



**Reach Native**  
**Chapter 04 - Challenge**

---



### SKILL METRICS

Kemampuan teknis  
yang akan kamu  
pelajari

01

### ACCEPTANCE CRITERIA

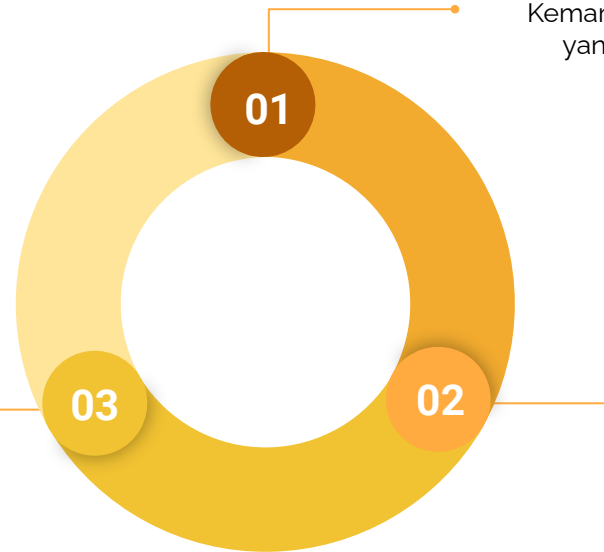
Kriteria pengumpulan  
challenge yang harus  
kamu penuhi untuk  
dapat melewati  
chapter

03

02

### DELIVERY

Hal yang akan kamu  
lakukan untuk dapat  
melewati chapter





### SKILL METRICS

(Kemampuan teknis yang akan kamu pelajari)

- Redux State Management 1
  - Concepts and Data Flow
  - State, Actions, and Reducers
  - Redux Store
- Redux State Management 2
  - Integrasi UI dan Redux
  - Async Logic dan Data Fetching ke komponen
- Redux Middleware
  - Redux thunk middleware
  - Implementasi Redux thunk
- Implementasi API ke redux
  - CRUD di UI
- Redux Debugging
  - Redux DevTools



### DELIVERY

(Hal yang akan kamu lakukan untuk dapat melewati chapter)

- Buat sebuah project react-native baru, beri nama project tersebut **bookapp-redux**
- Kemudian tulis semua kode react-native kamu di project tersebut
- Inisialisasi git pada folder project tersebut, kemudian **push** semua file jawaban yang telah kamu buat di dalam folder tersebut ke **remote-repository**
- Submit link remote-repository yang telah dibuat pada form yang disediakan oleh fasilitator
- Fasilitator akan menunjuk kamu untuk melakukan persentasi/demo singkat terkait project yang kamu buat kali ini.



### CRITERIA

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

1. Mampu mengimplementasikan cara menampilkan data dari RESTFul API
2. Mampu mengimplementasikan Authentication
3. Mampu membuat Loading Indicator pada **Home Screen** dan **Book Detail Screen**
4. Mampu mengimplementasikan Fitur RefreshControl pada **Home Screen** dan **Book Detail Screen**
5. Mampu mengimplementasikan Penerapan Fitur Internet connection checker (Menampilkan pesan error apabila aplikasi tidak terkoneksi dengan internet)
6. Mampu mengimplementasikan fitur Share Content
7. Mampu mengimplementasikan fitur Push notification



Pada challenge kali ini kamu akan membuat sebuah **aplikasi yang menampilkan list dari buku-buku.**

Ingat challenge kali ini kalian wajib menggunakan arsitektur Redux dan Redux-thunk!



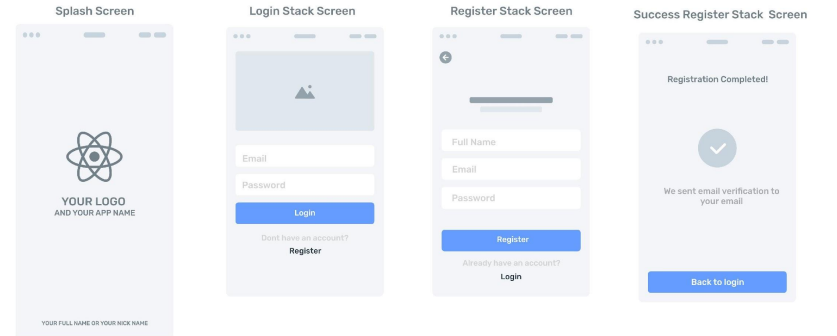


Kamu diberikan kebebasan untuk menentukan desain aplikasi kamu seperti apa, kamu bisa memilih warna, jenis font, dan beberapa unsur desain lainnya sesuai dengan keinginan kamu. Akan tetapi layout dasar dari aplikasi ini harus memenuhi kriteria desain wireframe pada gambar disamping ini.

