

Full-Stack Web Challenge

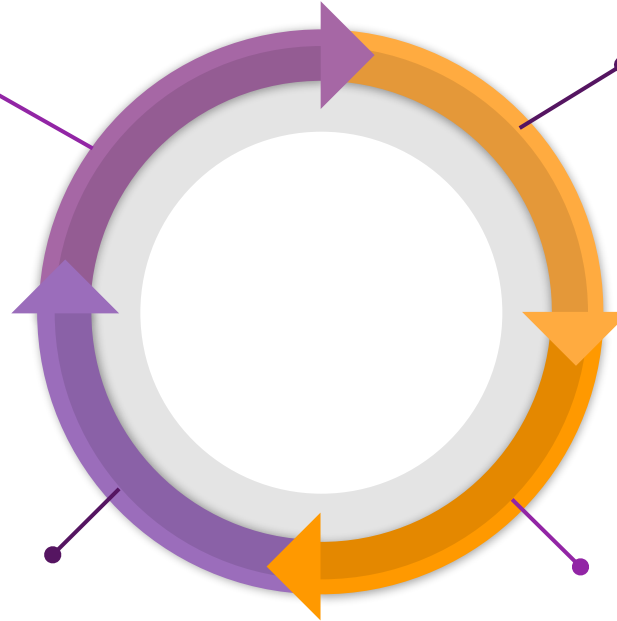
Platinum - Chapter 9

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)

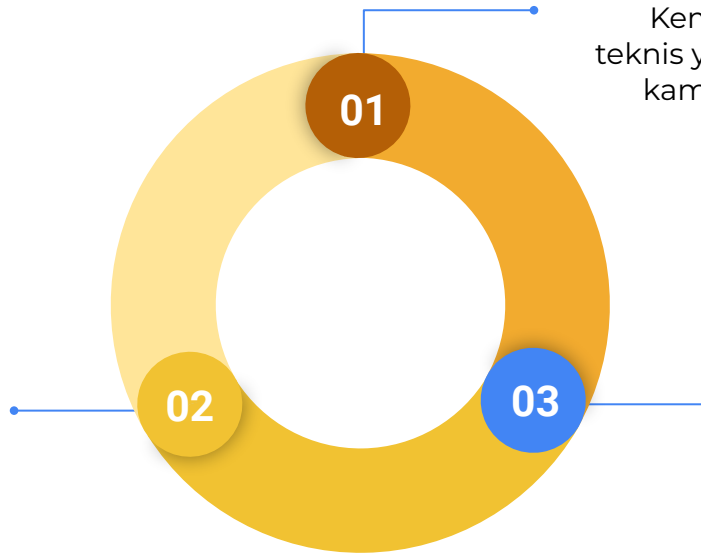
Masing-masing tim akan menggunakan metode Scrum untuk mengerjakan challenge.

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)

[P.9.FSW.SPA Menerapkan Single Page Application 2]

1. REACTJS [SPA] 2

- Menggunakan UI framework
- Routing
- Data
- GET data dari server
- File processing
- Auth
- Middleware

[P.9.FSW.RDS Menggunakan Realtime Database Service]

2. FIREBASE INTEGRATION

- Firebase auth
- Firebase storage
- Firebase Database

DELIVERY

(Hal yang akan kamu lakukan untuk dapat melewati chapter)

1. Mengimplementasikan routing pada aplikasi client-side React
2. Menggunakan HTTP GET request untuk mengakses back end atau API eksternal
3. Mengupload file ke back end atau Firebase
4. Mengimplementasikan auth lewat back end atau Firebase
5. Mengakses data dari database via back end atau Firebase
6. Merancang UI dari aplikasi client-side React

