

LAMPIRAN

Lampiran 1. Surat Ketersediaan Pembimbing

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Muhammad Rizqi
NIM : 20090059
Program Studi : Teknik Informatika

Pihak Kedua

Nama : Dyah Apriliani, S.T., M.Kom
Status : Dosen / Praktisi
NIDN : 0614049002
Jabatan Fungsional : Lektor
Pangkat/Golongan : III d

Pada hari ini jum'at tanggal 15 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan **minimal bimbingan satu kali dalam seminggu**. Adapun waktu dan tempat pelaksanaan disepakati antar pihak. Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 15 Maret 2024

Pihak Pertama



Muhammad Rizqi

Pihak Kedua



Dyah Apriliani, S.T., M.Kom

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Apriliani, S.T., M.Kom
09.015.225

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Muhammad Rizqi
NIM : 20090059
Program Studi : Teknik Informatika

Pihak Kedua

Nama : Dairoh, M.Sc
Status : Dosen / Praktisi
NIDN : 0612108701
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata Tk I/IIIid

Pada hari ini Senin tanggal 25 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak. Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 25 Maret 2024

Pihak Pertama



Muhammad Rizqi

Pihak Kedua



Dairoh, M.Sc

Mengetahui
Ketua Program Studi Sajana Terapan Teknik Informatika



Dyah Apriyani, S.T., M.Kom
NIPY. 09.015.225

Lampiran 2. Surat Pernyataan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

1. Nama : Muhammad Rizqi
Kewarganegaraan : Indonesia
Alamat : Jl. Letjen S.Parman Pesanggrahan Brebes RT/RW 007/017, Kecamatan Brebes, Kabupaten Brebes, Provinsi Jawa Tengah
2. Nama : Dyah Apriliani, S.T., M.Kom
Kewarganegaraan : Indonesia
Alamat : Perumahan Mutiara Indah Blok C4, Jalan Nangka gang 2 RT/RW 002/002, Kelurahan Procot, Kecamatan Slawi, Kabupaten Tegal
2. Nama : Dairoh, M.Sc
Kewarganegaraan : Indonesia
Alamat : Jl. Garuda No 3 RT/RW 002/009 Randugunting Tegal Selatan Kota Tegal

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan :
Berupa : Program Komputer
Berjudul : Aplikasi Studiobook Untuk Pemesanan Jasa Foto Berbasis Website di Daerah Brebes
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
 - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan
4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami melanggar, maka saya / kami bersedia secara sukarela bahwa:
 - a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau

- b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
- c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, Juli 2024



(Muhammad Rizqi)

(Dyah Apiliani, S.T., M.Kom)

(Dairoh, M.Sc)

Lampiran 3. Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini:

1. Nama : Muhammad Rizqi
Kewarganegaraan : Indonesia
Alamat : Jl. Letjen S.Parman Pesanggrahan Brebes RT/RW 007/017, Kecamatan Brebes, Kabupaten Brebes, Provinsi Jawa Tengah
2. Nama : Dyah Apriliani, S.T., M.Kom
Kewarganegaraan : Indonesia
Alamat : Perumahan Mutiara Indah Blok C4, Jalan Nangka gang 2 RT/RW 002/002, Kelurahan Procot, Kecamatan Slawi, Kabupaten Tegal
2. Nama : Dairoh, M.Sc
Kewarganegaraan : Indonesia
Alamat : Jl. Garuda No 3 RT/RW 002/009 Randugunting Tegal Selatan Kota Tegal

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada:

Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M)
Alamat : Jl. Mataram No. 9 Pesurungan Lor Kota Tegal

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul “Aplikasi Studiobook Untuk Pemesanan Jasa Foto Berbasis Website di Daerah Brebes”. untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikian Surat pernyataan ini kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Pemegang Hak Cipta
Ketua P3M



(Dr. Aldi Budi Riyanta, S.Si., M.T)

Tegal, Juli 2024
Pencipta



(Muhammad Rizqi)



(Dyah Apriliani, S.T., M.Kom)



(Dairoh, M.Sc)



MANUAL BOOK

BUKU PETUNJUK PENGGUNAAN APLIKASI (USER MANUAL)

Aplikasi Studiobook Untuk Pemesanan Jasa Foto Berbasis Website di Daerah Brebes

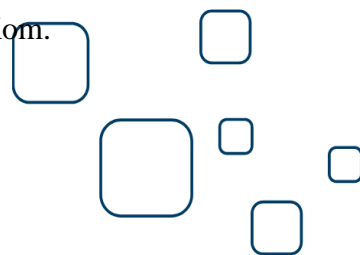


Oleh :

Muhammad rizqi

Dyah Apriliani, S.T., M.Kom.

Dairoh, M.Sc



1. PENDAHULUAN

1.1 Tujuan Pembuatan Dokumen

Dokumen user manual Aplikasi Studiobook untuk pemesanan Jasa Foto di Daerah Brebes dibuat dengan tujuan menggambarkan, menjelaskan, dan memberikan panduan penggunaan aplikasi Studiobook. Aplikasi ini dirancang untuk mempermudah pengguna dalam melakukan pemesanan studio foto dan jasa fotografi di daerah brebes. Panduan ini ditujukan untuk semua pengguna aplikasi, termasuk pelanggan dan pemilik studio foto, dengan tujuan untuk memastikan mereka dapat memanfaatkan semua fitur yang tersedia dengan maksimal.

1.2 Deskripsi Umum Sistem

Aplikasi Studiobook untuk Pemesanan Jasa Foto di Daerah Brebes merupakan sistem berbasis *website* yang dirancang untuk mempermudah proses pemesanan studio foto dan jasa fotografi. Aplikasi ini memungkinkan pengguna untuk mencari, membandingkan, dan memesan berbagai studio foto yang tersedia di Daerah Brebes. Tujuan dibuatnya sistem ini adalah untuk menyediakan platform yang efisien dan *user-friendly* bagi pelanggan dan pemilik studio, sehingga dapat meningkatkan kenyamanan dan kemudahan dalam mengatur sesi foto secara profesional.

1.3 Deskripsi Dokumen

Dokumen ini dibuat untuk memberikan panduan penggunaan “Aplikasi Studiobook Untuk Pemesanan Jasa Foto Berbasis *Website* di Daerah Brebes”. Dokumen ini berisikan informasi sebagai berikut:

1. BAB I.

Berisikan informasi umum yang merupakan bagian pendahuluan, yang meliputi tujuan pembuatan dokumen, deskripsi umum sistem serta deskripsi dokumen.

2. BAB II.

Berisikan perangkat yang dibutuhkan Aplikasi Studiobook untuk Pemesanan Jasa Foto Berbasis *Website* di Daerah Brebes, termasuk perangkat lunak dan perangkat keras yang diperlukan untuk menjalankan aplikasi ini dengan optimal.

3. BAB III.

Berisikan user manual untuk menggunakan Aplikasi Studiobook untuk Pemesanan Jasa Foto Berbasis *Website* di Daerah Brebes.

2. PERANGKAT YANG DIBUTUHKAN

2.1 Perangkat Lunak

Perangkat lunak yang digunakan adalah sebagai berikut:

1. *System operasi seperti Windows*
2. *Web browser (Google Chrome, GoogleMaps, dll)*
3. *Visual Studio Code*

2.2 Perangkat Keras

Perangkat keras (Hardware) yang digunakan sebagai berikut:

1. Laptop MSI Modern 14

2.3 Penggunaan Aplikasi

Pengguna aplikasi yang akan menggunakan aplikasi adalah sebagai berikut:

1. User : Pengguna ini memiliki fitur untuk melihat studio dan dapat mencari, membandingkan antara studio satu dan lainnya, User juga dapat melukan pemesanan studio dan pembayaran sesuai yang ditentukan.
2. Owner : Pengguna ini memiliki fitur yang dapat mengelola studio seperti menambahkan, mengedit dan menghapus, serta owner juga dapat mengelola pemesanan.

3. Admin : Admin memiliki fitur yang dapat mengelola user dan mengelola pemesanan.

3. MENU DAN CARA PENGGUNAAN

3.1 Struktur Menu

Adapun struktur halaman pada Aplikasi Studiobook Untuk Pemesanan Jasa Foto Berbasis Website di Daerah Brebes adalah sebagai berikut:

1. Halaman Utama Beranda

Pada halaman beranda akan muncul tampilan awal pada saat mengakses Aplikasi Studiobook adapun menu *navbar* yang dapat digunakan untuk sumber informasi penggunaan aplikasi dan menu tombol *login* dan daftar yang dapat digunakan untuk masuk sebagai user, owner maupun admin.

2. Halaman Navbar

Pada halaman navbar ada beberapa menu yang tersedia adalah

- a. Booking untuk memilih studio dan memesan studio
- b. Testimonial berisi testimonial user pengguna
- c. Riwayat pemesanan berisi pemesanan yang pernah dilakukan

3. Halaman Utama Admin

Pada halaman utama admin ada beberapa menu yang tersedia sebagai berikut:

- a. Menambahkan Studio
- b. Mengupdate Studio
- c. Menghapus Studio
- d. Logout

3.2 Pengguna

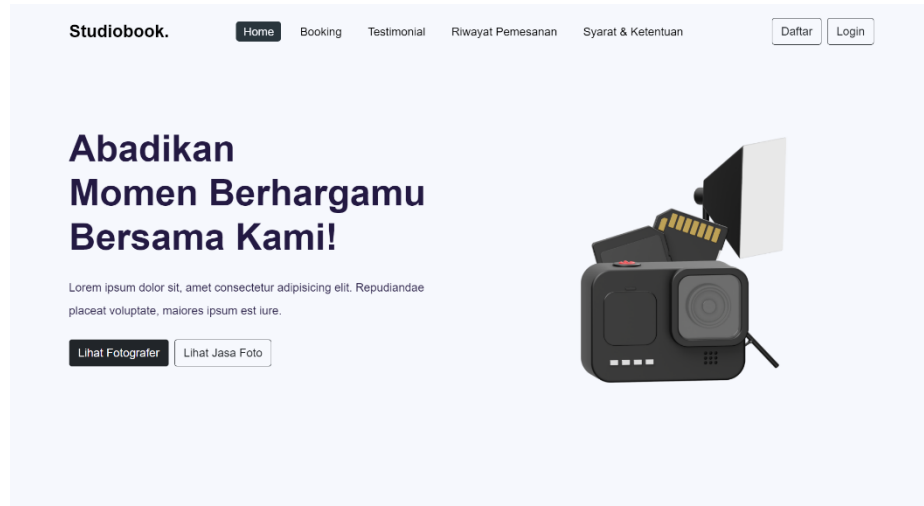
Pada bagian ini akan dijelaskan mengenai menu-menu yang tersedia disistem sebagai berikut:

3.2.1 Cara membuka Sistem

Untuk menjalankan *website* Studiobook adalah sebagai berikut:

1. Bukalah *website* sistem melalui *web browser* yang tersedia seperti (*Goggle Chrome* dll)

2. Tekan tombol *enter* pada *keyboard*
3. Setelah itu, akan muncul tampilan utama *website* sistem. Pada halaman terdapat menu di *navbar*. Adapun tampilan utama dapat dilihat pada gambar 1.



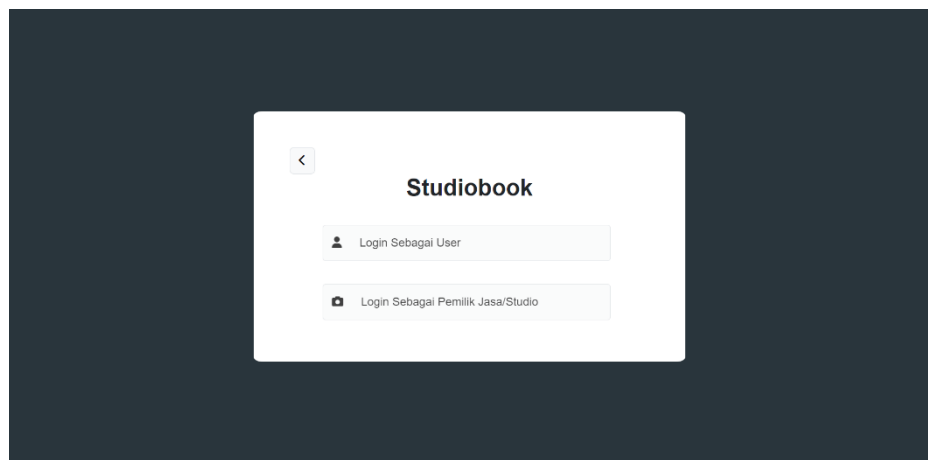
Gambar 1 Halaman Header

3.2.2 Halaman Utama

Pada sistem ini terdapat halaman yang terdiri dari berbagai menu sebagai berikut:

1. Menu pemilihan *Login*

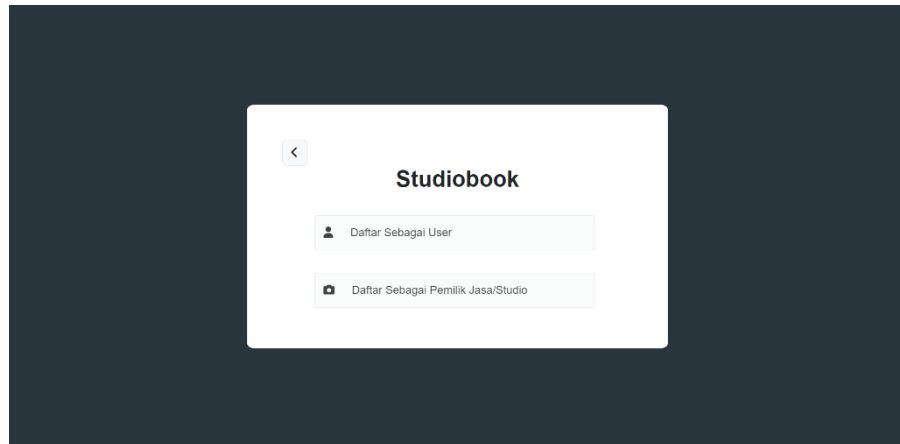
Menu pemilihan login berisi 2 form terdapat login sebagai user dan login sebagai owner yang digunakan untuk masuk sebagai user, owner dan admin. Adapun tampilan halaman login dapat dilihat pada gambar.



