[BAB III PEMBAHASAN MATERI 8](#_Toc121248788)

[3.1 Definisi *Web* Aplikasi 8](#_Toc121248789)

[3.2 Teknologi Pembangun Aplikasi Berbasis *Web* 8](#_Toc121248790)

[3.2.1 Hyper Text Makrup Language (HTML) 8](#_Toc121248791)

[3.2.2 Cascading Style Sheets (CSS) 8](#_Toc121248792)

[3.2.3 PHP Hypertext Preprocessor (PHP) 8](#_Toc121248793)

[3.2.4 Javascript 8](#_Toc121248794)

[3.2.5 MySQL 8](#_Toc121248795)

[3.2.6 Laravel 8](#_Toc121248796)

[3.2.7 Composer 8](#_Toc121248797)

[3.3 Prosedur Pembuatan Sistem 8](#_Toc121248798)

[3.4 Implementasi Pembuatan Aplikasi Rental Buku 8](#_Toc121248799)

[3.5 Bukti Hasil Pekerjaan 8](#_Toc121248800)

[3.5.1 Struktur Table 8](#_Toc121248801)

[3.5.2 Model 8](#_Toc121248802)

[3.5.3 Migration 8](#_Toc121248803)

[3.5.4 Controller 8](#_Toc121248804)

[3.5.5 View 8](#_Toc121248805)

[3.5.6 Tampilan Default Header 8](#_Toc121248806)

[3.5.7 Tampilan Default Daftar Buku 8](#_Toc121248807)

[3.5.8 Tampilan Login Aplikasi 8](#_Toc121248808)

[3.5.9 Tampilan Register Aplikasi 8](#_Toc121248809)

[3.5.10 Tampilan Dashboard 8](#_Toc121248810)

[3.5.11 Tampilan Kategori 8](#_Toc121248811)

[3.5.12 Tampilan Tambah Kategori 8](#_Toc121248812)

[3.5.13 Tampilan Manajemen Buku 8](#_Toc121248813)

[3.5.14 Tampilan Tambah Buku 16](#_Toc121248814)

[3.5.15 Tampilan User Baru 16](#_Toc121248815)

[3.5.16 Tampilan Approve User Baru 17](#_Toc121248816)

[3.5.17 Tampilan User Approved 17](#_Toc121248817)

[3.5.18 Tampilan Form Peminjaman Buku 18](#_Toc121248818)

[3.5.19 Tampilan Form Pengembalian Buku 18](#_Toc121248819)

[3.5.20 Tampilan Peminjaman dan Pengembalian Buku 19](#_Toc121248820)

[3.6 Kendala Implementasi 19](#_Toc121248821)

[3.7 Pemecahan Atas Kendala 19](#_Toc121248822)

# PEMBAHASAN MATERI

## Definisi *Web* Aplikasi

Aplikasi berbasis *web* merupakan sebuah program atau perangkat lunak yang di akses melalui internet dengan menggunakan *web browser*. Pada awalnya aplikasi *web* dibangun dengan hanya menggunakan bahasa markah yang disebut HTML (*Hyper Text Markup Langauge*). Pada perkembangan berikutnya, sejumlah *script* dan objek dikembangkan untuk memperluas kemampuan HTML seperti PHP dan ASP pada *script* dan Apllet pada objek.

Aplikasi *web* dapat dibagi menjadi dua jenis yaitu aplikasi *web* statis dan dinamis. *Web* statis dibentuk dengan menggunakan HTML. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perkembangan yang terjadi. Kelemahan ini diatasi oleh model aplikasi *web* dinamis. Pada aplikasi *web* dinamis, perubahan informasi dalam halaman *web* dilakukan tanpa perubahan program tetapi melalui perubahan data. Sebagai implementasi, aplikasi *web* dapat dikoneksikan ke basis data sehingga perubahan informasi dapat dilakukan oleh operator dan tidak menjadi tanggung- jawab dari *web* master. Arsitektur aplikasi *web* meliputi *client*, *web* *server*, *moddleware* dan basis data. *Client*  berinteraksi dengan *web* *server*. Secara internal, *web* *server* berkomunikasi dengan *moddleware* dan *moddleware* yang berkomunikasi dengan basis data. Contoh *moddleware* adalah PHP dan ASP. Pada mekanisme aplikasi *web* dinamis, terjadi tambahan proses yaitu *server* menerjemahkan kode PHP ataupun ASP menjadi kode HTML. Kode PHP diterjemahkan oleh mesin PHP yang akan diterima oleh *client* melalui sebuah perangkat komputer.

Keunggulan dari aplikasi berbasis *web* ringan serta dapat diakses dengan cepat melalui *browser* dan koneksi internet atau intranet ke *server*.

## Teknologi Pembangun Aplikasi Berbasis *Web*

Pada umumnya untuk membangun sebuah aplikasi berbasis *web* menggunakan teknologi dan beberapa bahasa pemerograman sebagai berikut:

### Hyper Text Makrup Language (HTML)

HTML memungkinkan seorang pengguna dapat membuat dan menyusun bagian *heading*, paragraf, *link* atau tautan, dan *blockquote* untuk halaman sebuah *website*.

Halaman *website* yang kita lihat dan kita baca disusun dengan menggunakan bahasa ini dan kemudian diterjemahkan oleh komputer agar dapat dipahami oleh penggunanya. HTML disusun dengan kode dan simbol tertentu yang dimasukkan kedalam sebuah file atau dokumen. Jadi setiap membuka *website* apapun dengan menggunakan *browser* maka *web* tersebut dibuat dengan struktur kode menggunakan HTML.

### Cascading Style Sheets (CSS)

*Cascading Style Sheets* (CSS) merupakan salah satu kode pemerograman yang bertujuan untuk menghias dan mengatur gaya tampilan atau layout halaman *web* supaya terlihat menarik untuk dilihat oleh pengguna *web* tersebut. CSS biasanya selalu dikaitkan dengan HTML, karena keduanya memang saling melengkapi.

### PHP Hypertext Preprocessor (PHP)

*PHP Hypertext Prepocessor* (PHP) merupakan bahasa pemerograman *web* *server-side* yang bersifat *open source* yang proses penerjemahannya dimengerti oleh computer secara langsung pada saat baris kode dijalankan. PHP itu sendiri merupakan singkatan dari *Personal Home Page Tools. Script* ini akan membuat suatu aplikasi yang dapat diintegrasikan ke dalam HTML sehingga suatu halaman HTML tidak lagi bersifat statis, namun bersifat dinamis.

### Javascript

*Javascript* merupakan bahasa pemerograman bersifat *client-side* sehingga dapat diolah langsung di *browser* tanpa harus terhubung ke *server*  terlebih dahulu, yang berfungsi merubah tampilan *website* menjadi lebih interaktif.

### MySQL

MySQL merupakan *database* yang menghubungkan *script* PHP menggunakan perintah query dan *escaps character* yang sama dengan PHP. MySQL adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas).

MySQL masuk ke dalam jenis RDBMS *(Relational Database Management System*). Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada aplikasi database ini. Contohnya di dalam MySQL sebuah database terdapat satu atau beberapa tabel.

SQL sendiri merupakan suatu bahasa yang dipakai di dalam pengambilan data pada *relational database* atau *database* yang terstruktur. Jadi MySQL adalah database management system yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server*.

### Laravel

Laravel adalah kerangka kerja aplikasi *web* berbasis PHP yang sumber terbuka, menggunakan konsep Model-View-Controller. Laravel berada dibawah lisensi MIT, dengan menggunakan GitHub sebagai tempat berbagi kode.

