



**Kampus
Merdeka**
INDONESIA JAYA



3. Kelas dan Objek dalam JavaScript

PEMROGRAMAN BERBASIS OBJEK

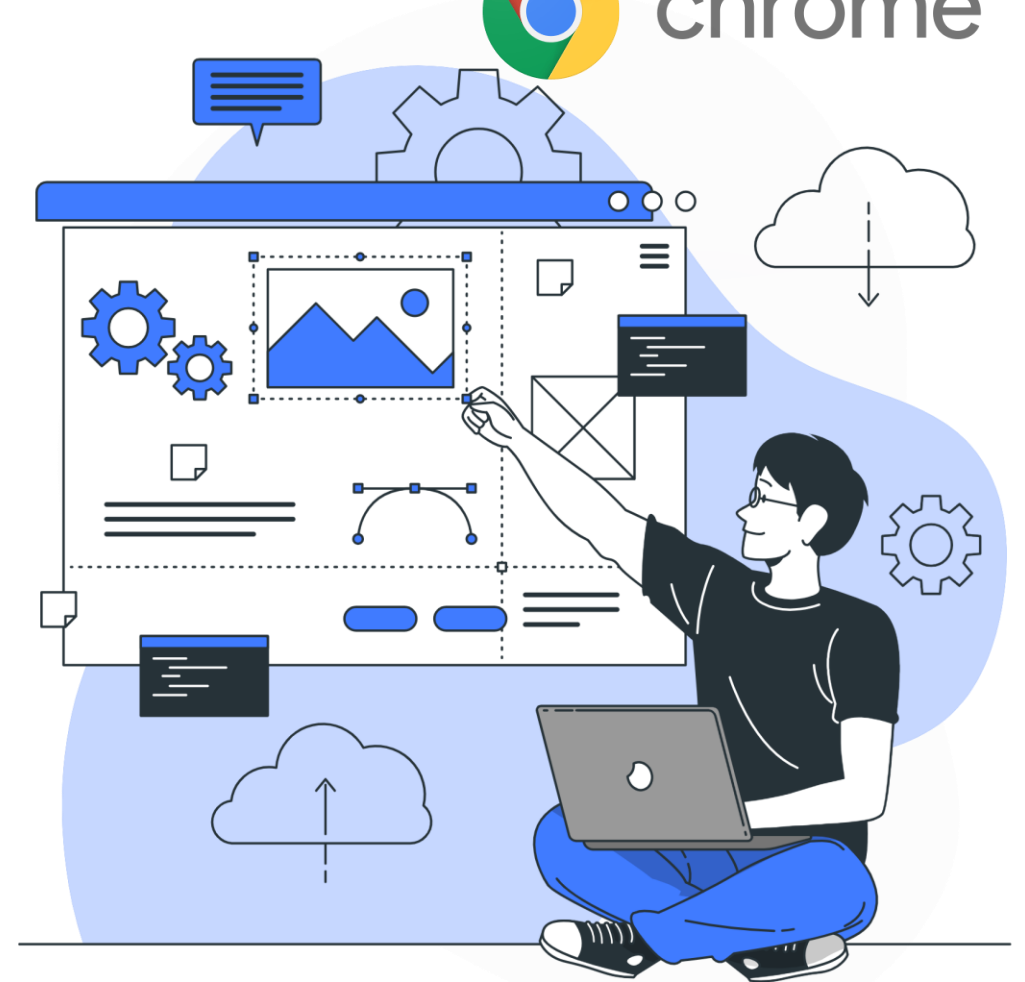


1. Definisi Kelas dan Objek

Dalam OOP,

- ▶ kelas (class) adalah cetak biru (blueprint) atau **template** yang digunakan untuk membuat objek. Sementara itu,
- ▶ objek adalah instansiasi dari kelas, yang berarti setiap objek adalah **salinan** konkret yang memiliki properti dan metode dari kelasnya.