Gambar 2 Halaman pemilihan login

2. Menu pemilihan daftar

Menu pemilihan daftar berisi 2 form terdapat daftar sebagai user dan daftar sebagai owner yang digunakan untuk masuk sebagai user, owner dan admin. Adapun tampilan halaman daftar dapat dilihat pada gambar.



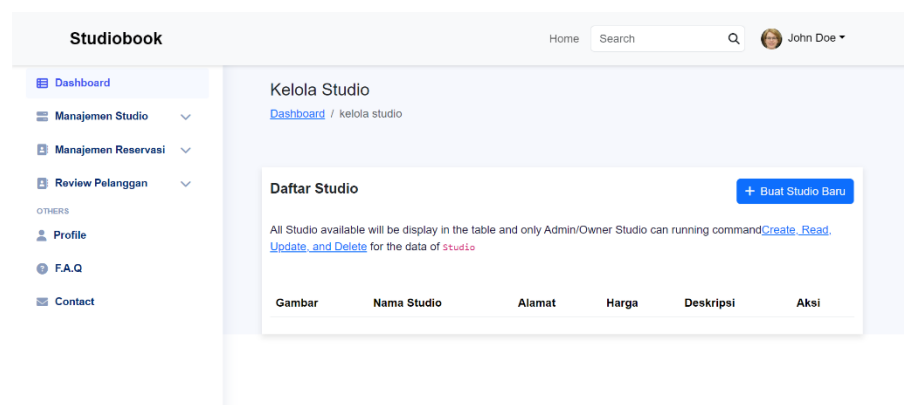
Gambar 3 Halaman pemilihan daftar

3.2.3 Halaman Admin/Owner

Pada sistem ini terdapat halaman admin dan owner yang terdiri dari berbagai menu fitur sebagai berikut:

1. Menu kelola Studio

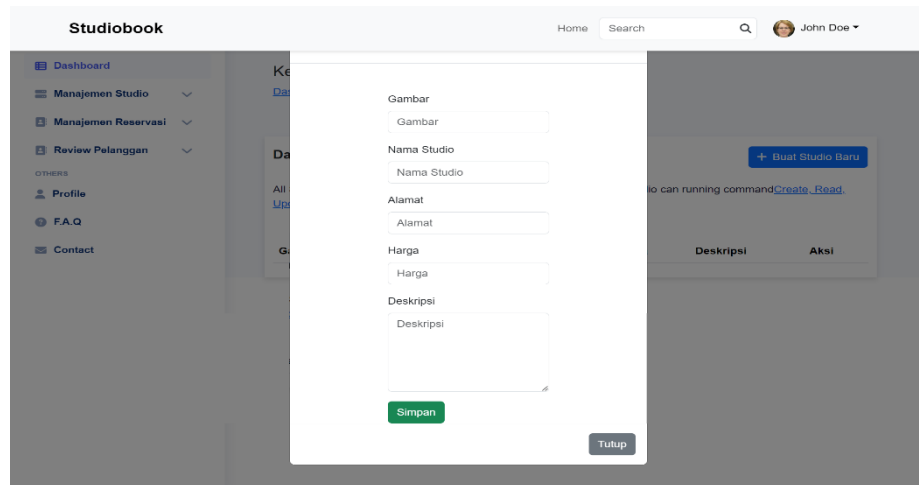
Fitur kelola studio untuk admin dan owner dapat menambahkan data studio dengan menekan buat studio baru. Menu tersebut dapat dilihat pada gambar.



Gambar 4 Halaman kelola studio

2. Menu tambah studio

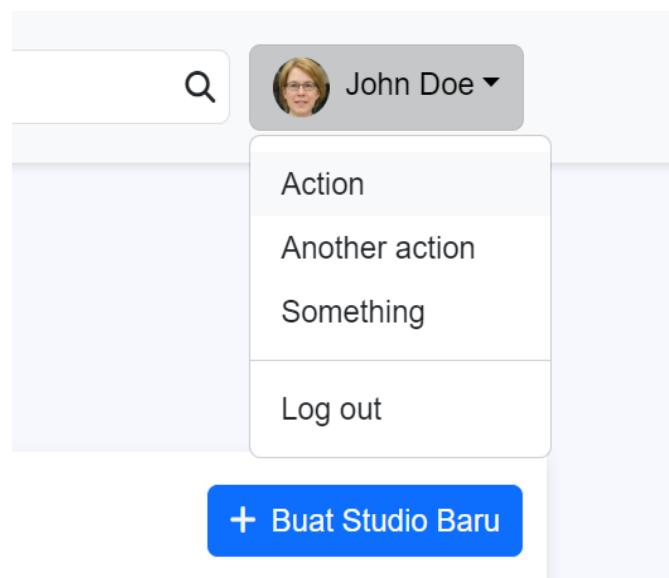
Fitur tambah studio bisa dilihat pada gambar.



Gambar 5 Halaman modal tambah studio

3. Menu Logout

Menu logout digunakan untuk keluar dari halaman admin dan jika diklik akan kembali kehalaman utama. Adapun tampilan halaman logout dapat dilihat pada gambar.



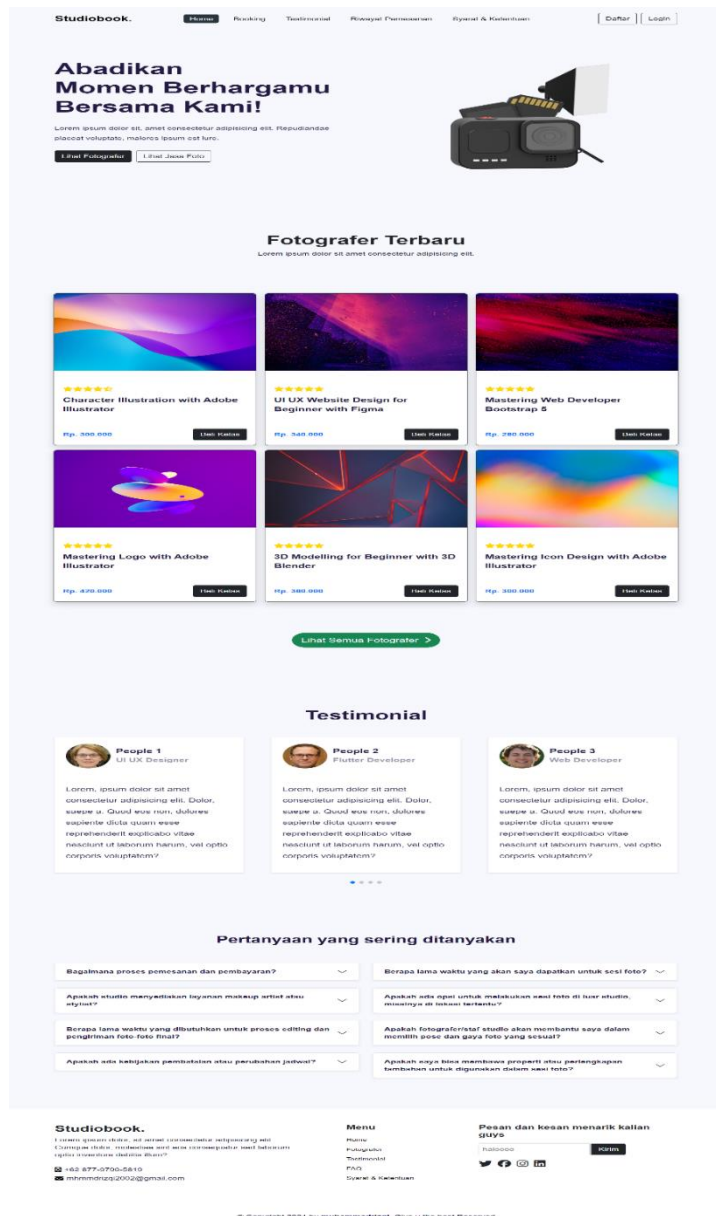
Gambar 6 Logout

3.2.4 Halaman User

Pada halaman user terdapat beberapa fitur yang bisa pengguna pakai yang sebagai berikut:

1. Dashboard

Dashboard adalah tampilan awal user ketika mengunjungi website pertama kali dilihat pada gambar.



Gambar 7 Dashboard

2. Fitur Detail Studio

User dapat memilih fitur ini dengan mngklik tombol booking untuk melihat daftar semua studio dilihat pada gambar.

Studiobook. Home Booking Testimonial Riwatay Pemesanan Syarat & Ketentuan [Daftar](#) [Login](#)

Detail

Lorem ipsum dolor sit amet.

Deskripsi Product

Lorem ipsum dolor sit amet consectetur adipiscing elit. Placeat expedita minima kure assumenda nulla, laboriosam neque et sed nisi sint dignissimos rem amet error voluptatem perspiciatis quia ipsa id ullam. Enim eveniet, exercitationem cupiditate accusantium fuga obcaecati laboriosam pariatur. Totam, nisi? Voluptate veritatis veniam voluptatem omnis quae voluptates, quas nihil quo nostrum inventore nam ex! Necessitatibus, et magnam facilis, a nam animi blanditis placeat ducimus voluptas tempore beatae eveniet alias? Ducimus, veniam soluta debitis ullam, temporibus reiciendis quasi doloremque quo aliquid, omnis excepturi. Rem, iste delectus, explicabo ea minima assumenda earum debitis maxime incidunt, dignissimos numquam sapiente nobis molestias neque.

- Free Parking
- 10+ Mbps/s
- AC
- Free Drinks
- Free Toilet

100% Professional

Mulai Pemesanan Studio

RP 20K per jam

Berapa lama ingin sewa studio?

Pilih tanggal

Pilih jam

Anda akan membayar sewa studio sebesar: **Rp20000 per 1 jam**

[Lanjut Pembayaran](#)

Studiobook.
Lorem ipsum dolor, sit amet consectetur adipiscing elit.
Cumque dolor, molestiae sint eos consequatur sed laborum opto inventore debitis illum?
☎ +62 877-9790-5819
✉ mhmmdrizq2002@gmail.com

Menu
Home
Fotografer
Testimonial
FAQ
Syarat & Ketentuan

Pesanan dan kesan menarik kalian guys

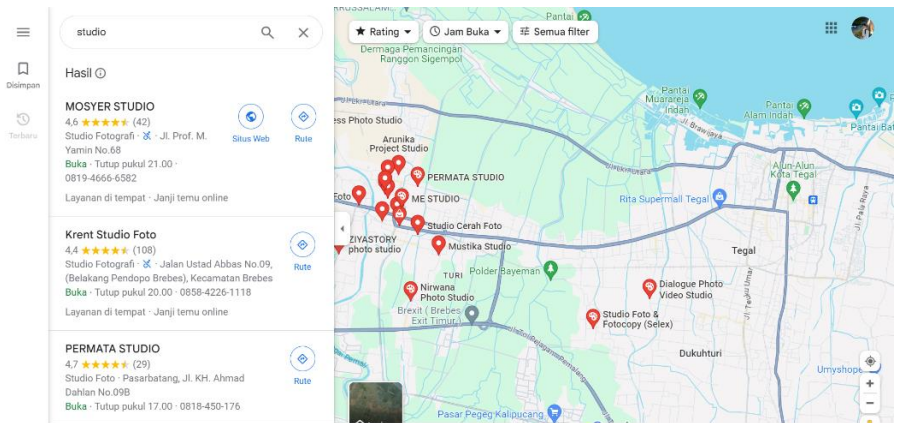
🐦 📷 📱 🌐

© Copyright 2024 by muhammadrizq. Give u the best Reserved

3. Fitur Studio Terdekat

Gambar 8 Detail studio

Fitur studio terdekat bisa diakses dengan klik tombol studio terdekat di awal chatbot, setelah di klik akan mengarah ke gmaps dan mencari apotek terdekat berdasarkan gps.