### Composer

Composer adalah alat *dependency manager* untuk bahasa pemrograman PHP. Dengan kata lain, Composer adalah aplikasi yang diinstal ke perangkat untuk memfasilitasi developer menggunakan library *open source* milik orang lain ke dalam projek yang sedang dibangun “*Dependency Manager*.” Dalam sebuah project PHP, penggunaan library sangat memudahkan dalam proses penulisan kode. Namun terkadang, library satu dengan yang lainnya saling membutuhkan agar bisa digunakan. Hal ini disebut dependensi atau ketergantungan. Di sinilah peran Composer untuk menghubungkan project PHP dan Laravel dengan *library* eksternal yang dibutuhkan.

## Prosedur Pembuatan Sistem

Model pengembangan perangkat lunak atau sistem yang digunakan adalah metode *waterfall.* Proses pengembangan waterfall model terdiri dari analisis, desain, pembuatan program, pengujian dan pemeliharaan.

Berikut adalah penjelasan mengenai analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support)* atau pemeliharaan (*maintence).*

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mengekspresikan kebutuhan perangkat lunak afar dapat dipahami perangkat lunak seprti apayang dibutuhkan oleh user. Spesifikasikan kebutuhan perangkat lunak pada tahap ini perlu untuk di dokumentasikan.

2. Desain

Desain perangkat lunak adalah proses langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan kerepresentasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

3. Pembuatan kode

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tehnik ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada peranagkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (error) dan memastikan keluaran yang dihasilkan sesuai dengan diinginkan.

5. Pendukung (*support)* atau pemeliharaan (*Maintence)*

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karna adanya analisis kesalahan yang muncul dan tidak terdeteksi saat peujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak. Berikut merupakan gambar dari *waterfall model.*



Gambar 3.3.1 *Waterfall Model*

## Implementasi Pembuatan Aplikasi Rental Buku

Penulis membuat sebuah sistem aplikasi rental buku dengan menggunakan *framework* Laravel. Adapun *tools* dan bahasa pemerograman yang digunakan untuk membuat sistem tersebut sebagai berikut:

