

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **A. Analisa Hasil**

Sistem ini menggunakan metode pengembangan *Extreme programming*. Dengan metode ini, pengembang dan klien dapat saling bekerjasama demi keberhasilan sebuah system. Seperti yang telah dijelaskan sebelumnya bahwa metode *Extreme Programming* ini mempunyai empat tahapan, yaitu perencanaan, perancangan, pengkodean dan pengujian. Berikut adalah tahapan-tahapan yang dilalui oleh peneliti dalam membangun aplikasi *webstite* Penerapan *Framework* Laravel pada Pembuatan Aplikasi Profil dan Penyewaan Studio Berbasis *Website* pada studio Vibesroom:

##### **1. Perencanaan (*Planning*)**

Perencanaan merupakan tahapan awal yang dilakukan untuk memenuhi kebutuhan sistem yang akan dibuat berdasarkan hasil wawancara dan daftar pustaka yang telah diperoleh sebelumnya dari pihak Studio Vibesroom. Kemudian peneliti melakukan Analisa dan identifikasi kebutuhan pengguna dan kebutuhan untuk membangun sistem, telah dijabarkan identifikasi kebutuhan pengguna berdasarkan hasil dari Teknik perolehan data yang telah dilakukan terkait dalam pembuatan sistem ini, telah disimpulkan bahwa belum adanya sistem yang dapat memudahkan pihak Studio Vibesroom dalam mengelola data *booking*. Berikut tahapan analisis untuk mengetahui kebutuhan sistem yang akan dirancang:

### a. Analisis Kebutuhan Sistem

Sesuai dengan hasil wawancara yang telah dilakukan bahwa pihak programmer mitra di Studio Vibesroom membutuhkan suatu sistem yang dapat memudahkan petugas dalam melakukan pengelolaan data *booking*.

Berdasarkan hasil wawancara yang dilakukan dapat disimpulkan bahwa Sistem Pengelolaan Data *booking* pada Studio Vibesroom ini didasari pada kebutuhan sistem informasi yang digunakan untuk mengolah data *booking* di studio Vibesroom. Adapun syarat-syarat fitur yang harus ada di sistem pengelolaan data *booking* ini adalah sebagai berikut:

#### 1) Fitur Halaman Petugas

a) <i>Login</i> Petugas	:	Berisi <i>form</i> melakukan <i>login</i> petugas Studio.
b) <i>Dashboard</i>	:	<i>Dashboard</i> ini merupakan halaman yang pertama kali ditampilkan, berisi fitur utama aplikasi seperti fitur <i>menu</i> ( <i>dashboard</i> , kategori produk, produk, data <i>booking</i> , jadwal <i>booking</i> ), personalisasi, kelola akun, dan fitur <i>logout</i> .
c) Kategori Produk	:	Berisi fitur yang menampilkan info kategori dari produk studio
d) Produk	:	Berisi fitur yang menampilkan data Produk. Pada <i>menu</i> ini petugas dapat menambahkan, mengubah dan menghapus data produk.
e) Data <i>booking</i>	:	Berisi fitur yang menampilkan data <i>booking</i> . Pada <i>menu</i> ini petugas studio dapat melihat riwayat <i>booking</i> dari <i>username</i> , waktu, status, detail dari pembayaran atau membatalkan jadwal <i>booking</i> yang sudah di tetapkan.
f) Jadwal <i>Booking</i>	:	Berisi fitur yang menampilkan data <i>booking</i> studio. Pada <i>menu</i> ini petugas dapat melihat jadwal berupa kalender dalam format bulan, minggu, atau hari.
g) Personalisasi	:	Berisi menu yang menampilkan banner dari halaman landing page dan halaman portofolio serta mengatur <i>hyperlink</i> dari masing-masing

	<i>icon</i> sosial media yang tertera pada halaman <i>website</i>
h) <i>Role User</i>	: Berisi fitur yang menampilkan role dari akun yang sudah pernah dibuat
i) <i>User</i>	: Berisi fitur yang menampilkan semua akun <i>user</i> yang sudah terdaftar pada <i>website</i> studio Vibesroom dan juga petugas dapat me-reset password dari user, mengubah informasi user atau menghapusnya.

## 2) Fitur Halaman Pengguna

a) <i>Login</i> Pengguna	: Berisi <i>form</i> melakukan <i>login</i> pengguna
	Studio.
b) <i>Dashboard</i>	: <i>Dashboard</i> ini merupakan halaman yang pertama kali ditampilkan, berisi fitur utama aplikasi seperti fitur <i>menu</i> ( <i>dashboard</i> , <i>orders</i> ), dan fitur <i>logout</i> .
c) <i>Orders</i>	: Berisi fitur yang menampilkan info data order booking yang sudah pernah dilakukan.

### b. Analisis Kebutuhan Pengguna

#### 1) Analisis Kebutuhan Fungsional

Analisis fungsional merupakan paparan mengenai fungsi dari fitur-fitur yang ditanamkan ke dalam sistem pengelolaan data *booking* pada Studio Vibesroom agar berjalan sesuai yang diharapkan. Berikut fungsi Fitur-fitur tersebut antara lain:

##### a) Petugas

(1) Mampu menampilkan info dan *detail* rencana *booking*.

(2) Mampu menampilkan, menambahkan, menghapus data *booking*.

(3) Mampu menampilkan, menambahkan, mengubah, dan menghapus data *user*.

(4) Mampu menampilkan, melihat detail, dan membatalkan data *booking*.

(5) Mampu mengunduh laporan dari data *booking user*.

b) Petugas

(1) Mampu menampilkan info dan *detail* rencana *booking*.

(2) Mampu menampilkan info riwayat *booking*.

## 2) Analisis Kebutuhan Non Fungsional

Kebutuhan *non* fungsional menjelaskan mengenai kebutuhan peneliti untuk membuat dan menjalankan sistem agar dapat berjalan dengan baik dan sebagaimana mestinya. Berikut adalah kebutuhan non-fungsional dari system ini:

a) Kebutuhan Perangkat Lunak

(1) Bahasa Pemrograman (*PHP*)

(2) Sistem Operasi (*Windows*)

(3) *Text Editor (Visual Studio Code)*

(4) *Local Server (XAMPP)*

(5) *Web Browser*

**b) Kebutuhan Perangkat Keras**

- (1) *Processor* : AMD APU A4-9125 2,3GHz
- (2) RAM : 8192 MB
- (3) *Harddisk* : 500GB
- (4) VGA : *AMD Radeon R3 Graphics*
- (5) Sistem operasi : *Windows 10*
- (6) Tipe sistem : *64-bit Operating System*

**2. Perancangan (Design)****a. Pemodelan**

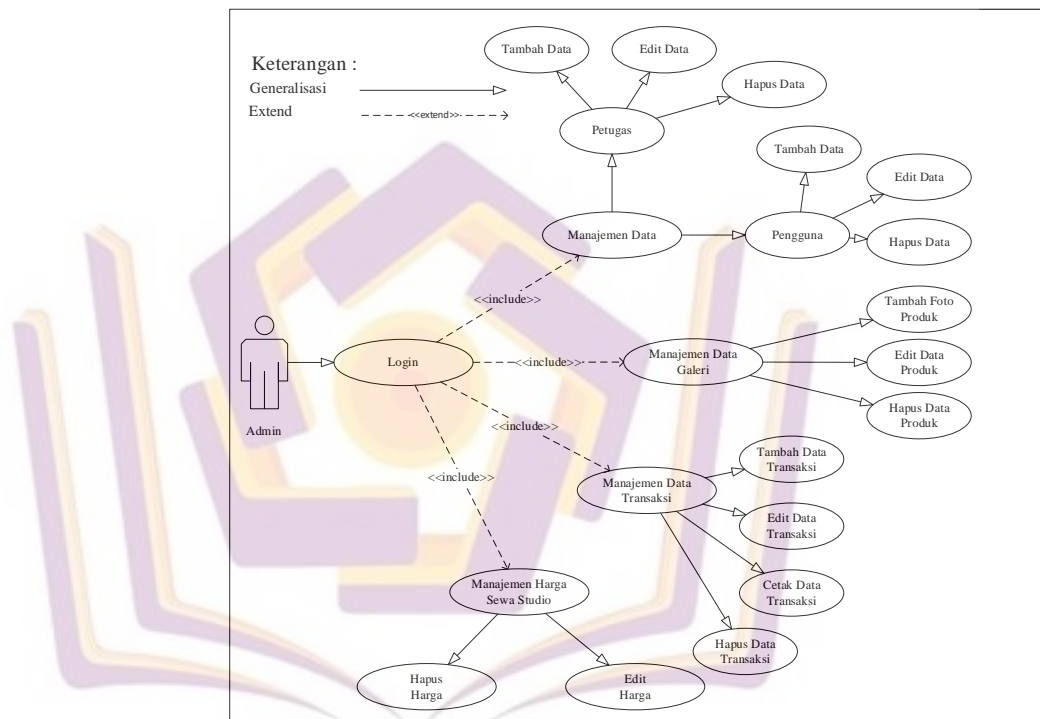
Tahap pemodelan terbagi menjadi 2, yaitu model analisis dan model desain. Pada model analisis menggunakan *Unified Modeling Language (UML)* dengan 4 diagram yaitu *use case diagram*, *activity diagram*, *class diagram* dan *sequence diagram*. Sedangkan pada model desain dilakukan desain antar muka (*design user interface*). Tahap pemodelan yang dilakukan adalah sebagai berikut:

## 1) Model Analisis

### a. Use Case Diagram

Use case yang dikembangkan dalam aplikasi ini dapat dilihat pada gambar berikut:

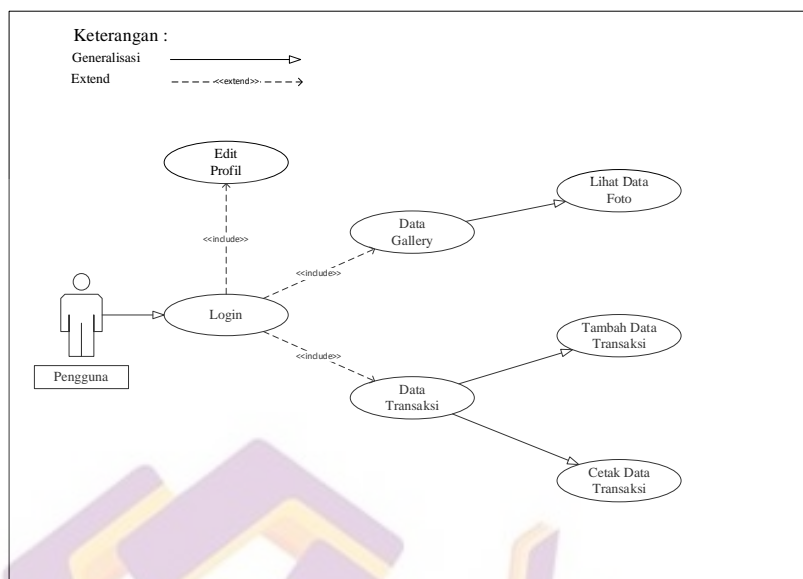
#### 1) Use Case Diagram Admin



Gambar 4.1 Use Case Diagram Admin

Gambar 4.1 menjelaskan aktivitas yang dilakukan oleh admin yaitu melakukan pengelolaan data *user*, pengelolaan foto yang akan di *upload*, pengelolaan transaksi, dan pengelolaan harga studio. Admin harus login *username* dan *password* untuk melakukan akses pengelolaan tersebut.

## 2) Use Case Diagram User



Gambar 4.2 Use Case Diagram User

Gambar 4.2 menjelaskan aktivitas yang dilakukan oleh user seperti melakukan edit profil dari user, melihat data *gallery* yang ada didalam website, dan melakukan transaksi penyewaan studio. Untuk mendapatkan akses tersebut user dapat melakukan login menggunakan *username* dan *password* yang telah dimiliki.

Adapun definisi aktor yang terdapat pada Use Case Diagram diatas dapat dilihat pada tabel 4.1

Tabel 4.1. Definisi aktor

No	Nama aktor	Definisi
1.	Admin	Admin memiliki akses penuh terhadap sistem, diantaranya manajemen (user, Studio), manajemen data <i>gallery</i> , data transaksi, dan harga studio.
2.	User	User hanya mempunyai hak akses edit profil, melihat data <i>gallery</i> yang ada di <i>website</i> serta melakukan proses transaksi.

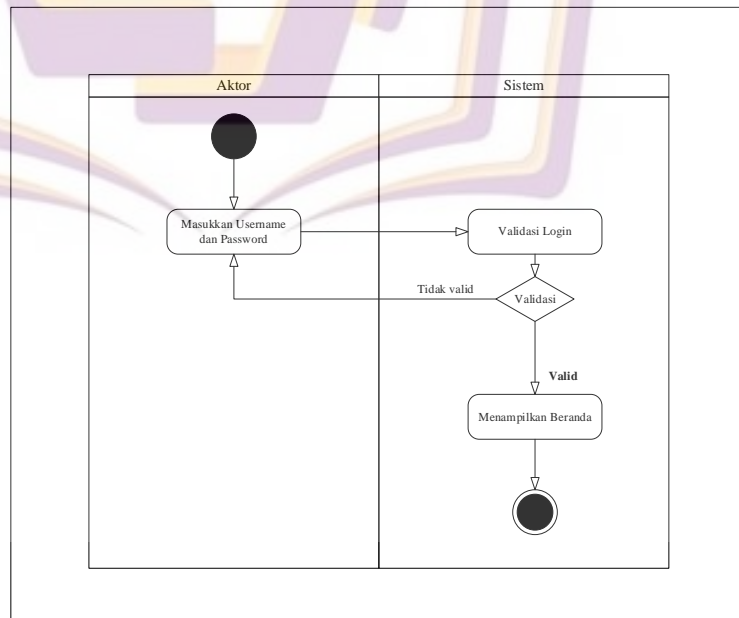
Adapun definisi *use case* yang terdapat pada *Use Case Diagram* dapat dilihat pada tabel 4.2.

