

Isyraq Sajid Adli

Mini Challenge Paragon

Pada kesempatan kali ini saya akan mencoba untuk menjelaskan hasil implementasi website dari pedoman mini challenge menggunakan framework React JS.

1. Hal yang pertama dilakukan yaitu pembuatan komponen Header dan komponen Home. Kemudian import cdn dari Bootstrap untuk memudahkan styling pada css, dan menginstall Reactrouter, ReactIcon, axios, dan import font style 'Effra'.
2. Pada komponen Home import Reactrouter dan Axios pertama kali dilakukan. Dimana Reactrouter berfungsi untuk berpindah halaman dari Home menuju halaman lainnya seperti Shop, Brand, Kahf Program, Discovery, Login Page, dan Shopping Cart. Dan axios akan mengambil data dari url 'https://paragon-training-api.herokuapp.com/cart' yang akan diberikan pada icon trolley yang tersimpan pada komponen HeaderTopComp, dimana memiliki fungsi untuk melihat berapa banyak product yang sudah dimasukkan kedalam trolley.
3. Kemudian Header terbagi lagi menjadi dua komponen yaitu: HeaderTopComp dan HeaderBottomComp.
4. Pada HeaderTopComp saya membagi menjadi 3 div yaitu: header-left, header-title dan header-cart. Kemudian HeaderTopComp diberikan style dengan library Bootstrap yaitu 'd-flex' dan styling khusus menggunakan className untuk memberikan nilai flex-grow:1. Kemudian import library Reactrouter agar icon trolley jika di *click* dapat berpindah ke halaman Cart.
 - Pada header-title menggunakan text-align: center
 - Pada header-cart terbagi menjadi 3 div yaitu: header-cart-trolley, header-cart-heart, header-cart-login. Kemudian styling menggunakan library Bootstrap yaitu 'd-flex' dan 'justify-content-end'.
 - Pada div header-cart-trolley diberikan link yang mana jika user *click* maka akan ter redirect ke halaman Cart. Dimana berfungsi untuk melihat summary dari produk yang telah dimasukkan melalui "Add to Cart" pada halaman Shop.
 - Pada div header-cart-login diberikan link yang mana jika user *click* maka akan ter redirect halaman Login Page

5. Kemudian pada HeaderBottomComp saya membagi menjadi 3 div yaitu: header-bottom-logo, header-bottom-link dan header-bottom-search. Dengan style menggunakan library Bootstrap yaitu 'd-flex' dan 'justify-content-between'. Lalu import library Reactrouter untuk memudahkan perpindahan halaman jika user meng*click* navbar yang tersedia.
 - Pada header-bottom-logo diberikan gambar logo sesuai dengan desain.
 - Pada header-bottom-link diberikan link atau navbar yang akan ter redirect ke tujuan page yaitu Home, Shop (komponen Product), Brand, Kahf program, Discovery.
 - Pada header-bottom-search diberikan input dengan style library Bootstrap, kemudian diberikan icon search pada input tersebut.
6. Pada komponen Main saya membaginya menjadi 2 komponen yaitu: LoginPage dan SignUpPage. Dimana pada komponen Main, image akan bersifat statis tetapi LoginPage dan SignUpPage akan bersifat dinamis. Komponen Main diberikan style dengan library Bootstrap yaitu 'container', agar image berada dibelakang komponen LoginPage/SignUpPage maka dilakukan position: relative pada komponen Main dan position: absolut pada komponen LoginPage/SignUpPage.
 - Pada LoginPage diberikan sebuah fungsi pada form yang akan memberikan validasi apakah berhasil login atau tidak, yang mana email dan password pertama kali akan dibuat dengan data dummy yang dapat di akses pada folder Account dengan nama file Account.js yang menyimpan sebuah object dengan nilai {'adlidimas12@gmail.com': '123456Ad'}. Apabila input pada form sesuai dengan key dan value dari file Account.js dan button 'LOGIN' di *click* maka akan *pop up* sebuah *alert*('Anda Berhasil Login') user akan akan langsung ter redirect ke halaman Shop. Tetapi jika salah satu dari email maupun password tidak memiliki nilai yang sama dengan key dan value dari file Account.js maka akan *pop up* sebuah *alert*('Email/Password Anda Salah').
 - Pada SingUpPage diberikan sebuah fungsi pada form yang akan memberikan name, email dan password ketika melakukan *signup*. Dimana *singup* akan berhasil apabila user memberikan input email yang berbeda

dari data dummy yang sudah dibuat. Apabila email sama dengan data dummy yang sudah dibuat maka akan *pop up* sebuah *alert*('Email Telah Terdaftar'). Kemudian akan ada sebuah validasi dimana panjang karakter password harus lebih dari 8, ketika user memberikan input password kurang dari 8 karakter maka akan *pop up* sebuah *alert*('Password yang anda masukkan kurang dari 8 karakter'). Ketika semua syarat diatas sudah terpenuhi dan button 'REGISTER' di *click* maka user akan ter redirect ke halaman Shop. Kemudian apabila user kembali ke Login Page. User dapat berhasil *login* dengan menginput email dan password yang sesuai pada saat melakukan *signup*.

- Pada input password di LoginPage maupun SignUpPage user dapat meng*click* icon mata yang berfungsi untuk melihat atau menyembunyikan password yang telah di ketik.
- LoginPage dan SignUpPage pertama kali diberikan style dengan library Bootstrap, yang kemudian dilakukan style khusus dengan *className* untuk setiap *div*, *input*, *label*, *button*. Pada footer setiap LoginPage dan SignUpPage diberikan sebuah *anchor text*, jika user *click* akan berpindah ke komponen LoginPage atau SignUpPage.

7. Pada halaman Shop (komponen Product) saya membaginya menjadi 1 komponen yaitu: CardProduct. Dimana style pada komponen Product menggunakan library Bootstrap 'container' agar komponen CardProduct berada di tengah page dan *display: flex* dengan method *flex-wrap: wrap* kemudian *justify-content: space-between*. Kemudian import *axios* lalu menjalankan method GET agar mendapatkan data dari url 'https://paragon-training-api.herokuapp.com/products'. Data yang sudah didapatkan lalu di looping pada komponen CardProduct untuk menampilkan image dan title pada halaman Shop. Kemudian pada CardProduct terdapat *button* 'Add to Cart' yang mana berfungsi untuk menambahkan product kedalam database dengan url 'https://paragon-training-api.herokuapp.com/cart' menggunakan method POST pada *axios*.
8. Pada halaman Cart (komponen Cart) saya membaginya menjadi 1 komponen yaitu: CartDetail yang berfungsi untuk menampilkan order summary pada halaman Shop

dimana data summary didapatkan dari method GET di dalam database dengan url 'https://paragon-training-api.herokuapp.com/cart'. Halaman Cart akan ditampilkan apabila user meng*click* icon trolley pada bagian header. Dimana pada icon tersebut akan memperlihatkan berapa banyak product yang sudah dimasukkan kedalam trolley dengan cara *click* 'Add to Cart' pada halaman Shop. Jika user meng*click* tombol 'Go Back to Shopping' maka user akan kembali ke halaman Shop. Styling yang diterapkan pada halaman Cart dengan menggunakan library Bootstrap 'container' dan display: inline-block.