A. Peralatan

1. Laptop

2. Mouse

3. Wi-Fi

B. Bahasa Pemerograman dan Aplikasi Pendukung

1. HTML

2. CSS

3. PHP

4. Javascript

5. *Framework* Laravel

6. Bootstrap

7. JQuery & Ajax

8. MySQL

9. Laragon

10. PhpMyadmin

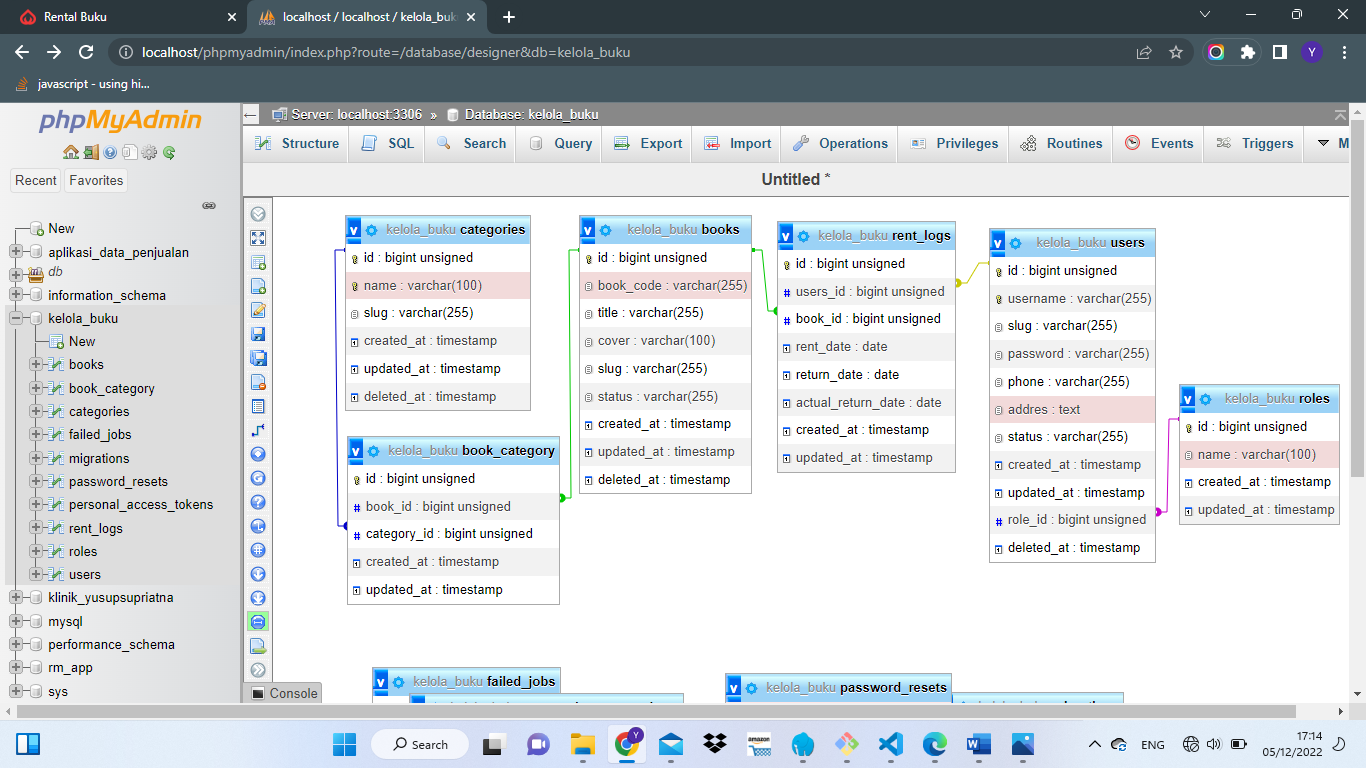
11. Git Bash

12. VSCODE

13. Composer

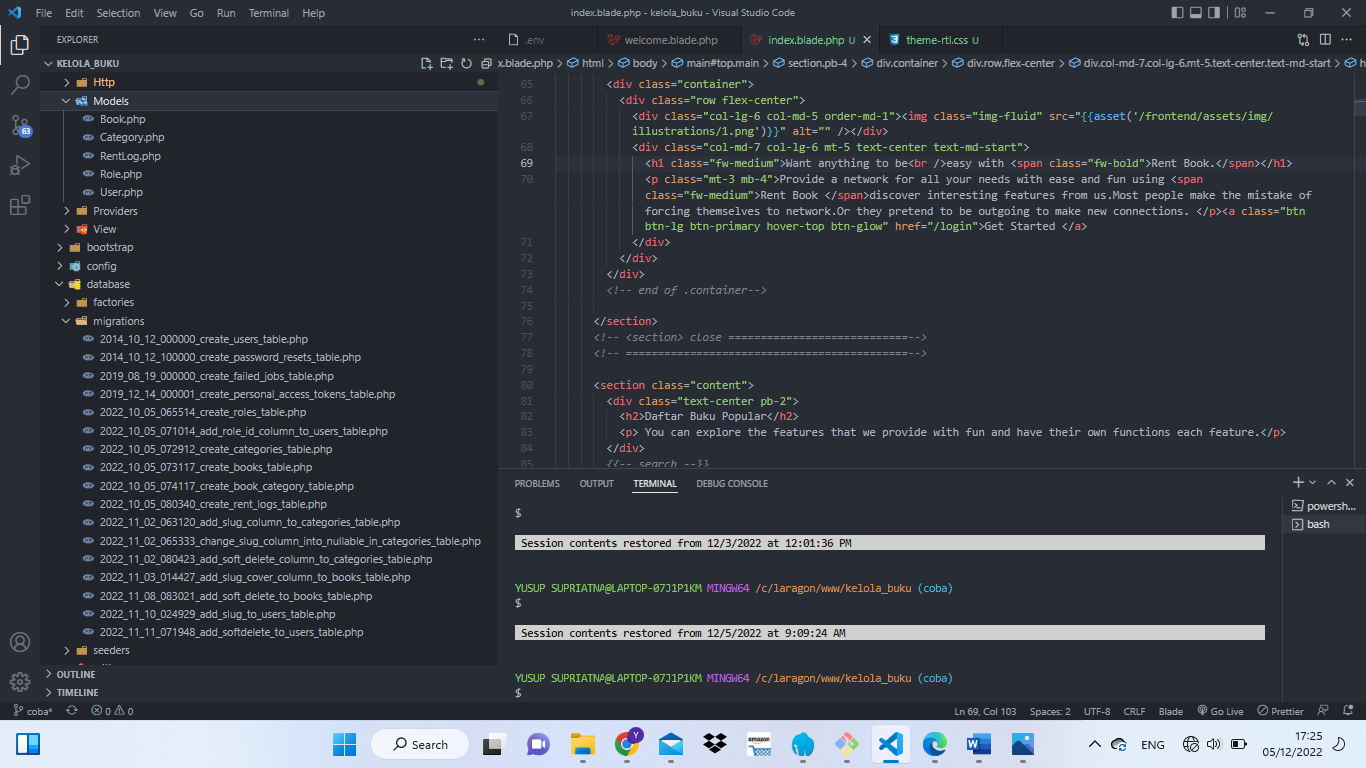
## Bukti Hasil Pekerjaan

### Struktur Table



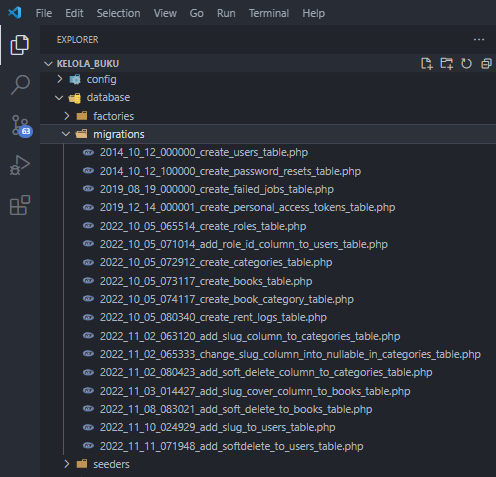
Gambar 3.5.1 Struktur Table Aplikasi Rental Buku

### Model



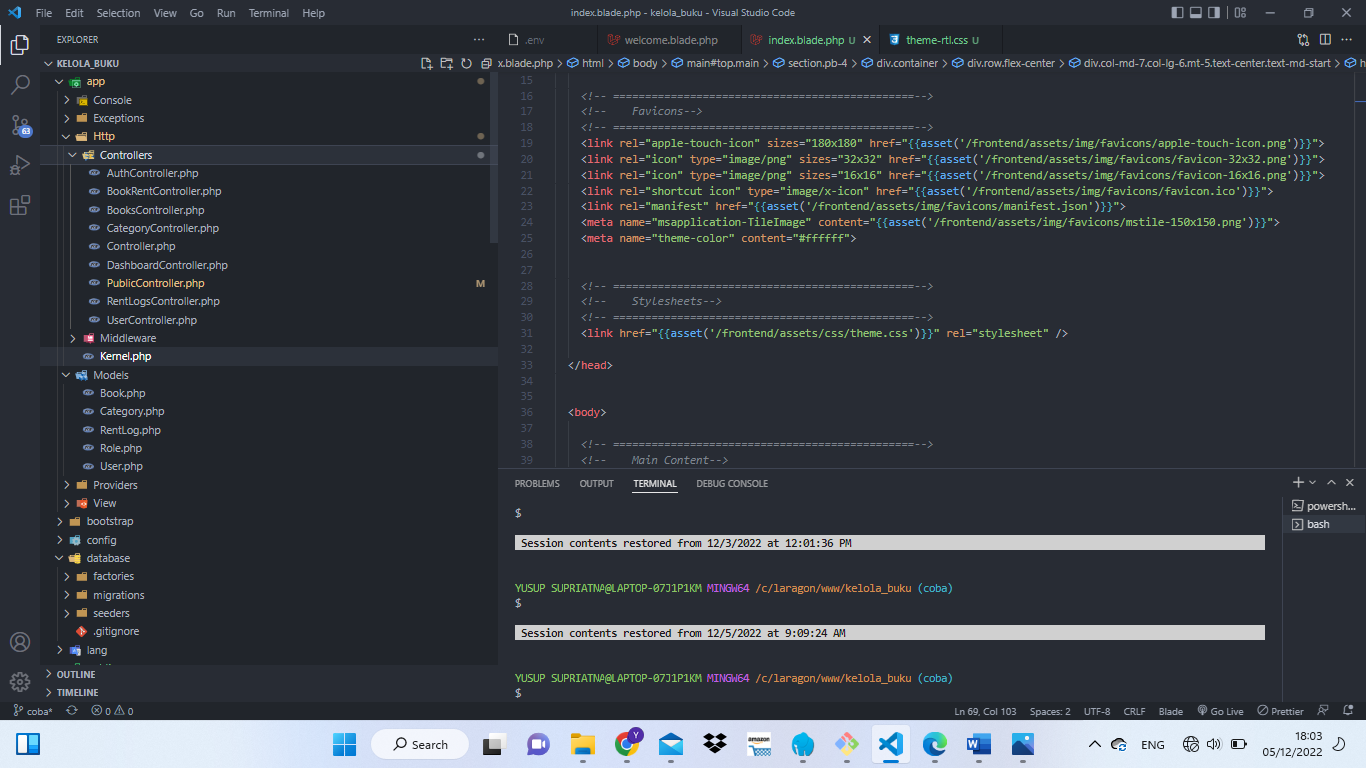
Gambar 3.5.2 Model Aplikasi Rental Buku

### Migration



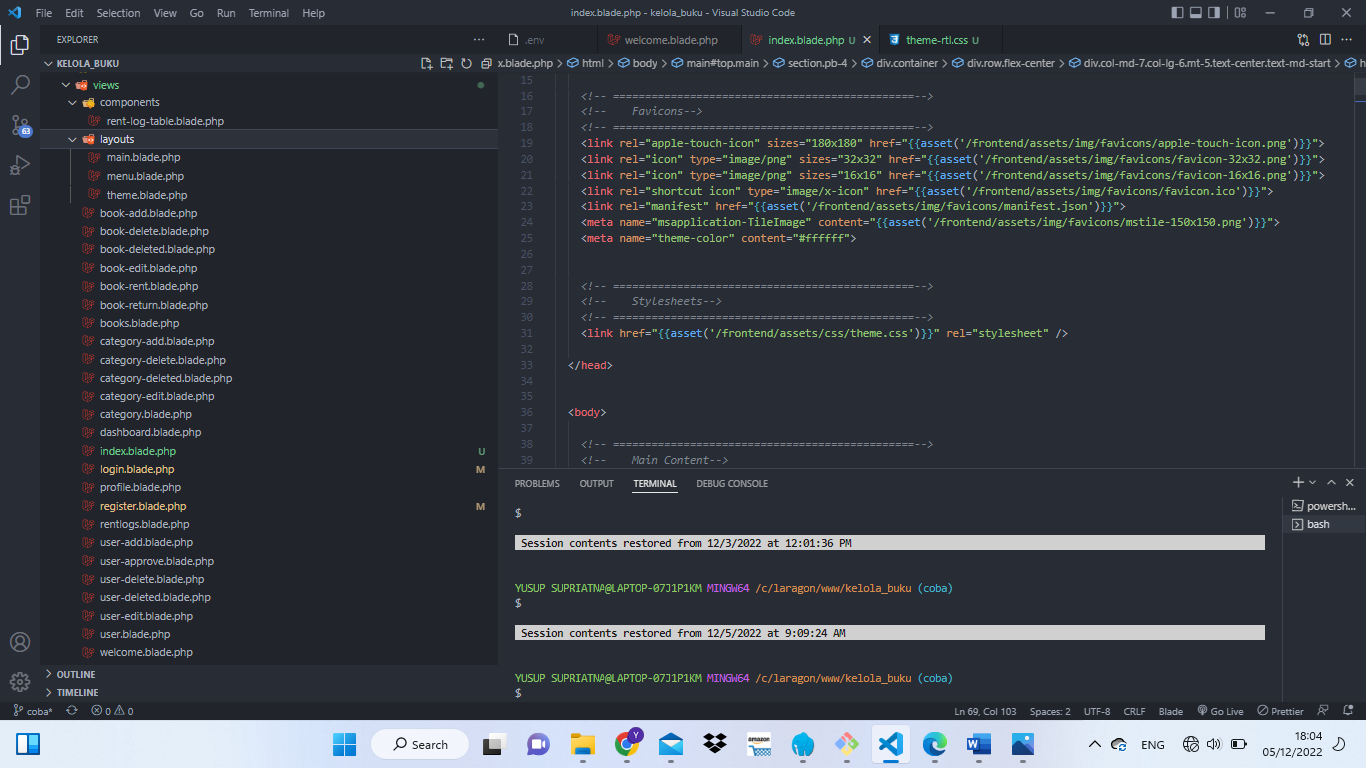
Gambar 3.5.3 Migration Aplikasi Rental Buku