CRITERIA

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

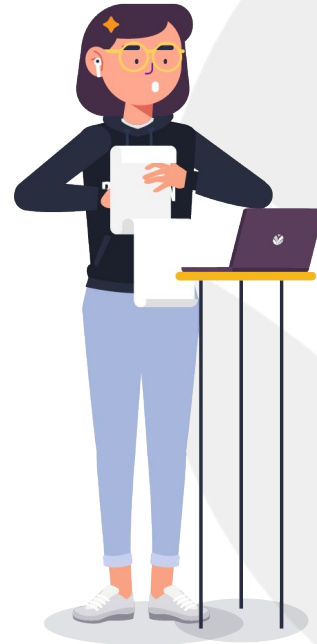
1. Merancang dan membangun project dengan tim
2. Menerapkan SDLC Agile Scrum dalam kerja tim
3. Project harus berkaitan dengan case study yang ada di gold phase
4. Mengimplementasikan semua aspek delivery
5. Mampu mengimplementasikan UI framework terhadap project yang dibangun
6. Mampu menerapkan routing untuk pembeda halaman
7. Mampu mengolah data dan auth yang berasal dari server
8. Mengimplementasikan middleware yang ada di backend ataupun di frontend
9. Mampu menggunakan fitur dari firebase
10. Membuat flowchart sesuai dengan fitur yang ditentukan
11. Membuat prototype lofi sesuai dengan fitur yang ditentukan
12. Design tidak jauh beda dengan yang pernah dibuat di Gold phase
13. Git flow berdasarkan fitur, dengan branch Feature -> Develop -> Master. Ketika sprint review semua branch harus sudah di push ke branch Master
14. Membuat dan melakukan presentasi kelompok untuk sprint review yang menjelaskan tentang hasil dari design process dan engineering process

Pada challenge Chapter 9 ini, tugas utama yang perlu dikerjakan adalah membuat **platform game online**.

Lebih detailnya, kita akan membuat website yang memungkinkan user bermain berbagai jenis game online lewat browser tanpa perlu mengunduh atau instalasi file apapun.

Beberapa contoh platform game online yaitu:

- [Crazy Games](#)
- [Miniclip](#)
- [Armor Games](#)
- [AARP Games](#)





Nantinya, akan tersedia tiga jenis 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. Auth Basics

- **Register**
- **Login**

2. Website Pages

- **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
- **Game list page** : halaman yang menunjukkan list game tersedia
- **Game detail page** : halaman yang berisi permainan beserta leaderboard



3. Business Rules

- User harus login untuk bisa main game, tapi bisa melihat daftar game yang tersedia
- Minimum tersedia game *rock-paper-scissors* untuk 1 player vs COM (lihat challenge chapter 4 untuk detail implementasi)
- User bisa melihat info user lain. Kemudian, user hanya bisa ubah info dirinya sendiri.
- User yang bermain game kemudian menang akan mendapat poin. Poin tersebut ditambahkan ke poin total yang didapat user



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
- **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, dll)



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.

| Role | Deskripsi Kerja |
|---|--|
| Koordinator / Scrum Master | Mengundang & memimpin pertemuan rutin |
| Repo Maintainer (dua 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 (dua 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 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.

- **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 tiap 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.

Contoh Implementasi* : Tabel User

User

- **Id (PK)**
- Email
- Username
- Password
- Total_score
- Bio
- City
- social_media_url

