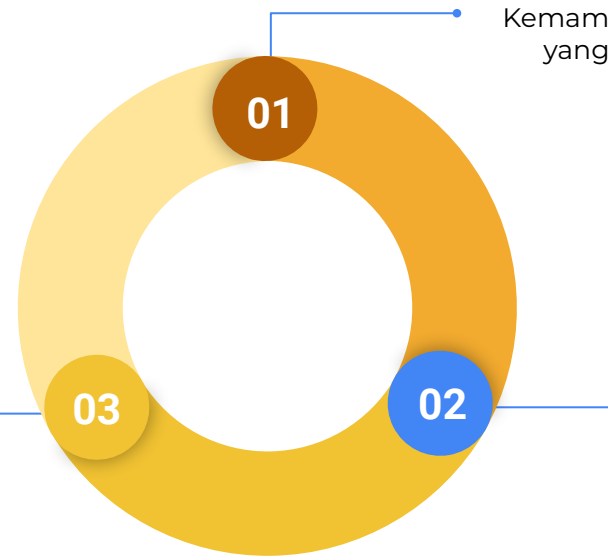


# Backend Javascript Challenge

**Silver** Chapter 2

### ACCEPTANCE CRITERIA

Kriteria pengumpulan challenge yang harus kamu penuhi untuk dapat melewati chapter



### SKILL METRICS

Kemampuan teknis yang akan kamu pelajari

### DELIVERY

Hal yang akan kamu lakukan untuk dapat melewati chapter

### SKILL METRICS

(Kemampuan teknis yang akan kamu pelajari)

- Programming Algorithm
- Melakukan error handling
- Menerapkan Asynchronous process
- Menerapkan OOP
- Menggunakan Git

### DELIVERY

(Hal yang akan kamu lakukan untuk dapat melewati chapter)

1. Membuat repositori baru di GitHub dengan nama "Basic-Banking-System"
2. Salin file bank\_account.js Challenge 1 ke direktori proyek di repositori baru
3. Membuat file JavaScript baru dengan nama banking\_system.js
4. Implementasikan kelas BankAccount dengan metode deposit() dan withdraw()
5. Gunakan setTimeout() untuk mensimulasikan operasi transaksi yang asynchronous
6. Commit dan push perubahan ke repositori GitHub

### CRITERIA

(Kriteria pengumpulan challenge yang harus kamu penuhi untuk dapat melewati chapter)

1. Menggunakan modular class OOP (40 points)
2. Mengurai proses berpikir dengan menggunakan flowchart (30 points)
3. Menggunakan GitHub untuk mengelola repository (30 points)

Sabrina jalan menyusuri gua  
Uhuy, saatnya challenge dua 🎉

### Studi Kasus

Dalam studi kasus ini, kita akan membuat sebuah **Basic Banking System** sederhana menggunakan konsep Pemrograman Berorientasi Objek (**OOP**) dan mensimulasikan operasi transaksi yang **asynchronous**.



### Ini dia! hal-hal yang perlu kamu lakukan di Challenge Chapter 2! 😎

- Buatlah repositori baru di GitHub dengan nama **Basic-Banking-System**.
- Salin file **bank\_account.js** dari **Challenge 1** ke direktori proyek di repositori baru tersebut. File ini akan berisi definisi kelas **BankAccount** yang akan digunakan dalam implementasi sistem perbankan.
- Buatlah file JavaScript baru dengan nama **banking\_system.js**. File ini akan menjadi pusat implementasi sistem perbankan menggunakan konsep **OOP**.



### Lanjutan langkah-langkah pengerjaan:

- Implementasikan kelas BankAccount dengan minimal dua metode:
  - **deposit(amount)**: Metode ini akan menerima jumlah uang yang akan disimpan ke dalam akun.
  - **withdraw(amount)**: Metode ini akan menerima jumlah uang yang akan ditarik dari akun, asalkan saldo mencukupi.





- Gunakan fungsi **setTimeout()** untuk mensimulasikan operasi transaksi yang **asynchronous**. Misalnya, setelah melakukan deposit atau withdrawal, Anda dapat menunda eksekusi selama beberapa detik sebelum mengembalikan hasil operasi.



### Langkah terakhir:

- Lakukan commit dan push perubahan yang telah Anda buat ke **repositori** GitHub.



