

Nama : Salman Alfarisi

Kelas : S1SE-07-01

NIM : 231114036

```

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

  increasePlayCount(jumlah) {
    this.playCount += jumlah;
  }

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

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

  addVideo(video) {
    this.uploadedVideos.push(video);
  }

  getTotalVideoPlayCount() {
    return this.uploadedVideos.reduce((total, video) => total + video.playCount, 0);
  }

  printAllVideoPlayCount() {
    console.log(`User: ${this.username}`);
    this.uploadedVideos.forEach((video, index) => {
      console.log(`Video ${index + 1} judul: ${video.title}`);
    });
  }
}

// Demo penggunaan
const user = new SayaTubeUser("Raka");
const judulFilm = [
  "Review Film Interstellar oleh Raka",
  "Review Film Inception oleh Raka",
  "Review Film The Dark Knight oleh Raka",
  "Review Film Tenet oleh Raka",
  "Review Film Dunkirk oleh Raka",
  "Review Film Oppenheimer oleh Raka",
  "Review Film The Prestige oleh Raka",
  "Review Film Memento oleh Raka",
  "Review Film The Batman oleh Raka",
  "Review Film Avatar oleh Raka"
];

judulFilm.forEach(judul => {
  const video = new SayaTubeVideo(judul);
  video.increasePlayCount(Math.floor(Math.random() * 1000));
  user.addVideo(video);
});

user.printAllVideoPlayCount();
console.log("Total Play Count:", user.getTotalVideoPlayCount());

```

- Penjelasan:
 - class SayaTubeVideo {
 Mendefinisikan sebuah kelas bernama SayaTubeVideo.
 - constructor(title) {

```

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

```

- constructor(title) adalah metode khusus untuk menginisialisasi objek saat dibuat.
- this.id: Menghasilkan ID acak 5 digit antara 10000 sampai 99999.
- this.title: Menyimpan judul video.
- this.playCount = 0: Menginisialisasi jumlah penayangan video ke nol.

- increasePlayCount(jumlah) {


```

        this.playCount += jumlah;
      }

```

Fungsi untuk menambah jumlah penayangan video (playCount) sebesar jumlah.

- printVideoDetails() {


```

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

```

Menampilkan detail video ke konsol: ID, judul, dan jumlah tayang.

- class SayaTubeUser {
 Kelas ini merepresentasikan pengguna yang memiliki beberapa video.
- constructor(username) {


```

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

```

- this.id: ID acak pengguna, seperti pada video.
- this.username: Nama pengguna.
- this.uploadedVideos: Array kosong untuk menyimpan daftar video yang diunggah.

Kode tersebut mensimulasikan sistem pengguna dan video seperti YouTube. Kelas SayaTubeVideo merepresentasikan video, sedangkan SayaTubeUser merepresentasikan pengguna yang memiliki video. Program ini juga menggunakan OOP (Object-Oriented Programming) seperti constructor, properti instance, dan method.

```

class SayaTubeVideo {
  constructor(title) {
    if (title == null || title.length > 200) {
      throw new Error("Judul tidak valid (null atau lebih dari 200 karakter)");
    }

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

  increasePlayCount(jumlah) {
    try {
      if (jumlah < 0 || jumlah > 25000000) {
        throw new Error("Jumlah play count tidak valid (negatif atau lebih dari 25.000.000)");
      }
      const result = this.playCount + jumlah;
      if (result > Number.MAX_SAFE_INTEGER) {
        throw new Error("Overflow play count");
      }
      this.playCount = result;
    } catch (e) {
      console.error("Error: " + e.message);
    }
  }

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

class SayaTubeUser {
  constructor(username) {
    if (username == null || username.length > 100) {
      throw new Error("Username tidak valid (null atau lebih dari 100 karakter)");
    }

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

  addVideo(video) {
    if (video == null) throw new Error("Video tidak boleh null");
    if (video.playCount >= Number.MAX_SAFE_INTEGER)
      throw new Error("Play count video terlalu besar");

    this.uploadedVideos.push(video);
  }

  getTotalVideoPlayCount() {
    return this.uploadedVideos.reduce((total, video) => total + video.playCount, 0);
  }

  printAllVideoPlayCount() {
    console.log(`User: ${this.username}`);
    for (let i = 0; i < Math.min(8, this.uploadedVideos.length); i++) {
      console.log(`Video ${i + 1} judul: ${this.uploadedVideos[i].title}`);
    }
  }
}

// Pengujian precondition dan postcondition
try {
  const user = new SayaTubeUser("Raka");

  for (let i = 0; i < 10; i++) {
    let title = `Review Film ke-${i + 1} oleh Raka`;
    let video = new SayaTubeVideo(title);
    video.increasePlayCount(25000000); // valid
    user.addVideo(video);
  }

  // Test overflow
  const videoOverflow = new SayaTubeVideo("Video Overflow Test");
  videoOverflow.increasePlayCount(Number.MAX_SAFE_INTEGER); // akan error
  user.addVideo(videoOverflow); // akan error

  user.printAllVideoPlayCount(); // hanya cetak 8 video
  console.log("Total Play Count:", user.getTotalVideoPlayCount());
} catch (e) {
  console.error("Exception caught: " + e.message);
}

```



```
if (title == null || title.length > 200) {  
    throw new Error("Judul tidak valid (null atau lebih dari 200 karakter)");  
}
```

Menambahkan pengecekan bahwa:

- title **tidak boleh null**
- title **maksimal 200 karakter**



```
if (jumlah < 0 || jumlah > 250000000) throw Error(...);  
if (result > Number.MAX_SAFE_INTEGER) throw Error(...);
```

Perubahan:

- Mencegah input negatif.
- Membatasi input maksimal 25.000.000.
- Mencegah **integer overflow** dengan memeriksa apakah totalnya melebihi Number.MAX_SAFE_INTEGER.



```
if (username == null || username.length > 100) {  
    throw new Error("Username tidak valid");  
}
```

Username sekarang divalidasi agar **tidak null** dan **maksimal 100 karakter**.



```
if (video == null) throw new Error("Video tidak boleh null");  
if (video.playCount >= Number.MAX_SAFE_INTEGER) throw new Error("Play count video terlalu besar");  
|
```

Perubahan:

- Menolak video yang null.
- Menolak video dengan play count terlalu besar (**mencegah overflow total**).

```
for (let i = 0; i < Math.min(8, this.uploadedVideos.length); i++) { ... }
```

Perubahan: Hanya mencetak **maksimal 8 video pertama**, bukan semua.

```
videoOverflow.increasePlayCount(Number.MAX_SAFE_INTEGER); // error|
```

Menguji sistem terhadap input ekstrem (test **postcondition/overflow**) dan melihat apakah error ditangani dengan benar.

```
try {  
  ...  
} catch (e) {  
  console.error("Exception caught: " + e.message);  
}
```

Perubahan: Menghindari crash program jika ada kesalahan—misalnya judul tidak valid atau overflow play count.