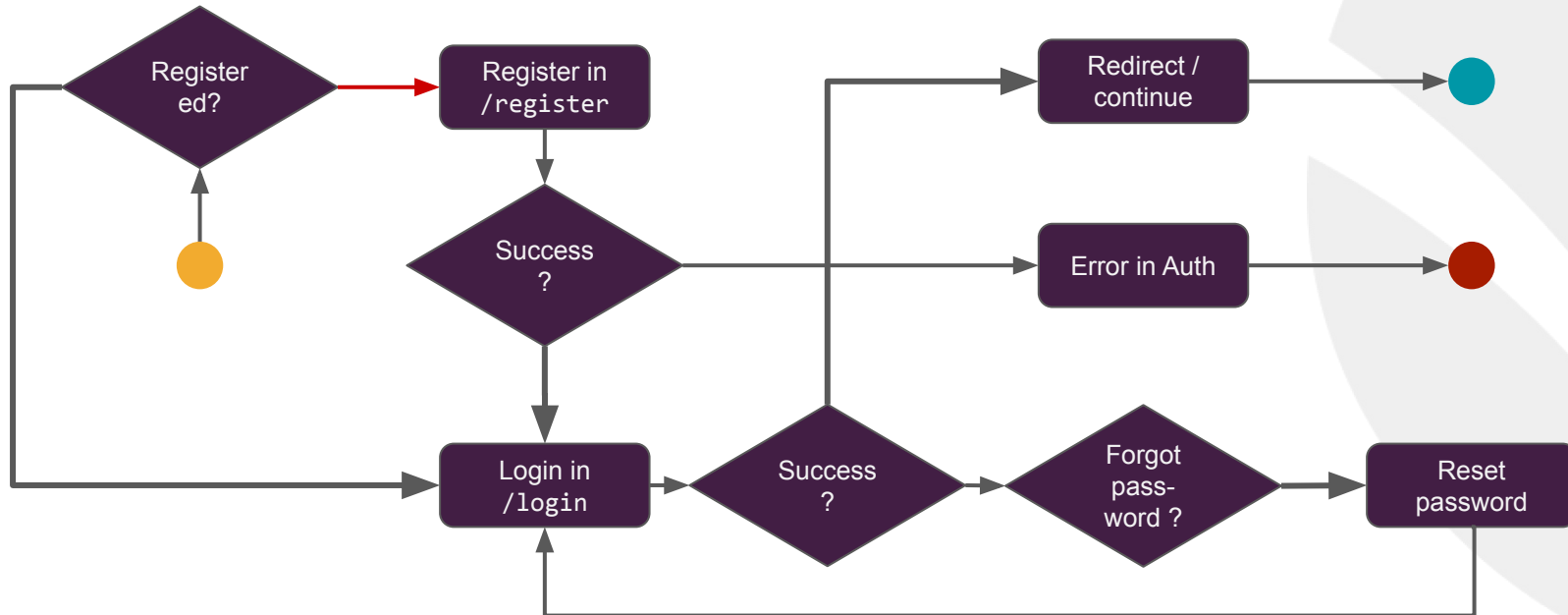
Contoh Implementasi* : Tabel Game

Game

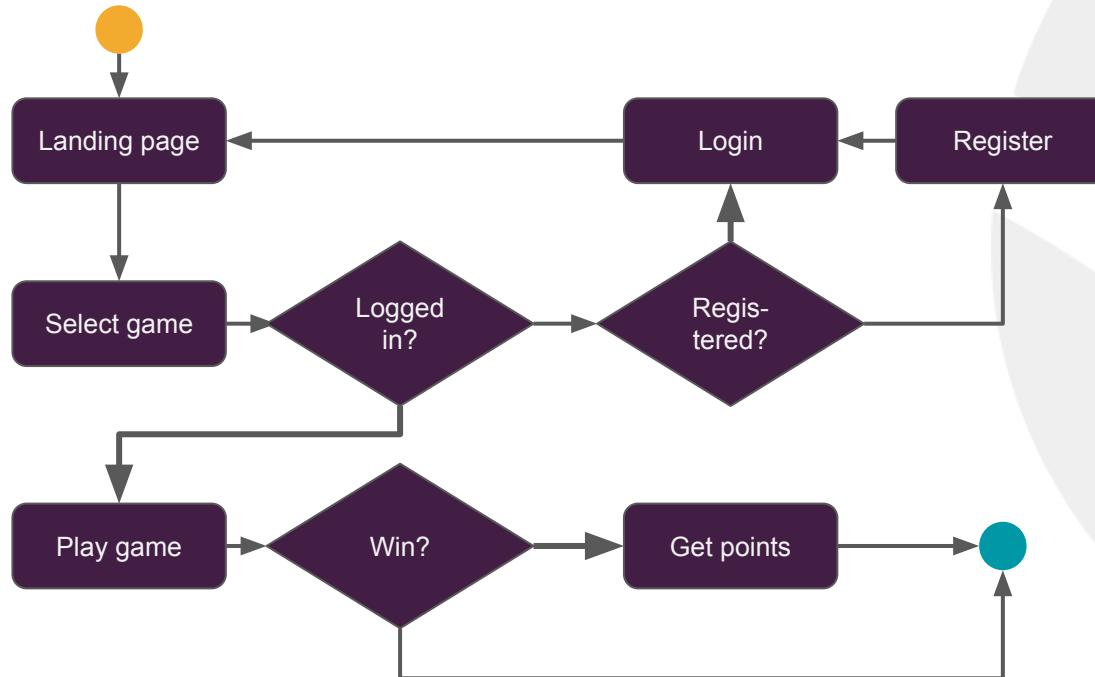
- **Id (PK)**
- name
- Description
- thumbnail_url
- Game_url
- play_count