Tabel 4.2. Definisi Use Case

No	Nama <i>Use Case</i>	Definisi
1.	Manajemen User	Mengelola data user meliputi tambah data user, mengedit data user dan menghapus data user.
2.	Manajemen Data <i>Gallery</i>	Mengelola data <i>gallery</i> meliputi menambahkan data foto, mengedit data foto dan menghapus data foto.
3.	Manajemen Data Transaksi	Mengelola data transaksi meliputi menambahkan transaksi, mengedit data transaksi dan menghapus data transaksi.
4.	Manajemen Harga Studio	Mengelola data harga Studio meliputi mengedit harga Studio dan menghapus harga studio.

## b. Activity Diagram

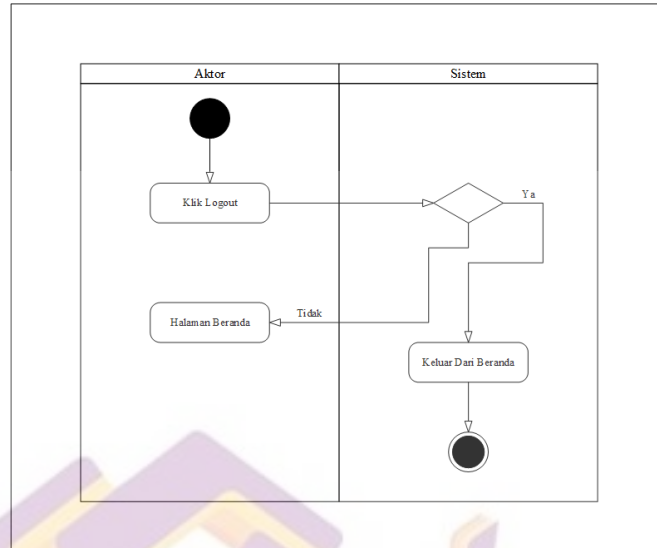
### 1) Activity Diagram Login



Gambar 4.3 Activity Diagram Login

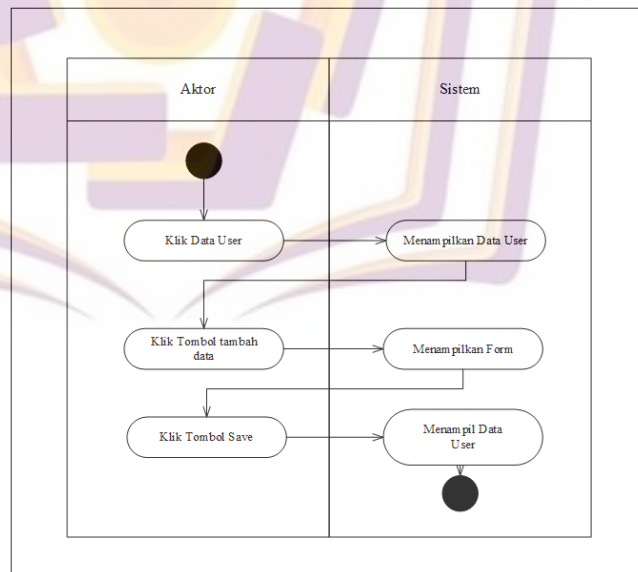


### 2) Activity Diagram Logout



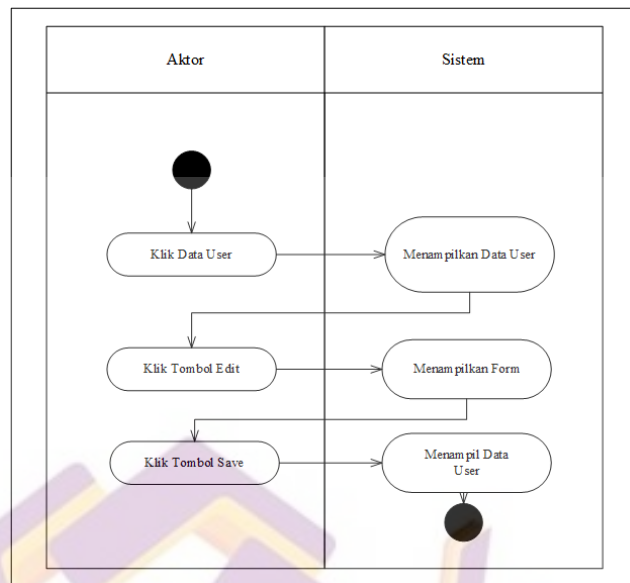
Gambar 4.4 Activity Diagram Logout

### 3) Activity Diagram Admin Tambah Data User



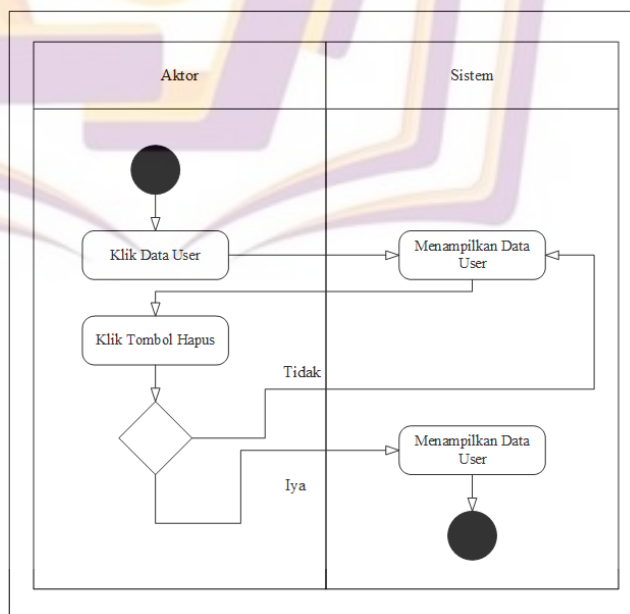
Gambar 4.5 Activity Diagram Tambah Data User

#### 4) Activity Diagram Admin Edit Data User



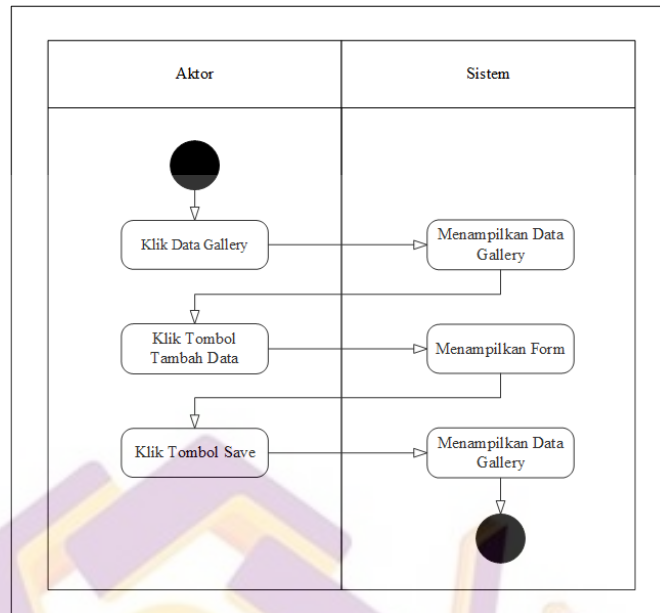
Gambar 4.6 Activity Diagram Edit Data User

#### 5) Activity Diagram Admin Hapus Data User



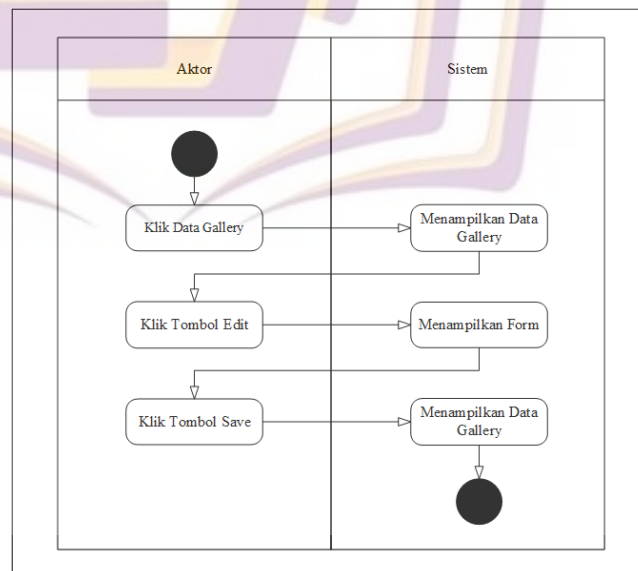
Gambar 4.7 Activity Diagram Hapus Data User

6) *Activity Diagram* Tambah Foto Gallery

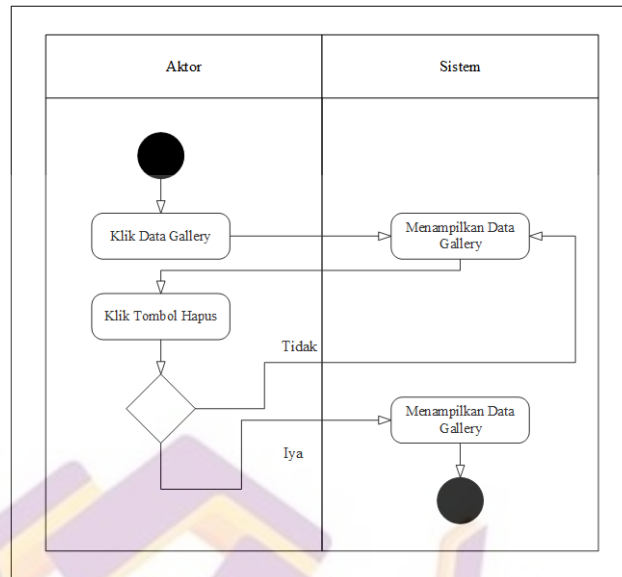
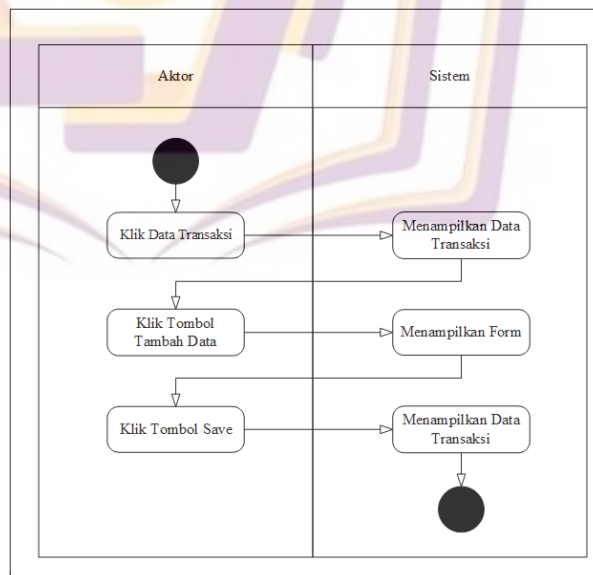


Gambar 4.8 Activity Diagram Tambah Foto Gallery

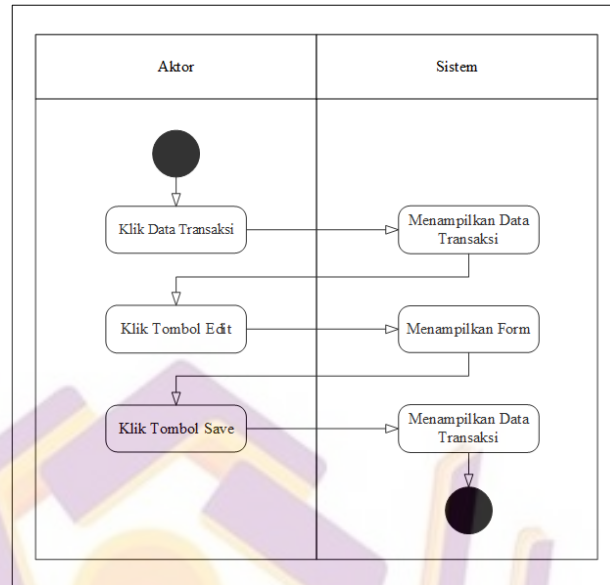
7) *Activity Diagram* Edit Data Foto Gallery



Gambar 4.9 Activity Diagram Edit Data Foto Gallery

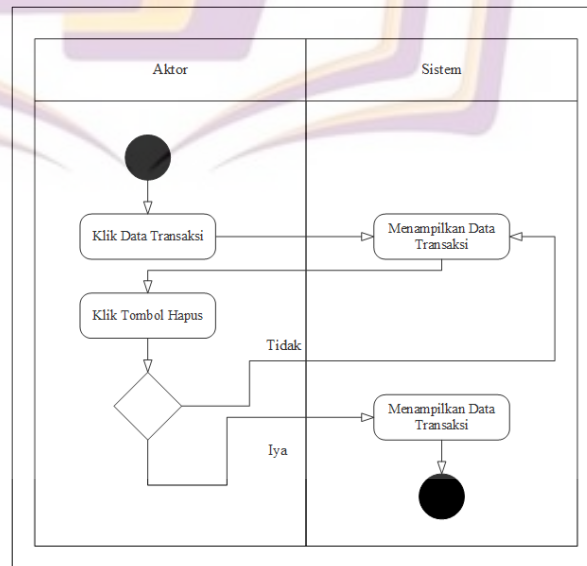
8) *Activity Diagram Hapus Data Foto Gallery*Gambar 4.10 *Activity Diagram Hapus Data Foto Gallery*9) *Activity Diagram Tambah Data Transaksi*Gambar 4.11 *Activity Diagram Tambah Data Transaksi*

