

Full-Stack Web Challenge

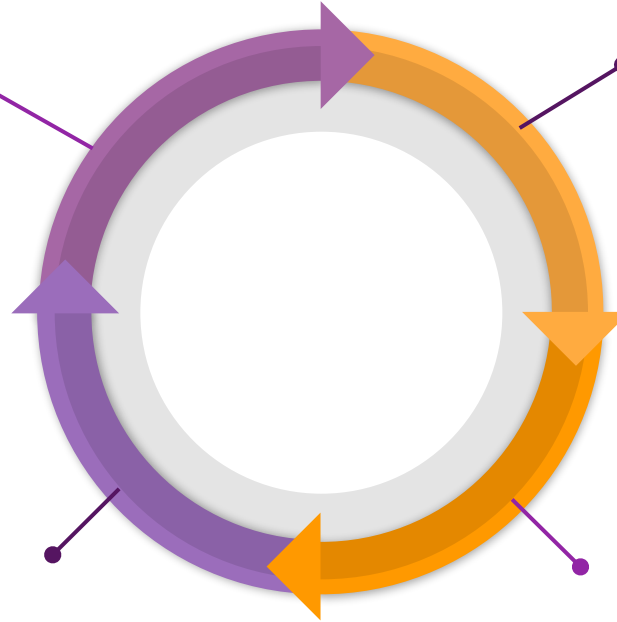
Platinum - Chapter 10

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)

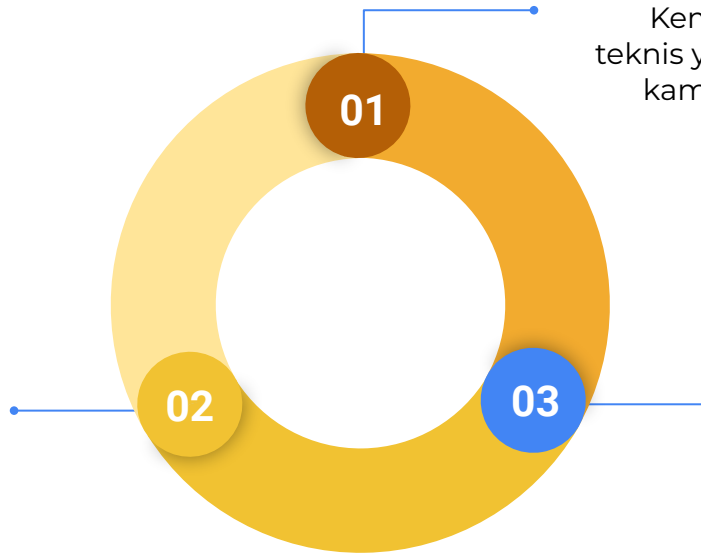
Masing-masing tim akan menggunakan metode Scrum untuk mengerjakan challenge.

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. Redux
 - Redux thunk (Middleware)
2. SSR (Server Side Rendering) pada React/dengan NEXTJS

DELIVERY

(Hal yang akan kamu lakukan untuk dapat melewati chapter)

1. Memahami library redux dan manfaatnya dalam pengembangan front-end
2. Memahami Redux Thunk dan manfaatnya dalam pengembangan front-end
3. Memahami server-side rendering dan perbedaannya dengan SPA
4. Rewrite aplikasi client-side dan mengimplementasi Redux serta server-side rendering

CRITERIA

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

1. Mengimplementasikan library Redux sebagai standard design pattern pada project yang dibuat pada chapter sebelumnya
2. Mengimplementasikan library Redux sebagai State management
3. Mampu menggunakan Redux thunk sebagai middleware antara server dan project
4. Converting project yang semula SPA (Single Page Application) ke dalam bentuk SSR (Server Side Rendering)
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

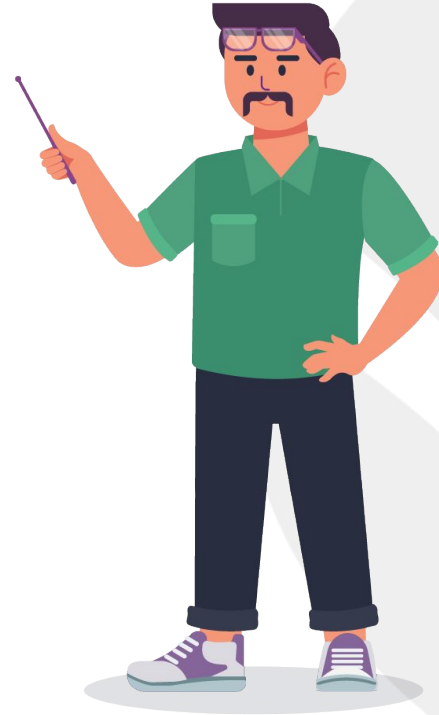
Tugas utama di Chapter 10 ini adalah menulis ulang platform game online yang telah diimplementasi di challenge Chapter 9.

Penulisan ulang bertujuan untuk menambahkan pattern berikut:

- **Redux state management**
- **Server-side rendering**

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. **Optional** → 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 1

- UI Improvements with **Redux**
 - a. **Top Navbar** → tampilkan navbar yang menampilkan tombol **login & register** bagi visitor, dan tampilkan **profil & score** bagi user yang login.
 - b. **Current Game Info** → dalam permainan rock-paper-scissors, tambahkan tampilan informasi seperti **ronde saat ini, status menang/kalah** di ronde-ronde dalam satu game, serta **score** yang diperoleh dalam satu game.
 - c. **Played Games** → apabila user sudah memainkan suatu game, maka akan muncul **tampilan informasi** “pernah dimainkan”.
 - d. **Dummy games** → buat list game baru yang “kosong” alias tidak bisa dimainkan, tapi bisa memunculkan score random, serta tetap mengimplementasi konsep platform seperti pada poin ©.
 - e. **Dynamic button** → buat **tombol dinamis** yang menampilkan info seperti “processing” dan “done” jika diklik. Berguna antara lain untuk : register, login, update profil, play game, dan lain-lain.