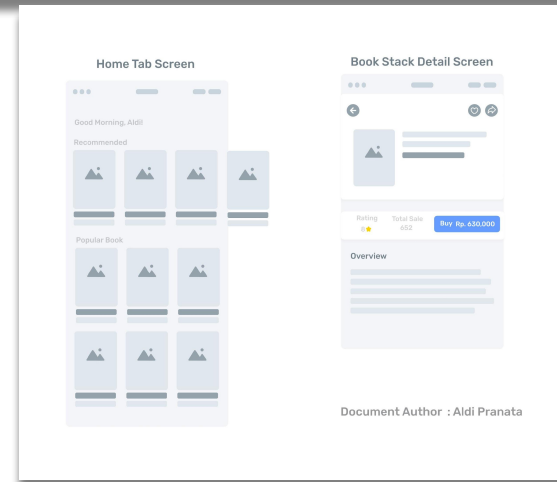
Ada 6 screen yang harus kamu buat, yaitu **Splash Screen**, **Login Screen**, **Register Screen**, **Success Register Screen**, **Home Tab Screen**, dan **Book Detail Screen**.

**Link Figma :**

**<https://www.figma.com/file/Mx2WpWBWdWAZjLie6VCewc/KM---Book-App?node-id=126%3A57>**



Document Author : Aldi Pranata



Document Author : Aldi Pranata

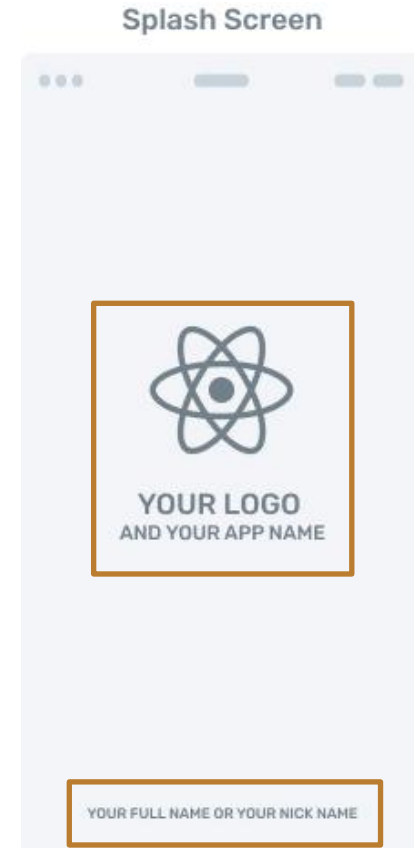


## Splash Screen - Requirement

**Splash Screen** adalah screen yang tampil Ketika aplikasi pertama kali dibuka, dan akan hilang secara otomatis dalam beberapa detik.

Pada splash screen ini kamu diminta untuk menampilkan logo dan nama aplikasi kamu.

Kemudian dibagian paling bawah **Splash Screen** tampilkan juga nama lengkap kamu.







## Login Screen - Requirement

**Login Screen** adalah screen yang akan muncul setelah splash screen selesai.

Pada screen ini terdapat 2 field yaitu Email dan Password, apabila user mengetikkan email dan password yang telah terdaftar maka User akan di arahkan ke halaman **Home Screen**, tapi apabila user memasukkan email dan password yang salah maka akan muncul message error.

Ketika user klik button "Register",  
aplikasi akan mengarahkan user ke  
**Register Screen**

### Login Stack Screen

Mockup of a mobile app login screen titled "Login Stack Screen". The screen displays a large light blue rectangle with a mountain icon, followed by "Email" and "Password" input fields, a blue "Login" button, and a "Don't have an account?" link with a "Register" button below it. An orange arrow points from the "Register" button to the explanatory text box on the left.



## Login Screen - API

Dengan API ini user dapat melakukan login, dan mendapatkan **access\_token** yang berfungsi sebagai **Authorization Header** untuk melakukan API request nantinya.

