

Nama : M. Faris

NIM : 2311104017

## Jurnal Modul 6

### 1. Source Kode :

```
class SayaTubeVideo {
  constructor(title) {
    if (!title || title.length > 200) {
      throw new Error("Judul video tidak boleh null dan maksimal 200 karakter.");
    }

    this.id = Math.floor(10000 + Math.random() * 90000);
    this.title = title;
    this.playCount = 0;
  }

  increasePlayCount(count) {
    if (count < 0 || count > 25000000) {
      throw new Error("Play count harus antara 0 dan 25 juta.");
    }

    if (this.playCount + count > Number.MAX_SAFE_INTEGER) {
      console.warn("Play count akan melebihi batas maksimum integer JavaScript.");
      return;
    }

    this.playCount += count;
  }

  printVideoDetails() {
    console.log(`Video ID : ${this.id}`);
    console.log(`Judul Video : ${this.title}`);
    console.log(`Play Count : ${this.playCount}`);
  }

  getPlayCount() {
    return this.playCount;
  }
}
```

```

    getTitle() {
        return this.title;
    }
}

class SayaTubeUser {
    constructor(username) {
        if (!username || username.length > 100) {
            throw new Error("Username tidak boleh null dan maksimal 100 karakter.");
        }

        this.id = Math.floor(10000 + Math.random() * 90000);
        this.username = username;
        this.uploadedVideos = [];
    }

    addVideo(video) {
        if (!video || video.getPlayCount() > Number.MAX_SAFE_INTEGER) {
            throw new Error("Video tidak boleh null dan playCount terlalu besar.");
        }

        this.uploadedVideos.push(video);
    }

    getTotalVideoPlayCount() {
        return this.uploadedVideos.reduce((sum, v) => sum + v.getPlayCount(), 0);
    }

    printAllVideoPlaycount() {
        console.log(`\nUser: ${this.username}`);
        const limit = Math.min(this.uploadedVideos.length, 8);
        for (let i = 0; i < limit; i++) {
            console.log(`Video ${i + 1} judul: ${this.uploadedVideos[i].getTitle()}`);
        }
    }
}

try {
    const user = new SayaTubeUser("Faris");

    const titles = [
        "Review Film Interstellar oleh Faris",
        "Review Film Inception oleh Faris",
    ]
}

```

```

    "Review Film Tenet oleh Faris",
    "Review Film The Dark Knight oleh Faris",
    "Review Film Oppenheimer oleh Faris",
    "Review Film Dunkirk oleh Faris",
    "Review Film Prestige oleh Faris",
    "Review Film Memento oleh Faris",
    "Review Film Parasite oleh Faris",
    "Review Film The Matrix oleh Faris"
];

titles.forEach(title => {
    const video = new SayaTubeVideo(title);
    video.increasePlayCount(10);
    user.addVideo(video);
});

user.printAllVideoPlaycount();
console.log("\nTotal Play Count:", user.getTotalVideoPlayCount());
const overflowVideo = new SayaTubeVideo("Video Uji Overflow");
overflowVideo.increasePlayCount(Number.MAX_SAFE_INTEGER);

} catch (error) {
    console.error("Error:", error.message);
}

```

## 2. Output :

```

User: Faris
Video 1 judul: Review Film Interstellar oleh Faris
Video 2 judul: Review Film Inception oleh Faris
Video 3 judul: Review Film Tenet oleh Faris
Video 4 judul: Review Film The Dark Knight oleh Faris
Video 5 judul: Review Film Oppenheimer oleh Faris
Video 6 judul: Review Film Dunkirk oleh Faris
Video 7 judul: Review Film Prestige oleh Faris
Video 8 judul: Review Film Memento oleh Faris

Total Play Count: 100
Error: Play count harus antara 0 dan 25 juta.

```

3. Penjelasan :

Kode di atas merupakan implementasi **Design by Contract** melalui dua kelas: `SayaTubeVideo` dan `SayaTubeUser`. Kelas `SayaTubeVideo` merepresentasikan video dengan kontrak pada konstruktor untuk memastikan judul tidak null dan maksimal 200 karakter, serta pembatasan pada method `increasePlayCount` agar nilai penambahan tidak melebihi 25 juta dan tidak menyebabkan overflow. Kelas `SayaTubeUser` merepresentasikan pengguna dengan validasi username maksimal 100 karakter, serta kontrak dalam method `addVideo` untuk memastikan objek video valid dan play count-nya aman. Program ini membuat objek user "Faris", menambahkan 10 video dengan play count 10 masing-masing, mencetak 8 judul video teratas, menghitung total play count, dan menguji batas overflow dengan nilai `Number.MAX_SAFE_INTEGER`. Semua kesalahan ditangani melalui blok try-catch.