10) *Activity Diagram* Edit Data Transaksi



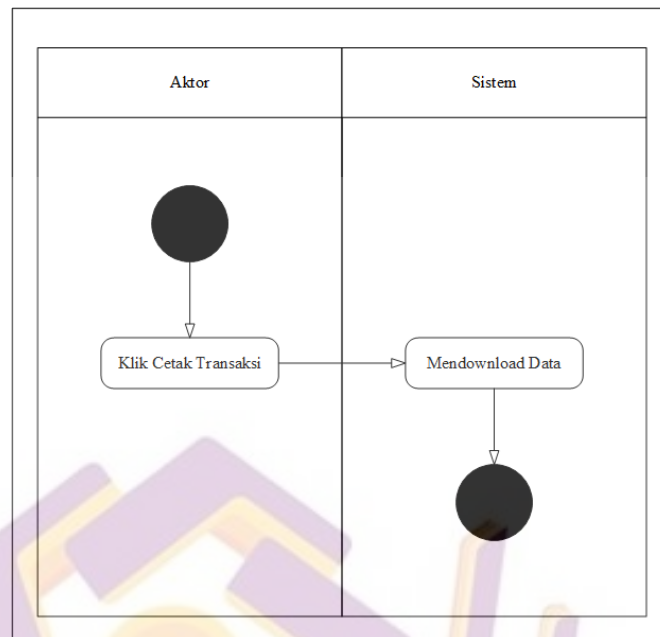
Gambar 4.12 *Activity Diagram* Edit Data Transaksi

11) *Activity Diagram* Hapus Data Transaksi



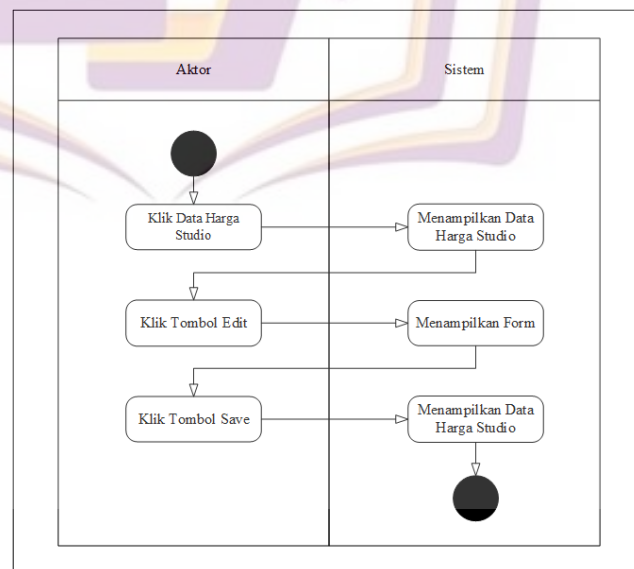
Gambar 4.13 *Activity Diagram* Hapus Data Transaksi

12) *Activity Diagram* Cetak Data Transaksi



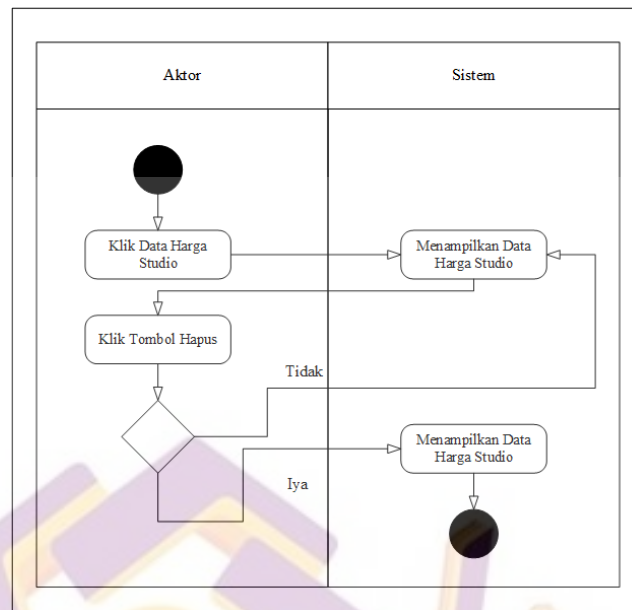
Gambar 4.14 *Activity Diagram* Cetak Data Transaksi

13) *Activity Diagram* Edit Data Harga studio



Gambar 4.15 *Activity Diagram* Edit Data Harga Studio

#### 14) Activity Diagram Hapus Data Harga Studio

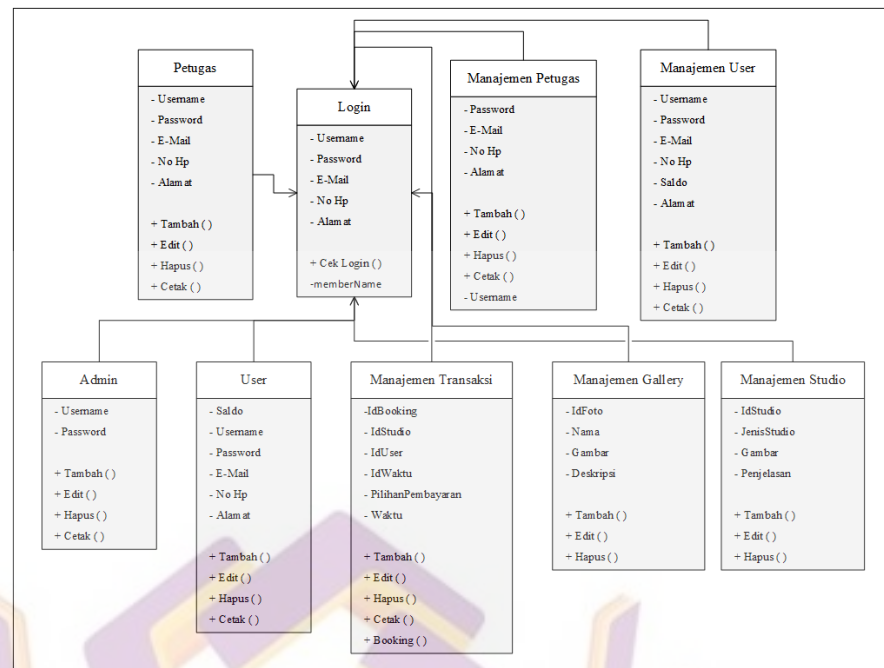


Gambar 4.16 Activity Diagram Hapus Data Harga Studio

#### c. Class Diagram

*Class diagram* adalah model statis yang menggambarkan struktur dan deskripsi *class* serta hubungannya antara *class*.

*Class diagram* menggambarkan struktur dan deskripsi *class* serta hubungan satu sama lain, seperti hubungan *generalisasi*, *asosiasi*.



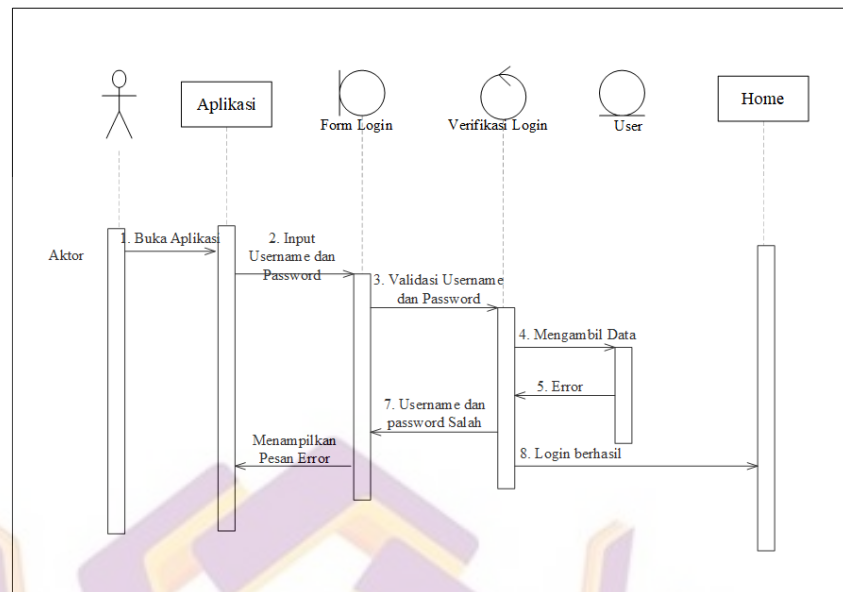
Gambar 4.17 Class Diagram

#### d. Sequence Diagram

*Sequence Diagram* digunakan untuk menggambarkan interaksi antar objek di dalam dan sekitar sistem (termasuk pengguna, *display*, dan sebagainya). *Sequence diagram* dibuat berdasarkan *activity diagram* dan *class diagram*.

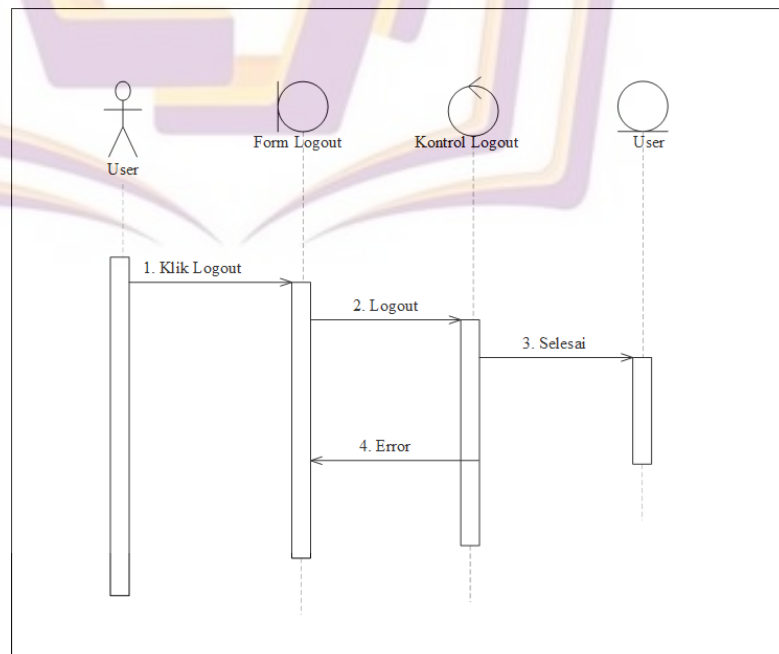


### 1) Sequence Diagram Login



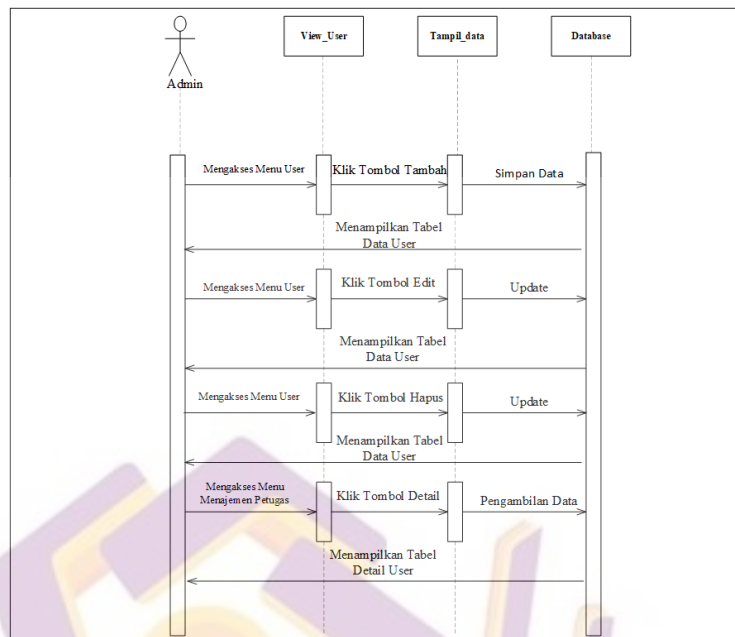
Gambar 4.18 Sequence Diagram Login

### 2) Sequence Diagram Logout



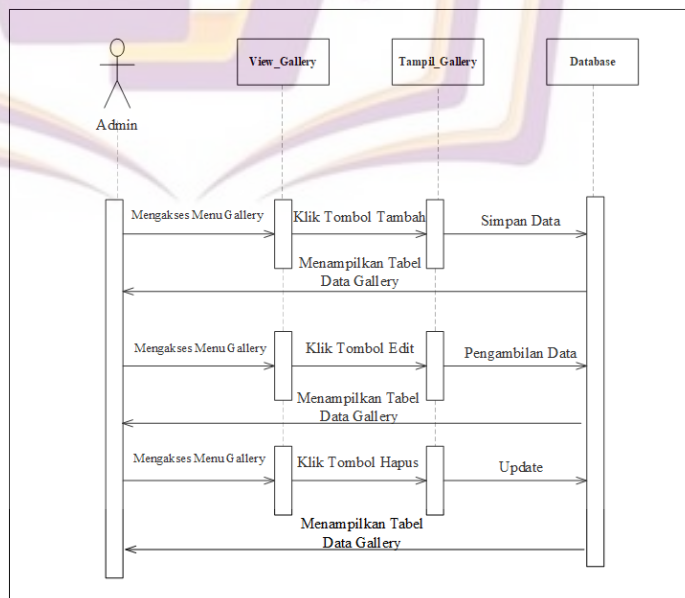
Gambar 4.19 Sequence Diagram Logout

### 3) Sequence Diagram Manajemen User



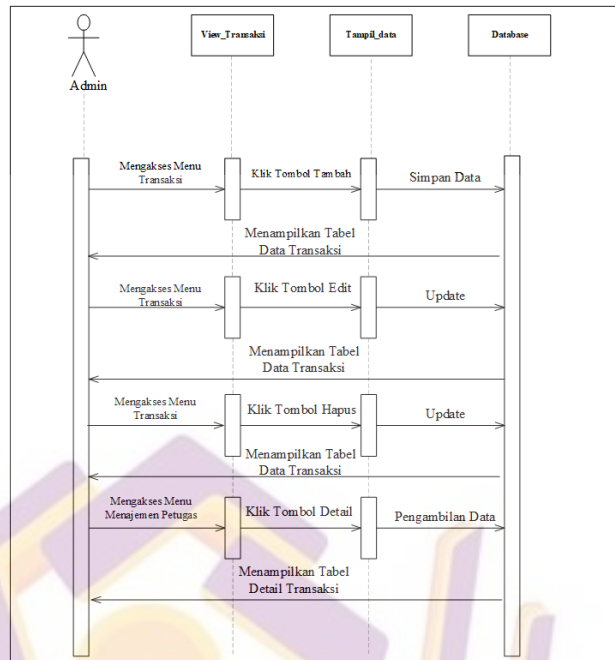
Gambar 4.20 Sequence Diagram Manajemen User