### Controller



Gambar 3.5.4 Controller Aplikasi Rental Buku

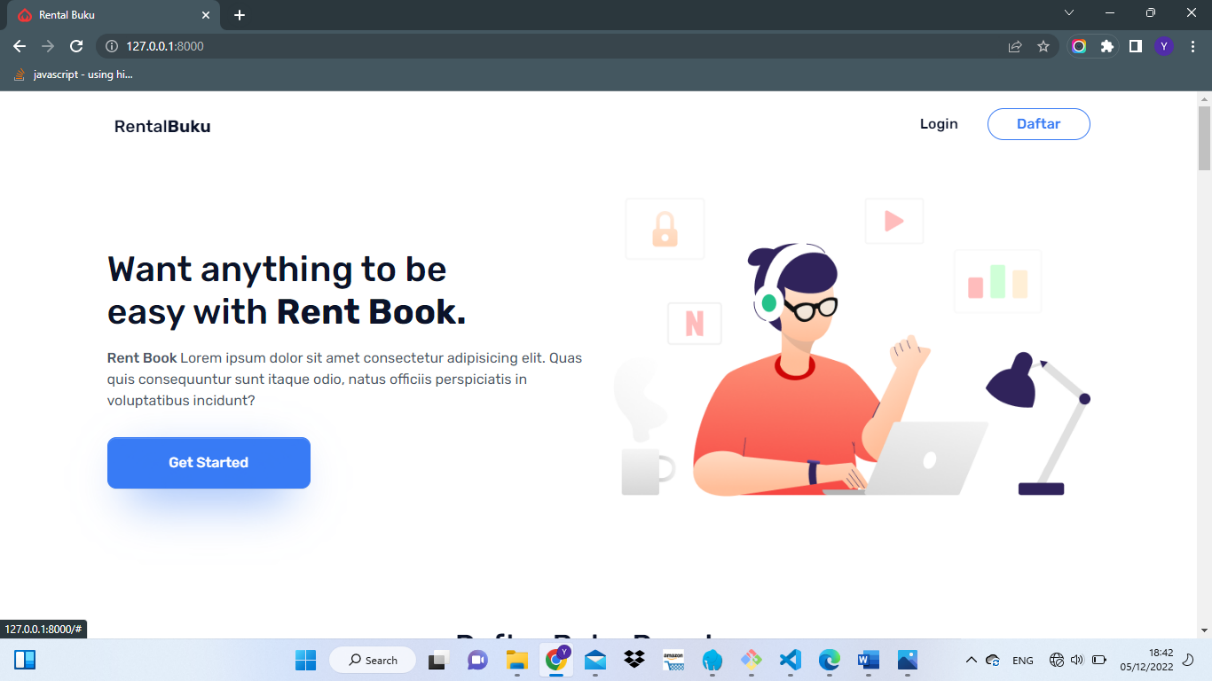
### View



Gambar 3.5.5 View Aplikasi Rental Buku

### Tampilan Default Header

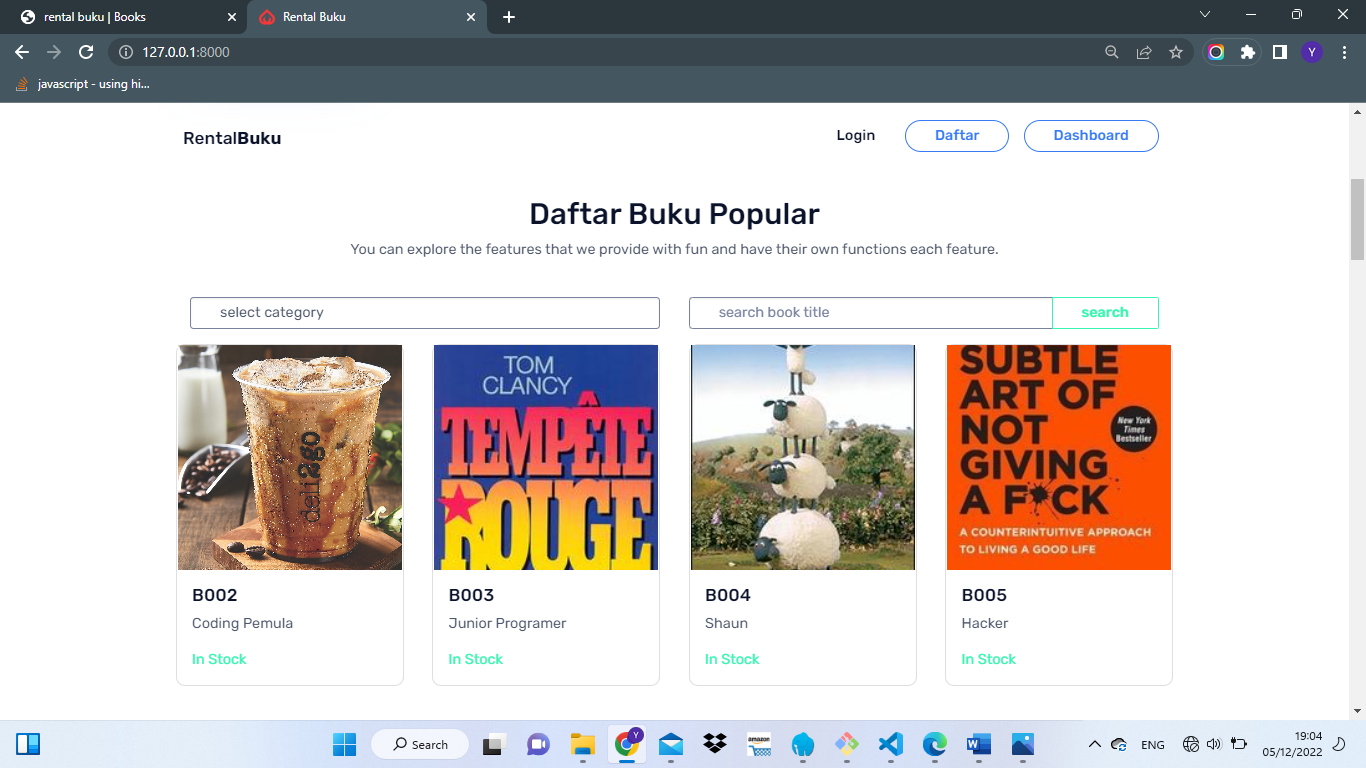
Tampilan default atau tampilan *home* pada saat *web*site pertama kali di akses.



