
510.	if(pelanggan[idx] instanceof pelangganRegular){
511.	for(Film x: DaftarFilm){
512.	if(Integer.parseInt(cmd) == id&&!x.kategori.equals("New")&&!x.kategori.equals("Ori")){
513.	if(pelanggan[idx].getUmur().equals("")){
514.	System.out.println("\nIsi data diri lengkat terlebih dahulu pada menu 6!\n");
515.	return;
516.	}
517.	
518.	int umr = Integer.parseInt(pelanggan[idx].getUmur());
519.	int batasan = Integer.parseInt(x.batasUmur);
520.	if(umr<batasan){
521.	System.out.println("Kamu belum cukup umur!\n");
522.	return;
523.	}
524.	
525.	x.displayFULL();
526.	ada = true;

527.	pelanggan[idx].addHistory(x);
528.	}
529.	id++;
530.	}
531.	}else if(pelanggan[idx] instanceof pelangganGold){
532.	for(Film x: DaftarFilm){
533.	if(Integer.parseInt(cmd) == id&&!x.kategori.equals("Ori")){
534.	if(pelanggan[idx].getUmur().equals("")){
535.	System.out.println("\nIsi data diri lengkat terlebih dahulu pada menu 6!\n");
536.	return;
537.	}
538.	
539.	int umr = Integer.parseInt(pelanggan[idx].getUmur());
540.	int batasan = Integer.parseInt(x.batasUmur);
541.	if(umr<batasan){
542.	System.out.println("Kamu belum cukup umur!\n");
543.	return;
544.	}
545.	
546.	x.displayFULL();
547.	ada = true;
548.	pelanggan[idx].addHistory(x);

```
549.                    }
550.                    id++;
551.                }
552.            }else{
553.                for(Film x: DaftarFilm){
554.                    if(Integer.parseInt(cmd)     ==
    id){
555.
    if(pelanggan[idx].getUmur().equals("")){
556.
    System.out.println("\nIsi  data  diri  lengkap
    terlebih dahulu pada menu 6!\n");
557.                                return;
558.                            }
559.                            int          umr          =
    Integer.parseInt(pelanggan[idx].getUmur());
560.                            int        batasan        =
    Integer.parseInt(x.batasUmur);
561.                            if(umr<batasan){
562.
    System.out.println("Kamu belum cukup umur!\n");
563.                                return;
564.                            }
565.
566.                            x.displayFULL();
567.                            ada = true;
568.
    pelanggan[idx].addHistory(x);
569.                    }
570.                    id++;
571.                }
572.            }
```

```
573.          if(!ada){
574.              System.out.println("Anda      tidak
dapat menonton Film ini\n");
575.          }else{
576.              System.out.println("Sedang
menonton film....\n\n");
577.              System.out.println("Film
Selesai!\n");
578.          }
579.
580.      }
581.
582.
583.
584. }
585.
```

5. DFilm

DFilm.java

```
1.  package komponen;
2.  public class Dfilm {
3.      String[] judul = new String[15];
4.      Film.genreFilm[] genre = new
Film.genreFilm[15];
5.      String[] tahun = new String[15];
6.      String[] kategori = new String[15];
7.      String[] sinopsis = new String[15];
8.      String[] batas = new String[15];
9.      Dfilm(){
10.         this.initialize();
11.     }
```

12.	`    void initialize(){`
13.	`        judul[0] = "Spirited Away";`
14.	`        judul[1] = "My Neigbor Totoro";`
15.	`        judul[2] = "Princess Mononoke";`
16.	`        judul[3] = "Avengers: End Game";`
17.	`        judul[4] = "Spider-Man: No Way Home";`
18.	`        judul[5] = "Parasite";`
19.	`        judul[6] = "Black Panther: Wakanda Forever";`
20.	`        judul[7] = "Attack on Titan";`
21.	`        judul[8] = "Boku no Hero";`
22.	`        judul[9] = "Wednesday";`
23.	`        judul[10] = "Papa Frangku";`
24.	`        judul[11] = "Mobu";`
25.	`        judul[12] = "Manusia Ikan";`
26.	`        judul[13] = "Zanite";`
27.	`        judul[14] = "Avatar: The Way of Water";`
28.	
29.	`        genre[0] = Film.genreFilm.FANTASY;`
30.	`        genre[1] = Film.genreFilm.FANTASY;`
31.	`        genre[2] = Film.genreFilm.FANTASY;`
32.	`        genre[3] = Film.genreFilm.SCIFI;`
33.	`        genre[4] = Film.genreFilm.SCIFI;`
34.	`        genre[5] = Film.genreFilm.THRILLER;`
35.	`        genre[6] = Film.genreFilm.SCIFI;`
36.	`        genre[7] = Film.genreFilm.FANTASY;`
37.	`        genre[8] = Film.genreFilm.ACTION;`
38.	`        genre[9] = Film.genreFilm.MYSTERY;`
39.	`        genre[10] = Film.genreFilm.COMEDY;`
40.	`        genre[11] = Film.genreFilm.FANTASY;`