### 4) Sequence Diagram Manajemen Data Gallery



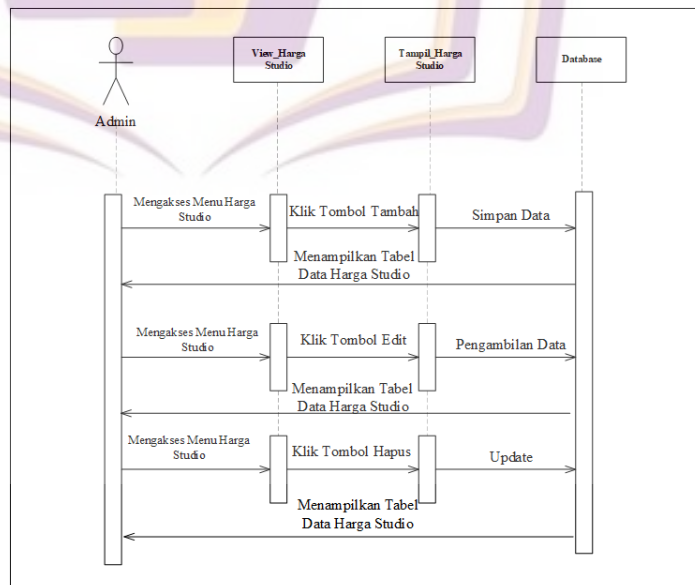
Gambar 4.21 Sequence Diagram Manajemen Data Gallery

### 5) Sequence Diagram Manajemen Data Transaksi



Gambar 4.22 Sequence Diagram Manajemen Data Transaksi

### 6) Sequence Diagram Manajemen Harga Studio




Gambar 4.23 Sequence Diagram Manajemen Harga Studio

## 2) Model Desain

*Design User Interface* merupakan desain tampilan aplikasi secara sederhana untuk mengetahui gambaran dari sistem yang akan dibuat adalah sebagai berikut:

### a) *Form Halaman Login*

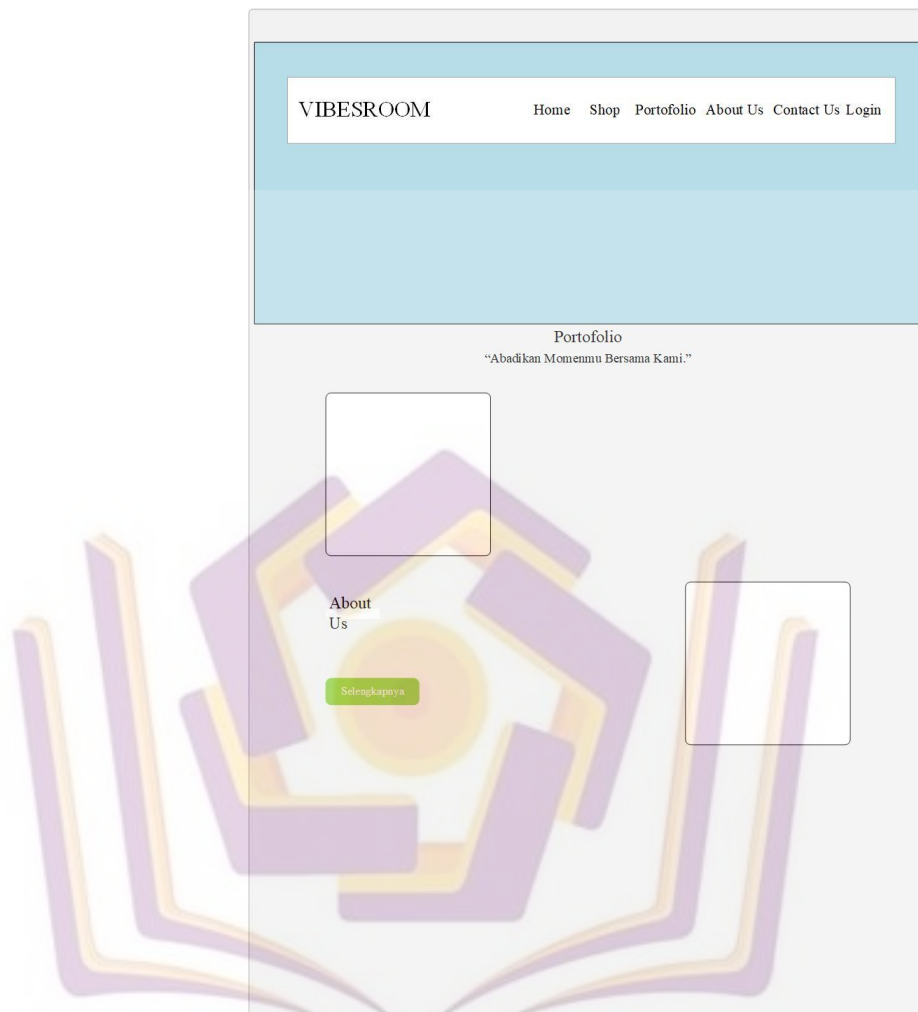


The image shows a login form for 'VIBESROOM'. The form is titled 'VIBESROOM Sign In' and contains two input fields: 'Email' and 'Password'. Below the fields is a green button labeled 'Masuk'. At the bottom left, there is a link 'Belum Punya Akun? Sign Up'. The form is set against a light blue background with the text 'Join Our Community' on the right side.

Gambar 4.24 *Form Halaman Login User*

*Form Login* merupakan *form* untuk masuk ke akun yang telah dibuat atau telah dimiliki. Untuk melakukan *login*, user maupun admin perlu menginputkan username dan password. Dalam *form login user* ada hak akses untuk melakukan penyewaan di halaman booking, melihat profil yang ditampilkan dalam *website* dan terdapat akses untuk melakukan pengelolaan website bagi *Admin*.

b) Halaman Utama *Website*



Gambar 4.25 Halaman Utama *Website*

Merupakan gambaran halaman utama *website* ketika diakses oleh user untuk melakukan penyewaan maupun melihat profil dari Vibesroom Studio.

## c) Halaman Buat Akun

Gambar 4.26 Halaman Buat Akun

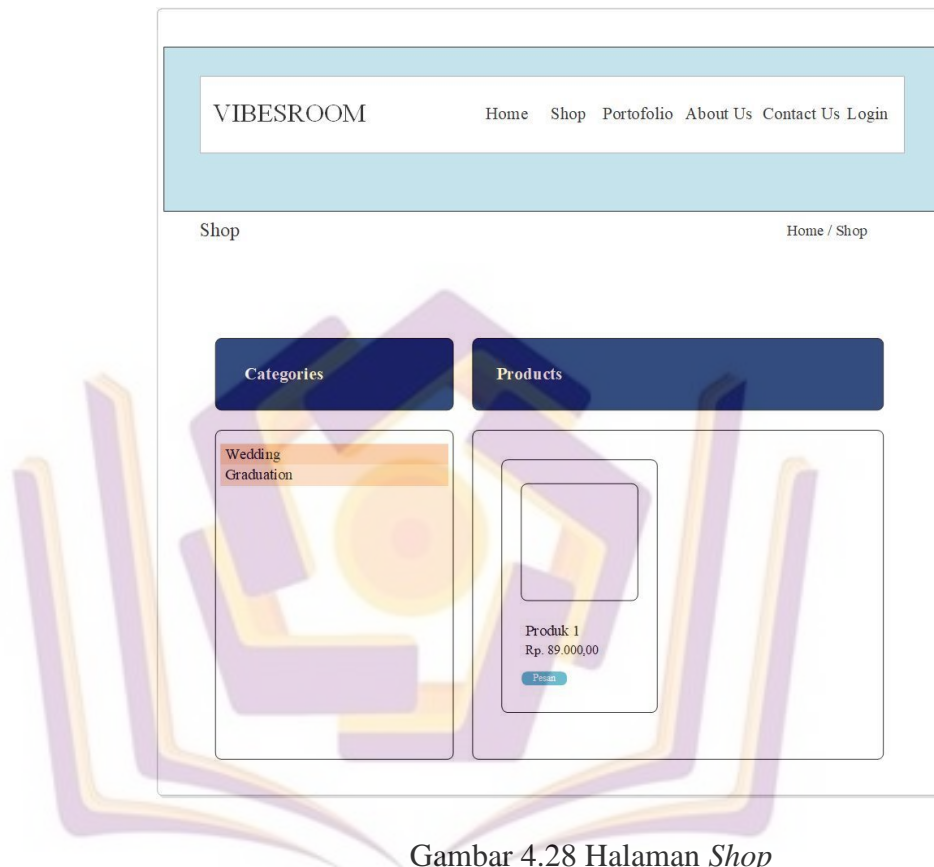
Merupakan halaman membuat akun untuk melakukan akses penyewaan studio saat sudah memiliki akun.

d) Halaman *About Us*Gambar 4.27 Halaman *About Us*

Pada halaman *about us* terdapat penjelasan tentang orang-orang yang berperan dalam keberjalanannya Vibesroom Studio.

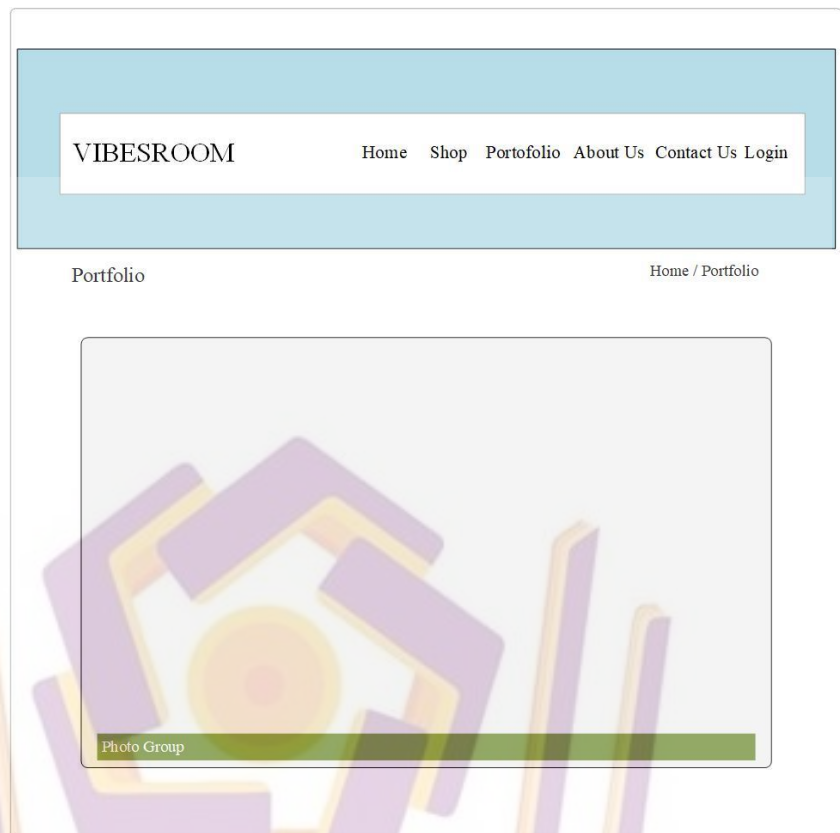
User dapat mengetahui informasi yang ditampilkan dalam website tersebut.