* (tidak perlu implementasi sama persis)

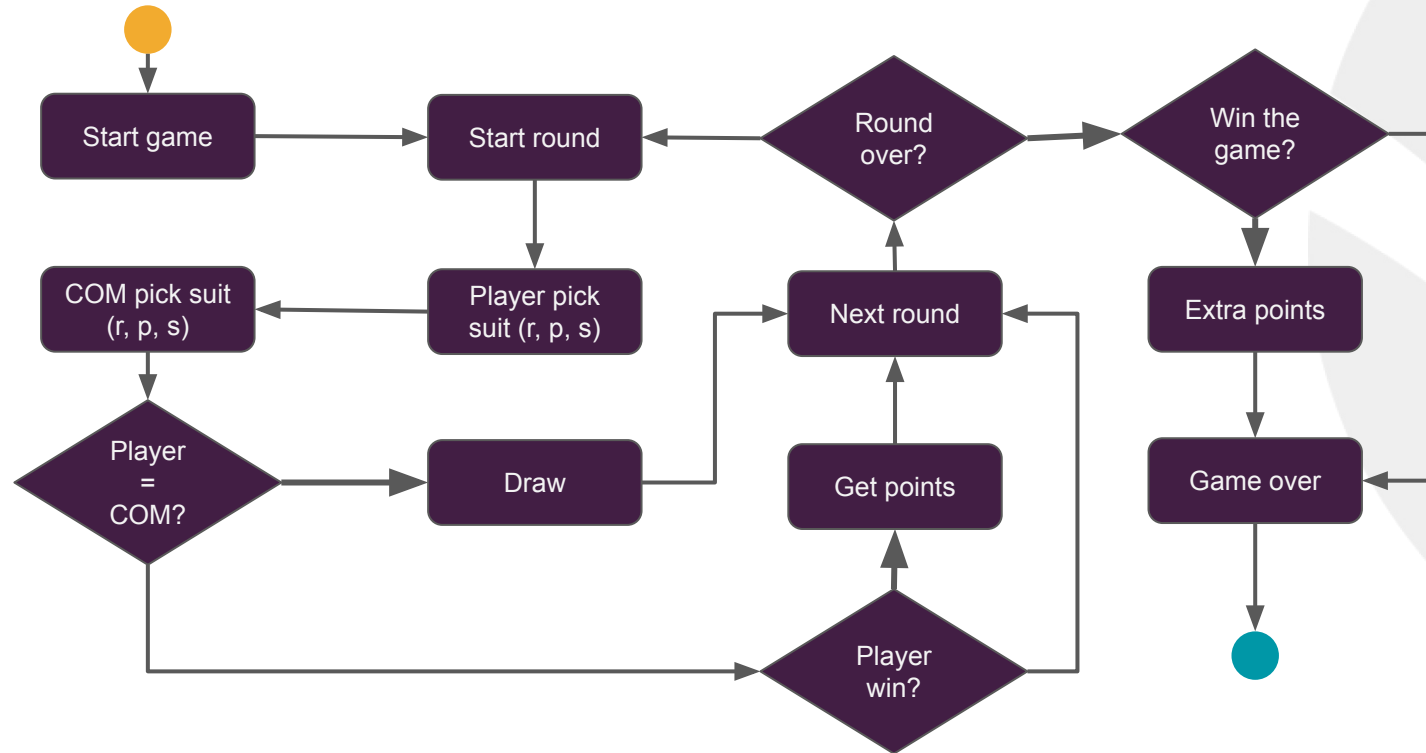
Flowchart: Basic User Auth



Flowchart: Playing A Game



Flowchart: Rock-Paper-Scissors Game



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! 🙌

