



DISKUSIKAN MATERI

LAPORKAN MATERI

## Daftar Modul

Persetujuan Hak Cipta ✓

Modul 1: Introduction Course (Gratis) ✓

Prasyarat Kelas (Gratis) ✓

Apa yang Akan Kita Pelajari? (Gratis) ✓

Tools Requirement ✓

Modul 2: ECMAScript 6 (Gratis) ✓

Running Snippet Code (Gratis) ✓

Persiapan Project Latihan (Gratis) ✓

## Static Method

Seluruh kendaraan pasti butuh yang namanya perawatan bukan? Jika iya, tentu kita perlu membuat method **repair** untuk memperbaiki kendaraan tersebut. Dalam analogi dunia nyata, ketika kendaraan mengalami kerusakan maka kendaraan tersebut akan diperbaiki di bengkel (factory), sehingga kita perlu membuat class baru yang berperan sebagai factory, sebutlah class tersebut **VehicleFactory**. Di dalam kelas **VehicleFactory** terdapat satu method **repair()** yang dapat menerima banyak kendaraan sebagai parameternya.

```
1. class Vehicle {
2.   constructor(licensePlate, manufacture) {
3.     this.licensePlate = licensePlate;
4.     this.manufacture = manufacture;
5.     this.engineActive = false;
6.   }
7.   /*
8.     kode lainnya
9.   */
10. }
11.
12. /* kode lainnya dalam pembuatan class Car,
13.   Motorcycle, dsb. */
14.
15.
16. class VehicleFactory {
17.   repair(vehicles) {
18.     vehicles.forEach(vehicele => {
19.       console.log(`Kendaraan ${vehicele.licensePlate} sedang melakukan perawatan`)
20.     })
21.   }
22. }
```

Untuk mengakses method dari class, sejauh ini kita perlu membuat instance dari classnya terlebih dahulu. Sehingga untuk memanggil **repair()**, kita perlu membuat instance dari class **VehicleFactory**

```
1. const johnCar = new Car("H121S", "Honda", 4);
2. const tomMotor = new Motorcycle("GF121J", "Yamaha", 2);
3. const dimasCar = new Car("TA1408K", "Tesla", 4);
4.
5. /* Membuat instance untuk memanggil fungsi repair */
6. const vehicleFactory = new VehicleFactory();
7. vehicleFactory.repair([johnCar, tomMotor, dimasCar]);
8.
9. /* output:
10.  Kendaraan H121S sedang melakukan perawatan
11.  Kendaraan GF121J sedang melakukan perawatan
12.  Kendaraan TA1408K sedang melakukan perawatan
13.  */
```

Kode tersebut berjalan sesuai harapan namun tidak efektif. Mengapa? Karena kita perlu membuat **instance** untuk sekedar memanggil satu fungsi dari class-nya tersebut. Membuat **instance** adalah membuat sebuah objek baru yang terbentuk melalui blueprint sehingga membutuhkan memori ekstra. Jika kita dapat mengakses method tersebut tanpa melalui instance mengapa tidak? Pada kasus inilah kita membutuhkan sebuah **static method**. Static method merupakan method yang tidak dapat dipanggil oleh instance dari class, namun dapat dipanggil melalui class-nya sendiri.

Pada ES6 class kita dapat membuat static method dengan menambahkan keyword static sebelum deklarasi method-nya:

```
1. class Vehicle {
2.   constructor(licensePlate, manufacture) {
3.     this.licensePlate = licensePlate;
4.     this.manufacture = manufacture;
5.     this.engineActive = false;
6.   }
7.   /*
8.     kode lainnya
9.   */
10. }
11.
12. /* kode lainnya dalam pembuatan class Car,
13.   Motorcycle, dsb. */
14.
15. class VehicleFactory {
16.   static repair(vehicles) {
17.     vehicles.forEach(vehicele => {
18.       console.log(`Kendaraan ${vehicele.licensePlate} sedang melakukan perawatan`)
19.     })
20.   }
21. }
```

Kemudian untuk memanggil methodnya kita bisa panggil melalui class **VehicleFactory** kemudian

*repair()*.

```
1. const johnCar = new Car("H121S", "Honda", 4);
2. const tomMotor = new Motorcycle("GF121J", "Yamaha", 2);
3. const dimasCar = new Car("TA1408K", "Tesla", 4);
4.
5. /* Pemanggilan method repair langsung dari class-nya */
6. VehicleFactory.repair([johnCar, tomMotor, dimasCar]);
7.
8. /* output:
9. Kendaraan H121S sedang melakukan perawatan
10. Kendaraan GF121J sedang melakukan perawatan
11. Kendaraan TA1408K sedang melakukan perawatan
12. */
```

← KEMBALI KE MATERI SEBELUMNYA

LANJUTKAN KE MATERI BERIKUTNYA →



image  
click here  
to view image

image  
click here  
to view image

#### PERUSAHAAN

Tentang Kami

Blog

Berita Terbaru



#### PROGRAM

Academy

Challenge

Event

Job

Rewards

#### SUPPORT

Bantuan

FAQ

Hubungi Kami