PATH ENDPOINT	METHOD
http://code.aldipeeee.com/api/v1/auth/login	POST

```
// Contoh Request Body untuk Login

{
  "email": "aldialdi@gmail.com",
  "password": "pisangisbanana2021"
}
```



## Register Screen - Requirement

Register Screen adalah screen yang akan muncul ketika user mengklik text “Register” pada **Login Screen**.

Pada screen ini terdapat 3 field yaitu **Full Name, Email dan Password**.

Pada proses ini user akan melakukan registrasi dengan mengirimkan data nama lengkap, email dan password. Ketika proses registrasi berhasil, user nantinya akan dikirimkan sebuah email untuk verifikasi.

Barulah setelah proses verifikasi selesai, user dapat melakukan login.

Ketika user klik text “Login”, aplikasi akan mengarahkan user ke Login Screen

### Register Stack Screen

←

Full Name

Email

Password

Register

Already have an account?

Login



## Register Screen - Requirement

- Pada Screen ini kamu wajib membuat Validasi Field, seperti apabila user memasukkan data email yang tidak valid contoh 'emailkamu@kamu'; maka akan muncul sebuah error pada form inputnya, dan apabila user memasukkan password yang panjang karakter nya kurang dari delapan karakter, maka akan muncul error pada input fieldnya.
- Setelah data register user berhasil dikirim ke Server, maka user akan diarahkan ke screen Success Register.
- Apabila data register user gagal dikirim ke Server, maka user tidak di arahkan ke **Succes Register Screen** dan kamu harus memunculkan message pop up error.

### Register Stack Screen

Register Stack Screen

←

[Redacted]

Full Name

Email

Password

Register

Already have an account?  
Login



## Register Screen - API

Register digunakan untuk registrasi user baru pada sistem. Setelah register user akan dikirimkan sebuah email verifikasi. Apabila email tidak masuk pada inbox, coba check di spam.

PATH ENDPOINT	METHOD
<code>http://code.aldipeeee.com/api/v1/auth/register</code>	POST

```
// Contoh Request Body untuk Register
```

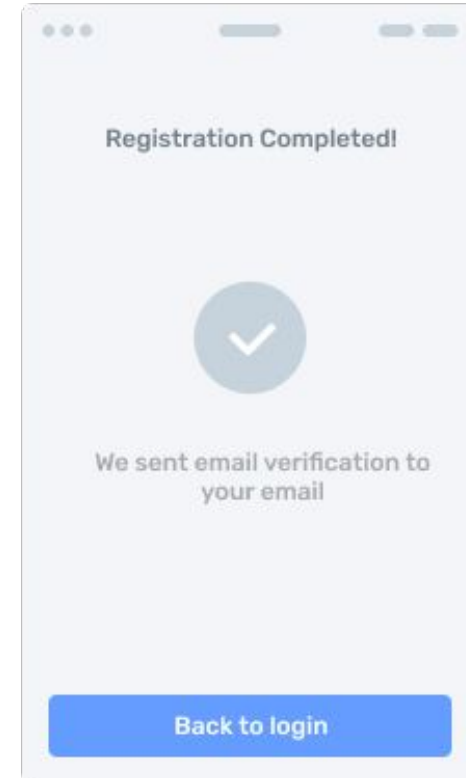
```
{  
  "email": "aldialdi@gmail.com",  
  "password": "pisangisbanana2021",  
  "name" : "Aldi Daniela Pranata"  
}
```



## Success Register Screen - Requirement

- **Success Register Screen** hanya akan muncul apabila proses registrasi user pada Register Screen Berhasil.
- Tidak ada proses pemanggilan API pada screen ini.
- Ketika button “Back to login” di klik, aplikasi harus menampilkan “**Login Screen**”.

