

# Modul 7: Modularitas Program dengan Fungsi

## Function Declaration, Expression, ES6 Arrow Functions, dan Parameter Default

Kusuma Web Academy

28 Juni 2026

# Function Declaration vs Function Expression

Fungsi membungkus blok logika kerja yang dapat dipanggil berkali-kali tanpa menulis ulang:

## 1. Function Declaration

Dideklarasikan secara formal. Mendukung fitur *\*hoisting\** penuh (fungsi dapat dipanggil di baris program yang letaknya di atas baris deklarasi fungsinya).

```
function hitungLuas(p, l) {  
    return p * l;  
}
```

## 2. Function Expression

Fungsi dibuat tanpa nama lalu dimasukkan sebagai nilai dari sebuah variabel. Tidak terpengaruh hoisting (tidak bisa dipanggil sebelum dideklarasikan).

```
const hitungKeliling = function(p, l) {  
    return 2 * (p + l);  
};
```

# ES6 Arrow Function: Pemangkasan Sintaksis

Arrow function menyajikan penulisan fungsi yang sangat bersih, modern, dan ringkas:

- **Implisit Return:** Jika isi fungsi hanya berupa satu baris pengembalian nilai, kita boleh menghapus tanda kurung kurawal serta kata kunci `return`.

```
// Sintaksis konvensional:  
const kuadratkan = (x) => { return x * x; };
```

```
// Sintaksis implisit satu baris (ES6):  
const kuadratkanSingkat = x => x * x;
```

- **Ketiadaan binding 'this':** Arrow function tidak mendefinisikan konteks `this` miliknya sendiri, melainkan mewarisi konteks lexical induk tempat ia dilahirkan.

# Parameter Default & Fleksibilitas Argumen

Bagaimana mengamankan program jika pengguna lupa mengisi argumen input?

- **Parameter Default:** Menetapkan nilai cadangan otomatis pada parameter fungsi saat deklarasi dibuat.

```
const kirimPesan = (penerima = "Tamu") => {  
  return "Selamat Belajar, " + penerima;  
};
```

```
console.log(kirimPesan());           // Output: Selamat Belajar, Tamu  
console.log(kirimPesan("Deri"));    // Output: Selamat Belajar, Deri
```

- **Return Statement:** Kata kunci `return` secara mutlak mengakhiri eksekusi logika fungsi dan mengembalikan paket nilai ke baris pemanggilnya.

# Latihan Praktik Pembuatan Fungsi Utilitas

Rakitlah program utilitas perhitungan tarif pajak berikut menggunakan gaya ES6:

```
const hitungPajakBelanja = (subtotal, persentasePajak = 0.1) => {  
  return subtotal + (subtotal * persentasePajak);  
};
```

```
let tagihanAkhir = hitungPajakBelanja(150000);  
console.log("Total tagihan setelah pajak: Rp" + tagihanAkhir);  
// Output: Total tagihan setelah pajak: Rp165000
```