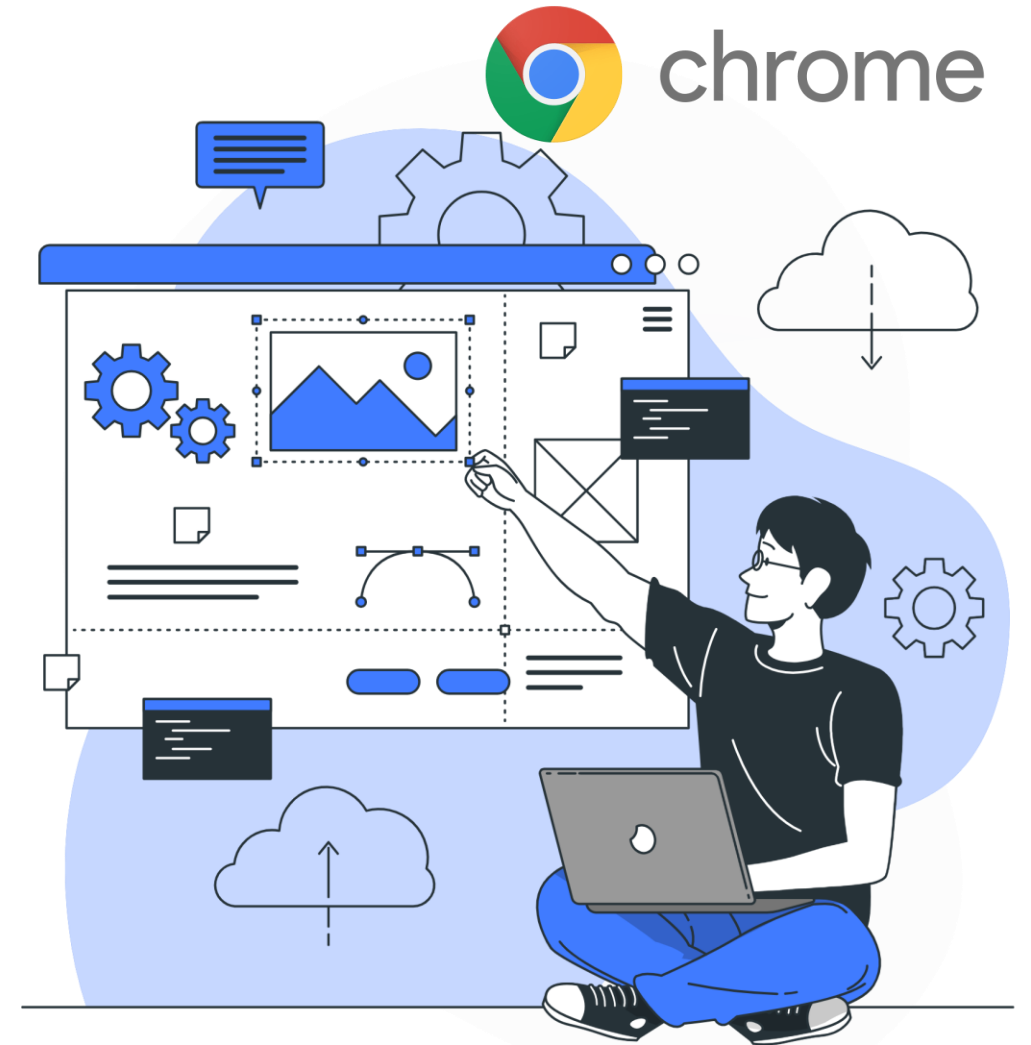
Sebagai analogi, bayangkan kelas seperti cetak biru mobil, yang berisi informasi tentang bagaimana mobil itu seharusnya dibuat, seperti bentuk, mesin, dan jumlah roda. Sementara itu, objek adalah mobil nyata yang dibuat berdasarkan cetak biru tersebut, seperti Toyota Avanza, Honda Civic, atau Tesla Model 3.





2. Struktur Kelas dalam JavaScript (ES6 Class)

- ▶ Sebelum ES6, kita harus menggunakan function **constructor** untuk membuat objek. Namun, dengan ES6, kita bisa langsung mendefinisikan kelas dengan sintaks yang lebih rapi dan mudah dipahami.
- ▶ Contoh perbedaan antara function constructor dan ES6 class:





Menggunakan Function Constructor (Sebelum ES6):

```
function Mobil(merk, warna) {  
  this.merk = merk;  
  this.warna = warna;  
  this.info = function () {  
    return `Mobil ini adalah ${this.merk} berwarna  
    ${this.warna}.`;   
  };  
}  
  
const avanza = new Mobil("Toyota Avanza", "Hitam");  
console.log(avanza.info());
```