### Success Register Stack Screen





## Home Tab Screen - Requirement

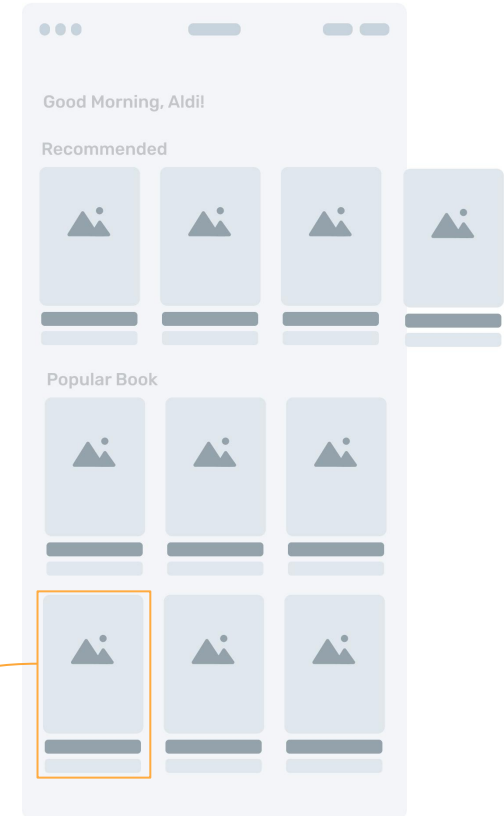
Home Tab Screen akan menampilkan list-list buku yang terdapat pada API

Terdapat dua bagian pada Home Tab Screen yaitu Recommended dan Popular Book.

- Pada Bagian Recommended list buku yang ditampilkan dapat di scroll kesamping / secara horizontal.
- Pada Bagian Popular Book list buku yang ditampilkan dapat di scroll kebawah.
- Khusus pada bagian Recommended, data list buku-buku yang ditampilkan pada bagian Recommended adalah data yang telah di sorting berdasarkan **average\_rate** tertinggi ke terendah, dan yang tampilkan hanya 6 data buku saja.

Apabila gambar buku di klik akan navigasi ke Book Detail Screen

### Home Tab Screen





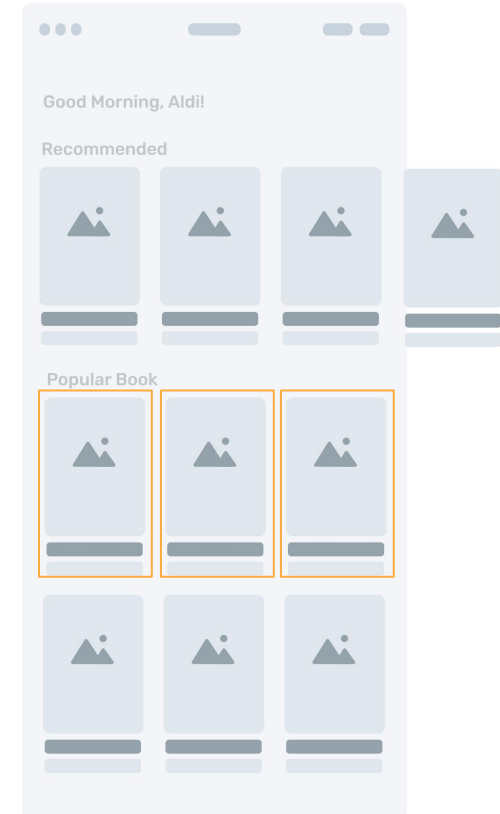
## Home Tab Screen - Requirement

Data yang harus kamu tampilkan dalam setiap item buku adalah

- Gambar Buku
- Judul Buku
- Author Buku
- Publisher Buku
- Rating Buku
- Harga Buku

Khusus untuk **Harga Buku**, data yang diberikan dari server akan berbentuk "292532", kamu harus mengubah data harga buku tersebut dalam bentuk format rupiah "Rp. 292.532", kemudian baru datanya kamu tampilkan.

### Home Tab Screen







## Home Tab Screen - API

Untuk menampilkan list data buku, gunakan API ini.  
API ini bersifat private dan membutuhkan

### Authorization

PATH ENDPOINT	METHOD
<a href="http://code.aldipee.com/api/v1/books">http://code.aldipee.com/api/v1/books</a>	GET

## Query Parameter

Query Name	Deskripsi
title	Untuk mencari buku berdasarkan judul
limit	Untuk mengatur berapa banyak jumlah data yang akan ditampilkan dalam sekali request

Contoh penggunaan query parameter:

**<http://code.aldipee.com/api/v1/books?title=Harry Potter>**



## Book Detail Screen - Requirement

Berikut kriteria data yang harus ditampilkan pada **Book Detail Screen**

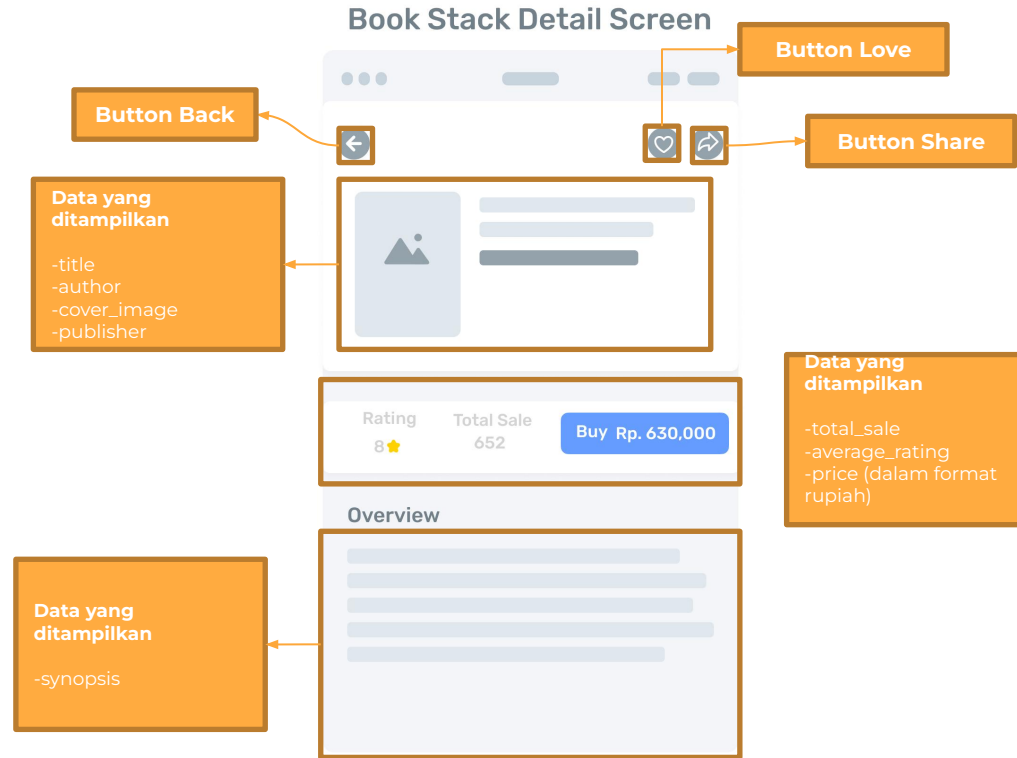
Aplikasi kamu minimal harus menampilkan data-data yang telah ditentukan pada gambar disamping.

Perhatikan pada bagian atas screen pada gambar disamping, terdapat tiga button; paling kiri itu back button, kemudian paling kanan itu ada dua button yang terdiri dari icon love dan icon share.

Requirement fungsionalitas untuk button-button tersebut adalah :

**Button Back** ⇒ Ketika diklik harus navigasi ke **Home Screen**

**Button Share** ⇒ Ketika diklik harus memunculkan fitur share

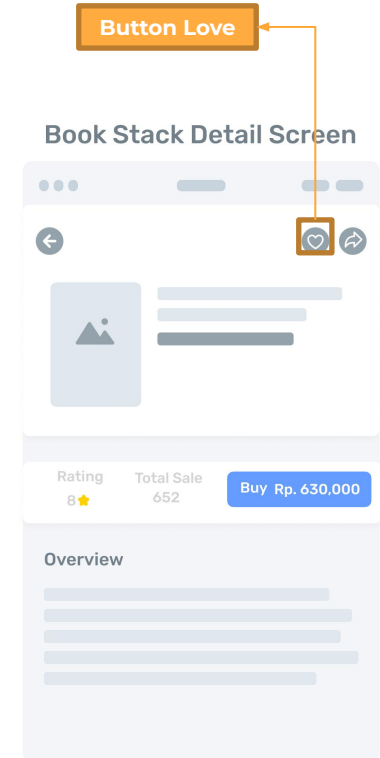
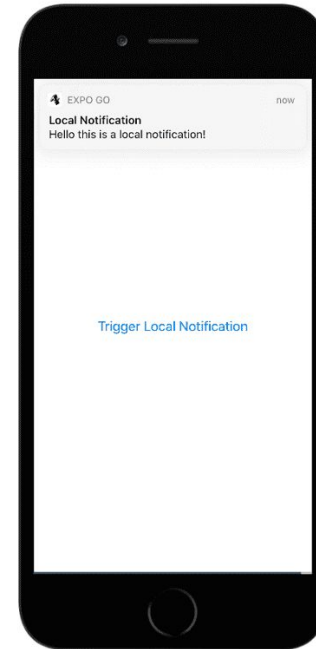
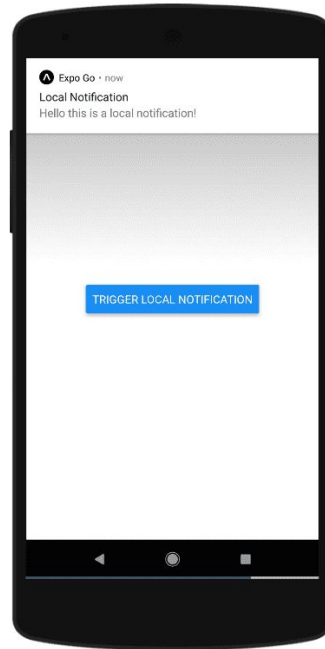