e) Halaman Shop



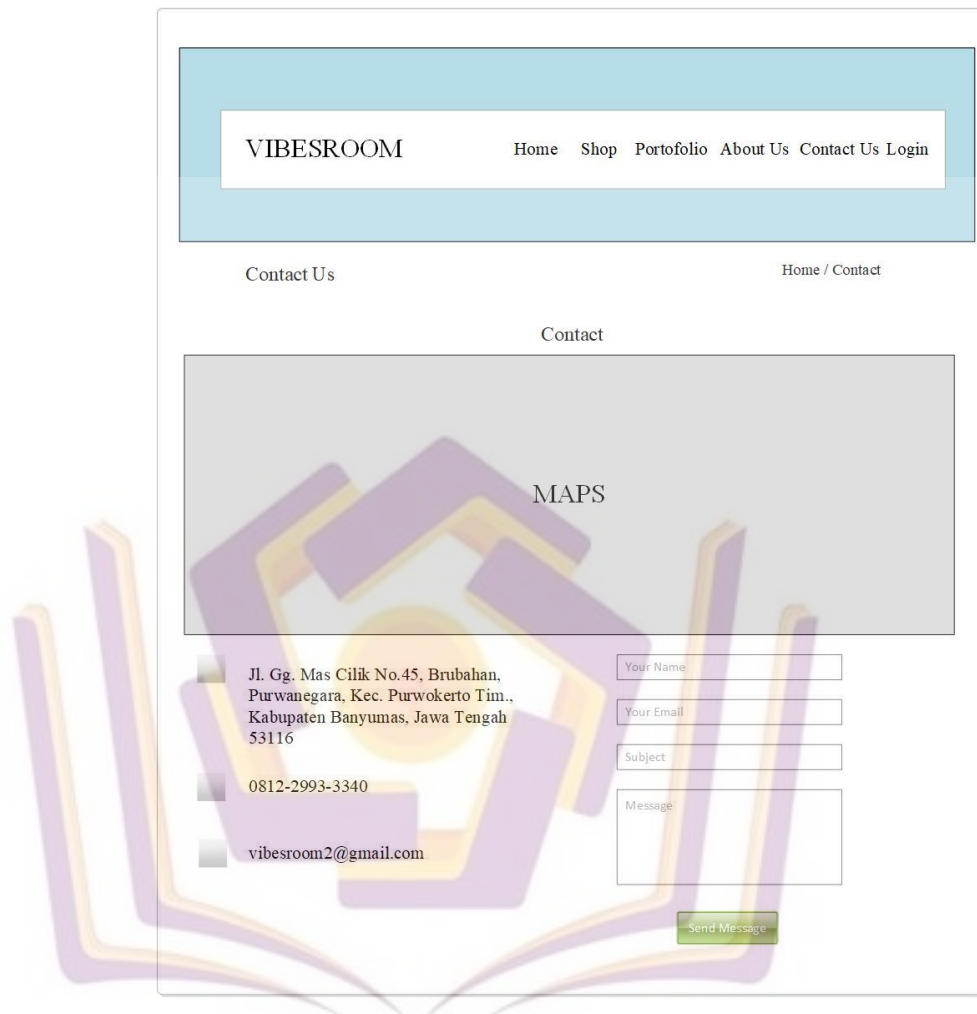
Gambar 4.28 Halaman *Shop*

Pada halaman shop, user dapat melihat penawaran paket yang disediakan untuk melakukan penyewaan dengan memilih produk dan harga yang akan dipesan. Dalam proses penyewaan user harus memiliki saldo untuk melakukan penyewaan dan apabila user belum melakukan login akun maka proses penyewaan studio tidak dapat berlangsung.

f) Halaman *Portofolio*Gambar 4.29 Halaman *Portofolio*

Pada halaman *Portofolio*, admin dapat menambahkan foto berkaitan dengan kegiatan pemotretan yang ada atau telah diselenggarakan di Vibesroom Studio agar dapat menarik minat pengunjung yang akan menggunakan jasa dari Vibesroom Studio.

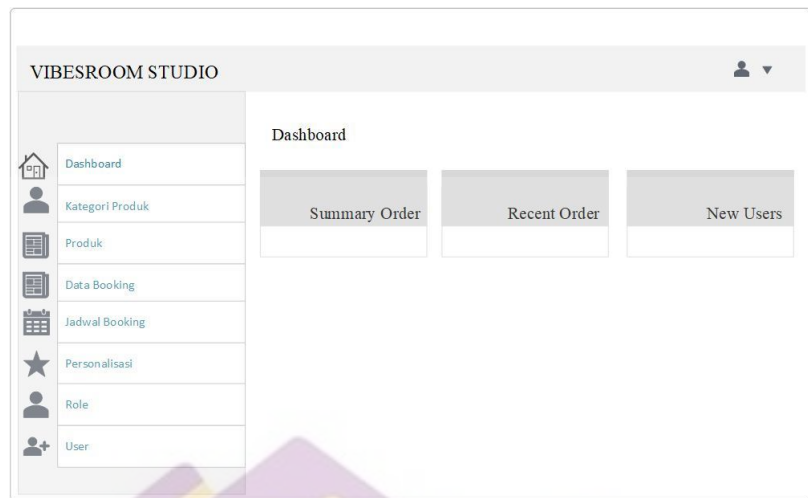


g) Halaman *Contact Us*

Gambar 4.30 Halaman Contact Us

Pada halaman *Contact*, terdapat informasi mengenai alamat, nomor handphone dan email dari Vibesroom Studio.

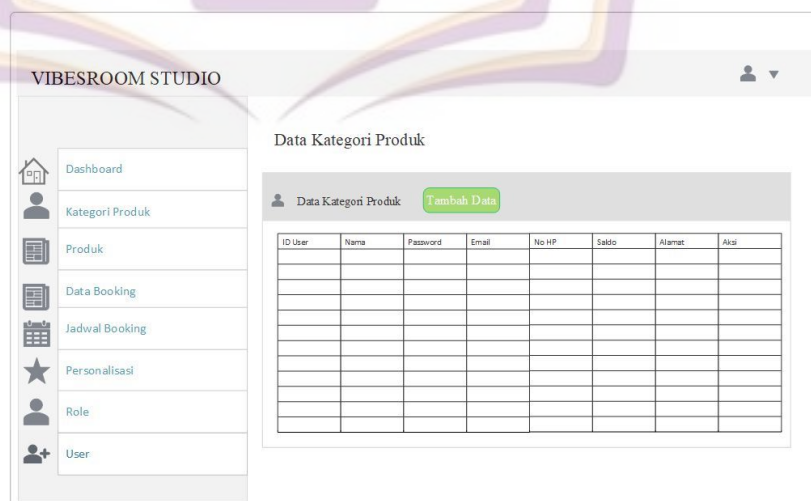
## h) Halaman *Dashboard* Admin



Gambar 4.31 Halaman *Dashboard* Admin

Pada halaman *dashboard* terdapat jumlah data yang ada pada data user, *gallery*, transaksi dan harga di Vibesroom Studio.

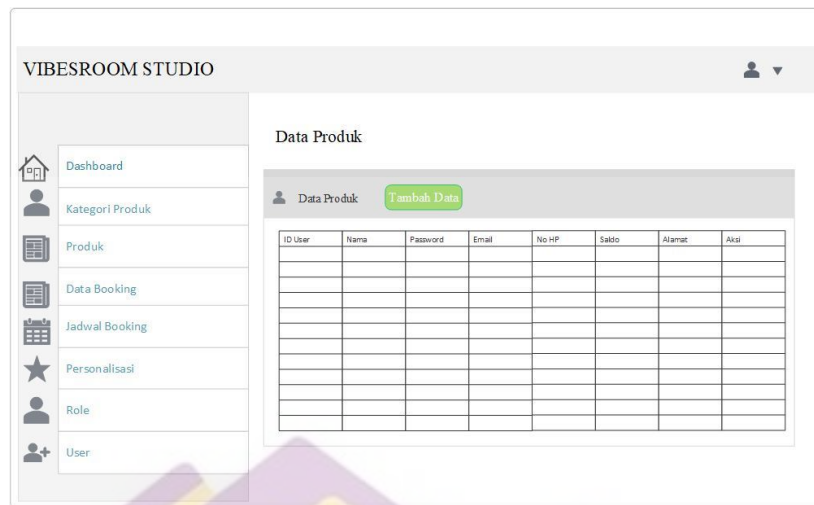
## i) Halaman Data Kategori Produk



Gambar 4.32 Halaman Data Kategori Produk

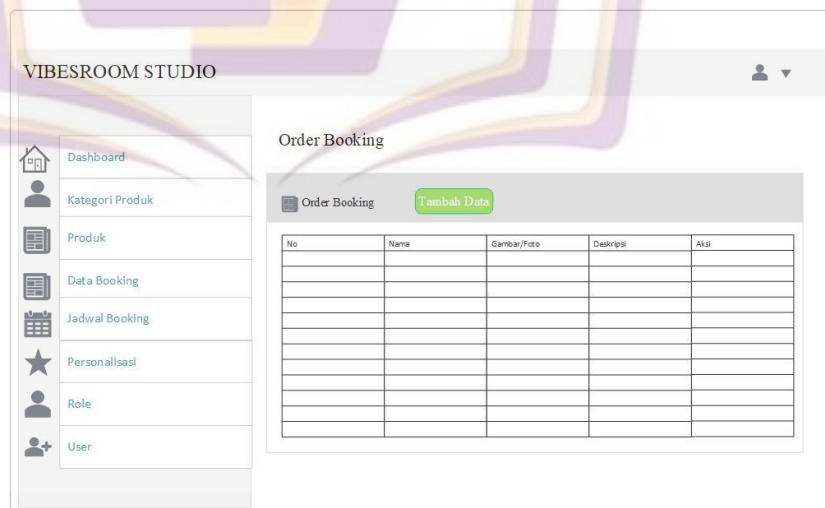
Pada halaman data kategori produk, admin dapat menambah, mengedit atau menghapus data kategori produk.

## j) Halaman Data Produk



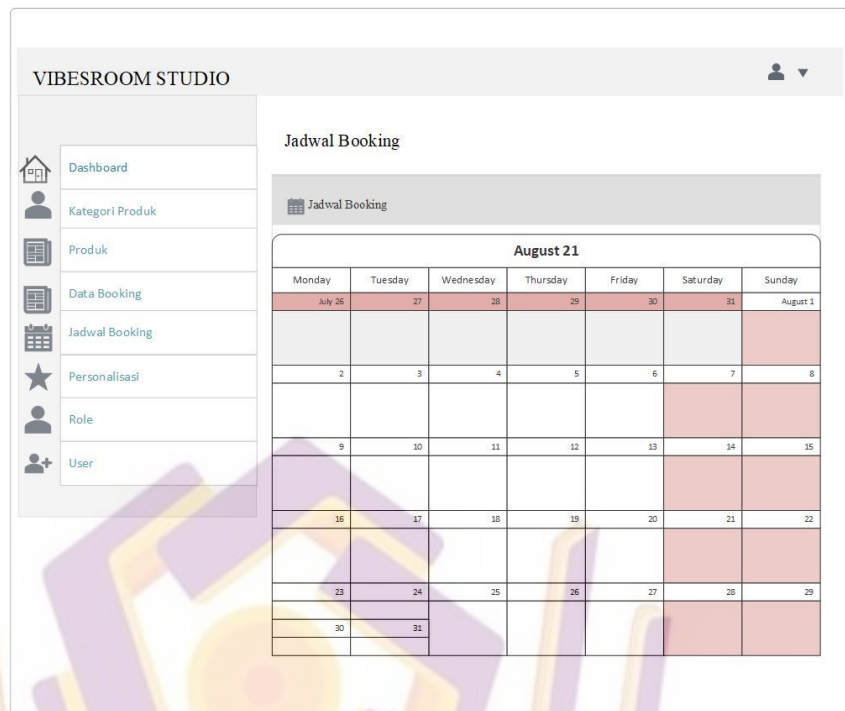
Gambar 4.33 Halaman Data Produk

Pada halaman data produk, admin dapat menambahkan data, mengedit data atau menghapus data produk.

k) Halaman Data *Booking*.Gambar 4.34 Halaman Data *Booking*

Pada halaman data *Order Booking*, admin dapat menambahkan data, mengedit data, atau menghapus data order booking yang sudah dilakukan oleh pengunjung.

## 1) Halaman Jadwal Booking



Gambar 4.35 Halaman Jadwal Booking

Pada halaman data transaksi, admin dapat melihat transaksi penyewaan studio dalam bentuk calendar berdasarkan status yang tertera baik itu sudah dilaksanakan, penundaan, maupun belum terlaksana.