Gambar 3.5.6 Tampilan Default Header Aplikasi Rental Buku

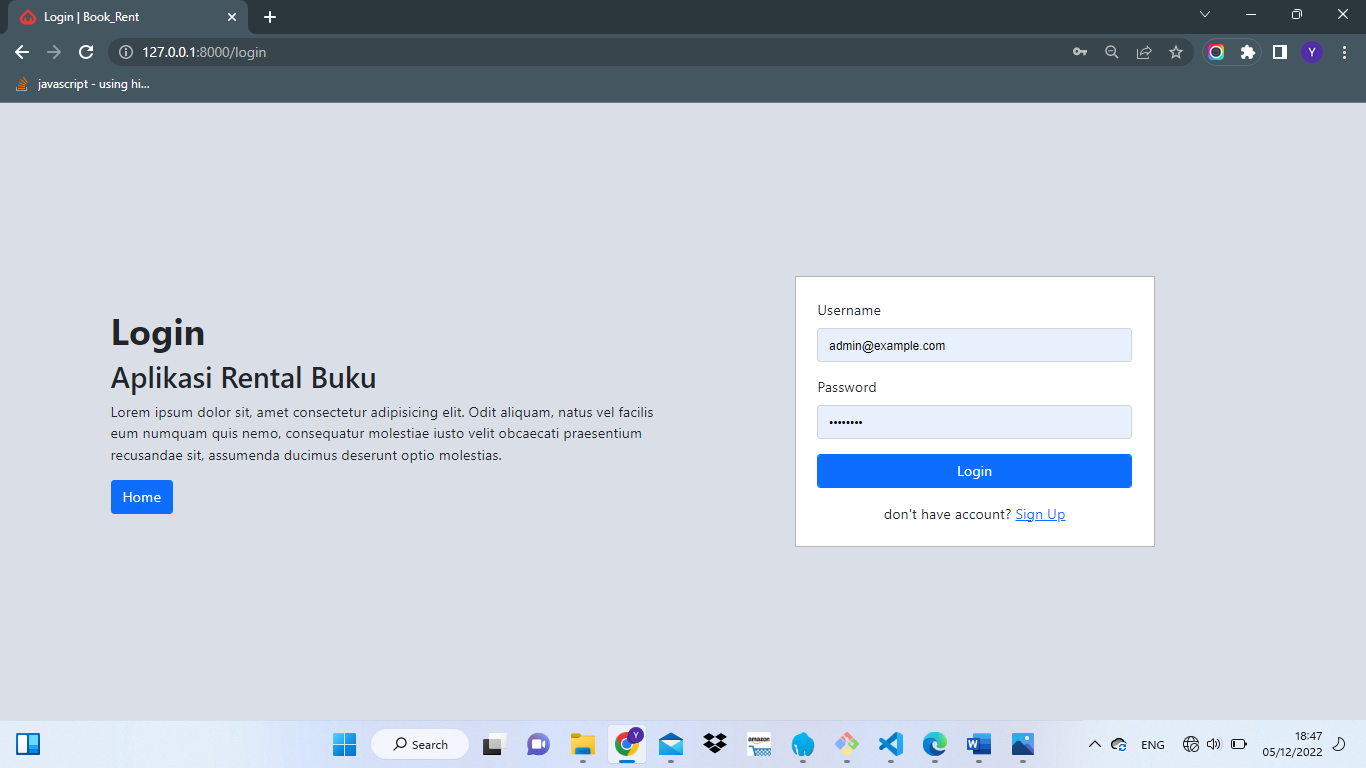
### Tampilan Default Daftar Buku

Tampilan daftar buku yang dibuat secara dinamis di bagian *back-end.*



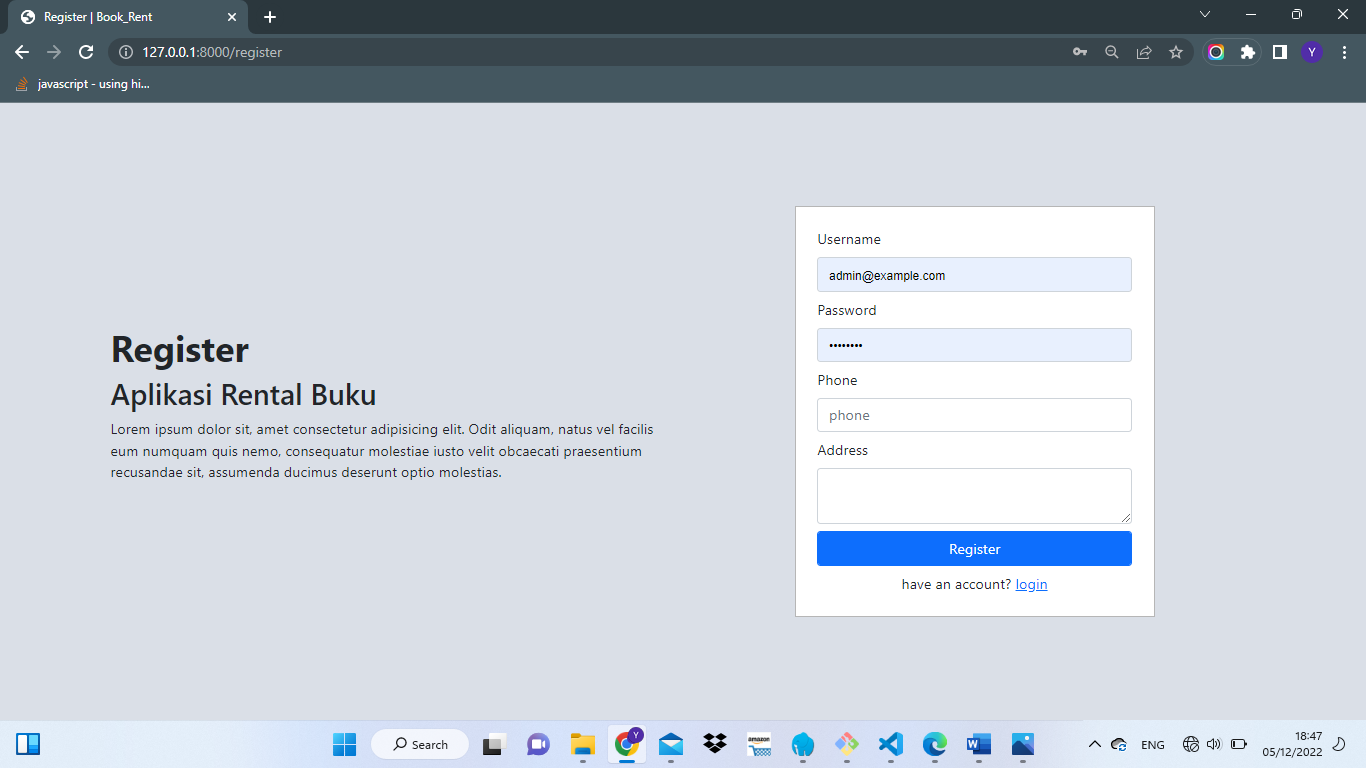
Gambar 3.5.7 Tampilan Default Daftar Buku