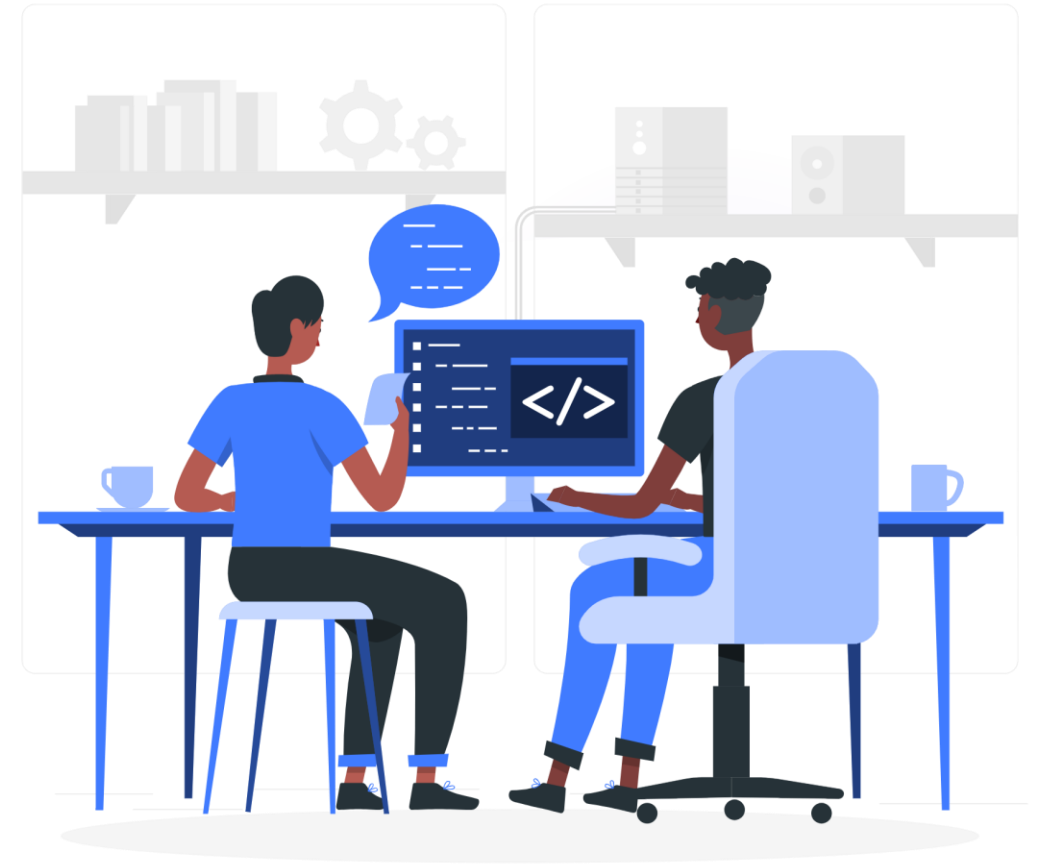
Menggunakan ES6 Class:

```
class Mobil {  
  constructor(merk, warna) {  
    this.merk = merk;  
    this.warna = warna;  
  }  
  info() {  
    return `Mobil ini adalah ${this.merk} berwarna ${this.warna}.`;   
  }  
}  
  
const avanza = new Mobil("Toyota Avanza", "Hitam");  
console.log(avanza.info());
```



Perbedaan utama:

- ▶ Pada function **constructor**, kita mendefinisikan metode langsung di dalam function, sehingga setiap objek memiliki salinan metodenya sendiri.
- ▶ Pada ES6 class, metode ditambahkan ke **prototype** secara otomatis, sehingga lebih efisien dalam penggunaan memori.





3. Pembuatan dan Penggunaan Objek

Setelah kita memiliki kelas, kita bisa membuat **objek** dengan menggunakan kata kunci **new**.

Penting: Setiap kali kita menggunakan new, JavaScript akan:

- Membuat objek kosong {}.

- Memanggil constructor() dengan nilai yang diberikan.

- Mengembalikan objek yang telah dibuat.

```
class Mahasiswa {  
    constructor(nama, nim) {  
        this.nama = nama;  
        this.nim = nim;  
    }  
    perkenalan() {  
        return `Halo, saya ${this.nama} dengan NIM  
        ${this.nim}.`;   
    }  
}  
  
// Membuat objek dari kelas Mahasiswa  
const mahasiswa1 = new Mahasiswa("Aldi",  
"2203123456");  
  
console.log(mahasiswa1.perkenalan());  
  
const mahasiswa2 = new Mahasiswa("Budi",  
"2203123457");  
  
console.log(mahasiswa2.perkenalan());
```



4. Constructor dan Method dalam Kelas

Constructor adalah metode khusus dalam kelas yang dipanggil pertama kali saat objek dibuat. Tujuannya untuk menginisialisasi properti objek.

```
class Buku {  
    constructor(judul, penulis) {  
        this.judul = judul;  
        this.penulis = penulis;  
    }  
}  
  
const buku1 = new Buku("Clean Code", "Robert C.  
Martin");  
console.log(buku1.judul); // Output: Clean Code
```



4. Constructor dan Method dalam Kelas

Method adalah fungsi yang berada di dalam kelas dan bisa diakses oleh objek yang dibuat dari kelas tersebut.