Gambar 9 Fitur studio terdekat

4. Fitur Pemesanan

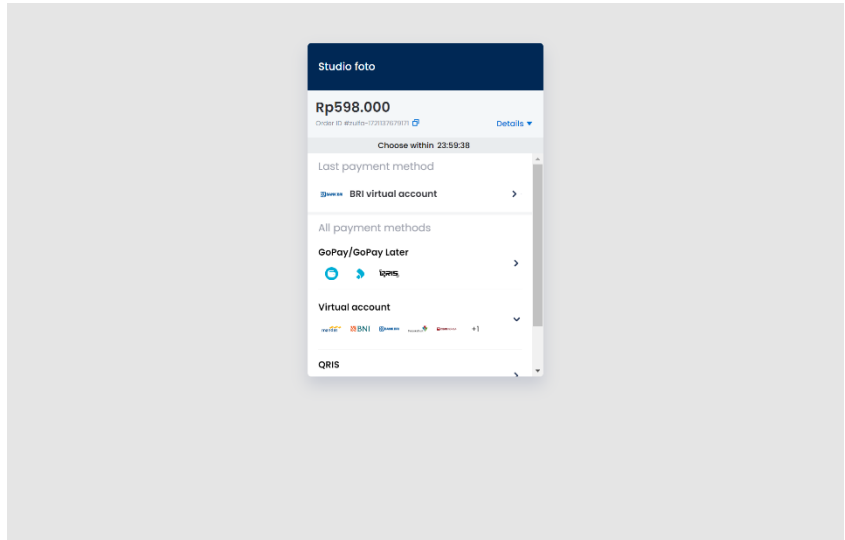
Fitur pemesanan yaitu user bisa memesan studio yang dipilih, berapa lama ingin memesan studio, memilih tanggal dan jam.



Gambar 10 Fitur pemesanan

5. Fitur Pembayaran

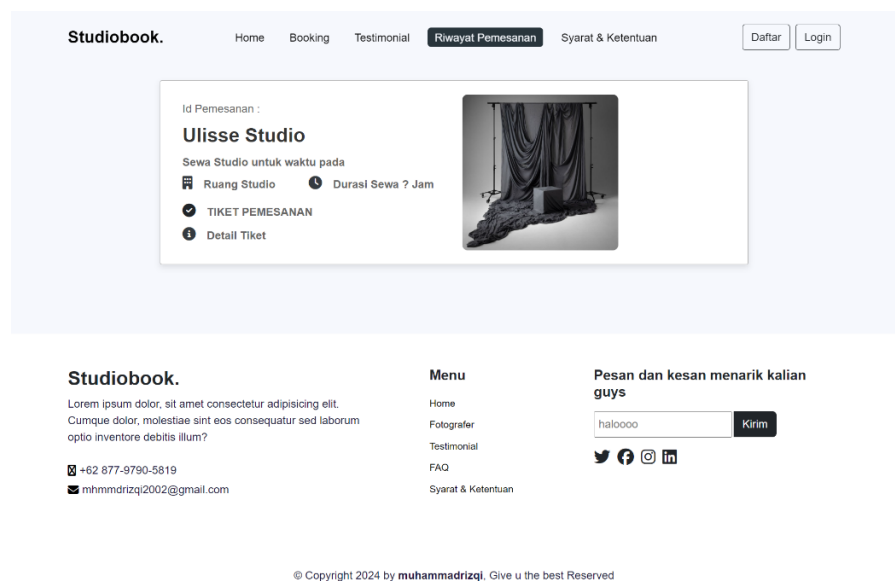
Fitur pembayaran memungkinkan pengguna untuk memilih metode pembayaran sesuai keinginan mereka.



Gambar 11 Fitur Pembayaran

6. Fitur Riwayat Pemesanan

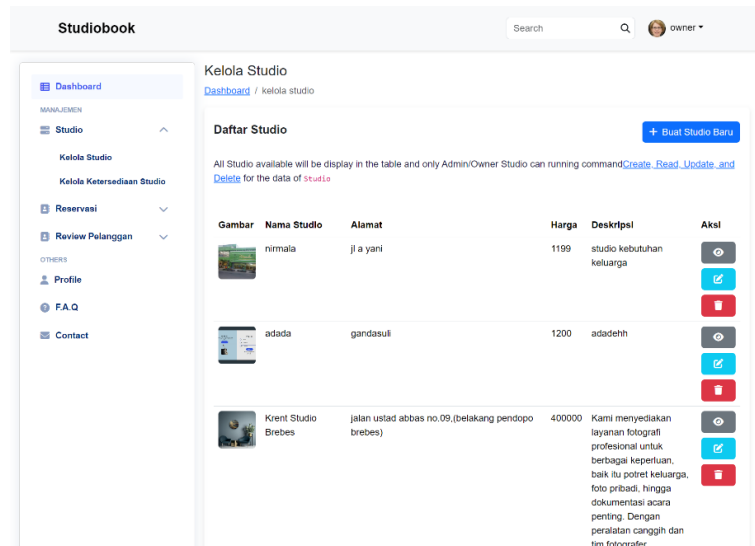
Fitur riwayat pemesanan adalah fungsi dalam aplikasi atau sistem yang memungkinkan pengguna untuk melihat daftar semua pemesanan yang pernah mereka lakukan di masa lalu. Dengan fitur ini, pengguna dapat melacak detail pemesanan mereka, seperti tanggal dan waktu pemesanan, jenis layanan atau produk yang dipesan, jumlah pembayaran, dan status pemesanan (misalnya, selesai, dibatalkan, atau masih dalam proses).



Gambar 12 riwayat pemesanan

7. Halaman Kelola Studio owner

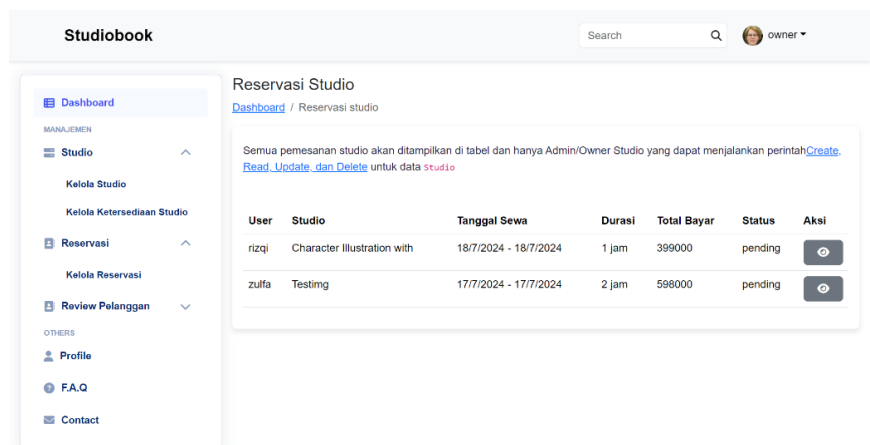
Halaman kelola studio owner adalah halaman yang berisi crud studio yang dilakukan oleh owner ketika sudah mendaftar menjadi owner.



Gambar 13 kelola studio

8. halaman Reservasi Owner

halaman ini berisikan pesanan dari user jika studio tersebut telah dipesan oleh user.



Gambar 14 reservasi



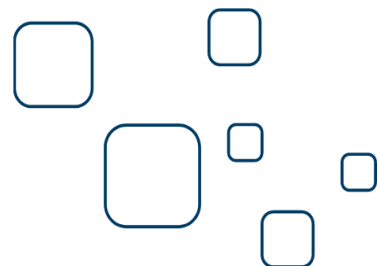
TECHNICAL BOOK

BUKU PETUNJUK TEKNIK PEMBUATAN APLIKASI

APLIKASI STUDIOBOOK UNTUK PEMESANAN
JASA FOTO BERBASIS WEBSITE DI DAERAH
BREBES



PENULIS :
Muhammad Rizqi



Bagian Front-End

1. Halaman Dashboard

Kodingan

```
1 import { Container, Row, Col } from 'react-bootstrap';
2 import PropTypes from 'prop-types';
3 import { Link, NavLink } from 'react-router-dom';
4 import { useAuth } from 'react-auth-kit';
5 import { useAlert } from 'react-alert';
6
7 // Import helper hooks
8 import { usePageTitle } from 'react-page-title';
9
10 // Import helper styles
11 import 'react-bootstrap';
12 import 'react-bootstrap';
13 // Import required modules
14 import { PageHeader } from 'react-page-header';
15 import { FooterPage } from 'react-page-footer';
16 import { HeaderComponent } from 'react-components/HeaderComponent';
17 import { FooterComponent } from 'react-components/FooterComponent';
18
19 // Main function
20 const Dashboard = () => {
21   // Get user info
22   const user = useAuth().getUser();
23
24   return (
25     <div className="dashboard">
26       <Header className="header" data-cs="12" data-kind="parent" style={{background-color: '#f8d7da'}}>
27         <Container>
28           <div style="display: flex; justify-content: space-between; align-items: center; padding: 5px 0">
29             <div style="font-size: 1.2em; font-weight: bold; color: #007bff">
30               Selamat datang!
31             </div>
32             <div style="font-size: 0.9em; color: #6c757d">
33               <span>Hi, <span style="font-weight: bold; color: #007bff">{user.name}</span>!</span>
34             </div>
35           </div>
36           <div style="display: flex; justify-content: space-between; align-items: center; padding: 5px 0">
37             <div style="font-size: 0.8em; color: #6c757d">
38               <span>Total Pengguna: <span style="font-weight: bold; color: #007bff">{totalUsers}</span></span>
39             </div>
40             <div style="font-size: 0.8em; color: #6c757d">
41               <span>Total Produk: <span style="font-weight: bold; color: #007bff">{totalProducts}</span></span>
42             </div>
43             <div style="font-size: 0.8em; color: #6c757d">
44               <span>Total Order: <span style="font-weight: bold; color: #007bff">{totalOrders}</span></span>
45             </div>
46           </div>
47         </Container>
48       </Header>
49
50       <div style="padding: 10px 0">
51         <Container>
52           <div style="display: flex; justify-content: space-between; align-items: flex-start; padding: 10px 0">
53             <div style="width: 45%; padding: 5px 0">
54               <div style="border: 1px solid #dee2e6; border-radius: 10px; padding: 10px; margin-bottom: 10px">
55                 <div style="display: flex; justify-content: space-between; align-items: center; padding: 5px 0">
56                   <div style="font-size: 0.9em; color: #6c757d">
57                     Total Pengguna
58                   </div>
59                   <div style="font-size: 1.2em; font-weight: bold; color: #007bff">
60                     {totalUsers}
61                   </div>
62                 </div>
63                 <div style="font-size: 0.8em; color: #6c757d; margin-top: 5px">
64                   <span>+ <span style="font-weight: bold; color: #007bff">{totalUsers}</span> pengguna baru</span>
65                 </div>
66               </div>
67               <div style="border: 1px solid #dee2e6; border-radius: 10px; padding: 10px; margin-bottom: 10px">
68                 <div style="display: flex; justify-content: space-between; align-items: center; padding: 5px 0">
69                   <div style="font-size: 0.9em; color: #6c757d">
70                     Total Produk
71                   </div>
72                   <div style="font-size: 1.2em; font-weight: bold; color: #007bff">
73                     {totalProducts}
74                   </div>
75                 </div>
76                 <div style="font-size: 0.8em; color: #6c757d; margin-top: 5px">
77                   <span>+ <span style="font-weight: bold; color: #007bff">{totalProducts}</span> produk baru</span>
78                 </div>
79               </div>
80               <div style="border: 1px solid #dee2e6; border-radius: 10px; padding: 10px">
81                 <div style="display: flex; justify-content: space-between; align-items: center; padding: 5px 0">
82                   <div style="font-size: 0.9em; color: #6c757d">
83                     Total Order
84                   </div>
85                   <div style="font-size: 1.2em; font-weight: bold; color: #007bff">
86                     {totalOrders}
87                   </div>
88                 </div>
89                 <div style="font-size: 0.8em; color: #6c757d; margin-top: 5px">
90                   <span>+ <span style="font-weight: bold; color: #007bff">{totalOrders}</span> order baru</span>
91                 </div>
92               </div>
93             </div>
94             <div style="width: 45%; padding: 5px 0">
95               <div style="border: 1px solid #dee2e6; border-radius: 10px; padding: 10px; margin-bottom: 10px">
96                 <div style="display: flex; justify-content: space-between; align-items: center; padding: 5px 0">
97                   <div style="font-size: 0.9em; color: #6c757d">
98                     Total Pendapatan
99                   </div>
100                  <div style="font-size: 1.2em; font-weight: bold; color: #007bff">
101                    {totalRevenue}
102                  </div>
103                </div>
104                <div style="font-size: 0.8em; color: #6c757d; margin-top: 5px">
105                  <span>+ <span style="font-weight: bold; color: #007bff">{totalRevenue}</span> pendapatan</span>
106                </div>
107              </div>
108              <div style="border: 1px solid #dee2e6; border-radius: 10px; padding: 10px">
109                <div style="display: flex; justify-content: space-between; align-items: center; padding: 5px 0">
110                  <div style="font-size: 0.9em; color: #6c757d">
111                    Total Stok
112                  </div>
113                  <div style="font-size: 1.2em; font-weight: bold; color: #007bff">
114                    {totalStock}
115                  </div>
116                </div>
117                <div style="font-size: 0.8em; color: #6c757d; margin-top: 5px">
118                  <span>+ <span style="font-weight: bold; color: #007bff">{totalStock}</span> stok</span>
119                </div>
120              </div>
121            </div>
122          </div>
123          <div style="display: flex; justify-content: space-between; align-items: flex-start; padding: 10px 0">
124            <div style="width: 45%; padding: 5px 0">
125              <div style="border: 1px solid #dee2e6; border-radius: 10px; padding: 10px; margin-bottom: 10px">
126                <div style="display: flex; justify-content: space-between; align-items: center; padding: 5px 0">
127                  <div style="font-size: 0.9em; color: #6c757d">
128                    Total Pengguna
129                  </div>
130                  <div style="font-size: 1.2em; font-weight: bold; color: #007bff">
131                    {totalUsers}
132                  </div>
133                </div>
134                <div style="font-size: 0.8em; color: #6c757d; margin-top: 5px">
135                  <span>+ <span style="font-weight: bold; color: #007bff">{totalUsers}</span> pengguna baru</span>
136                </div>
137              </div>
138              <div style="border: 1px solid #dee2e6; border-radius: 10px; padding: 10px; margin-bottom: 10px">
139                <div style="display: flex; justify-content: space-between; align-items: center; padding: 5px 0">
140                  <div style="font-size: 0.9em; color: #6c757d">
141                    Total Produk
142                  </div>
143                  <div style="font-size: 1.2em; font-weight: bold; color: #007bff">
144                    {totalProducts}
145                  </div>
146                </div>
147                <div style="font-size: 0.8em; color: #6c757d; margin-top: 5px">
148                  <span>+ <span style="font-weight: bold; color: #007bff">{totalProducts}</span> produk baru</span>
149                </div>
150              </div>
151              <div style="border: 1px solid #dee2e6; border-radius: 10px; padding: 10px">
152                <div style="display: flex; justify-content: space-between; align-items: center; padding: 5px 0">
153                  <div style="font-size: 0.9em; color: #6c757d">
154                    Total Order
155                  </div>
156                  <div style="font-size: 1.2em; font-weight: bold; color: #007bff">
157                    {totalOrders}
158                  </div>
159                </div>
160                <div style="font-size: 0.8em; color: #6c757d; margin-top: 5px">
161                  <span>+ <span style="font-weight: bold; color: #007bff">{totalOrders}</span> order baru</span>
162                </div>
163              </div>
164            </div>
165            <div style="width: 45%; padding: 5px 0">
166              <div style="border: 1px solid #dee2e6; border-radius: 10px; padding: 10px; margin-bottom: 10px">
167                <div style="display: flex; justify-content: space-between; align-items: center; padding: 5px 0">
168                  <div style="font-size: 0.9em; color: #6c757d">
169                    Total Pendapatan
170                  </div>
171                  <div style="font-size: 1.2em; font-weight: bold; color: #007bff">
172                    {totalRevenue}
173                  </div>
174                </div>
175                <div style="font-size: 0.8em; color: #6c757d; margin-top: 5px">
176                  <span>+ <span style="font-weight: bold; color: #007bff">{totalRevenue}</span> pendapatan</span>
177                </div>
178              </div>
179              <div style="border: 1px solid #dee2e6; border-radius: 10px; padding: 10px">
180                <div style="display: flex; justify-content: space-between; align-items: center; padding: 5px 0">
181                  <div style="font-size: 0.9em; color: #6c757d">
182                    Total Stok
183                  </div>
184                  <div style="font-size: 1.2em; font-weight: bold; color: #007bff">
185                    {totalStock}
186                  </div>
187                </div>
188                <div style="font-size: 0.8em; color: #6c757d; margin-top: 5px">
189                  <span>+ <span style="font-weight: bold; color: #007bff">{totalStock}</span> stok</span>
190                </div>
191              </div>
192            </div>
193          </div>
194        </Container>
195      </div>
196
197      <FooterComponent />
198    </div>
199  );
200};
201
202export default Dashboard;
```