41.	genre[12] = Film.genreFilm.FANTASY;
42.	genre[13] = Film.genreFilm.ADVENTURE;
43.	genre[14] = Film.genreFilm.ADVENTURE;
44.	
45.	tahun[0] = "2001";
46.	tahun[1] = "1988";
47.	tahun[2] = "1997";
48.	tahun[3] = "2019";
49.	tahun[4] = "2021";
50.	tahun[5] = "2019";
51.	tahun[6] = "2022";
52.	tahun[7] = "2013";
53.	tahun[8] = "2016";
54.	tahun[9] = "2022";
55.	tahun[10] = "2003";
56.	tahun[11] = "2005";
57.	tahun[12] = "2018";
58.	tahun[13] = "2022";
59.	tahun[14] = "2022";
60.	
61.	sinopsis[0] = "During her family's move to the suburbs, a sullen 10-year-old girl wanders into a world ruled by gods, witches, and spirits, and where humans are changed into beasts.";
62.	sinopsis[1] = "When two girls move to the country to be near their ailing mother, they have adventures with the wondrous forest spirits who live nearby.";
63.	sinopsis[2] = "On a journey to find the cure for a Tatarigami's curse, Ashitaka finds himself in the middle of a war between

	the forest gods and Tatara, a mining colony. In this quest he also meets San, the Mononoke Hime.";
64.	sinopsis[3] = "After the devastating events of Avengers: Infinity War (2018), the universe is in ruins. With the help of remaining allies, the Avengers assemble once more in order to reverse Thanos' actions and restore balance to the universe.";
65.	sinopsis[4] = "With Spider-Man's identity now revealed, Peter asks Doctor Strange for help. When a spell goes wrong, dangerous foes from other worlds start to appear, forcing Peter to discover what it truly means to be Spider-Man.";
66.	sinopsis[5] = "Greed and class discrimination threaten the newly formed symbiotic relationship between the wealthy Park family and the destitute Kim clan.";
67.	sinopsis[6] = "The people of Wakanda fight to protect their home from intervening world powers as they mourn the death of King T'Challa.";
68.	sinopsis[7] = "After his hometown is destroyed and his mother is killed, young Eren Jaeger vows to cleanse the earth of the giant humanoid Titans that have brought humanity to the brink of extinction.";
69.	sinopsis[8] = "Deku si pahlawan desa konoha";
70.	sinopsis[9] = "Follows Wednesday Addams' years as a student, when she attempts