### Tampilan Login Aplikasi



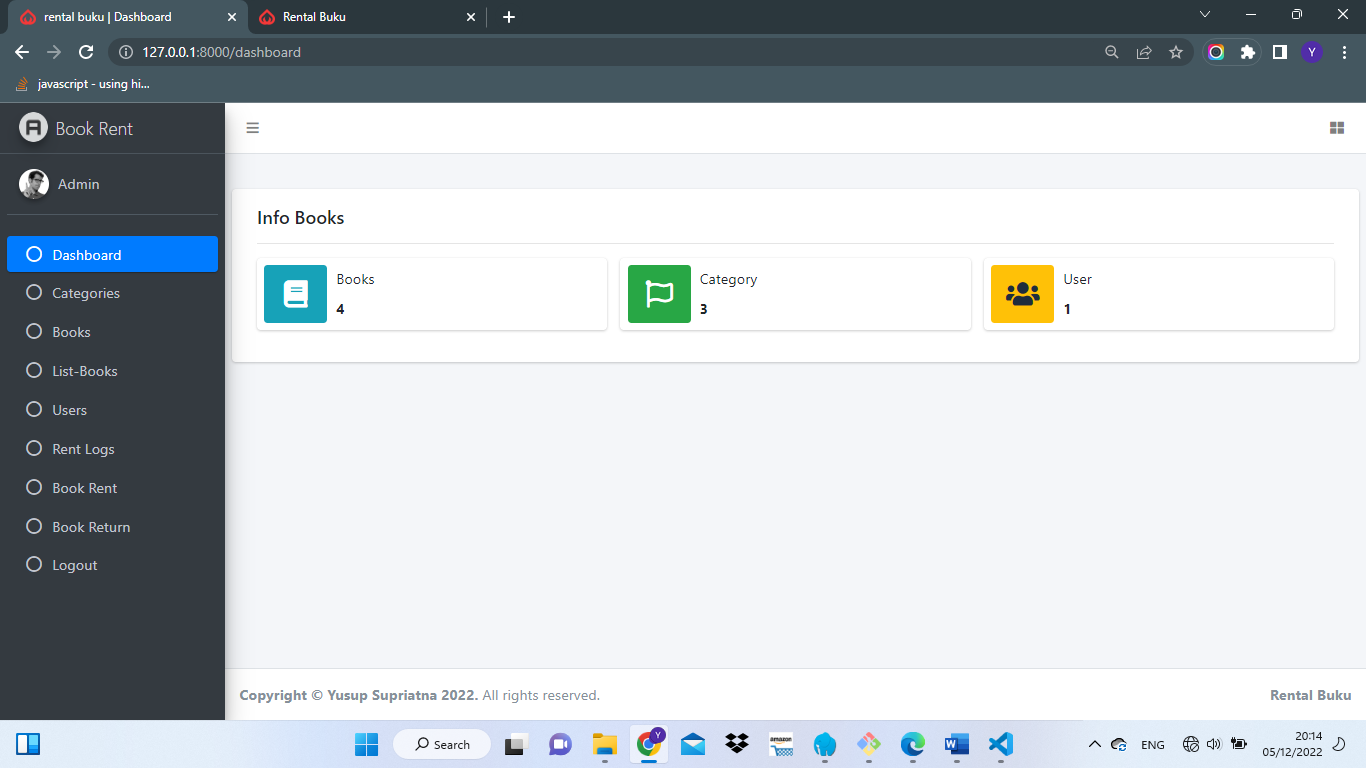
Gambar 3.5.8 Tampilan Login Aplikasi Rental Buku

### Tampilan Register Aplikasi



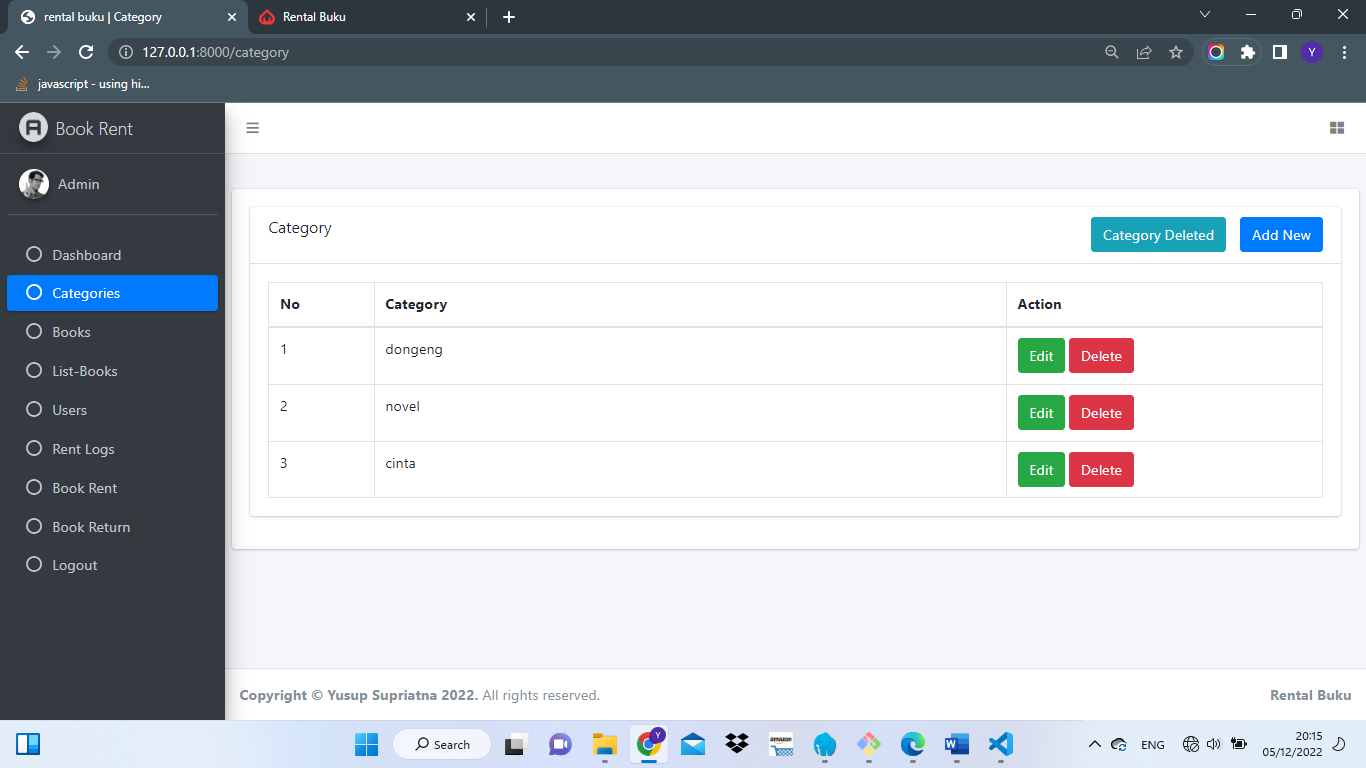
Gambar 3.5.9 Tampilan Registrasi Aplikasi Rental Buku

### Tampilan Dashboard



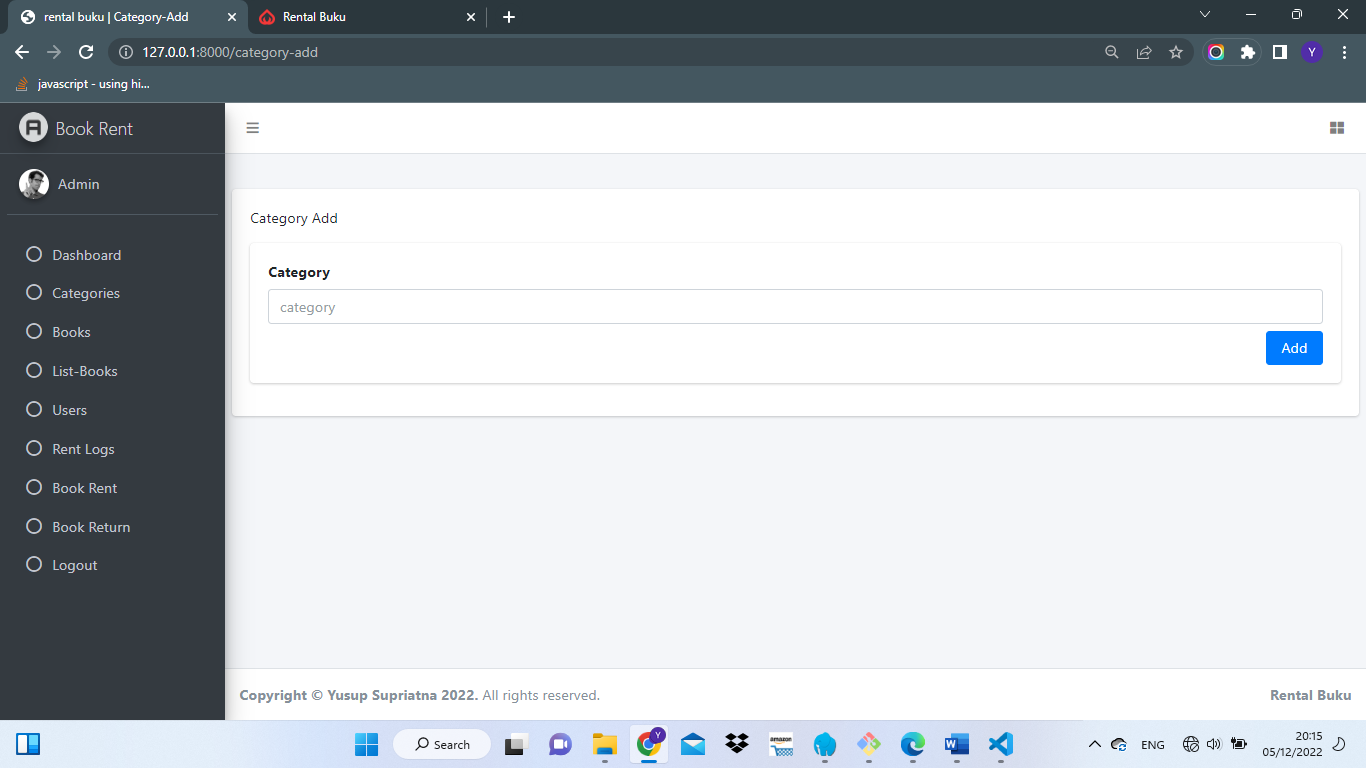
Gambar 3.5.10 Tampilan Dashboard Aplikasi Rental Buku