Penjelasan

Kode ini adalah sebuah komponen React untuk halaman utama dari sebuah situs web. Ini mengimpor berbagai modul dan komponen yang diperlukan, termasuk React Bootstrap untuk tata letak, Swiper untuk membuat karusel, dan komponen khusus seperti FaqComponent, NavbarComponent, dan FooterComponent.

- **Bagian Header:** Berisi pesan sambutan dengan tombol tindakan yang mengarahkan ke halaman fotografer dan jasa foto.
- **Bagian Fotografer Terbaru:** Menampilkan fotografer terbaru dengan menggunakan fungsi map yang mengiterasi data studio terbaru untuk menampilkan setiap fotografer dengan detail seperti gambar, bintang penilaian, judul, harga, dan tombol beli.
- **Bagian Testimonial:** Menampilkan testimonial dalam bentuk karusel menggunakan Swiper. Setiap testimonial berisi gambar, nama, keterampilan, dan deskripsi dari pelanggan.
- **Bagian FAQ:** Menampilkan komponen FAQ (Frequently Asked Questions) untuk menjawab pertanyaan umum dari pengguna.


Output

Studiobook. [Home](#) [About Us](#) [Testimoni](#) [Kerjasama](#) [Syarat & Ketentuan](#) [Daftar](#) [Login](#)

Abadikan Momen Berhargamu Bersama Kami!


Lorem ipsum dolor sit amet consectetur adipiscing elit. Tempus blandit accumsan, maecenas placerat elit.

[Lihat Foto](#) [Lihat Jasa](#)




Fotografer Terbaru

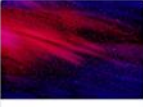
Lorem ipsum dolor sit amet consectetur adipiscing elit.




★★★★★
Character Illustration with Adobe Illustrator
Rp. 300.000 [Lihat Detail](#)




★★★★★
UI UX Website Design for Beginner with Figma
Rp. 240.000 [Lihat Detail](#)




★★★★★
Mastering Web Developer Bootstrap 5
Rp. 390.000 [Lihat Detail](#)



★★★★★
Mastering Logo with Adobe Illustrator
Rp. 420.000 [Lihat Detail](#)




★★★★★
3D Modeling for Beginner with 3D Blender
Rp. 380.000 [Lihat Detail](#)



★★★★★
Mastering Icon Design with Adobe Illustrator
Rp. 360.000 [Lihat Detail](#)


[Lihat Semua Fotografer >](#)

Testimonial




People 1
UI UX Designer

Lorem ipsum dolor sit amet consectetur adipiscing elit. Donec tempus a Quis egestas non, donec accumsan elit. Quis egestas non, donec accumsan elit. Quis egestas non, donec accumsan elit.



People 2
Flutter Developer

Lorem ipsum dolor sit amet consectetur adipiscing elit. Donec tempus a Quis egestas non, donec accumsan elit. Quis egestas non, donec accumsan elit. Quis egestas non, donec accumsan elit.



People 3
Web Developer

Lorem ipsum dolor sit amet consectetur adipiscing elit. Donec tempus a Quis egestas non, donec accumsan elit. Quis egestas non, donec accumsan elit. Quis egestas non, donec accumsan elit.

.....

Pertanyaan yang sering ditanyakan

Bagaimana proses pemesanan dan pembayaran?	Berapa lama waktu yang akan saya dapatkan untuk sesi foto?
Apakah studio menyediakan layanan makeup artist atau stylist?	Apakah ada opsi untuk melakukan sesi foto di luar studio, misalnya di lokasi tertentu?
Berapa lama waktu yang dibutuhkan untuk proses editing dan pengiriman foto-foto final?	Apakah fotografer studio akan membantu saya dalam memilih pose dan gaya foto yang sesuai?
Apakah ada kebijakan pembatalan atau perubahan jadwal?	Apakah saya bisa membawa properti atau perlengkapan tambahan untuk digunakan dalam sesi foto?

Studiobook.

Lorem ipsum dolor sit amet consectetur adipiscing elit. Donec tempus a Quis egestas non, donec accumsan elit. Quis egestas non, donec accumsan elit.

03 5774000000
#studiobookid

Menu

Home
Tentang
Kerjasama
Syarat & Ketentuan

Pesan dan kesan menarik kalian

03 5774000000

[Twitter](#) [Facebook](#) [Instagram](#)

© Copyright 2024 by studiobookid. All rights reserved.

D-18

Halaman pemilihan login dan register

Kodingan pemilihan login

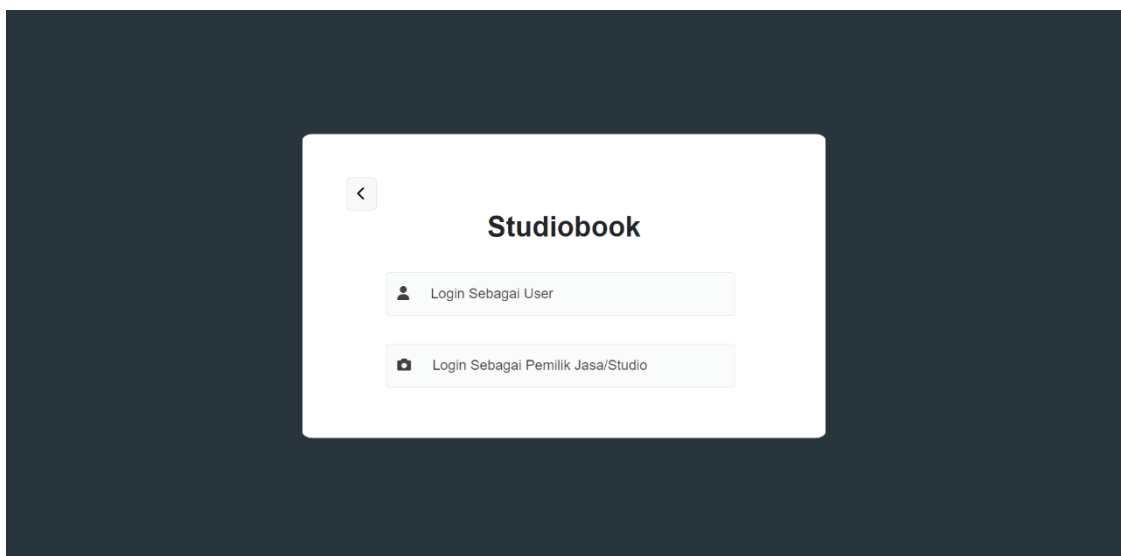
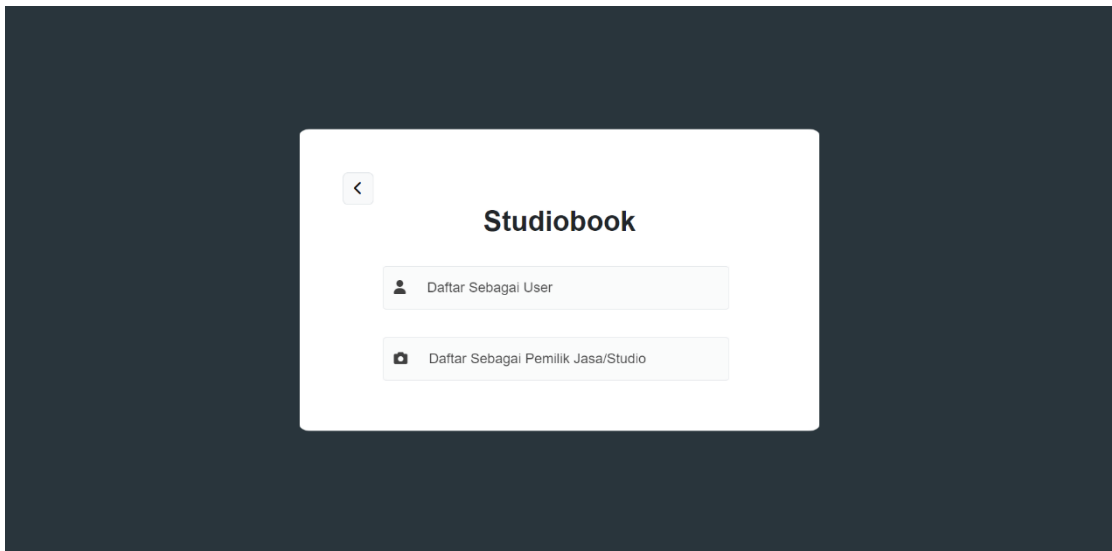
```
1 import React from "react";
2 import { Container, Row, Col, Button } from "react-bootstrap";
3 import { useNavigate } from "react-router-dom";
4
5 const PemilihanLogin = () => {
6   let navigate = useNavigate();
7   return (
8     <div className="body-pemilihan">
9       <div className="pemilihan">
10        <Container className="container-pemilihan">
11          <Button className="btn-ajadeh btn btn-light border" onClick={() => navigate("/")}>
12            <i className="fa fa-chevron-left"></i>
13          </Button>
14          <Row>
15            <Col>
16              <h2 className="text-center mb-2 fw-bold">Studiobook</h2>
17            </Col>
18          </Row>
19          <Row>
20            <Col className="text-center">
21              <Button variant="primary" type="submit" className="btn-pemilihan btn btn-light opacity-75 rounded-1 me-2 mb-xs-0 mb-2 mt-4 border" onClick={() => navigate("/login")}>
22                <i className="fa fa-user fw-bold"></i> Login Sebagai User
23              </Button>
24              <Button variant="primary" type="submit" className="btn-pemilihan btn btn-light opacity-75 rounded-1 me-2 mb-xs-0 mb-2 mt-4 border" onClick={() => navigate("/loginowner")}>
25                <i className="fa fa-camera fw-bold"></i> Login Sebagai Pemilik Jasa/Studio
26              </Button>
27            </Col>
28          </Row>
29        </Container>
30      </div>
31    </div>
32  );
33 };
34
35 export default PemilihanLogin;
```