Fitur Wajib 2

1. Server side rendering implementation with **Next.js**
 - **Landing page:** halaman utama yang diakses pengunjung web
 - **Register page:** daftar layanan platform
 - **Login page:** login ke layanan platform
 - **Home page:** halaman awal yang diakses pengunjung yang login
 - **Profile page:** halaman informasi user
 - **Profile update page:** halaman untuk ubah informasi user
 - **Game list page:** halaman yang menunjukkan list game tersedia
 - **Game detail page:** halaman yang berisi permainan beserta leaderboard
2. Student diberi kesempatan untuk berkreasi dengan fitur dan aspek tampilan di dalam masing-masing page. Yang penting semua fungsi wajib terpenuhi.

The image shows the Next.js logo, which consists of the word "NEXT" in a bold, sans-serif font, followed by ".JS". A diagonal line crosses through the "X" in "NEXT". The logo is set against a light gray background with a large, stylized leaf or petal shape.

Fitur Opsional

- **Game Leaderboard:** menampilkan daftar user dengan skor tertinggi di suatu permainan
- **New Game(s):** buat permainan baru dengan aturan dan scoring
- **Forgot password:** mengimplementasi logic untuk reset password dengan metode pengiriman link via email
- **Share button:** Tombol share permainan via media sosial
- **Badge/achievement:** daftar penghargaan yang bisa diterima user jika sudah menyelesaikan hal tertentu
- **Social login** (Facebook, Google, Twitter, Github, 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.

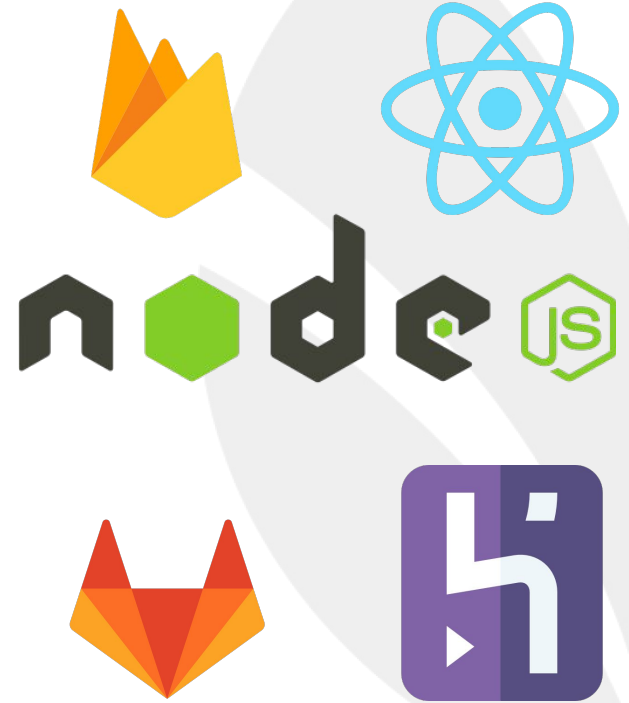
Anggota yang mengisi role disarankan berbeda dari challenge sebelumnya untuk memberikan kesempatan bagi anggota lain.

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**, yaitu:

- Aplikasi menggunakan arsitektur client-server
- Bagian client (front-end) diimplementasi menggunakan Next.js yang memiliki sintaks serupa ReactJS
- Bagian server (back-end) diimplementasi dengan NodeJS atau Firebase atau kombinasi keduanya
- Proses development dalam tim menggunakan Git. Repository yang dipakai adalah Gitlab.
- Repository back-end dan front-end boleh dijadikan satu maupun dipisah, sesuai kesepakatan tim.
- Alur kerja development dengan Gitlab mengikuti slide “Git Flow” dari file ini.
- Aplikasi **harus di-deploy**. Bisa gunakan layanan cloud hosting gratis seperti Heroku dan sejenisnya.



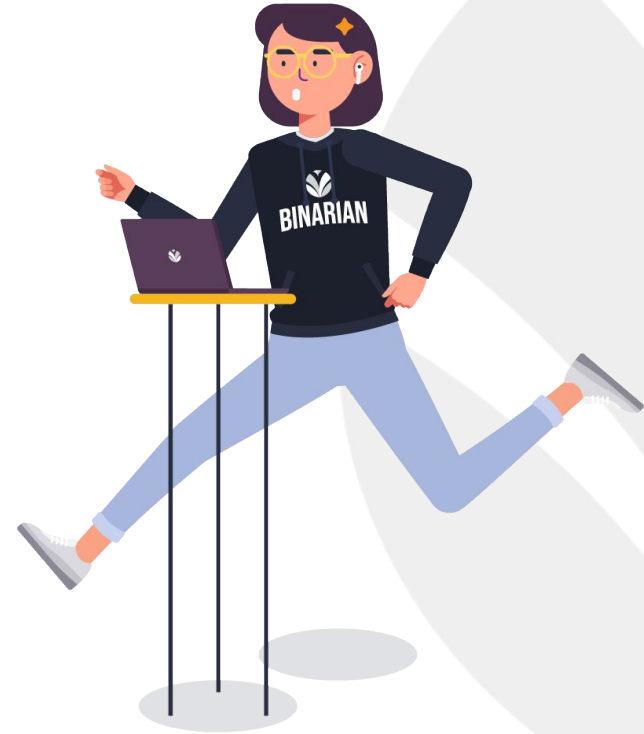
Git Flow

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! 🙌