	to master her emerging psychic ability, thwart a killing spree, and solve the mystery that embroiled her parents.";
71.	sinopsis[10] = "Papa frangku adalah orang jepang yang memberikan edukasi dari sudut pandang yang berbeda. Ternyata komedi hanyalah sebuah tragedi dari sudut pandang berbeda";
72.	sinopsis[11] = "Mobu adalah seorang anak wibu di filkom yang tidak punya teman. Akankah dia punya teman sebelum lulus?";
73.	sinopsis[12] = "Legenda manusia ikan di sungai di bawah jembatan Suhat. Konon katanya iya hidup di sekitar mahasiswa UB dan hendak mencari mangsa";
74.	sinopsis[13] = "Zanite adalah anak lugu yang sekarang menjadi Seorang mahasiswa. FILKOM sebagai tempat untuk berproses akan memberikan petualangan tak terduga dan pengalaman yang sarat makna";
75.	sinopsis[14] = "Jake Sully lives with his newfound family formed on the extrasolar moon Pandora. Once a familiar threat returns to finish what was previously started, Jake must work with Neytiri and the army of the Na'vi race to protect their home.";
76.	
77.	kategori[0] = "Regular";
78.	kategori[1] = "Regular";
79.	kategori[2] = "Regular";
80.	kategori[3] = "New";
81.	kategori[4] = "New";

```
82.             kategori[5] = "New";
83.             kategori[6] = "New";
84.             kategori[7] = "Regular";
85.             kategori[8] = "Regular";
86.             kategori[9] = "Ori";
87.             kategori[10] = "Regular";
88.             kategori[11] = "Ori";
89.             kategori[12] = "Ori";
90.             kategori[13] = "Ori";
91.             kategori[14] = "New";
92.
93.             batas[0] = "13";
94.             batas[1] = "13";
95.             batas[2] = "13";
96.             batas[3] = "17";
97.             batas[4] = "17";
98.             batas[5] = "18";
99.             batas[6] = "17";
100.            batas[7] = "15";
101.            batas[8] = "13";
102.            batas[9] = "18";
103.            batas[10] = "18";
104.            batas[11] = "15";
105.            batas[12] = "18";
106.            batas[13] = "17";
107.            batas[14] = "18";
108.        }
109.
110.    }
111.
```

6. ExceptionHandling

ExceptionHandling.java	
1.	`package komponen;`
2.	
3.	
4.	`class UmurNegatifException extends Exception {`
5.	`    public UmurNegatifException(String errorMessage) {`
6.	`        super(errorMessage);`
7.	`    }`
8.	`}`
9.	
10.	`class TahunTooOldException extends Exception{`
11.	`    public TahunTooOldException(String errorMessage){`
12.	`        super(errorMessage);`
13.	`    }`
14.	`}`
15.	

2. Access Modifier

2.1. Public

Access modifier public merupakan access modifier paling fleksibel dalam java. Method atau atribut yang memiliki access modifier public dapat diakses pada seluruh class yang ada.

Pada program yang telah dibuat, terdapat beberapa method yang menggunakan access modifier public. Alasan menggunakan access modifier public antara lain:

- Fleksibel dapat diakses di kelas mana saja.
- Sebagai method getter dan setter untuk atribut dengan access modifier private.
- Pada kelas menu, Array Deque daftar film di set sebagai public karena daftar film dapat dilihat oleh siapa saja. Implementasi

menggunakan access modifier lain tidak akan terlalu berpengaruh signifikan pada kasus ini. Karena Film yang ada di dalam daftar film memiliki atribut dengan access modifier protected. Kita dapat memilih access modifier default sebagai alternatif

2.2. Private

Access modifier private merupakan access modifier yang hanya mengizinkan pengaksesan melalui kelas itu sendiri. Pada Program, terdapat beberapa skenario penggunaan access modifier private, antara lain:

1. Atribut pada kelas Pelanggan

   Pelanggan memiliki beberapa atribut berupa data diri, password dan histori yang sifatnya rahasia. Oleh karena itu, penggunaan access modifier private pada pelanggan difungsikan untuk mencegah agar tidak ada yang dapat mengubah value atribut selain dari kelas pelanggan. Dengan demikian, informasi pelanggan menjadi lebih aman.

2. Array Of Object Pelanggan pada kelas menu

   Array dari objek Pelanggan yang dapat berisi beberapa pelanggan juga menggunakan access modifier private. Alasannya adalah agar tidak ada kelas luar yang dapat melakukan perubahan terhadap isi di dalam array pelanggan.

2.3. Protected

Access modifier protected merupakan access modifier yang hanya mengizinkan pengaksesan pada kelas itu sendiri atau subclass dari tempat atribut atau method protected itu berada.

Pada program, implementasi access modifier protected digunakan pada atribut kelas Film. Film memiliki beberapa subclass yang dapat melakukan perubahan pada atribut di parent class. Access modifier protected dipilih karena access modifier ini dapat membatasi ruang lingkup yang dapat melakukan perubahan pada atribut ini, tetapi tetap memberikan access untuk subclass yang merupakan bagian dari parent class. Subclass ini diberikan akses untuk memodifikasi atribut atau memanggil method protected.