## b. Perancangan Tabel

## 1) Tabel Admin dan user

Tabel 4.3. Tabel Admin dan User

Nama Kolom	Tipe data	Keterangan
<b>Id</b>	<i>Bigint(20)</i>	<i>Primary Key</i>
Username	<i>varchar(255)</i>	
Email	<i>varchar(255)</i>	
Password	<i>varchar(255)</i>	
Created_At	<i>Timestamp</i>	
Updated_At	<i>Timestamp</i>	

## 2) Tabel Booking

Tabel 4.4. Tabel Booking

<b>Nama Kolom</b>	<b>Tipe data</b>	<b>Keterangan</b>
<b>id</b>	<i>bigint(20)</i>	<i>Primary Key</i>
<i>user_id</i>	<i>bigint(20)</i>	
<i>tanggal_booking</i>	<i>date</i>	
<i>waktu_booking</i>	<i>time</i>	
<i>catatan</i>	<i>text</i>	
<i>status</i>	<i>enum</i>	
<i>pembayaran</i>	<i>Tinyint(1)</i>	
<i>total</i>	<i>Double</i>	
<i>bukti_transfer</i>	<i>varchar(255)</i>	
<i>created_at</i>	<i>Timestamp</i>	
<i>updated_at</i>	<i>Timestamp</i>	

## 3) Tabel Detail Booking

Tabel 4.5. Tabel Detail Booking

<b>Nama Kolom</b>	<b>Tipe data</b>	<b>Keterangan</b>
<b>id</b>	<i>bigint(20)</i>	<i>Primary Key</i>
<i>booking_id</i>	<i>bigint(20)</i>	
<i>product_id</i>	<i>varchar(255)</i>	
<i>first_name</i>	<i>varchar(255)</i>	
<i>last_name</i>	<i>varchar(255)</i>	
<i>address</i>	<i>varchar(255)</i>	
<i>phone</i>	<i>varchar(255)</i>	
<i>email</i>	<i>varchar(255)</i>	
<i>created_at</i>	<i>Timestamp</i>	
<i>updated_at</i>	<i>Timestamp</i>	

#### 4) Tabel Produk

Tabel 4.6. Tabel Produk

Nama Kolom	Tipe data	Keterangan
<b>id</b>	<i>bigint(20)</i>	<i>Primary Key</i>
category_id	<i>bigint(20)</i>	
nama	<i>Varchar(150)</i>	
deskripsi	<i>Text</i>	
harga	<i>Double</i>	
diskon	<i>Double</i>	
diskon_type	<i>Enum</i>	
thumbnail	<i>Varchar(255)</i>	
slug	<i>Varchar(255)</i>	
status	<i>Varchar(255)</i>	

#### 5) Tabel produk kategori

Tabel 4.7. Tabel Produk Kategori

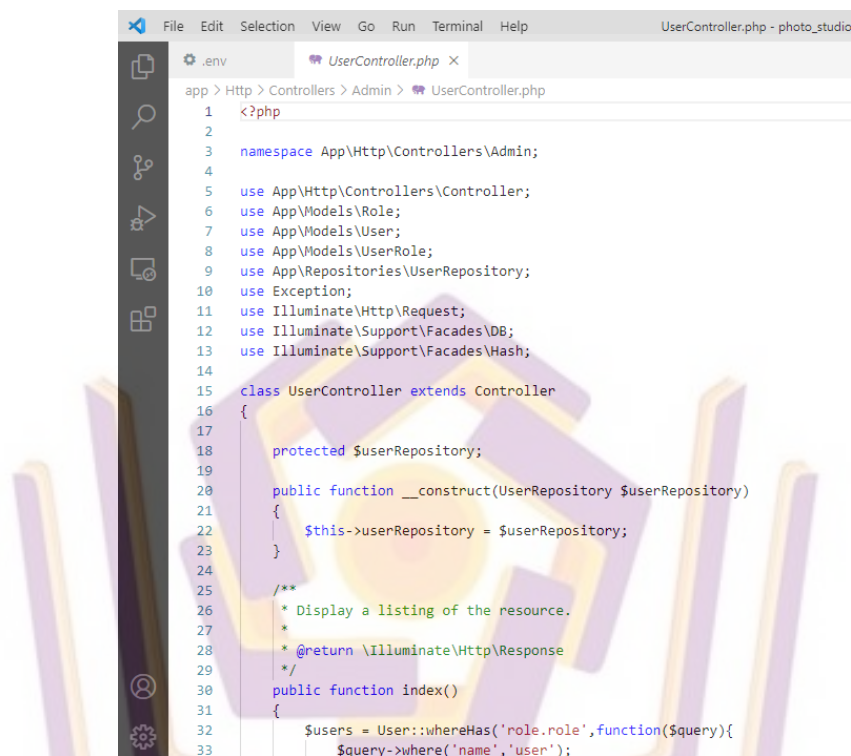
Nama Kolom	Tipe data	Keterangan
<b>id</b>	<i>Bigint(20)</i>	<i>Primary Key</i>
nama	<i>Varchar(100)</i>	
style	<i>Enum</i>	
deskripsi	<i>Text</i>	
thumbnail	<i>Text</i>	

### 3. Pengkodean (Coding)

Berikut ini salah satu contoh *script* kode yang digunakan dalam proses pengkodean Penerapan *Framework* Laravel untuk Pembuatan Aplikasi Profil dan Booking Studio berbasis *Website* (Studi Kasus: Vibesroom Studio Purwokerto):

### a) Pembuatan Data User

Dalam proses pengkodean berisi update, input dan hapus data. Berikut adalah pengkodeannya:



```

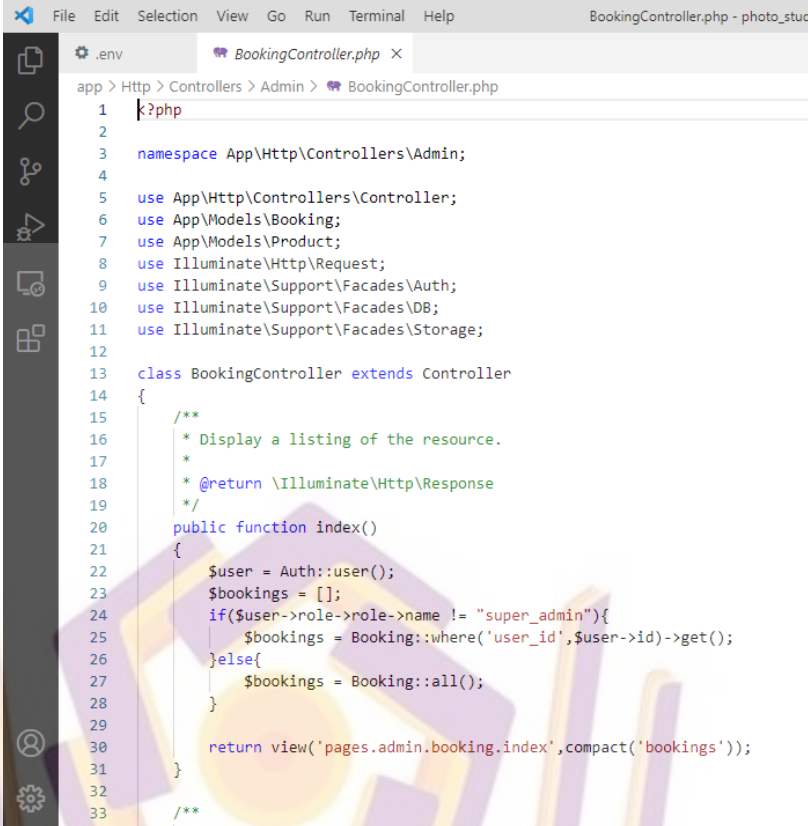
1  <?php
2
3  namespace App\Http\Controllers\Admin;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Role;
7  use App\Models\User;
8  use App\Models\UserRole;
9  use App\Repositories\UserRepository;
10 use Exception;
11 use Illuminate\Http\Request;
12 use Illuminate\Support\Facades\DB;
13 use Illuminate\Support\Facades\Hash;
14
15 class UserController extends Controller
16 {
17     protected $userRepository;
18
19     public function __construct(UserRepository $userRepository)
20     {
21         $this->userRepository = $userRepository;
22     }
23
24     /**
25      * Display a listing of the resource.
26      *
27      * @return \Illuminate\Http\Response
28      */
29     public function index()
30     {
31         $users = User::whereHas('role.role', function($query){
32             $query->where('name', 'user');
33

```

Gambar 4.36 Gambar Kode Data User

### b) Pembuatan Transaksi atau Booking

Dalam proses pengkodean berisi data-data waktu dan studio yang akan ditampilkan. Berikut pengkodeannya:



```

1  k?php
2
3  namespace App\Http\Controllers\Admin;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Booking;
7  use App\Models\Product;
8  use Illuminate\Http\Request;
9  use Illuminate\Support\Facades\Auth;
10 use Illuminate\Support\Facades\DB;
11 use Illuminate\Support\Facades\Storage;
12
13 class BookingController extends Controller
14 {
15     /**
16      * Display a listing of the resource.
17      *
18      * @return \Illuminate\Http\Response
19      */
20     public function index()
21     {
22         $user = Auth::user();
23         $bookings = [];
24         if($user->role->role->name != "super_admin"){
25             $bookings = Booking::where('user_id',$user->id)->get();
26         }else{
27             $bookings = Booking::all();
28         }
29
30         return view('pages.admin.booking.index',compact('bookings'));
31     }
32
33     /**

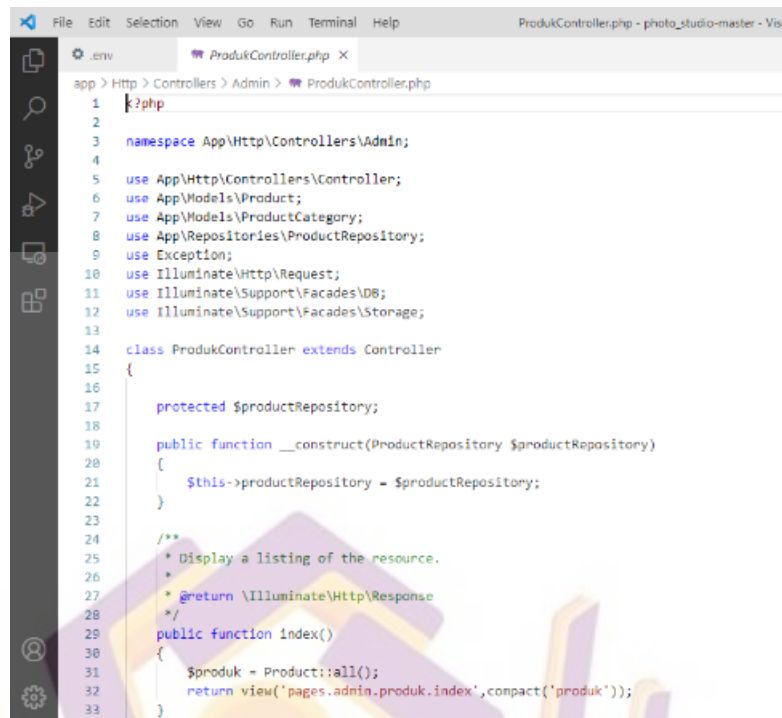
```

Gambar 4.37 Gambar Kode *Booking*

### c) Pembuatan Produk

Dalam proses pengkodean berisi perintah untuk upload, edit dan hapus data produk. Berikut pengkodeannya:





```

1  k?php
2
3  namespace App\Http\Controllers\Admin;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Product;
7  use App\Models\ProductCategory;
8  use App\Repositories\ProductRepository;
9  use Exception;
10 use Illuminate\Http\Request;
11 use Illuminate\Support\Facades\DB;
12 use Illuminate\Support\Facades\Storage;
13
14 class ProdukController extends Controller
15 {
16
17     protected $productRepository;
18
19     public function __construct(ProductRepository $productRepository)
20     {
21         $this->productRepository = $productRepository;
22     }
23
24     /**
25      * Display a listing of the resource.
26      *
27      * @return \Illuminate\Http\Response
28      */
29     public function index()
30     {
31         $produk = Product::all();
32         return view('pages.admin-produk.index', compact('produk'));
33     }
34 }

```

Gambar 4.38 Gambar Kode Data Produk

## a. Implementasi Tabel

### 1) Tabel Admin dan User

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	email_verified_at	timestamp			Yes	NULL			Change Drop More
5	password	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
6	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
7	created_at	timestamp			Yes	NULL			Change Drop More
8	updated_at	timestamp			Yes	NULL			Change Drop More

Gambar 4.39 Implementasi Tabel Admin dan User

## 2) Tabel Boking

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	2 user_id	bigint(20)		UNSIGNED	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	3 tanggal_booking	date			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	4 waktu_booking	time			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	5 catatan	text	utf8mb4_unicode_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	6 status	enum('acc','pending','cancel')	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	7 is_pay	tinyint(1)			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	8 total	double			No	0			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	9 bukti_transfer	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	10 created_at	timestamp			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	11 updated_at	timestamp			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Gambar 4.40 Implementasi Tabel Boking

## 3) Tabel Detail Booking

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	2 booking_id	bigint(20)		UNSIGNED	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	3 product_id	varchar(255)	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	4 first_name	varchar(255)	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	5 last_name	varchar(255)	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	6 address	varchar(255)	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	7 phone	varchar(255)	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	8 email	varchar(255)	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	9 created_at	timestamp			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	10 updated_at	timestamp			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Gambar 4.41 Implementasi Tabel Detail Booking

## 4) Tabel Produk

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	2 category_id	bigint(20)		UNSIGNED	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	3 nama	varchar(150)	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	4 deskripsi	text	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	5 harga	double			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	6 diskon	double			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	7 diskon_type	enum('persen','harga')	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	8 thumbnail	varchar(255)	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	9 slug	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	10 status	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	11 created_at	timestamp			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	12 updated_at	timestamp			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Gambar 4.42 Implementasi Tabel Produk

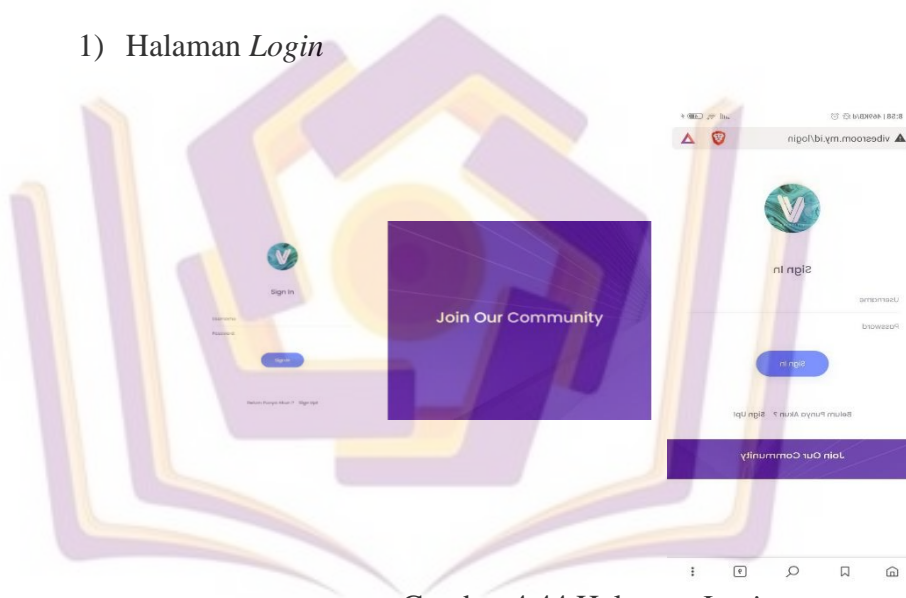
## 5) Tabel Kategori Produk

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 nama	varchar(100)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 style	enum('parent', 'sub', 'single')	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4 parent_id	bigint(20)		UNSIGNED	Yes	NULL			Change  Drop  More
<input type="checkbox"/>	5 deskripsi	text	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	6 thumbnail	text	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	7 slug	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	8 created_at	timestamp			Yes	NULL			Change  Drop  More
<input type="checkbox"/>	9 updated_at	timestamp			Yes	NULL			Change  Drop  More

Gambar 4.43 Implementasi Tabel Kategori Produk

### b. Screenshots Implementasi Design User Interface

#### 1) Halaman *Login*



Gambar 4.44 Halaman *Login*

Halaman *Login* merupakan halaman yang digunakan untuk masuk ke akun yang telah dibuat atau dimiliki. Untuk melakukan *login*, user maupun admin perlu menginputkan username dan password seperti yang tertera pada Gambar 4.44 di atas. Apabila pengguna belum memiliki akun maka bisa mendaftar dengan cara meng-klik pada text *Sign Up!* yang berada di bawah button *Sign in*.