Kodingan pemilihan register

```
1 import React from "react";
2 import { Container, Row, Col, Button } from "react-bootstrap";
3 import { useNavigate } from "react-router-dom";
4
5 const PemilihanDaftar = () => {
6   let navigate = useNavigate();
7   return (
8     <div className="body-pemilihan">
9       <div className="pemilihan">
10        <Container className="container-pemilihan">
11          <Button className="btn-ajadeh btn btn-light border" onClick={() => navigate("/")}>
12            <i className="fa fa-chevron-left"></i>
13          </Button>
14          <Row>
15            <Col>
16              <h2 className="text-center mb-2 fw-bold">Studiobook</h2>
17            </Col>
18          </Row>
19          <Row>
20            <Col className="text-center">
21              <Button variant="primary" type="submit" className="btn-pemilihan btn btn-light opacity-75 rounded-1 me-2 mb-xs-0 mb-2 mt-4 border" onClick={() => navigate("/register")}>
22                <i className="fa fa-user fw-bold"></i> Daftar Sebagai User
23              </Button>
24              <Button variant="primary" type="submit" className="btn-pemilihan btn btn-light opacity-75 rounded-1 me-2 mb-xs-0 mb-2 mt-4 border" onClick={() => navigate("/registerowner")}>
25                <i className="fa fa-camera fw-bold"></i> Daftar Sebagai Pemilik Jasa/Studio
26              </Button>
27            </Col>
28          </Row>
29        </Container>
30      </div>
31    </div>
32  );
33 };
34
35 export default PemilihanDaftar;
```


Kode diatas adalah sebuah komponen React yang bernama PemilihanLogin dan PemilihanDaftar. Komponen ini digunakan untuk menampilkan halaman pilihan login bagi pengguna

Output



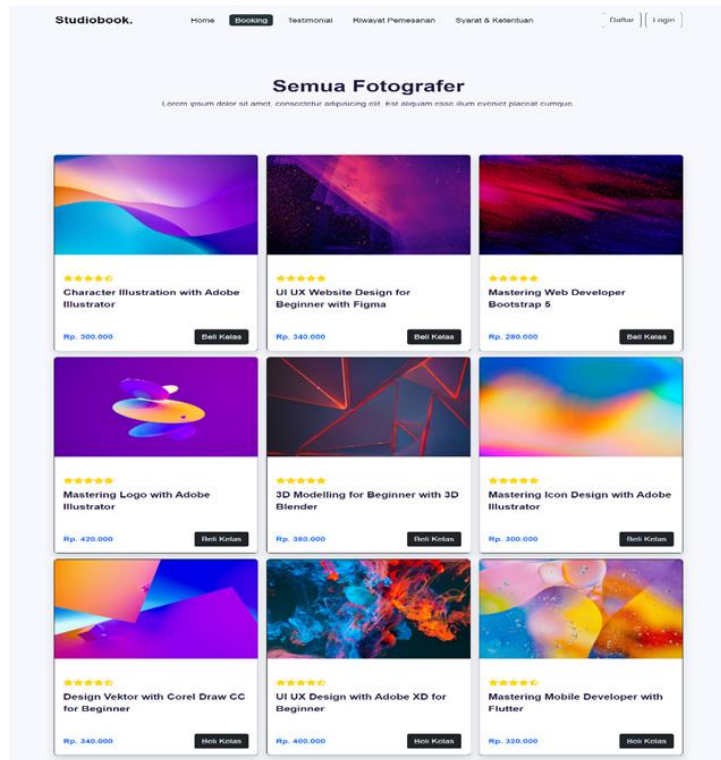
2. Halaman booking

Kodingan

```
1 import React from "react";
2 import { Container, Row, Col } from "react-bootstrap";
3 import { semuaKelas } from "../data/index";
4 import FaqComponent from "../components/FaqComponent";
5
6 const FotograferPage = () => {
7   return (
8     <div className="studios-page">
9       <div className="studios min-vh-100">
10        <Container>
11          <Row>
12            <Col>
13              <h1 className="text-center fu-bold animate__animated animate__fadeInUp animate__delay-1s">Semua fotografer</h1>
14              <p className="text-center animate__animated animate__fadeInUp animate__delay-1s">Iorem ipsum dolor sit amet, consectetur adipiscing elit. Est aliquam esse illum eveniet placeat cumque.</p>
15            </Col>
16          </Row>
17          <Row>
18            {semuaKelas.map((kelas) => {
19              return (
20                <Col key={kelas.id} className="border border-dark shadow rounded" data-aos="fadeUp" data-aos-duration="1000" data-aos-delay={kelas.delay}>
21                  <img src={kelas.image} alt="unsplash.com" className="w-100 mb-5 rounded-top" />
22                  <div className="star mb-2 px-3">
23                    <i className={kelas.star1}></i>
24                    <i className={kelas.star2}></i>
25                    <i className={kelas.star3}></i>
26                    <i className={kelas.star4}></i>
27                    <i className={kelas.star5}></i>
28                  </div>
29                  <h5 className="mb-5 px-3">{kelas.title}</h5>
30                  <div className="ket d-flex justify-content-between align-items-center px-3 pb-3">
31                    <p className="m-0 text-primary fu-bold">{kelas.price}</p>
32                    <button className="btn btn-dark rounded-1">{kelas.buy}</button>
33                  </div>
34                </Col>
35              );
36            });
37          </Row>
38        </Container>
39      </div>
40      <FaqComponent />
41    </div>
42  );
43 };
44
45 export default FotograferPage;
46
47
```

Kode ini adalah sebuah komponen React yang bernama FotograferPage. Komponen ini digunakan untuk menampilkan halaman yang berisi daftar semua fotografer

Output

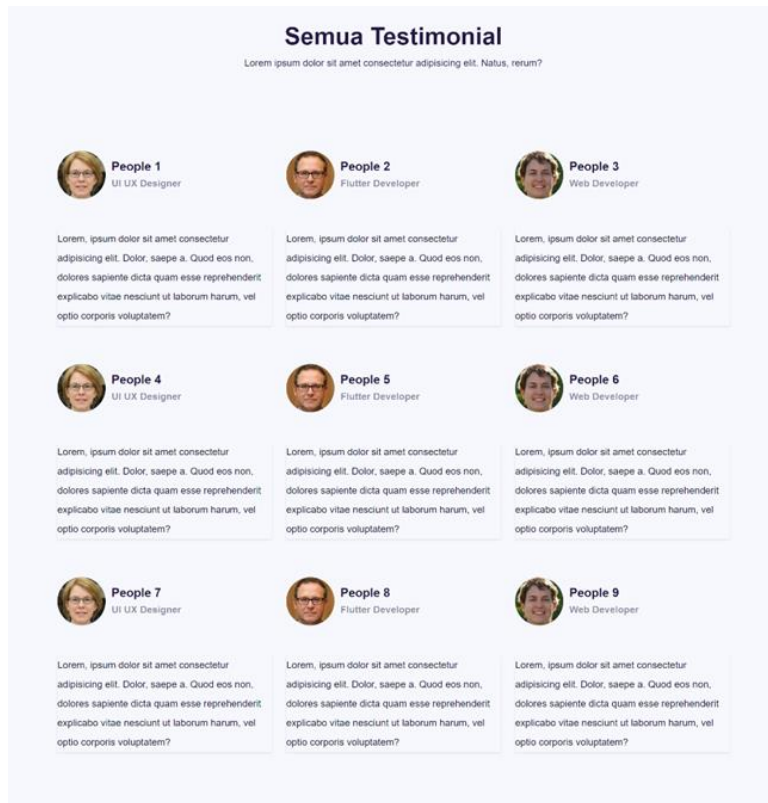


3. Halaman testimonial Kodingan

```
1 import React from "react";
2 import { Container, Row, Col } from "react-bootstrap";
3 import { testimonial } from "../data/index";
4
5 import FaqComponent from "../components/FaqComponent";
6
7 const TestimonialPage = () => {
8   return (
9     <div className="testimonial-page">
10      <div className="testimonial">
11        <Container>
12          <Row className="mb-5">
13            <Col>
14              <h1 className="text-center fw-bold animate__animated animate__fadeInUp animate__delay-1s">Semua Testimonial</h1>
15              <p className="text-center animate__animated animate__fadeInUp animate__delay-1s">Lorem ipsum dolor sit amet consectetur adipiscing elit. Natus, rerum</p>
16            </Col>
17          </Row>
18          <Row className="row-cols-lg-3 row-cols-1">
19            {testimonial.map((data) => {
20              return (
21                <Col key={data.id} className="mb-5">
22                  <div className="people">
23                    <img src={data.image} alt="" />
24                    <div>
25                      <h5 className="mb-1">{data.name}</h5>
26                      <p className="m-0 fw-bold">{data.skill}</p>
27                    </div>
28                  </div>
29                  <p className="desc shadow-sm">{data.desc}</p>
30                </Col>
31              );
32            })}
33          </Row>
34        </Container>
35      </div>
36      <FaqComponent />
37    </div>
38  );
39 };
40
41 export default TestimonialPage;
42
```

Kodingan di atas digunakan untuk menampilkan semua testimonial

Output



4. Halaman riwayat pemesanan

Kodingan

```
1 import React from "react";
2 import { Container, Row, Col } from "react-bootstrap";
3
4 const RiwayatSewaStudio = () => {
5   return (
6     <div className="riwayat-page">
7       <Container className="ticket-container">
8         <Row className="ticket-row d-flex justify-content-between align-items-center">
9           <Col className="ticket-details">
10            <p>Id Pemesanan :</p>
11            <h3 className="fw-bold">Ulisse Studio</h3>
12            <p className="fw-semibold mb-2">Sewa Studio untuk waktu pada </p>
13            <div className="d-flex gap-5">
14              <span className="d-flex gap-3 text-dark">
15                <i className="fa fa-building"></i>
16                <p className="fw-semibold m-0">Ruang Studio </p>
17              </span>
18              <span className="d-flex gap-3 text-dark">
19                <i className="fa fa-clock"></i>
20                <p className="fw-semibold m-0">Durasi Sewa ? Jam</p>
21              </span>
22            </div>
23            <span className="d-flex gap-3 mt-3">
24              <i className="fa fa-check-circle"></i>
25              <p className="fw-semibold text-uppercase">TIKET PEMESANAN </p>
26            </span>
27            <a href="/detail-ticket/<%= reservation.id %>" className="d-flex gap-3 text-dark">
28              <i className="fa fa-info-circle"></i>
29              <p className="fw-semibold m-0">Detail Tiket</p>
30            </a>
31          </Col>
32          <Col>
33            
34          </Col>
35        </Row>
36      </Container>
37    </div>
38  );
39 };
40
41 export default RiwayatSewaStudio;
42
```

Kodingan diatas digunakan untuk menampilkan semua riwayat pemesanan yang telah dilakukan sebelumnya.



Output


Studiobook. [Home](#) [Booking](#) [Testimonial](#) [Riwayat Pemesanan](#) [Syarat & Ketentuan](#) [Daftar](#) [Login](#)


Id Pemesanan :


Ulisse Studio

Sewa Studio untuk waktu pada

 Ruang Studio  Durasi Sewa 7 Jam

 **TIKET PEMESANAN**

 Detail Tiket



Studiobook.

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Cumque dolor, molestiae sint eos consequatur sed laborum optio inventore debitis illum?

+62 877-9790-5819
mhmmdrizq2002@gmail.com

Menu

[Home](#)
[Fotografer](#)
[Testimonial](#)
[FAQ](#)
[Syarat & Ketentuan](#)

Pesan dan kesan menarik kalian guys

[Kirim](#)




Output

Studiobook. [Home](#) [Booking](#) [Testimonial](#) [Riwayat Pemesanan](#) [Syarat & Ketentuan](#) [Daftar](#) [Login](#)

Detail

Lorem ipsum dolor sit amet.



Deskripsi Product

Lorem ipsum dolor sit amet consectetur adipisicing elit. Placeat expedita minima iure assumenda nulla, laboriosam neque et sed nisi sint dignissimos rem amet error voluptatem perspiciatis quasi ipsa id ullam. Enim eveniet, exercitationem cupiditate accusantium fuga obcaecati laboriosam pariatur. Totam, nisi? Voluptate veritatis veniam voluptatem omnis quae voluptates, quas nihil quo nostrum inventore nam ex! Necessitatibus, et magnam facilis, a nam animi blanditiis placeat ducimus voluptas tempore beatae eveniet alias? Ducimus, veniam soluta debitis ullam, temporibus reiciendis quasi doloremque quo aliquid, omnis excepturi. Rem, iste delectus, explicabo ea minima assumenda earum debitis maxime incidunt, dignissimos numquam sapiente nobis molestias neque.