Setelah selesai mengerjakan challenge, kamu harus mengumpulkan outputnya kepada Tim BINAR.

Bagaimana caranya?

Cek slide berikutnya yaa...



## Timeline Pelaksanaan dan Pengumpulan Challenge

Biar makin paham dengan tahapan penyelesaian challenge dan estimasi waktu yang diperlukan untuk setiap tahap tersebut, yuk amati **timeline** berikut!

### Hari Pertama di Minggu Pertama

Penjelasan tentang Challenge 1 oleh Facilitator secara langsung ke student.

### Challenge Review

Pembahasan progress challenge bersama fasilitator dilakukan di FD terakhir di setiap chapter.

### Penilaian

Proses penilaian oleh fasilitator dilakukan H+1 s/d H+7 pengumpulan challenge oleh students.

### Pengerjaan Challenge

Dilakukan secara individu oleh Students selama Chapter berjalan. Pengerjaan challenge juga bisa didiskusikan pada saat live session dengan Facilitator.

### Pengumpulan Challenge (FD Terakhir chapter 1)

Student diberikan kesempatan untuk memperbaiki challenge setelah sesi review. Tim BINAR akan menyiapkan Form Submission untuk pengumpulan challenge.

## Info Pengumpulan Challenge

Setiap peserta dapat mengirimkan output challenge melalui **form submission** disamping ini 👉

**maksimal pada Jumat, 27 September 2024 pukul 23.59 WIB.**



### Untuk kamu yang menggunakan code repository sebagai dokumentasi project...

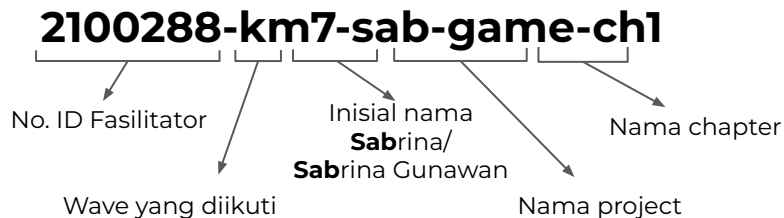
Ada **ketentuan\* tertentu nih yang perlu diikuti dalam penamaan project** kamu. Biar penamaan repository-nya spesifik sesuai akun dan project kamu di BINAR, juga mempermudah BINAR dalam tracking pengerjaan challenge-mu~

Langsung aja cek **ketentuan dan contohnya di bawah ini** ya 📌

**[No. ID fasilitator]-[km7]-[3 huruf inisial nama student]-[nama project]-[ch #]**

\*pastikan menggunakan lowercase semua

#### Contoh penamaan challenge:



### Eh, tapi Sab...

- **Gimana kita bisa tahu no. ID Fasilitator kelas?**  
Kamu bisa tanya langsung ke Fasilitator atau melalui tim SLS BINAR.
- **Ketentuan penulisan inisial nama gimana ya?**  
Seperti contoh di halaman sebelumnya, kamu bisa ikuti cara ini, sob 📌
  - **Nama <3 kata:** pakai tiga huruf pertama dari nama paling depan
  - **Nama ≥3 kata:** pakai huruf pertama dari tiga nama paling depan
- **Setelah lulus dari BINAR, boleh nggak kita ubah nama project di code repository-nya?**  
Tentu boleh, terlebih itu project yang kamu kerjakan hihi. Namun, pastikan tidak ada identitas BINAR maupun yang berkaitan dengan Bootcamp BINAR ya.



## Ketentuan Pengumpulan Challenge

Dengan mempelajari seluruh topic di chapter ini kamu diharapkan mampu menyelesaikan challenge dengan **maksimal dan tepat waktu**. Jika kamu terlambat mengumpulkan challenge maka akan ada pengurangan point berdasarkan lamanya waktu keterlambatan:

- Terlambat maksimal 3 hari: - **5 point**
- Terlambat maksimal 7 hari: - **10 point**
- Terlambat maksimal 14 hari: - **15 point**
- Terlambat lebih dari 14 hari: - **20 point**

### Standar Kelulusan: 60 point

So, pastikan kamu mengumpulkan challenge tepat waktu yaa. Kalau ada kendala, segera hubungi fasilitatormu.





**Sabrina cosplay jadi Avatar...**  
**Yuk, kita kerjain challenge sambil belajar! 🥰**