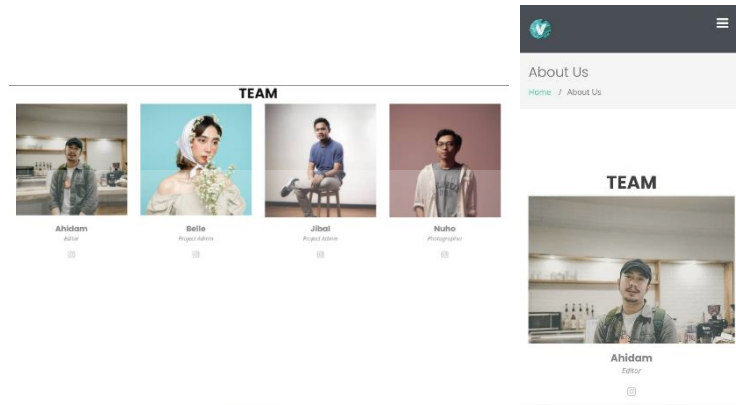
## 2) Halaman Buat Akun



Gambar 4.45 Halaman Buat Akun

Apabila seorang pengguna baru yang belum memiliki akun hendak mendaftar maka tampilan halaman pendaftaran akan tertera seperti pada Gambar 4.45 dan pengguna akan diminta untuk mengisikan data berupa username, email, password serta konfirmasi password untuk memverifikasi agar tidak terjadi kesalahan dalam pembuatan password.

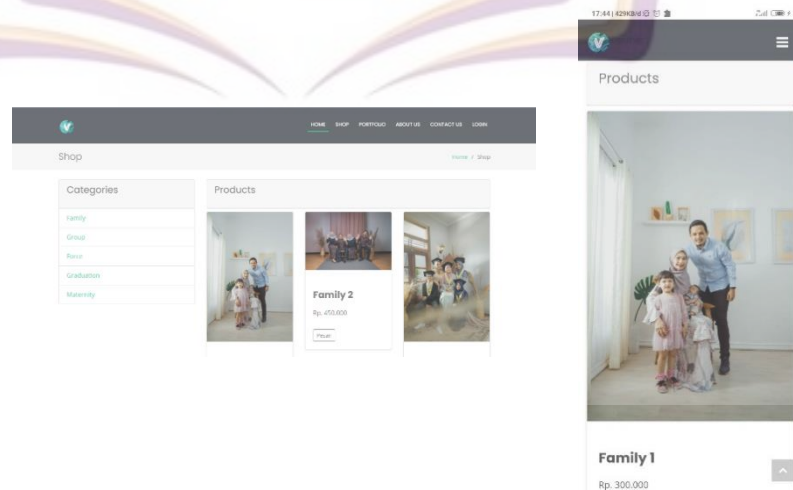
### 3) Halaman About Us



Gambar 4.46 Halaman *About Us*

Pada Halaman About Us akan memberikan informasi mengenai tim yang bertugas pada studio serta perannya masing-masing dalam keberlangsungan operasional studio seperti yang tertera pada Gambar 4.46

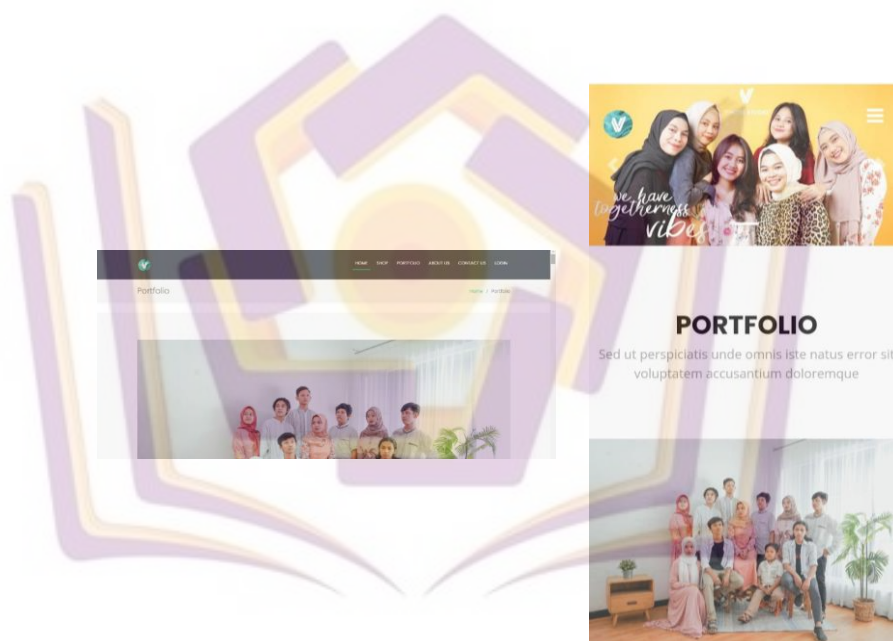
### 4) Halaman Shop



Gambar 4.47 Halaman *Shop*

Pengguna yang ingin langsung memeriksa katalog ataupun menyewa studio bisa langsung menuju halaman shop seperti tertera pada gambar 4.47. Informasi yang di ditampilkan adalah kategori dan paket yang tersedia untuk disewakan. Pada saat proses penyewaan pengguna diharuskan untuk terlebih dahulu log in dan memiliki cukup saldo sebagai syarat pemesanan.

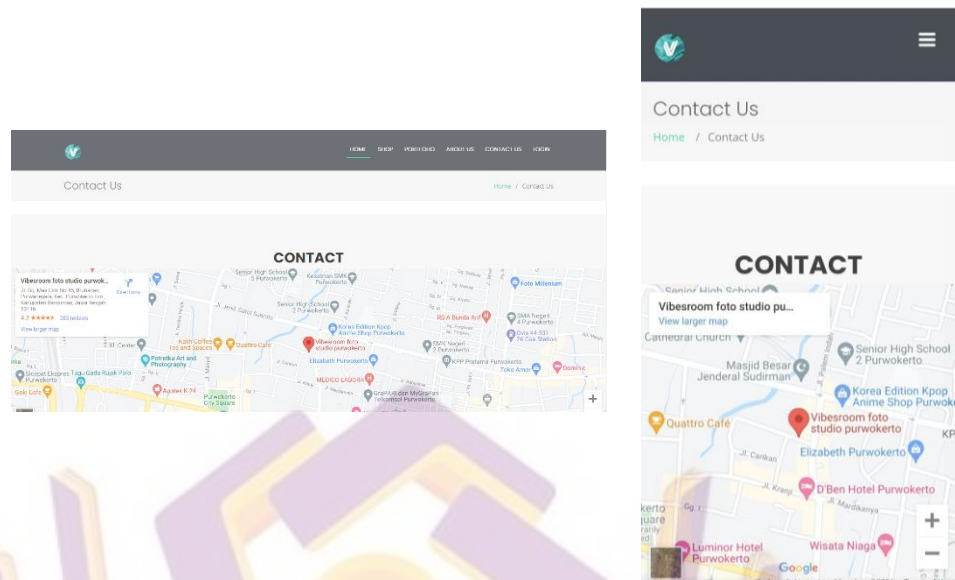
#### 5) Halaman *Gallery*



Gambar 4.48 Halaman Portofolio

Pada halaman portofolio akan menampilkan foto yang berkaitan dengan hasil pemotretan di Vibesroom Studio agar dapat menarik minat pengunjung yang akan menggunakan jasa dari Vibesroom Studio.

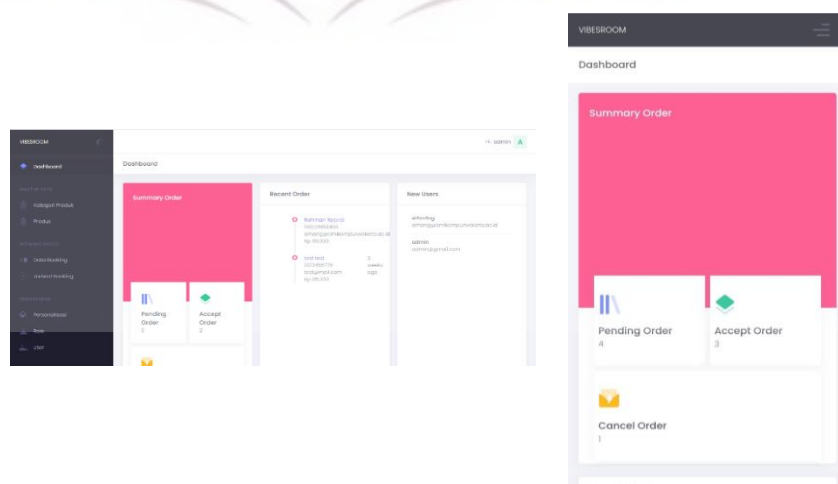
## 6) Halaman *Contact Us*



Gambar 4.49 Halaman *Contact Us*

Pada halaman *Contact Us* akan tertera informasi yang berkaitan dengan alamat lokasi, No Handphone dan Email guna memudahkan pengguna mengetahui letak studio Vibesroom.

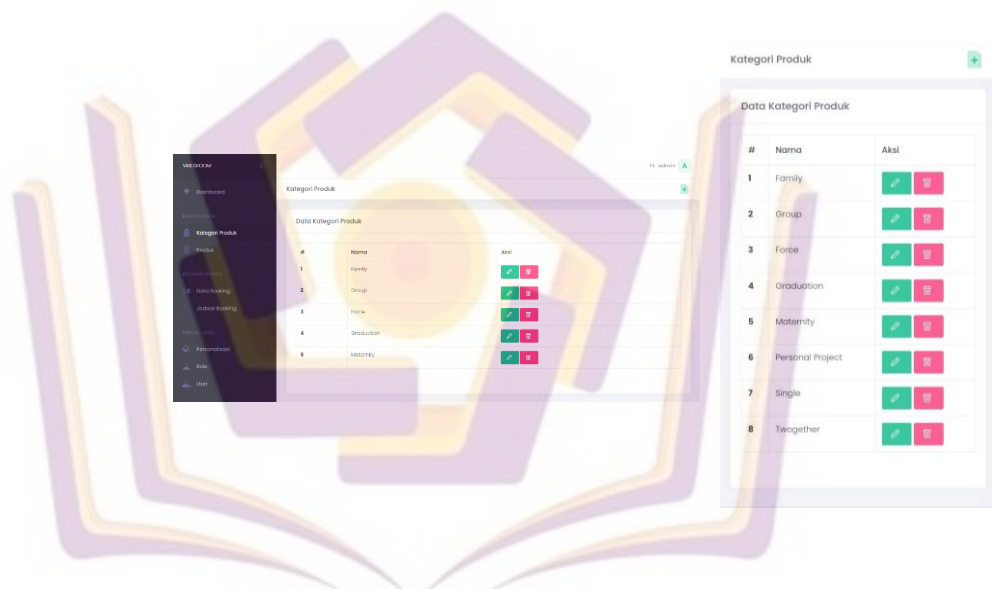
## 7) Halaman *Dashboard*



Gambar 4.50 Halaman *Dashboard*

Pada tampilan halaman dashboard admin seperti yang tertera pada Gambar 4.50 terdapat informasi mengenai update pendaftaran pengguna baru, pemesanan terbaru dan juga perolehan pesanan sementara. Halaman ini hanya dapat di akses oleh admin saja sedangkan halaman dashboard user memiliki tampilan yang berbeda.

#### 8) Halaman Kategori Produk



Gambar 4.51 Halaman Kategori Produk

Pada halaman kategori produk yang di tunjukkan oleh Gambar 4.51 terdapat beberapa menu untuk menambah kategori baru, mengubah atau menghapus kategori yang sudah ada.



