

# Full-Stack Web Challenge

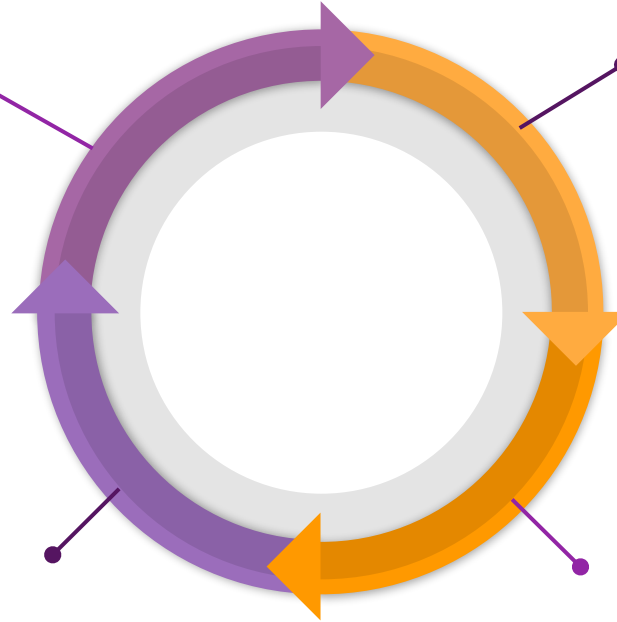
**Platinum** - Chapter 11

### PRODUCT DESIGN & PROTOTYPING

Masing-masing tim akan menentukan fitur apa saja yang akan ada di aplikasi sekaligus desainnya secara langsung.

### TEAM WORK

Setiap student akan bekerja dalam sebuah tim untuk berjuang bersama mencapai goals dan titik temu.



### AGILE METHODOLOGY (SCRUM)

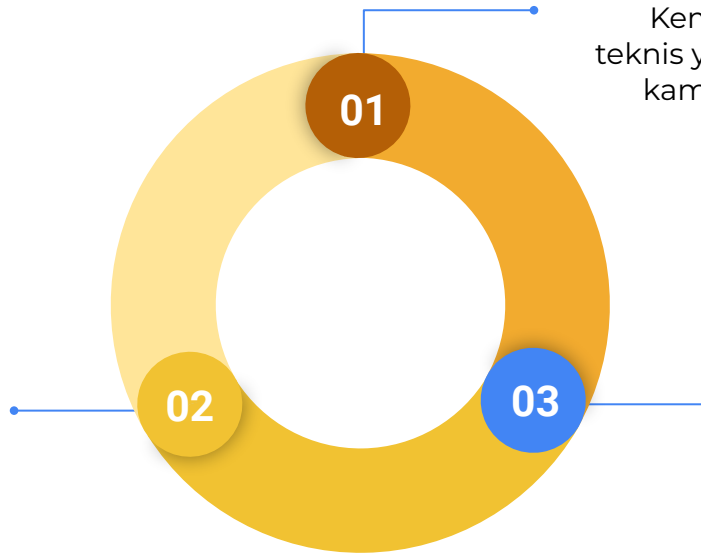
Masing-masing tim akan menggunakan metode Scrum untuk mengerjakan challenge.

### SDLC BEST PRACTICE

Setiap student diharapkan memiliki pemahaman tentang bagaimana menciptakan produk dengan aturan SDLC.

## ACCEPTANCE CRITERIA

Kriteria pengumpulan challenge yang harus kamu penuhi untuk dapat melewati chapter



## SKILL METRICS

Kemampuan teknis yang akan kamu pelajari

## GAME DESIGN DOCUMENT

Hal yang akan kamu buat untuk dapat melewati chapter

### SKILL METRICS

(Kemampuan teknis yang akan kamu pelajari)

#### 1. Media Handling

- Video
- Image
- PDF

#### 2. ESLint

#### 3. TDD (Test-driven Development)

#### 4. CI/CD

### DELIVERY

(Hal yang akan kamu lakukan untuk dapat melewati chapter)

1. Memahami konsep multimedia dan mengimplementasikan solusi untuk handling file multimedia (foto, audio, video, pdf, dan file lainnya)
2. Memahami konsep linting dan mengimplementasikan ESLint dalam project
3. Memahami konsep TDD dan menambahkan test case ke dalam project dengan Jest
4. Memahami konsep CI/CD dan mengimplementasi CI/CD dengan GitlabCI untuk deploy ke Heroku

### CRITERIA

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

1. Menambahkan fitur media handling yang berupa image, video dan pdf viewer ke dalam project
2. Mampu menerapkan prinsip ESLINT sebagai standar penulisan kode yang dibuat
3. Mengimplementasikan TDD(Test Driven Development) ke dalam project menggunakan Jest dengan hasil testing besar/sama dengan 80%
4. Mampu mengimplementasikan Gitlab CI/CD
5. Git flow berdasarkan fitur, dengan branch Feature-Develop-Master. Ketika sprint review semua branch harus sudah di push ke branch Master
6. Membuat dan melakukan presentasi kelompok untuk sprint review yang menjelaskan tentang hasil dari design process dan engineering process

Tugas utama di Chapter 11 ini adalah mengembangkan platform game online yang telah diimplementasi di challenge Chapter 9 dan 10.