### Tampilan Kategori



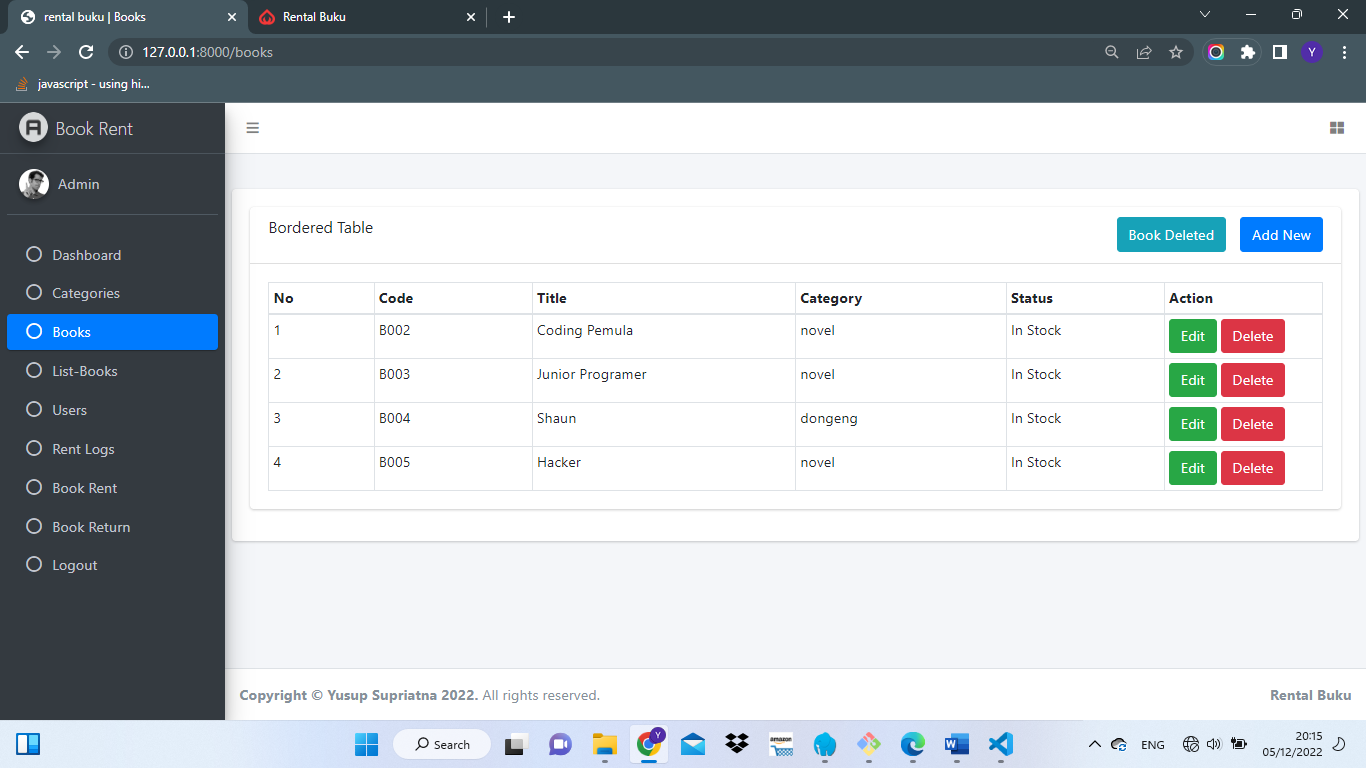
Gambar 3.5.11 Tampilan Kategori Buku

### Tampilan Tambah Kategori



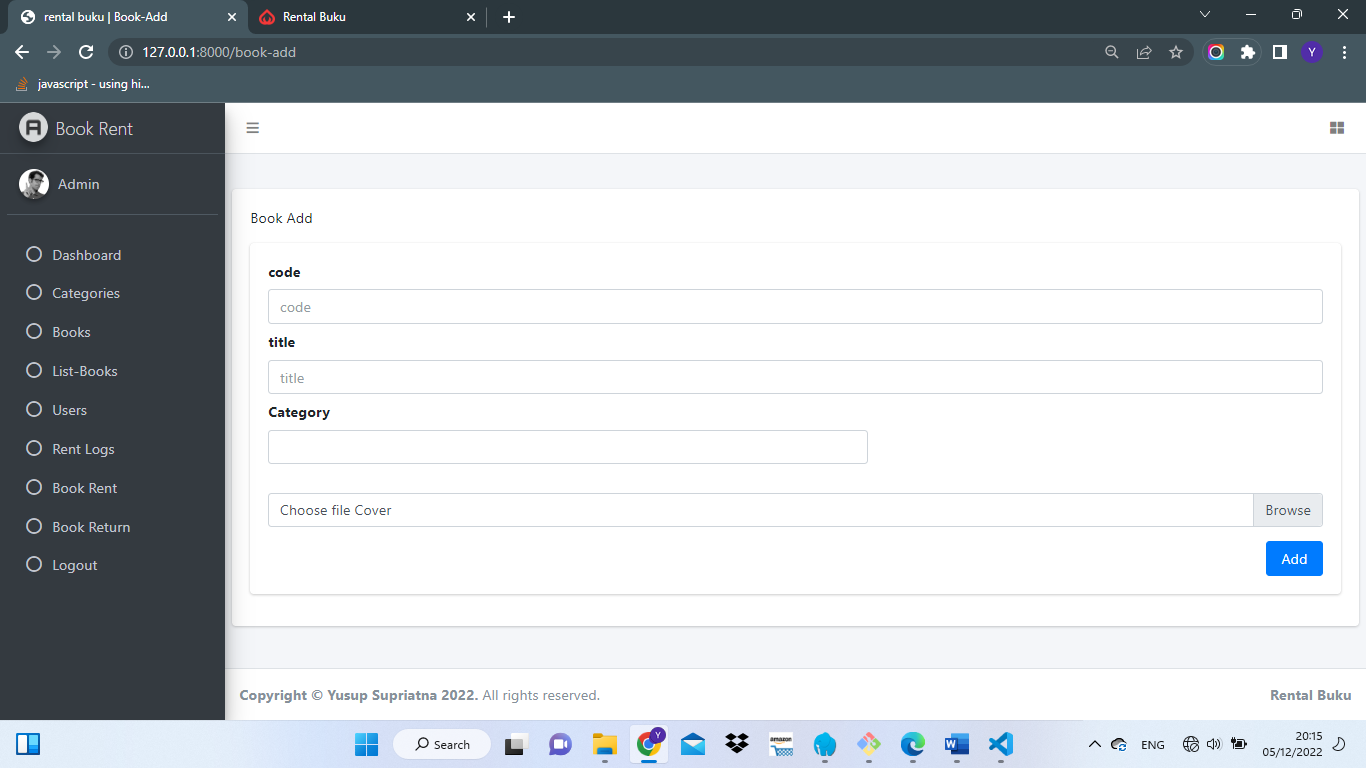
Gambar 3.5.12 Tampilan Tambah Kategori Buku