## 9) Halaman Data Produk

#	Nama	Thumbnail	Kategori	Harga	Aksi
1	Family 1		Family	3000	
2	Family 2		Family	4500	
3	Group A		Group	1500	
4	Group B		Group	2500	

Gambar 4.52 Halaman Data Produk

Pada halaman data produk terdapat tampilan untuk nama produk, *thumbnail* dari produk, harga produk, Riwayat *update* produk serta aksi untuk mengubah maupun menghapus produk. Selain itu juga terdapat menu untuk menambahkan produk yang berada di bawah *icon* admin.

## 10) Halaman Data *Booking*

#	Username	Waktu	Status	Aksi
1	admin	Tanggal: 17 07 2021 Pukul: 10:00:00	ada	
2	silwang	Tanggal: 18 07 2021 Pukul: 22:00:00	ada	
3	test03	Tanggal: 19 07 2021 Pukul: 10:00:00	cancel	
4	Omoibrahim	Tanggal: 19-08-2021 Pukul: 04:16:00	pending	

Gambar 4.53 Halaman Data *Booking*

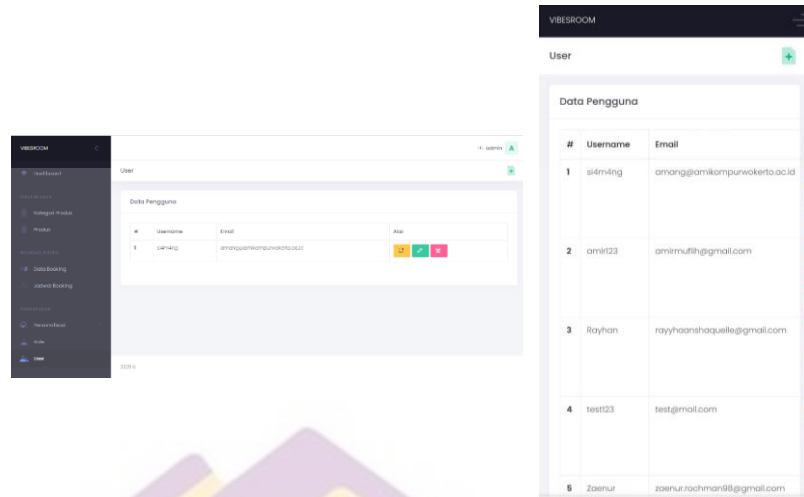
Pada halaman data booking admin dapat melihat informasi mengenai aktifitas booking yang sudah dilakukan oleh user pada akun mereka. Disana terdapat informasi mengenai username yang sudah melakukan booking, waktu booking dan status booking. Pada kolom aksi admin bisa melihat detail dari produk yang di booking dengan menekan *button details* serta memiliki hak untuk membatalkan booking dengan menekan *button batal*.

### 11) Tampilan Antrian *Booking*



Gambar 4.54 Halaman Antrian *Booking*

Pada halaman antrian booking terdapat informasi kalender dengan beberapa kriteria. Apabila menunjukkan tanda hijau maka berarti *booking* yang dilakukan sudah dalam status di terima, bila menunjukkan tanda *orange* maka berarti *booking* yang dilakukan masih dalam proses pending dikarenakan beberapa syarat belum terpenuhi sedangkan apabila menunjukkan tanda merah maka *booking* yang dilakukan telah dibatalkan oleh admin.

12) Halaman Manajemen Data *User*

The image displays two screenshots of a web application's user management interface. The left screenshot shows a sidebar menu with options like 'User', 'Produk', 'Data Booking', 'Pesanan Baru', 'User', and 'Item'. The right screenshot shows a 'User' management page with a table of users.

#	Username	Email	Aksi
1	si4m1ng	amang@amikompurwokerto.ac.id	[Reset Password] [Ubah] [Hapus]
2	amir123	amirmu1th@gmail.com	[Reset Password] [Ubah] [Hapus]
3	Rayhan	rayhaamshaquille@gmail.com	[Reset Password] [Ubah] [Hapus]
4	test123	test@gmail.com	[Reset Password] [Ubah] [Hapus]
5	Zaenur	zaenur.rochmar08@gmail.com	[Reset Password] [Ubah] [Hapus]

Gambar 4.55 Halaman Manajemen Data *User*

Pada halaman manajemen data *user* terdapat informasi mengenai data pengguna yang sudah terdaftar pada *system*. Dapat dilihat pada gambar 4.55 informasi mengenai pengguna berupa username dan email. Pada kolom aksi terdapat beberapa button yaitu *reset password*, *ubah* dan *hapus* akun.

#### 4. Pengujian (Testing)

Pengujian ini dilakukan untuk mengetahui apakah fungsi – fungsi yang ada pada aplikasi telah berjalan dengan baik dan sesuai. Tujuan dari pengujian *black box testing* adalah untuk menentukan kesalahan-kesalahan fungsi pada program. Adapun fungsi yang akan di uji.

##### a. Rencana Pengujian

Tabel 4.8 Rencana Pengujian

No	Akses	Kelas Uji	Butir Uji
1.	Admin	Login	Verifikasi Data
		Mengelola Data User	Tambah Data, Edit Data, Hapus Data
		Mengelola Data Transaksi	Tambah Data, Edit Data, Hapus Data, Cetak Data
		Mengelola Data Gallery	Tambah Data, Edit Data, Hapus Data
		Mengelola Data Harga Studio	Edit Data
2.	User	Melakukan Transaksi	Tambah Transaksi, Cetak Transaksi
		Mengubah Profil	Tampil Data, Edit Data

##### b. Implementasi Pengujian

###### 1) Pengujian Fitur *Login*

Tabel 4.9 Tabel Pengujian Fitur Login

Hak Akses	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Kenyataan
Admin dan User	<i>Input Username dan password benar</i>	Akan langsung menuju halaman <i>dashboard</i> atau halaman utama <i>website</i>	Sesuai Harapan
	<i>Username dan password belum diisi</i>	Akan muncul pesan <i>error</i>	Sesuai Harapan
	<i>Username belum diisi atau salah</i>	Akan muncul pesan <i>error</i>	Sesuai Harapan
	<i>Password belum diisi atau salah</i>	Akan muncul pesan <i>error</i>	Sesuai Harapan

Berdasarkan tabel 4.10 menjelaskan bahwa pengujian fungsi fitur *login* pada semua hak akses mendapatkan hasil sesuai harapan.

## 2) Pengujian Fitur *Dashboard*

Tabel 4.10 Tabel Pengujian Fitur Dashboard

Hak Akses	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Kenyataan
Admin, User	Klik <i>menu dashboard</i> atau halaman awal saat <i>login</i>	Akan muncul halaman <i>dashboard</i>	Sesuai Harapan

Berdasarkan tabel 4.10 menjelaskan bahwa pengujian fungsi fitur *dashboard* pada hak akses mendapatkan hasil sesuai harapan.

## 3) Pengujian Fitur Menambah Data User

Tabel 4.11 Tabel Pengujian Fitur Data User

Fungsi Yang Diuji	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Kenyataan
Tambah Data User	Klik tambah data pada halaman data user	Akan muncul <i>form</i> tambah data user	Sesuai Harapan
	<i>Input-an</i> sudah diisi	Data ditambahkan dan akan tampil di <i>view</i>	Sesuai Harapan
	<i>Input-an</i> salah satu atribut belum diisi	Akan muncul pesan <i>error</i>	Sesuai Harapan
	<i>Input-an</i> belum diisi semua	Akan muncul pesan <i>error</i>	Sesuai Harapan
<i>Edit</i> Data User	Klik tombol <i>edit</i> pada salah satu user	Akan tampil <i>form edit</i> data user	Sesuai Harapan
	<i>Input-an</i> sudah diisi	Data di- <i>update</i> dan akan tampil di daftar user	Sesuai Harapan
	<i>Input-an</i> salah satu atribut belum diisi	Akan muncul pesan <i>error</i>	Sesuai Harapan
	<i>Input-an</i> belum diisi semua	Akan muncul pesan <i>error</i>	Sesuai Harapan
Hapus Data User	Klik tombol hapus pada salah satu User	Akan muncul pesan konfirmasi	Sesuai Harapan
	Pilih Yakin	Data akan dihapus	Sesuai Harapan
	Pilih Batal	Data batal dihapus	Sesuai Harapan

Berdasarkan tabel 4.11 menjelaskan bahwa pengujian fungsi fitur data user pada hak akses admin yang meliputi tambah, *edit* dan hapus data user mendapatkan hasil sesuai harapan.

#### 4) Pengujian Fitur Menambah Data Transaksi

Tabel 4.12 Tabel Pengujian Fitur Data Transaksi

<b>Fungsi Yang Diuji</b>	<b>Skenario Pengujian</b>	<b>Hasil Yang Diharapkan</b>	<b>Hasil Kenyataan</b>
Tambah Data Transaksi	Klik tambah data pada halaman data transaksi	Akan muncul <i>form</i> tambah data admin	Sesuai Harapan
	<i>Input-an</i> sudah diisi	Data ditambahkan dan akan tampil di <i>view</i>	Sesuai Harapan
	<i>Input-an</i> salah satu atribut belum diisi	Akan muncul pesan <i>error</i>	Sesuai Harapan
	<i>Input-an</i> belum diisi semua	Akan muncul pesan <i>error</i>	Sesuai Harapan
<i>Edit</i> Data Transaksi	Klik tombol <i>edit</i> pada salah satu transaksi	Akan tampil <i>form edit</i> data transaksi	Sesuai Harapan
	<i>Input-an</i> sudah diisi	Data di- <i>update</i> dan akan tampil di daftar transaksi	Sesuai Harapan
	<i>Input-an</i> salah satu atribut belum diisi	Akan muncul pesan <i>error</i>	Sesuai Harapan
	<i>Input-an</i> belum diisi semua	Akan muncul pesan <i>error</i>	Sesuai Harapan
Hapus Data Transaksi	Klik tombol hapus pada salah satu transaksi	Akan muncul pesan konfirmasi	Sesuai Harapan
	Pilih Yakin	Data akan dihapus	Sesuai Harapan
	Pilih Batal	Data batal dihapus	Sesuai Harapan

Berdasarkan tabel 4.12 menjelaskan bahwa pengujian fungsi fitur data transaksi pada hak akses admin yang meliputi tambah, *edit* dan hapus data transaksi mendapatkan hasil sesuai harapan.

5) Pengujian Fitur Menambah Data *Portofolio*

Tabel 4.13 Tabel Pengujian Fitur Data Portofolio

<b>Fungsi Yang Diuji</b>	<b>Skenario Pengujian</b>	<b>Hasil Yang Diharapkan</b>	<b>Hasil Kenyataan</b>
Tambah Data <i>Portofolio</i>	Klik tambah data pada halaman data foto	Akan muncul <i>form</i> tambah data foto	Sesuai Harapan
	<i>Input</i> -an sudah diisi	Data ditambahkan dan akan tampil di <i>view</i>	Sesuai Harapan
	<i>Input</i> -an salah satu atribut belum diisi	Akan muncul pesan <i>error</i>	Sesuai Harapan
	<i>Input</i> -an belum diisi semua	Akan muncul pesan <i>error</i>	Sesuai Harapan
<i>Edit</i> Data <i>Portofolio</i>	Klik tombol <i>edit</i> pada salah satu foto	Akan tampil <i>form edit</i> data foto	Sesuai Harapan
	<i>Input</i> -an sudah diisi	Data di- <i>update</i> dan akan tampil di daftar foto	Sesuai Harapan
	<i>Input</i> -an salah satu atribut belum diisi	Akan muncul pesan <i>error</i>	Sesuai Harapan
	<i>Input</i> -an belum diisi semua	Akan muncul pesan <i>error</i>	Sesuai Harapan
Hapus Data <i>Portofolio</i>	Klik tombol hapus pada salah satu foto	Akan muncul pesan konfirmasi	Sesuai Harapan
	Pilih Yakin	Data akan dihapus	Sesuai Harapan
	Pilih Batal	Data batal dihapus	Sesuai Harapan

Berdasarkan tabel 4.13 menjelaskan bahwa pengujian fungsi fitur data *gallery* pada hak akses admin yang meliputi tambah, *edit* dan hapus data foto mendapatkan hasil sesuai harapan.

## 6) Pengujian Fitur Mengubah Data Harga Produk

Tabel 4.14 Tabel Pengujian Fitur Data Harga Produk

<b>Fungsi Yang Diuji</b>	<b>Skenario Pengujian</b>	<b>Hasil Yang Diharapkan</b>	<b>Hasil Kenyataan</b>
<i>Edit</i> Data Harga Produk	Klik tombol <i>edit</i> pada salah satu harga produk	Akan tampil <i>form edit</i> data harga produk	Sesuai Harapan
	<i>Input</i> -an sudah diisi	Data di- <i>update</i> dan akan tampil di daftar harga produk	Sesuai Harapan

Tabel 4.14 Tabel Pengujian Fitur Data Harga Produk (Lanjutan)

<b>Fungsi Yang Diuji</b>	<b>Skenario Pengujian</b>	<b>Hasil Yang Diharapkan</b>	<b>Hasil Kenyataan</b>
<i>Edit Data Harga Produk</i>	<i>Input-an salah satu atribut belum diisi</i>	Akan muncul pesan <i>error</i>	Sesuai Harapan
	<i>Input-an belum diisi semua</i>	Akan muncul pesan <i>error</i>	Sesuai Harapan

Berdasarkan tabel 4.14 menjelaskan bahwa pengujian fungsi fitur data harga produk pada hak akses admin yaitu edit data harga produk mendapatkan hasil sesuai harapan.