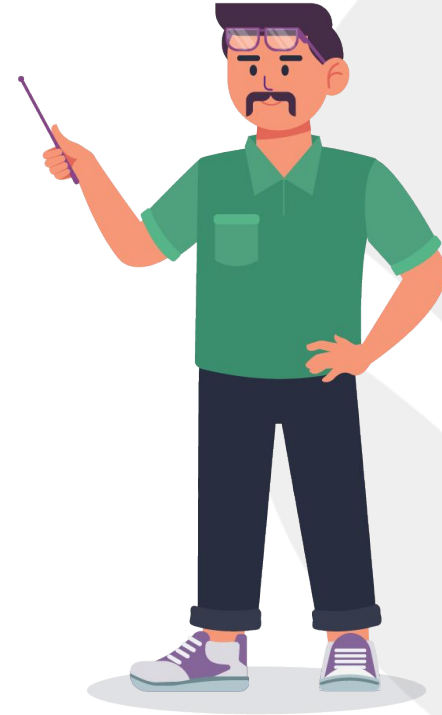
Produk yang baik adalah produk yang mengalami peningkatan/improvement. Naikkan level app kalian dengan menambahkan flow berikut:

- Test case
- Linting
- CI/CD
- Media handling



Seperti biasa, akan ada tiga kategori spesifikasi challenge, yaitu:

1. **Wajib** → fitur utama yang wajib dikerjakan oleh tim. Jika ada fitur wajib yang tidak selesai, akan dikenakan pengurangan nilai tim/kolektif
2. **Opsional** → boleh dikerjakan jika fitur wajib sudah selesai 100%. Jika diselesaikan akan menjadi poin bagi yang mengerjakan
3. **Role** → peran tambahan yang diambil untuk memperlancar kerja tim

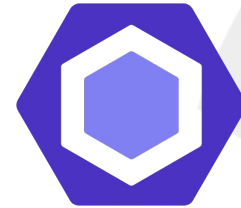




### Fitur Wajib

Fitur yang wajib dikerjakan oleh tim. Jika tidak diselesaikan, akan ada pengurangan nilai secara kolektif.

1. Implementasi ESLint, kemudian perbaiki code yang tidak memenuhi standar ESLint
2. Implementasi GitlabCI



ESLint



### 3. Implementasi unit test untuk BE dan FE:

- BE : test setiap controller/query dengan, minimum dua positive tests dan tiga negative tests. Pakai Jest.
- FE : test setiap component dengan Jest/Enzyme. Minimum dua positive tests dan tiga negative tests.

### 4. Implementasi handling multimedia file. Pilih dua dari requirements berikut.

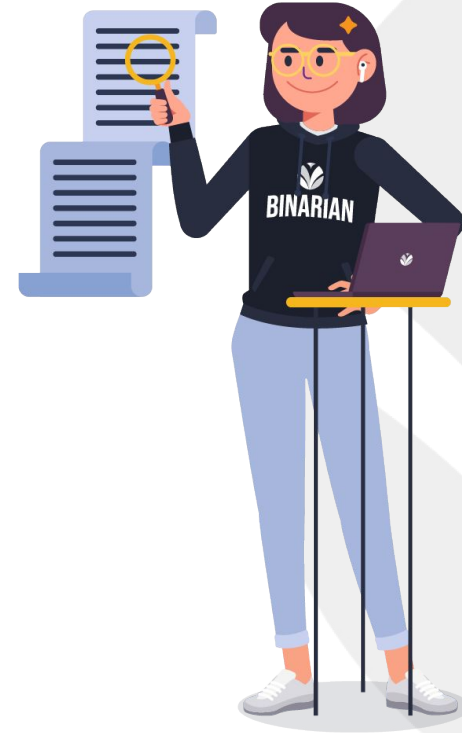
- Upload, tampilkan, dan manipulasi gambar dengan Cloudinary
- Upload & stream video
- Upload & stream audio
- Generate, tampilkan, dan download PDF
- Upload dan download file lain (.excel, .ppt, .zip, .rar, dsb). Gunakan layanan cloud storage



### Fitur Opsional

Fitur yang bisa dikerjakan tim setelah fitur wajib selesai. Poin ekstra diberikan kepada anggota yang mengerjakan.

1. Sisa dari implementasi multimedia file handling
2. Tambahan unit test untuk BE maupun FE
3. Deploy ke cloud server alternatif : Amazon, GCP, Azure, Netlify, dan lain-lain



### Role

Role merupakan peran yang perlu ada dalam tim untuk memperlancar alur kerja. Satu role diisi oleh satu anggota. Anggota yang mengambil role akan dapat poin tambahan jika melaksanakannya dengan baik.