- Free Parking
- 10+ Mbps/s
- AC
- Free Drinks
- Free Toilet

- 100% Professional

Mulai Pemesanan Studio

RP 20K per jam

Berapa lama ingin sewa studio?

Pilih tanggal

Pilih jam

Anda akan membayar sewa studio sebesar: **Rp20000 per 1 jam**

[Lanjut Pembayaran](#)

Studiobook.

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Cumque dolor, molestiae sint eos consequatur sed laborum optio inventore debitis illum?

+62 877-9790-5819
mhmmrdizqi2002@gmail.com

Menu

- Home
- Fotografer
- Testimonial
- FAQ
- Syarat & Ketentuan

Pesan dan kesan menarik kalian guys

[Twitter](#) [Facebook](#) [Instagram](#) [LinkedIn](#)

6. Halaman pemesanan

```
1 // src/components/BookingForm.js
2 import React, { useState } from "react";
3
4 const BookingForm = () => {
5   const [duration, setDuration] = useState(1);
6   const [rentDate, setRentDate] = useState("");
7   const [rentTime, setRentTime] = useState("");
8   const user = true; // ganti dengan kondisi autentikasi pengguna Anda
9
10  const handleDurationChange = (event) => {
11    setDuration(event.target.value);
12  };
13
14  const handleRentDateChange = (event) => {
15    setRentDate(event.target.value);
16  };
17
18  const handleRentTimeChange = (event) => {
19    setRentTime(event.target.value);
20  };
21
22  const handleMinusClick = () => {
23    setDuration((prevDuration) => Math.max(1, prevDuration - 1));
24  };
25
26  const handlePlusClick = () => {
27    setDuration((prevDuration) => Math.min(4, prevDuration + 1));
28  };
29
30  const totalPrice = duration * 20000;
31
32  return (
33    <div className="col-lg-12 p-3">
34      <div className="box-payment box border border-secondary rounded-3">
35        <form>
36          <h4 className="fw-bold fs-5 text-center">Mulai Pemesanan Studio</h4>
37          <input type="hidden" value="" name="studioId" id="studioId" />
38          <h3 className="fw-bold text-success fs-2 text-center">
39            Rp 20K <span className="fw-light fs-3 text-secondary">per jam</span>
40          </h3>
41          <input type="hidden" value="" name="priceStudio" id="priceStudio" />
42          <input type="hidden" value="0" name="totalPayment" id="totalPayment" />
43
44          <p className="mt-4">Siapa yang ingin sewa studio?</p>
45          <div className="btn-package d-flex justify-content-center">
46            <button type="button" className="fw-bold btn btn-danger" id="btn-minus" onClick={handleMinusClick}>
47              -
48            </button>
49            <input className="bg-body-secondary" type="number" value={duration} min="1" max="4" onChange={handleDurationChange} name="duration" id="duration" required />
50            <button type="button" className="fw-bold btn btn-success" id="btn-plus" onClick={handlePlusClick}>
51              +
52            </button>
53          </div>
54
55          <p className="mt-4">Pilih tanggal</p>
56          <div className="btn-date d-flex">
57            <button className="btn btn-dark">
58              <svg width="24" height="24" viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg">
59                <rect width="11" height="11" x="11" y="11" stroke="white" strokeWidth="2" />
60                <rect width="11" height="11" x="11" y="11" stroke="white" strokeWidth="2" />
61                <rect width="11" height="11" x="11" y="11" stroke="white" strokeWidth="2" />
62                <rect width="11" height="11" x="11" y="11" stroke="white" strokeWidth="2" />
63                <rect width="11" height="11" x="11" y="11" stroke="white" strokeWidth="2" />
64                <rect width="11" height="11" x="11" y="11" stroke="white" strokeWidth="2" />
65              </svg>
66            </button>
67            <input className="bg-body-secondary" type="text" value={rentDate} onChange={handleRentDateChange} name="rentDate" id="rentDate" required />
68          </div>
69
70          <p className="mt-4">Pilih jam</p>
71          <div className="btn-time d-flex">
72            <select className="bg-body-secondary form-control" value={rentTime} onChange={handleRentTimeChange} name="rentTime" id="rentTime" required>
73              <option value="">Pilih jam</option>
74              {arrayFrom(24, 1, 1) => {
75                <option key={i} value={` ${i}:00`}>
76                  { `${i}:00` }
77                </option>
78              )}
79            </select>
80          </div>
81
82          <p className="mt-5">anda akan membayar sewa studio sebesar: { " " }
83          <span className="fw-semi-bold" id="showTotalPrice">
84            {totalPrice}
85          </span> { " " }
86          <span>{ " " }
87          <span className="fw-semi-bold" id="hourRentStudio">
88            {duration} jam
89          </span>
90        </p>
91
92        <div style="text-align: center;">
93          <button type="submit" className="btn btn-primary w-100 p-2" disabled>
94            Lanjut Pembayaran
95          </button>
96          <button type="submit" className="btn btn-primary w-100 p-2" disabled>
97            Lanjut Pembayaran
98          </button>
99          <p className="text-danger mt-2">Silahkan login dahulu sebelum Anda melakukan pemesanan Studio</p>
100        </div>
101      </div>
102    </div>
103  );
104}
105
106 export default BookingForm;
107
108
109
110
111
112
```

Kodingan

Komponen BookingForm dalam kode ini merupakan formulir pemesanan studio yang menggunakan React dan useState untuk mengelola durasi, tanggal, dan waktu sewa studio; menampilkan harga sewa berdasarkan durasi yang dipilih pengguna; serta menyediakan tombol untuk mengurangi atau menambah durasi sewa, memilih tanggal dan waktu sewa, serta melanjutkan pembayaran, dengan pengecekan apakah pengguna sudah login atau belum, dan semua elemen ini diatur dalam sebuah formulir yang dirender dengan elemen-elemen React Bootstrap untuk tata letaknya.

Output

Deskripsi Product

Lorem ipsum dolor sit amet consectetur adipiscing elit. Placeat expedita minima iure assumenda nulla, laboriosam neque et sed nisi sint dignissimos rem amet error voluptatem perspiciatis quasi ipsa id ullam. Enim eveniet, exercitationem cupiditate accusantium fuga obcaecati laboriosam pariatur. Totam, nisi? Voluptate veritatis veniam voluptatem omnis quae voluptates, quas nihil quo nostrum inventore nam ex! Necessitatibus, et magnam facilis, a nam animi blanditiis placeat ducimus voluptas tempore beatae eveniet alias? Ducimus, veniam soluta debitis ullam, temporibus reiciendis quasi doloremque quo aliquid, omnis excepturi. Rem, iste delectus, explicabo ea minima assumenda earum debitis maxime incidunt, dignissimos numquam sapiente nobis molestias neque.

- Free Parking
- 10+ Mbps/s
- AC
- Free Drinks
- Free Toilet
- 100% Professional

Mulai Pemesanan Studio

RP 20K per jam

Berapa lama ingin sewa studio?

- 1 +

Pilih tanggal

hh/bb/tttt

Pilih jam

Pilih jam

Anda akan membayar sewa studio sebesar: **Rp20000** per 1 jam

Lanjut Pembayaran

Bagian Back-End

1. Model

a. Model user

```
1 model User {
2   id          Int      @id @default(autoincrement())
3   name        String
4   email       String
5   password    String
6   role        Role     @default(USER)
7   instagram   String?
8   accessToken String?
9
10  // Relasi: One-to-Many
11  pemesanan   Pemesanan[]
12 }
13
14 enum Role {
15   USER
16   ADMIN
17   OWNER
18 }
```

b. Model studio

```
1 model Studio {
2   id          Int          @id @default(autoincrement())
3   image       Bytes       // Menggunakan Bytes untuk menyimpan gambar
4   name        String
5   address     String
6   price       Float
7   description String
8
9   createdAt   DateTime @default(now())
10  updatedAt   DateTime @updatedAt
11
12  // Relasi: One-to-Many
13  pemesanan   Pemesanan[]
14 }
```

c. Model pemesanan

```
1 model Pemesanan {
2   id          Int          @id @default(autoincrement())
3   totalHarga  Float
4   waktuPesan  DateTime @default(now())
5   tanggalMulai DateTime
6   tanggalSelesai DateTime
7   durasi      Int
8   studioId    Int
9   userId      Int
10
11  // Relasi: Many-to-One
12  studio       Studio @relation(fields: [studioId], references: [id])
13  user         User   @relation(fields: [userId], references: [id])
14
15  // Relasi: One-to-Many ke Pembayaran
16  pembayaran  Pembayaran[]
17
18  @@map("pemesanan")
19 }
```

d. Model pembayaran

```
1 model Pembayaran {
2   id          Int          @id @default(autoincrement())
3   jumlah     Float
4   metode     String
5   status     String
6   waktuBayar DateTime @default(now())
7
8   // Relasi: Many-to-One
9   pemesanan  Pemesanan @relation(fields: [pemesananId], references: [id])
10
11  // Field foreign key ke Pemesanan
12  pemesananId Int
13  @@map("pembayaran")
14 }
```

2. Controller

a. Controller user

```
1 import { PrismaClient } from "@prisma/client";
2 // import prisma from "../prismaClient.js"; // Import Prisma Client
3 import bcrypt from "bcrypt";
4 import jwt from "jsonwebtoken";
5 import express from "express";
6
7 const prisma = new PrismaClient();
8
9 export const getUsers = async (req, res) => {
10   try {
11     const users = await prisma.user.findMany({
12       select: {
13         id: true,
14         name: true,
15         email: true,
16         role: true,
17       },
18     });
19     res.json(users);
20   } catch (error) {
21     console.log(error);
22   }
23 };
24
25 export const Register = async (req, res) => {
26   const { name, email, password, confirmPassword, role, instagram } = req.body;
27
28   // Validasi password
29   if (password !== confirmPassword) {
30     return res.status(400).json({ msg: "Password dan Konfirmasi Password tidak cocok" });
31   }
32
33   // Validasi instagram untuk role OWNER
34   if (role === "OWNER" && !instagram) {
35     return res.status(400).json({ msg: "Instagram username diperlukan untuk pendaftaran owner" });
36   }
37
38   const salt = await bcrypt.genSalt();
39   const hashPassword = await bcrypt.hash(password, salt);
40
41   try {
42     await prisma.user.create({
43       data: {
44         name,
45         email,
46         password: hashPassword,
47         role: role || "USER", // Default role to USER if not provided
48         instagram,
49       },
50     });
51     res.json({ msg: "Register Berhasil" });
52   } catch (error) {
53     console.log(error);
54     res.status(500).json({ msg: "Terjadi kesalahan saat registrasi" });
55   }
56 };
57
58 export const Login = async (req, res) => {
59   try {
60     const user = await prisma.user.findFirst({
61       where: {
62         email: req.body.email,
63       },
64     });
65
66     if (!user) {
67       return res.status(404).json({ msg: "Email tidak ditemukan" });
68     }
69
70     const match = await bcrypt.compare(req.body.password, user.password);
71     if (!match) return res.status(400).json({ msg: "Password salah" });
72
73     const userId = user.id;
74     const name = user.name;
75     const email = user.email;
76     const role = user.role;
77     const accessToken = jwt.sign({ userId, name, email, role }, process.env.ACCESS_TOKEN_SECRET, {
78       expiresIn: "1h",
79     });
80
81     res.json({ "accessToken": accessToken });
82   } catch (error) {
83     res.status(500).json({ msg: error.message });
84   }
85 };
86
87 export const Logout = async (req, res) => {
88   const refreshToken = req.cookies.refreshToken;
89   if (!refreshToken) return res.sendStatus(204);
90   const user = await prisma.user.findUnique({
91     where: {
92       refreshToken: refreshToken,
93     },
94   });
95
96   if (!user) return res.sendStatus(204);
97
98   await prisma.user.update({
99     where: {
100       id: user.id,
101     },
102     data: {
103       refreshToken: null,
104     },
105   });
106   res.clearCookie("refreshToken");
107   return res.sendStatus(200);
108 };
109
110
```

Controller user yang berisi getUser, Register, Login, Logout.

Pada kode di atas, fungsi getUsers mengelola permintaan untuk mendapatkan daftar pengguna dengan menggunakan Prisma untuk berinteraksi dengan database. Fungsi ini mengambil data dari tabel pengguna dengan mengatur kolom-kolom yang diambil seperti id, name, email, dan role. Data pengguna yang berhasil diambil kemudian dikirimkan sebagai respons dalam format JSON. Jika ada kesalahan dalam proses pengambilan data, fungsi akan menangkap error yang terjadi dan mencatatnya di konsol untuk penanganan lebih lanjut.

fungsi Register mengelola permintaan pendaftaran pengguna baru. Pertama, ia mengambil data seperti name, email, password, confirmPassword, role, dan instagram dari req.body. Fungsi kemudian melakukan validasi antara password dan konfirmasi password untuk memastikan kesesuaian. Jika tidak sesuai, akan mengembalikan respons dengan status 400 dan pesan yang sesuai.