## Book Detail Screen - Requirement

**Button Love** ⇒ Ketika diklik tampilkan push notification dengan title “Kamu menyukai \${JUDUL BUKU}”. Contoh “Kamu menyukai Harry Potter”.

Fitur Push notification yang dimaksud seperti gambar disamping ini.





## Book Detail Screen - API

Gunakanlah API ini untuk menampilkan detail data buku berdasarkan ID . API ini bersifat private dan membutuhkan **Authorization**

PATH ENDPOINT	METHOD
http://code.aldiptee.com/api/v1/books/\${BOOK-ID}	GET



## Resources Desain

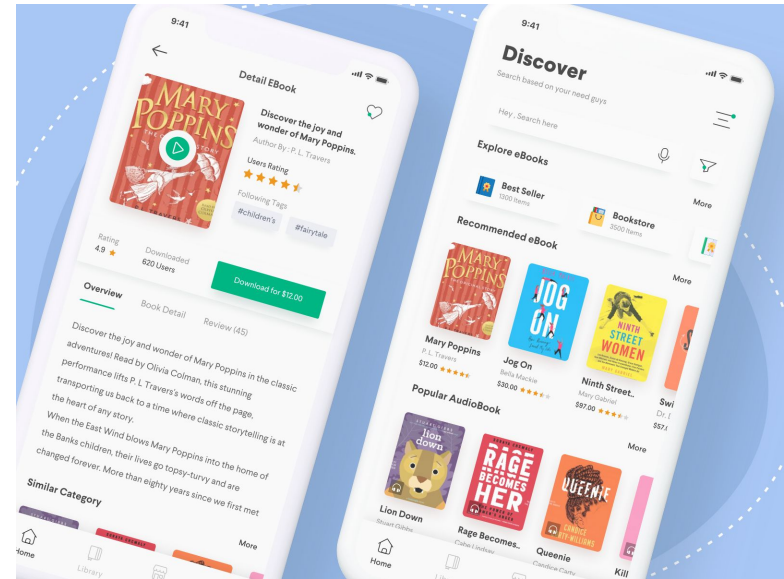
Untuk kamu agar mudah dalam mencari inspirasi desain buat aplikasi kamu, kamu dapat mencari ide-ide desain yang berkaitan dengan Book App' pada website Dribbble.com.

Kamu bisa mendapatkan ide ide desain dari sana, kemudian menggabungkan ide desain yang kamu dapat dengan ide desain yang kamu punya

Link

Dribbble:

<https://dribbble.com/shots/6212888-eBook-Audiobook-app>





## Resources Tutorial Local Push Notification

Untuk dapat mengimplementasi Local Push Notification pada aplikasi kamu, kamu dapat nonton dan mengikuti step-by-step video pada link youtube disamping.

<https://www.youtube.com/watch?v=IdXhIsYANbY>



Nah apabila kamu telah selesai mengerjakan semua soal, maka persiapkan diri kamu untuk melakukan persentasi dan menjawab pertanyaan yang akan diberikan oleh fasilitator ya.

Facilitator akan memberikan kamu pertanyaan acak yang bertujuan untuk mengetes pemahaman koding kamu.

**Selamat mencoba dan Happy Explore!**





**Terima  
Kasih**