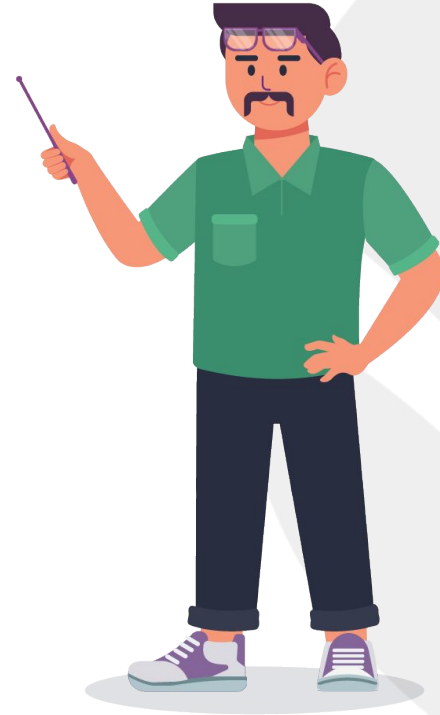
Anggota yang mengisi role disarankan berbeda dari challenge sebelumnya untuk memberikan kesempatan bagi anggota lain.

Role	Deskripsi Kerja
Koordinator / Scrum Master	Mengundang & memimpin pertemuan rutin
Repo Maintainer (2 orang jika repo back-end dan front-end dipisah)	Membuat repo, memberikan akses ke anggota tim, setup branch sesuai aturan git flow, review dan merge branch lain
Boilerplate Coder (2 orang jika repo back-end dan front-end dipisah)	Merancang dan implementasi boilerplate, lalu unggah ke repo
Devops	Setup server+DB untuk deploy, lakukan deploy di server

### Spesifikasi Teknis

Di bagian ini dijabarkan aspek engineering apa saja yang perlu dipenuhi:

- CI/CD menggunakan GitlabCI
- Testing menggunakan Jest
- Handling media foto/image menggunakan cloudinary
- Deploy fitur utama (**wajib**) dengan Heroku



### Git Flow

Berikut adalah panduan umum untuk developer yang akan mengembangkan satu fitur, yaitu:

1. Buat branch baru dan beri nama sesuai kode/fitur yang akan ditambah. Misal :  
`$ git checkout -b feature-login`
2. Lakukan coding seperti biasa. Tambahkan, hapus, atau ubah file yang diperlukan. Git akan tracking secara otomatis. Apabila ingin melihat file yang berubah, cek dengan `$ git status`
3. Untuk menambahkan perubahan, lakukan `$ git add`. atau `$ git add --all`, kemudian `$ git commit -m "add login feature"`.
4. Sebelum melakukan push, pastikan sudah update dengan versi development terbaru :  
`$ git pull origin development`



5. Jika terjadi conflict karena ada anggota tim lain yang mengedit baris code sama, lakukan **resolve di code editor** dengan memilih potongan code yang betul. Ada beberapa opsi:
  - pilih code existing (milik kita),
  - pilih code incoming (milik anggota lain), atau
  - keep both (dua-duanya dimasukkan).Pastikan rangkaian code masih berjalan ketika resolve conflict ini.
6. Lakukan push ke branch feature dengan `$ git push feature-login`
7. Lakukan merge request di Gitlab, pilih branch asal `feature-login` dan branch tujuan `development`. Pastikan tidak ada conflict, lalu **klik create merge request**
8. Maintainer review merge request dan approve apabila tidak ada conflict



### Arahan Kerja Sama Tim

Kerja sama tim yang baik adalah kunci penting agar challenge ini terselesaikan. Berikut adalah arahan dasar agar tim dapat bekerja efektif dan mampu melakukan product delivery secara memuaskan, yaitu:

- **Rancang waktu bertemu rutin** untuk cek progres, identifikasi hambatan, serta sampaikan action berikutnya.
- **Manfaatkan tools** seperti Trello atau Asana untuk mendefinisikan tugas, berbagi tugas, serta tracking tugas.
- **Tidak semua tugas memiliki beban sama.** Pastikan setiap anggota mendapat beban yang adil dan sesuai.
- **Mentor/akademik dapat dimintai tolong** apabila ada anggota tim yang tidak perform atau jarang hadir. Sementara itu, tim perlu segera mengelola beban tugas tersisa.
- **Pastikan fungsi dalam slide “Role” terpenuhi** semuanya, dan orangnya berkomitmen.
- **Komunikasi antar anggota tim** perlu dijaga selalu.





### Informasi Submission

- Challenge di submit paling lambat hari Senin di minggu ketiga chapter 9, pukul 23:59 WIB.
- Perwakilan tim mengirimkan deliverables lewat email kepada mentor dan bagian akademik.

### Deliverables

- Link repository di gitlab untuk source code.
- Link/tautan aplikasi yang telah di deploy.



Terima kasih, selamat  
mengerjakan! 🙌