Selanjutnya, fungsi melakukan validasi tambahan untuk memeriksa apakah username Instagram diperlukan untuk peran 'OWNER'. Jika peran yang diberikan adalah 'OWNER' namun username Instagram tidak disediakan, maka fungsi akan mengembalikan respons dengan status 400 dan pesan error yang sesuai.

Setelah validasi berhasil, fungsi menggunakan bcrypt untuk meng-generate salt dan meng-hash password sebelum menyimpannya ke database menggunakan Prisma. Data pengguna baru termasuk nama, email, password yang di-hash, peran (dengan peran default 'USER' jika tidak diberikan), dan username Instagram jika disediakan.

Jika proses pendaftaran berhasil, fungsi akan mengirimkan respons JSON dengan pesan "Register Berhasil". Namun, jika terjadi kesalahan selama proses, fungsi akan menangkap error, mencatatnya di konsol, dan mengembalikan respons dengan status 500 dan pesan error yang sesuai kepada pengguna.

Fungsi Login digunakan untuk memproses permintaan login pengguna. Pertama, ia mencari pengguna berdasarkan email yang diberikan menggunakan Prisma. Jika email tidak ditemukan, respons akan mengindikasikan bahwa email tidak terdaftar. Jika pengguna ditemukan, fungsi membandingkan password yang dikirim dengan yang di-hash di database. Jika password tidak cocok, pengguna akan mendapatkan pesan bahwa password salah. Jika autentikasi berhasil, fungsi membuat token akses menggunakan JSON Web Token (JWT) yang berisi informasi penting pengguna seperti ID, nama, email, dan peran. Token ini dikirimkan kembali ke pengguna untuk digunakan dalam mengakses sumber daya terproteksi. Jika terjadi kesalahan selama proses, respons akan mengembalikan pesan kesalahan yang sesuai.

b. Controller studio

```
1 import { PrismaClient } from "@prisma/client";
2 const prisma = new PrismaClient();
3 import multer from "multer";
4 import path from "path";
5
6 // Konfigurasi penyimpanan gambar menggunakan multer
7 const storage = multer.memoryStorage();
8 const upload = multer({ storage: storage }).single("image");
9
10 export const getStudio = async (req, res) => {
11   try {
12     const studios = await prisma.studio.findMany();
13     // Encode image buffer to base64
14     const studiosWithBase64Images = studios.map(studio => ({
15       ...studio,
16       image: studio.image ? studio.image.toString('base64') : null,
17     }));
18     res.status(200).json(studiosWithBase64Images);
19   } catch (error) {
20     console.error(error);
21     res.status(500).json({ error: "Internal Server Error" });
22   }
23 };
24
25 export const getStudioById = async (req, res) => {
26   const { id } = req.params;
27   try {
28     const studio = await prisma.studio.findUnique({
29       where: { id: parseInt(id) },
30     });
31     if (!studio) {
32       return res.status(404).json({ error: "Studio not found" });
33     }
34     studio.image = studio.image.toString('base64');
35     res.status(200).json(studio);
36   } catch (error) {
37     console.error(error);
38     res.status(500).json({ error: "Internal Server Error" });
39   }
40 };
41
42 export const createStudio = async (req, res) => {
43   upload(req, res, async (err) => {
44     if (err instanceof multer.MulterError) {
45       return res.status(400).json({ error: "Error uploading file." });
46     } else if (err) {
47       return res.status(500).json({ error: "Internal Server Error" });
48     }
49
50     const { name, address, price, description } = req.body;
51     const image = req.file.buffer; // Ambil data gambar dari req.file.buffer
52
53     // Konversi price dari string ke float
54     const parsedPrice = parseFloat(price);
55
56     if (isNaN(parsedPrice)) {
57       return res.status(400).json({ error: "Invalid price value" });
58     }
59
60     try {
61       await prisma.studio.create({
62         data: {
63           name,
64           address,
65           price: parsedPrice,
66           description,
67           image, // Simpan data gambar ke dalam kolom image
68         },
69       });
70       res.status(201).json({ msg: "Studio Created" });
71     } catch (error) {
72       console.error(error);
73       res.status(500).json({ error: "Internal Server Error" });
74     }
75   });
76 };
77
78 export const updateStudio = async (req, res) => {
79   const { id } = req.params;
80   const studioData = req.body;
81
82   // Jika terdapat file gambar baru yang diunggah
83   if (req.file) {
84     studioData.image = req.file.buffer; // Ambil data gambar dari req.file.buffer
85   }
86
87   try {
88     await prisma.studio.update({
89       where: { id: parseInt(id) },
90       data: studioData,
91     });
92     res.status(200).json({ msg: "Studio Updated" });
93   } catch (error) {
94     console.error(error);
95     res.status(500).json({ error: "Internal Server Error" });
96   }
97 };
98
99 export const deleteStudio = async (req, res) => {
100   const { id } = req.params;
101   try {
102     await prisma.studio.delete({
103       where: { id: parseInt(id) },
104     });
105     res.status(200).json({ msg: "Studio Deleted" });
106   } catch (error) {
107     console.error(error);
108     res.status(500).json({ error: "Internal Server Error" });
109   }
110 };
111 };
```

Fungsi `createStudio` mengelola pembuatan studio baru dengan mengelola pengunggahan gambar menggunakan `multer`. Setelah mengunggah gambar tanpa masalah, fungsi mengambil data seperti `name`, `address`, `price`, dan `description` dari permintaan. Harga diubah dari string ke float dan diverifikasi untuk validitas. Data studio disimpan dalam database menggunakan `prisma.studio.create`. Jika berhasil, respons dengan status 201 dikirimkan untuk mengkonfirmasi studio berhasil dibuat; jika gagal, respons dengan status 500 dikirim untuk menangani kesalahan server.

c. Controller pemesanan

```
1 import { PrismaClient } from "@prisma/client";
2 const prisma = new PrismaClient();
3 import midtransClient from 'midtrans-client';
4
5 const snap = new midtransClient.Snap({
6   isProduction: true,
7   serverKey: process.env.MIDTRANS_SERVER_KEY,
8   clientKey: process.env.MIDTRANS_CLIENT_KEY
9 });
10
11 // Endpoint untuk membuat pemesanan dan pembayaran
12 export const createPemesanan = async (req, res) => {
13   const { totalHarga, tanggalMulai, tanggalSelesai, durasi, studioId, userId } = req.body.datapemesanan;
14   const { amount, customerDetails } = req.body;
15
16   try {
17     // Buat pemesanan
18     const pemesanan = await prisma.pemesanan.create({
19       data: {
20         totalHarga,
21         tanggalMulai,
22         tanggalSelesai,
23         durasi,
24         studioId,
25         userId,
26       },
27     });
28
29     // Buat pembayaran
30     const pembayaran = await prisma.pembayaran.create({
31       data: {
32         jumlah: totalHarga,
33         metode: 'Midtrans',
34         status: 'pending',
35         waktuBayar: new Date(),
36         pemesananId: pemesanan.id,
37       },
38     });
39
40     // Siapkan parameter untuk transaksi Midtrans
41     const parameter = {
42       transaction_details: {
43         order_id: `order-id-${pemesanan.id}`,
44         gross_amount: amount
45       },
46       customer_details: customerDetails
47     };
48
49     // Buat transaksi Midtrans
50     snap.createTransaction(parameter)
51       .then((transaction) => {
52         const transactionToken = transaction.token;
53         const transactionRedirectUrl = transaction.redirect_url;
54         res.json({ token: transactionToken, redirect_url: transactionRedirectUrl });
55       })
56       .catch((e) => {
57         console.log('Midtrans error:', e);
58         res.status(500).json({ message: 'Something went wrong with Midtrans' });
59       });
60
61   } catch (error) {
62     res.status(500).json({ error: error.message });
63   }
64 };
65
```

Fungsi `createTransaction` menggunakan Midtrans untuk melakukan pembayaran. Jika berhasil, respons akan berisi token

transaksi dan URL redirect. Jika terjadi kesalahan, pesan kesalahan akan ditangkap dan ditampilkan.

Fungsi `getPemesananById` mengambil data pemesanan berdasarkan ID yang diberikan. Data pemesanan mencakup detail studio dan informasi pembayaran terkait. Jika pemesanan tidak ditemukan, respons akan menunjukkan bahwa pemesanan tidak ada. Sebelum dikirimkan, gambar studio diubah menjadi format base64 untuk memastikan dapat ditampilkan dengan benar dalam respons.

```

1 export const getPesanan = async (req, res) => {
2   try {
3     const pemesanan = await prisma.pemesanan.findMany();
4     res.json(pemesanan);
5   } catch (error) {
6     res.status(500).json({ error: error.message });
7   }
8 };
9
10 export const getPesananById = async (req, res) => {
11   const { id } = req.params;
12   try {
13     const pemesanan = await prisma.pemesanan.findUnique({
14       where: { id: parseInt(id) },
15       include: {
16         studio: true,
17         pembayaran: true,
18       },
19     });
20     console.log(pemesanan)
21     if (!pemesanan) {
22       return res.status(404).json({ message: 'Pemesanan tidak ditemukan' });
23     }
24
25     // Mengonversi gambar studio ke base64
26     let imageBase64 = null;
27     if (pemesanan.studio && pemesanan.studio.image) {
28       imageBase64 = pemesanan.studio.image.toString('base64');
29     }
30
31     const detailPemesanan = {
32       ...pemesanan,
33       studio: {
34         ...pemesanan.studio,
35         image: imageBase64,
36       },
37     };
38
39     res.json(detailPemesanan);
40   } catch (error) {
41     res.status(500).json({ error: error.message });
42   }
43 };
44
45 export const updatePemesanan = async (req, res) => {
46   const { id } = req.params;
47   const { totalHarga, tanggalMulai, tanggalSelesai, durasi, studioId, userId } = req.body;
48   try {
49     const pemesanan = await prisma.pemesanan.update({
50       where: { id: parseInt(id) },
51       data: { totalHarga, tanggalMulai, tanggalSelesai, durasi, studioId, userId },
52     });
53     res.json(pemesanan);
54   } catch (error) {
55     res.status(500).json({ error: error.message });
56   }
57 };
58
59 export const deletePemesanan = async (req, res) => {
60   const { id } = req.params;
61   try {
62     await prisma.pemesanan.delete({ where: { id: parseInt(id) } });
63     res.json({ msg: "Pemesanan berhasil dihapus" });
64   } catch (error) {
65     res.status(500).json({ error: error.message });
66   }
67 };

```

```

1 // Get all bookings for a specific studio
2 export const getBookingsForStudio = async (req, res) => {
3   const { studioId } = req.params;
4   try {
5     const bookings = await prisma.pemesanan.findMany({
6       where: { studioId: parseInt(studioId) },
7     });
8     res.json(bookings);
9   } catch (error) {
10    res.status(500).json({ error: error.message });
11  }
12 };
13 // Fungsi untuk mendapatkan riwayat pemesanan berdasarkan userId
14 export const getPesananByUserId = async (req, res) => {
15   const { userId } = req.params;
16   console.log(userId)
17   try {
18     const pemesanan = await prisma.pemesanan.findMany({
19       where: { userId: parseInt(userId) }
20     });
21     // Mengonversi gambar studio ke base64
22     const pemesananwithstudio = await Promise.all(pemesanan.map(async (pesanan) => {
23       const studio = await prisma.studio.findUnique({
24         where: { id: pesanan.studioId },
25       });
26
27       let imageBase64 = null;
28       if (studio && studio.image) {
29         imageBase64 = studio.image.toString('base64');
30       }
31
32       return {
33         ...pesanan,
34         studio: {
35           ...studio,
36           image: imageBase64,
37         },
38       };
39     }));
40
41     res.json(pemesananwithstudio);
42   } catch (error) {
43     console.log(error)
44     res.status(500).json({ error: error.message });
45   }
46 };