2.4. Default

Default adalah access modifier bawaan ketika kita tidak melakukan spesifikasi access modifier pada suatu atribut atau method. Pada program ini, penggunaan access modifier default dilakukan pada atribut kelas Dfilm, menus dan JalaFlix. Penggunaan access modifier default dan public tidak memberikan perbedaan yang signifikan pada implementasi program studi kasus aplikasi JalaFlix ini. Hal ini dikarenakan seluruh kelas dari program JalaFlix diletakkan dalam package yang sama.

3. Penjelasan Constructor

Setiap kelas selain kelas JalaFlix memiliki constructor dengan peran tersendiri, berikut adalah penjelasan mengenai kegunaan dari constructor yang terdapat pada kelas:

1. Pelanggan

   Kelas Pelanggan memiliki 3 konstruktor. Konstruktor pertama menerima 6 parameter yang berguna untuk menyimpan 5 atribut dari kelas pelanggan ketika suatu objek pelanggan diinstansiasi. Konstruktor kedua terdiri dari 8 parameter, yakni 5 parameter pada konstruktor sebelumnya ditambah dengan password dan histori. Konstruktor ini dibuat sesuai dengan kebutuhan pada program, dan memiliki 2 peran utama:

   - Mempersingkat proses penyimpanan data pada atribut private, sehingga tidak perlu melakukan pemanggilan method set untuk setiap atribut.
   - Menduplikasi data pelanggan ketika pelanggan melakukan unsubscribe.

   Konstruktor ketiga adalah konstruktor kosong berfungsi untuk melakukan instansiasi dari kelas pelanggan tanpa melakukan prosedur lainnya. Setelah melakukan modifikasi, konstruktor pelanggan dapat melakukan exception handling ketika dilakukan instansiasi suatu objek pelanggan dengan tahun yang negatif. Error ini ditangani dengan **UmurNegatifException**.

2. Film

   Kelas Film memiliki 2 buah Constructor, Constructor default (kosong) tidak menerima parameter dan akan digunakan ketika objek Film diinstansiasi. Constructor Film yang menerima 5 parameter (judul, genre, tahun, kategori, sinopsis dan batasUmur) dibuat untuk mempersingkat proses instansiasi Film dengan datanya dan meminimalisir penggunaan method getter dan setter.

   Masing-masing subclass Film memiliki constructor yang menerima parameter berupa atribut dari Film dan memiliki kegunaan untuk melakukan

assignment data pada atribut parent classnya ketika kita ingin membuat objek baru dari subclass.
Setelah melakukan modifikasi, konstruktor pada kelas Film dapat memeriksa dan melemparkan exception berupa **TahunTooOldException** yang akan dilemparkan ketika dilakukan instansiasi objek Film dengan nilai tahun kurang dari 1800.

3. Menus

   Kelas menus juga memiliki sebuah constructor yang bertugas untuk menginstansiasi setiap pelanggan index ke i dan menyimpan seluruh film pada Dfilm ke ArrayDeque DaftarFilm. Konstruktor ini akan otomatis terpanggil saat menginstansiasi objek menu pada kelas JalaFlix.

4. DFilm

   Dfilm memiliki sebuah constructor yang akan memanggil method initialize yang kemudian akan mengisi seluruh nilai dari array masing masing atribut. Sehingga, nantinya array tersebut dapat dimasukkan ke dalam ArrayDeque DaftarFilm di dalam class menus. ArrayDeque akan menampung Film.

4. Polimorfisme pada code

Pada program, penerapan konsep polimorfisme terdapat pada kelas Pelanggan dan Film.

4.1. Polimorfisme pada Kelas Pelanggan

Kelas pelanggan memiliki 4 subclass abstract yaitu pelanggan reguler, pelanggan gold, platinum dan biasa. Keempat subclass tersebut memiliki 4 method yang sama, yaitu platinum(), gold(), reguler() dan uns(). Keempat method ini juga terdapat pada parent class pelanggan yang bersifat abstract.

Method ini dipanggil ketika seorang pelanggan ingin mengganti status mereka. Konsep polimorfisme diterapkan karena seorang pelanggan dengan kategori tertentu akan mendapatkan perlakuan yang berbeda.

Saat seorang pelanggan reguler ingin berpindah ke kategori gold, program akan memanggil method gold(). Karena ia berasal dari kategori reguler, maka biaya yang harus dibayarkan akan lebih sedikit jika dibandingkan ketika dia ingin pindah dari user biasa ke pelanggan gold. Dengan kata lain, perlakuan fungsi regular, gold, platinum dan uns akan berbeda bergantung pada seorang pelanggan berasal dari kategori mana dan ingin pindah ke kategori apa.