Constructor tidak mengembalikan nilai secara eksplisit.

```
class Persegi {  
    constructor(sisi) {  
        this.sisi = sisi;  
    }  
  
    luas() {  
        return this.sisi * this.sisi;  
    }  
}  
  
const persegi1 = new Persegi(5);  
console.log(`Luas persegi: ${persegi1.luas()}`); //  
Output: Luas persegi: 25
```




4. Constructor dan Method dalam Kelas

Method adalah fungsi yang berada di dalam kelas dan bisa diakses oleh objek yang dibuat dari kelas tersebut.

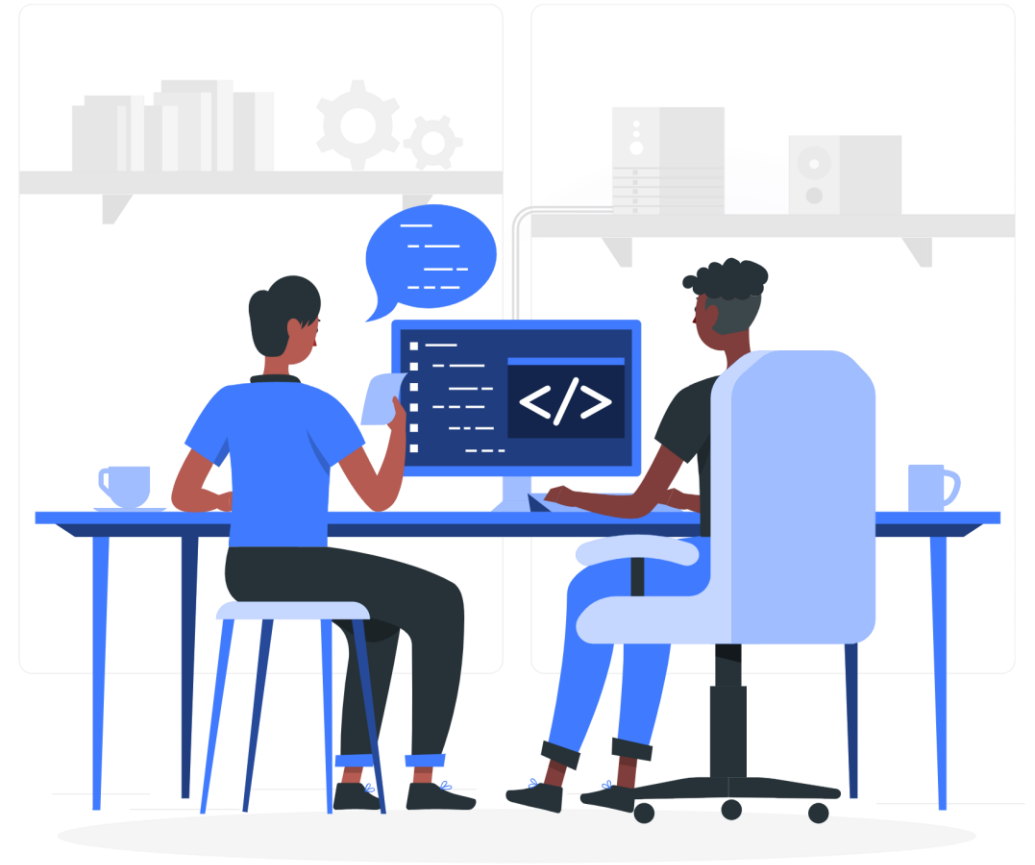
Method tidak perlu ditulis dengan kata kunci function, cukup nama metode langsung di dalam kelas.

```
class Persegi {  
    constructor(sisi) {  
        this.sisi = sisi;  
    }  
  
    luas() {  
        return this.sisi * this.sisi;  
    }  
}  
  
const persegi1 = new Persegi(5);  
console.log(`Luas persegi: ${persegi1.luas()}`); //  
Output: Luas persegi: 25
```



Kesimpulan

- ▶ Kelas adalah cetak biru untuk membuat objek dalam OOP.
- ▶ Objek adalah instansiasi dari kelas yang memiliki properti dan metode.
- ▶ JavaScript mendukung OOP dengan pendekatan prototype-based sebelum ES6, tetapi sekarang mendukung sintaks class yang lebih intuitif.
- ▶ Constructor digunakan untuk menginisialisasi properti saat objek dibuat.
- ▶ Method dalam kelas digunakan untuk memberikan perilaku kepada objek.





Praktikum:

- Membuat halaman web yang menarik secara visual