```

d. Controller pembayaran

```
1 // Import model Pembayaran dan dependencies yang diperlukan
2 import { PrismaClient } from "@prisma/client";
3 import midtransClient from 'midtrans-client';
4
5 const prisma = new PrismaClient();
6
7 // Mendapatkan semua pembayaran
8 export const getPembayaran = async (req, res) => {
9   try {
10     const pembayaran = await prisma.pembayaran.findMany();
11     res.json(pembayaran);
12   } catch (error) {
13     res.status(500).json({ error: "Gagal mendapatkan pembayaran." });
14   }
15 };
16
17 // Mendapatkan pembayaran berdasarkan ID
18 export const getPembayaranById = async (req, res) => {
19   const { id } = req.params;
20   try {
21     const pembayaran = await prisma.pembayaran.findUnique({
22       where: { id: parseInt(id) },
23     });
24     if (!pembayaran) {
25       res.status(404).json({ error: "Pembayaran tidak ditemukan." });
26     } else {
27       res.json(pembayaran);
28     }
29   } catch (error) {
30     res.status(500).json({ error: "Gagal mendapatkan pembayaran." });
31   }
32 };
33
34 // Membuat pembayaran baru
35 export const createPembayaran = async (req, res) => {
36   const { jumlah, metode, status, waktuBayar, pemesananId } = req.body;
37   try {
38     const newPembayaran = await prisma.pembayaran.create({
39       data: {
40         jumlah,
41         metode,
42         status,
43         waktuBayar,
44         pemesananId,
45       },
46     });
47     res.status(201).json(newPembayaran);
48   } catch (error) {
49     res.status(500).json({ error: "Gagal membuat pembayaran baru." });
50   }
51 };
```

```

1 // Mengupdate pembayaran berdasarkan ID
2 export const updatePembayaran = async (req, res) => {
3   const { id } = req.params;
4   const { jumlah, metode, status, waktuBayar, pemesananId } = req.body;
5   try {
6     const updatedPembayaran = await prisma.pembayaran.update({
7       where: { id: parseInt(id) },
8       data: {
9         jumlah,
10        metode,
11        status,
12        waktuBayar,
13        pemesananId,
14      },
15    });
16    res.json(updatedPembayaran);
17  } catch (error) {
18    res.status(500).json({ error: "Gagal mengupdate pembayaran." });
19  }
20 };
21
22 // Menghapus pembayaran berdasarkan ID
23 export const deletePembayaran = async (req, res) => {
24   const { id } = req.params;
25   try {
26     await prisma.pembayaran.delete({
27       where: { id: parseInt(id) },
28     });
29     res.json({ message: "Pembayaran berhasil dihapus." });
30   } catch (error) {
31     res.status(500).json({ error: "Gagal menghapus pembayaran." });
32   }
33 };
34
35 const snap = new midtransClient.Snap({
36   isProduction: true,
37   serverKey: process.env.MIDTRANS_SERVER_KEY,
38   clientKey: process.env.MIDTRANS_CLIENT_KEY
39 });
40
41 // Endpoint untuk membuat transaksi
42 export const createPayment = async (req, res) => {
43   try {
44     const { amount, customerDetails } = req.body;
45
46     const parameter = {
47       transaction_details: {
48         order_id: `order-id-${Math.floor(Math.random() * 1000000)}`, // unique order_id
49         gross_amount: amount
50       },
51       customer_details: customerDetails
52     };
53
54     snap.createTransaction(parameter)
55       .then((transaction) => {
56         console.log('Midtrans transaction:', transaction);
57         const transactionToken = transaction.token;
58         const transactionRedirectUrl = transaction.redirect_url;
59         res.json({ token: transactionToken, redirect_url: transactionRedirectUrl });
60       })
61       .catch((e) => {
62         console.log('Midtrans error:', e);
63         res.status(500).json({ message: 'Something went wrong' });
64       });
65   } catch (error) {
66     console.error('Server error:', error);
67     res.status(500).json({ message: 'Internal Server Error' });
68   }
69 }

```



```

1 // Endpoint untuk menerima notifikasi dari Midtrans
2 export const getNotification = async (req, res) => {
3   const receivedJson = req.body;
4
5   try {
6     const statusResponse = await snap.transaction.notification(receivedJson);
7     const { order_id: orderId, transaction_status: transactionStatus, fraud_status: fraudStatus } = statusResponse;
8
9     console.log(`Transaction notification received. Order ID: ${orderId}. Transaction status: ${transactionStatus}. Fraud status: ${fraudStatus}.`);
10
11     let updateStatus = 'unknown';
12     if (transactionStatus === 'capture') {
13       if (fraudStatus === 'challenge') {
14         updateStatus = 'challenge';
15       } else if (fraudStatus === 'accept') {
16         updateStatus = 'success';
17       }
18     } else if (transactionStatus === 'settlement') {
19       updateStatus = 'success';
20     } else if (transactionStatus === 'cancel' ||
21               transactionStatus === 'deny' ||
22               transactionStatus === 'expire') {
23       updateStatus = 'failure';
24     } else if (transactionStatus === 'pending') {
25       updateStatus = 'pending';
26     }
27
28     // Update status di database
29     await prisma.pemesanan.update({
30       where: { orderId }, // pastikan orderId sesuai dengan kolom yang ada di database
31       data: { status: updateStatus },
32     });
33
34     res.status(200).send('OK');
35   } catch (error) {
36     console.error('Failed to handle notification:', error);
37     res.status(500).send('Internal Server Error');
38   }
39 };
40 export const cancelPembayaran = async (req, res) => {
41   const { id } = req.params; // Mengambil id dari parameter rute
42   console.log(id);
43   try {
44     // Mencari pembayaran berdasarkan pemesananId
45     const pembayaran = await prisma.pembayaran.findFirst({
46       where: { pemesananId: parseInt(id) },
47     });
48
49     if (!pembayaran) {
50       return res.status(404).json({ error: 'Pembayaran tidak ditemukan' });
51     }
52
53     // Memperbarui status pembayaran
54     await prisma.pembayaran.update({
55       where: { id: pembayaran.id },
56       data: { status: 'cancelled' },
57     });
58
59     res.status(200).json({ message: 'Pemesanan berhasil dibatalkan' });
60   } catch (error) {
61     console.log(error);
62     res.status(500).json({ error: error.message });
63   }
64 };

```

3. Route

```
1 import express from "express";
2 import { getStudio, getStudioById, createStudio, updateStudio, deleteStudio } from "../controllers/StudioModel.js";
3 import { getUsers, Register, Login, Logout } from "../controllers/users.js";
4 import { verifyToken, authorizeRole } from "../middleware/VerifyToken.js";
5 import { createPemesanan, getPemesanan, getPemesananById, updatePemesanan, deletePemesanan, getBookingsForStudio, getPemesananByUserId } from "../controllers/PemesananController.js";
6 import { getPembayaran, getPembayaranById, createPembayaran, updatePembayaran, deletePembayaran, createPayment, getNotification, cancelPembayaran } from "../controllers/PembayaranController.js";
7 // import { refreshToken } from "../controller/refreshToken.js";
8
9 const router = express.Router();
10
11 // verifyToken, authorizeRole("OWNER")
12 // route studio
13 router.get("/studio", getStudio);
14 router.get("/studio/:id", getStudioById);
15 router.post("/studio", createStudio);
16 router.patch("/studio/:id", updateStudio);
17 router.delete("/studio/:id", deleteStudio);
18
19 // route user
20 router.get("/users", getUsers);
21 router.post("/users", Register);
22 router.post("/login", Login);
23 // router.get("/token", refreshToken);
24 router.delete("/logout", Logout);
25
26 // route pemesanan
27 router.post("/pemesanan", createPemesanan);
28 router.get("/pemesanan", getPemesanan);
29 router.get("/pemesanan/:id", getPemesananById);
30 router.put("/pemesanan/:id", updatePemesanan);
31 router.delete("/pemesanan/:id", deletePemesanan);
32 router.post("/pembayaran/cancel/:id", cancelPembayaran);
33
34 //route pembayaran
35 router.get("/pembayaran", getPembayaran);
36 router.get("/pembayaran/:id", getPembayaranById);
37 router.post("/pembayaran", createPembayaran);
38 router.put("/pembayaran/:id", updatePembayaran);
39 router.delete("/pembayaran/:id", deletePembayaran);
40
41 router.get("/pemesanan/studio/:studioId", getBookingsForStudio);
42
43 // api payment gateway
44 router.post("/api/trancation", createPayment);
45 router.post("/api/notification", getNotification);
46
47 // Tambahkan rute untuk mendapatkan riwayat pemesanan berdasarkan userid
48 router.get("/pemesanan/user/:userId", getPemesananByUserId);
49
50 export default router;
51
```

Kode ini mendefinisikan rute-rute API menggunakan Express di Node.js untuk mengelola operasi CRUD terkait studio, pengguna, pemesanan, dan pembayaran. Rute-rute ini mencakup pembuatan, pengambilan, pembaruan, dan penghapusan entitas seperti studio, pengguna, pemesanan, dan pembayaran. Middleware seperti `verifyToken` digunakan untuk memastikan akses yang aman. Terdapat juga rute khusus untuk integrasi dengan gateway pembayaran.

Lampiran 5. Sertifikat HKI yang telah Terbit

REPUBLIC INDONESIA KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA	
SURAT PENCATATAN CIPTAAN	
Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:	
Nomor dan tanggal permohonan	: EC00202467790, 18 Juli 2024
Pencipta	
Nama	: Muhammad Rizqi, Dyah Apriliani, S.T., M.Kom dkk
Alamat	: Jl. Letjen S.Parmen Pesanggrahan Brebes RT/RW 007/017, Kec./Brebes Kab.Brebes Provinsi Jawa Tengah 52212, Brebes, Jawa Tengah, 52212
Kewarganegaraan	: Indonesia
Pemegang Hak Cipta	
Nama	: Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama
Alamat	: Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana 52142, Margadana, Tegal, Jawa Tengah 52142
Kewarganegaraan	: Indonesia
Jenis Ciptaan	: Program Komputer
Judul Ciptaan	: Aplikasi Studiobook Untuk Pemesanan Jasa Foto Berbasis Website Di Daerah Brebes
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	: 18 Juli 2024, di Tegal
Jangka waktu perlindungan	: Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan	: 000643142
adalah benar berdasarkan keterangan yang diberikan oleh Pemohon. Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.	
	a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL u.b. Direktur Hak Cipta dan Desain Industri
	 IGNATIUS M.T. SILALAH NIP. 196812301996031001
Disclaimer: Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.	

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Muhammad Rizqi	Jl. Letjen S.Parman Pesanggrahan Brebes RT/RW 007/017, Kec./Brebes Kab.Brebes Provinsi Jawa Tengah 52212, Brebes, Brebes
2	Dyah Apriliani, S.T., M.Kom	Perumahan Mutiara Indah Blok C4 Jalan Nangka 2 RT/RW 002/002, Kelurahan Procot, Kecamatan Slawi Kab.Tegal Provinsi Jawa Tengah 52412, Slawi, Tegal
3	Dairoh, M.Sc	Jl. Garuda No 3 RT/RW 002/009 Randu Gunting Tegal Selatan Kec.Randugunting Kota Tegal Provinsi Jawa Tengah 52131, Tegal Selatan, Tegal



Lampiran 6. Lembar Bimbingan



**SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA**

LEMBAR BIMBINGAN SKRIPSI

Nama : Muhamad Rizqi
 Nim : 20090059
 No. Ponsel : 081329593435
 Judul TA : Aplikasi Studiobook Untuk Pemesanan Jasa Foto
 Menggunakan Algoritma Dijkstra Berbasis Website di
 Daerah Brebes
 Dosen Pembimbing I : Dyah Apriliani, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	22/3/2024	bab 1-3	<ul style="list-style-type: none"> Ambl data ↳ perancangan aplikasi ↳ aplikasi 	
2.	1/4/2024	Aplikasi	<ul style="list-style-type: none"> - Cari bukti y validasi fotogrametri - fotogrametri dg rumus foto - jadwal - level user: (?) - lanjutkan aplikasi 	
3.	22/4/2024	Aplikasi	<ul style="list-style-type: none"> - Validasi foto studi - Google scholar - Tamalan & dibuat lanjut aplikasi 	



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA

4.	17/5 2024	Aplikasi	Caruhun 1 user perilla Ordo-	
5.	28/6 2024	Aplikasi	Backend direksa -Integrasi	
6.	9/7 2024	Aplikasi	Backend - roles - (Integrasi) - - Report	
7.	17/7 2024	Manual + role	- direksa - Laporan 1,2	
8.	18/7 2024	Laporan.	Rinci laporan Bab 1-2.	
9.	22/7 2024	Laporan	Rinci + form + Payroll	
10.	24/7 2024	Laporan	Atc - -	



**SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA**

--	--	--	--	--

Tegal,
Dosen Pembimbing I

Dyah Apriliani, S.T., M.Kom.
NIPY. 09.015.225



SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA






LEMBAR BIMBINGAN SKRIPSI

Nama : Muhammad Rizqi
Nim : 20090059
No. Ponsel : 081329593435
Judul 1A : Aplikasi Studiobook Untuk Pemesanan Jasa Roto Menggunakan Algoritma Dijkstra Berbasis Website di Daerah Brebes
Dosen Pembimbing I : Dairoh, M.Sc

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	17/09 2024	Pemeriksaan Use case pada aplikasi	Membuat use case diagram	f
2	15/10/2024	use case	- use case use - Sequence - Diagram database - Diagram Entity Relationship - Menu - Diagram class	f
3	20/11/2024	skripsi	- Gambar dan - Aplikasi di tambah - page bintik - Rumus pada - dan di penuhi - studi kasus - deskripsi	f



**SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA**

4	02 Juni 2024	Aplikasi	-menambahkan jadwal -membuat dashboard admin -membuat dashboard owner	
5	17 Juni 2024	Aplikasi	-Pengecekan Aplikasi -Lanjutkan Back end	
6	10 Juli 2024	Aplikasi	-Pengecekan Aplikasi (selesaikan aplikasi)	
7	19 Juli 2024	Laporan dan Aplikasi	ACC aplikasi Perbaiki laporan bab 2	
8	21 Juli 2024	Laporan	ACC Laporan	



**SARJANA TERAPAN TEKNIK INFORMATIKA
POKITEKNIK HARAPAN BERSAMA**

--	--	--	--	--

Tegal,
Dosen Pembimbing II

Dairoh, M.Sc
NIPY. 04.014.178