Setiap method memiliki nama dan jumlah parameter yang sama, tetapi memiliki perintah yang berbeda. Ketika method pada child class dipanggil, method akan melakukan overriding terhadap method serupa yang terletak di parent class. Konsep ini lah yang disebut sebagai konsep polimorfisme.

4.2. Polimorfisme pada Kelas Film

Konsep polimorfisme juga diterapkan pada kelas Film. Kelas Film dan Child *class*-nya memiliki method displayFilm(int i). Dapat dilihat bahwa method ini memiliki nama dan parameter yang sama. Namun, perlakuan dari masing masing method akan berbeda, bergantung pada kategori dari suatu Film. Film dengan kategori regular akan dicetak berwarna kuning, New akan dicetak berwarna hijau, dan Original Series akan dicetak dengan warna biru (Warna mungkin berbeda tergantung pada

terminal atau IDE yang digunakan). Ketika program menunjukkan pilihan Film kepada user, user hanya dapat menonton Film yang memiliki warna. Untuk Film yang tidak memiliki warna akan dicetak oleh method displayFilm pada parent class.

Pada saat ini, akan terjadi proses overriding pada method displayFilm di parent class. Proses overriding ini yang menjadi salah satu penerapan dari polimorfisme. Hal ini bisa dilihat dari bagaimana setiap kelas dapat mencetak film dengan nama method yang sama, tetapi perlakuan saat mencetak teks memiliki perbedaan, dalam kasus ini adalah warna teksnya.

5. **Abstract Class dan Interface**

   Abstract class adalah sebuah class yang tidak bisa diinstansiasi. Kelas pelanggan, memiliki tipe kelas abstrak. Dalam kelas pelanggan, terdapat beberapa method abstrak seperti regular(), gold(), platinum(), dan uns(). Keberadaan method abstrak ini mengharuskan perluasan dari kelas ini, yaitu pelangganReguler, pelangganGold, pelangganPlatinum, dan pelangganBiasa harus setidaknya memiliki method-method abstrak tersebut.

   Dengan tujuan yang sama, sebuah interface customer dengan method tersebut didefinisikan dengan tujuan setiap kelas yang mengimplementasikan interface tersebut, harus memiliki method yang didefinisikan di dalam interface customer. Pada program, Pelanggan mengimplementasikan interface customer, sehingga ia harus memiliki seluruh interface yang terletak di dalam customer.

6. **Enumerasi**

   Enumerasi adalah sebuah jenis tipe data dalam bahasa pemrograman java yang berisi nilai-nilai konstanta. Sesuai dengan instruksi soal, enumerasi diimplementasikan untuk atribut genre film pada kelas Film. Dengan diterapkannya enumerasi ini, setiap instansiasi objek Film yang dilakukan, harus menggunakan konstanta yang terdapat di dalam enumerasi ini.

   Misalnya, dalam enumerasi terdapat beberapa konstanta seperti ACTION, ADVENTURE, FANTASY, dan lain sebagainya. Jika pada saat instansiasi objek Film genre yang dimasukkan tidak sesuai dengan satupun konstanta yang terdapat di dalam enumerasi, maka akan menyebabkan error.

7. **Exception Handling**

   Exception Handling suatu cara untuk menangani kesalahan atau kondisi yang tidak terduga pada saat program dijalankan. Ketika terjadi suatu kesalahan atau exception, program akan berhenti dan keluar dari jalannya secara abnormal. Untuk menghindari hal tersebut, maka proses Exception Handling harus dilakukan.

Pada program, terdapat kelas ExceptionHandling yang khusus difungsikan untuk mendefinisikan beberapa exception. Terdapat dua exception, yaitu: **TahunTooOldException** dan **UmurNegatifException**. Keduanya adalah perluasan dari kelas Exception yang secara default terdapat pada java dan masing masing memiliki fungsi menangani Film dengan waktu rilis < tahun 1800 dan Pelanggan dengan umur negatif.

Ketika kedua hal yang tidak diinginkan tersebut terjadi, exception ini akan mencetak pesan error ke layar.