Rakamin Academy

Javascript

Case Study: Simple Calculator for Calculating Savings Interest by applying OOP and DOM Level Intermediate

Raih Mimpi #TanpaBatas

Objektif sesi

- Peserta mengimplementasikan javascript OOP & DOM yang telah dipelajari dengan membuat Case Study: Simple Calculator for Calculating Savings Interest by applying OOP and DOM (Part 2)

Deskripsi

Pada tingkat ini, kita akan meningkatkan kalkulator bunga tabungan dengan menambahkan visualisasi grafik menggunakan Chart.js. Selain itu, kita akan memperkaya kelas SavingsCalculator dengan metode tambahan untuk memperoleh data per tahun dan membuat kelas baru untuk mengelola visualisasi grafik.

Contoh Output

OOP in JavaScript:

- Tambahkan metode calculateYearlyInterest di dalam kelas SavingsCalculator untuk mengembalikan data bunga setiap tahun.
- Buat kelas baru, misalnya ChartManager, yang akan menangani pembuatan dan pembaruan grafik menggunakan Chart.js.

javascript Copy code

```javascript
class SavingsCalculator {
  // ... (Kode sebelumnya)

  calculateYearlyInterest() {
    const yearlyInterest = [];
    let totalAmount = this._initialAmount;

    for (let year = 1; year <= this._period; year++) {
      const interest = (totalAmount * this._interestRate) / 100;
      totalAmount += interest;
      yearlyInterest.push({ year, interest: interest.toFixed(2), totalA
    }

    return yearlyInterest;
  }
}

class ChartManager {
  constructor() {
    this.chart = null;
  }
```

```javascript
class ChartManager {
  constructor() {
    this.chart = null;
  }

  createChart(data) {
    const ctx = document.getElementById('chart').getContext('2d');
    this.chart = new Chart(ctx, {
      type: 'bar',
      data: {
        labels: data.map(entry => `Year ${entry.year}`),
        datasets: [{
          label: 'Yearly Interest',
          data: data.map(entry => entry.interest),
          backgroundColor: 'rgba(75, 192, 192, 0.2)',
          borderColor: 'rgba(75, 192, 192, 1)',
          borderWidth: 1
        }]
      },
      options: {
        scales: {
          y: {
            beginAtZero: true
          }
        }
      }
    });
  }
```

Contoh Output

DOM Interaction:

- Tambahkan elemen HTML untuk menampilkan grafik.
- Modifikasi fungsi calculateInterest untuk menangani pembaruan grafik setelah perhitungan selesai.

```
<!-- Modifikasi elemen HTML -->

<body>

 <h1>Savings Interest Calculator</h1>

 <form id="savingsForm">

  <!-- ... (Kode sebelumnya) -->

 </form>

 <div id="result"></div>

 <canvas id="chart" width="400" height="200"></canvas> <!-- Tambahkan elemen untuk grafik
-->
```

```
 <script>
  const chartManager = new ChartManager();

  function calculateInterest() {
   const initialAmount =
parseFloat(document.getElementById('initialAmount').value);
   const interestRate =
parseFloat(document.getElementById('interestRate').value);
   const period = parseFloat(document.getElementById('period').value);

   const calculator = new SavingsCalculator(initialAmount, interestRate,
period);
   const yearlyInterest = calculator.calculateYearlyInterest();

   document.getElementById('result').innerHTML = `Total Interest Earned:
$${yearlyInterest[yearlyInterest.length - 1].totalAmount}`;

   // Tambahkan pemanggilan fungsi untuk membuat atau memperbarui grafik
   if (!chartManager.chart) {
    chartManager.createChart(yearlyInterest);
   } else {
    chartManager.updateChart(yearlyInterest);
   }
  }
 </script>
</body>
```

Output

Output:

Pengguna membuka halaman HTML yang telah diperbarui.

Pengguna memasukkan jumlah tabungan, tingkat bunga, dan periode tabungan.

Pengguna mengeklik tombol "Calculate Interest".

Hasil perhitungan bunga tabungan dan grafik per tahun ditampilkan di halaman menggunakan DOM.


Rakamin
Academy
Thank you
Raih Mimpi #TanpaBatas