### Tampilan Manajemen Buku



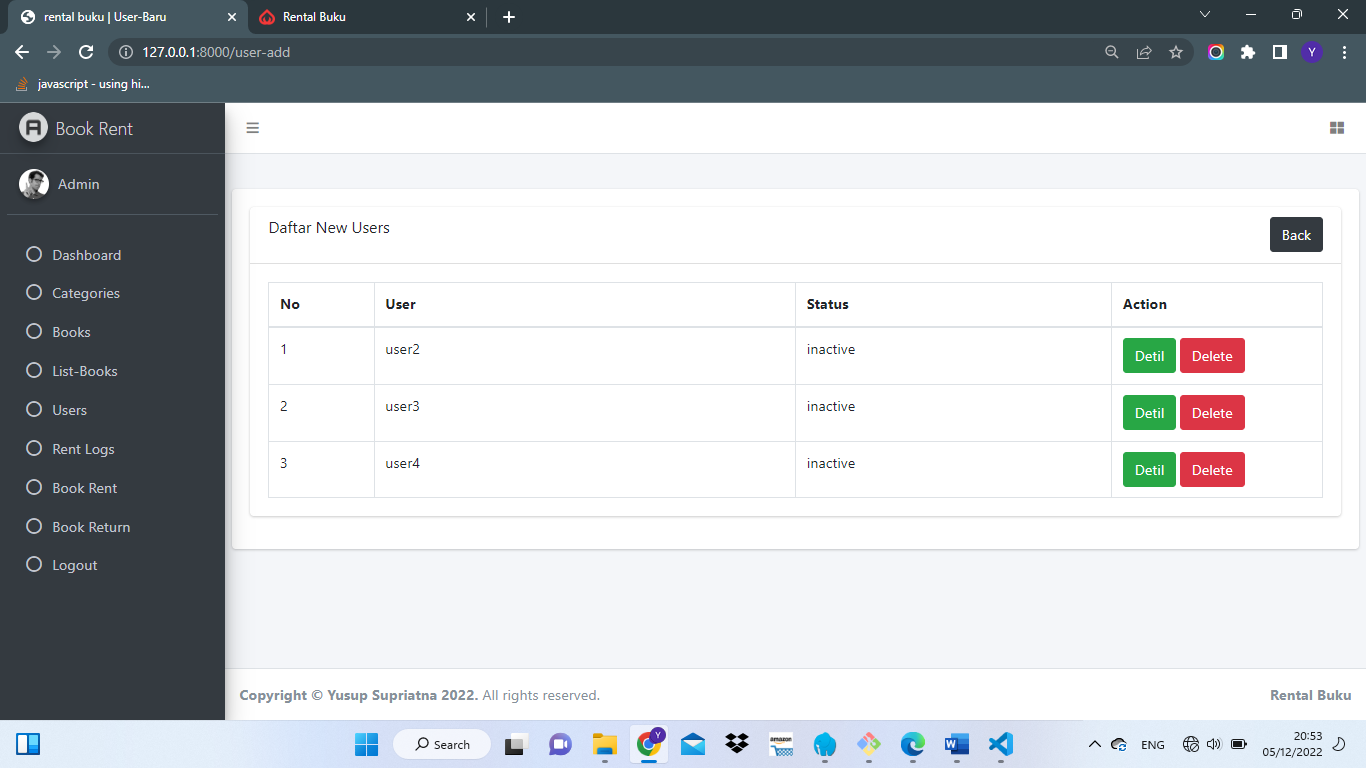
Gambar 3.5.13 Tampilan Manajemen Buku

### Tampilan Tambah Buku



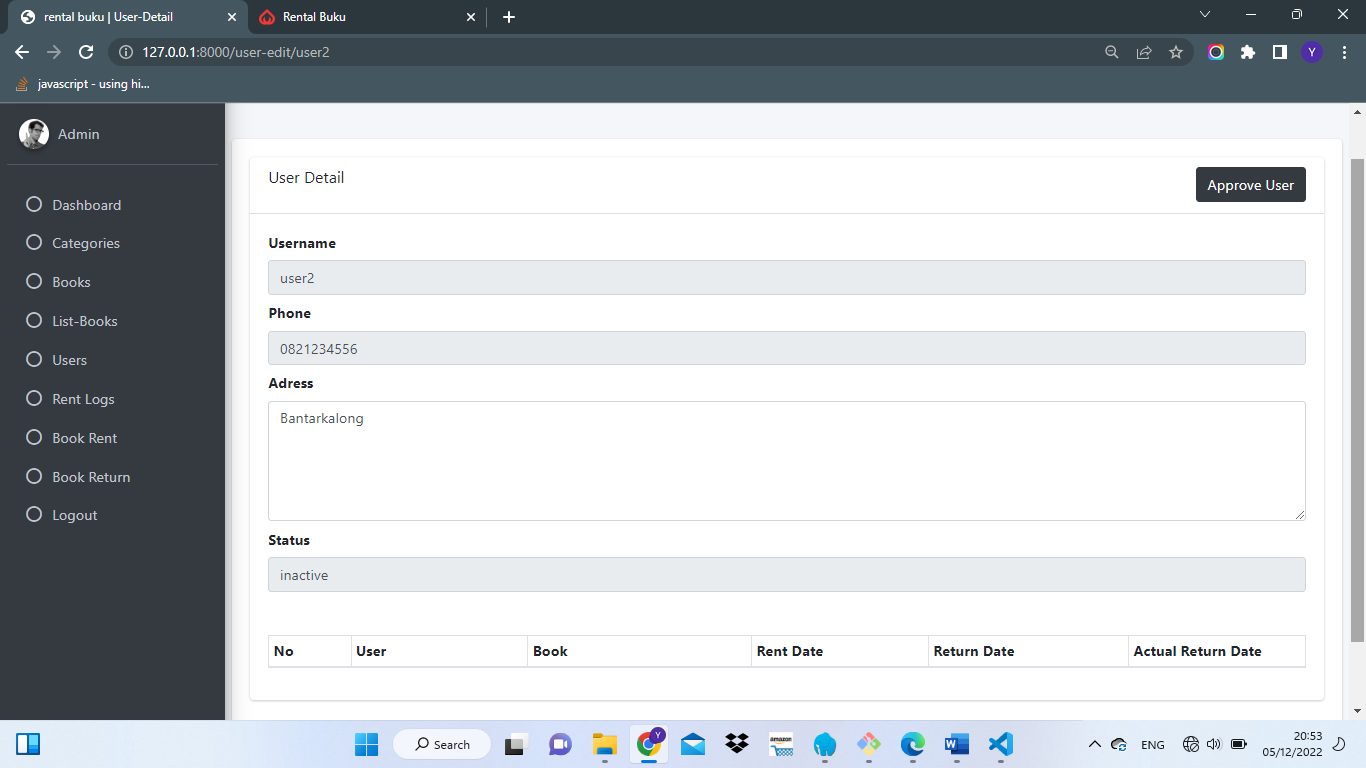
Gambar 3.5.14 Tampilan Tambah Buku

### Tampilan User Baru



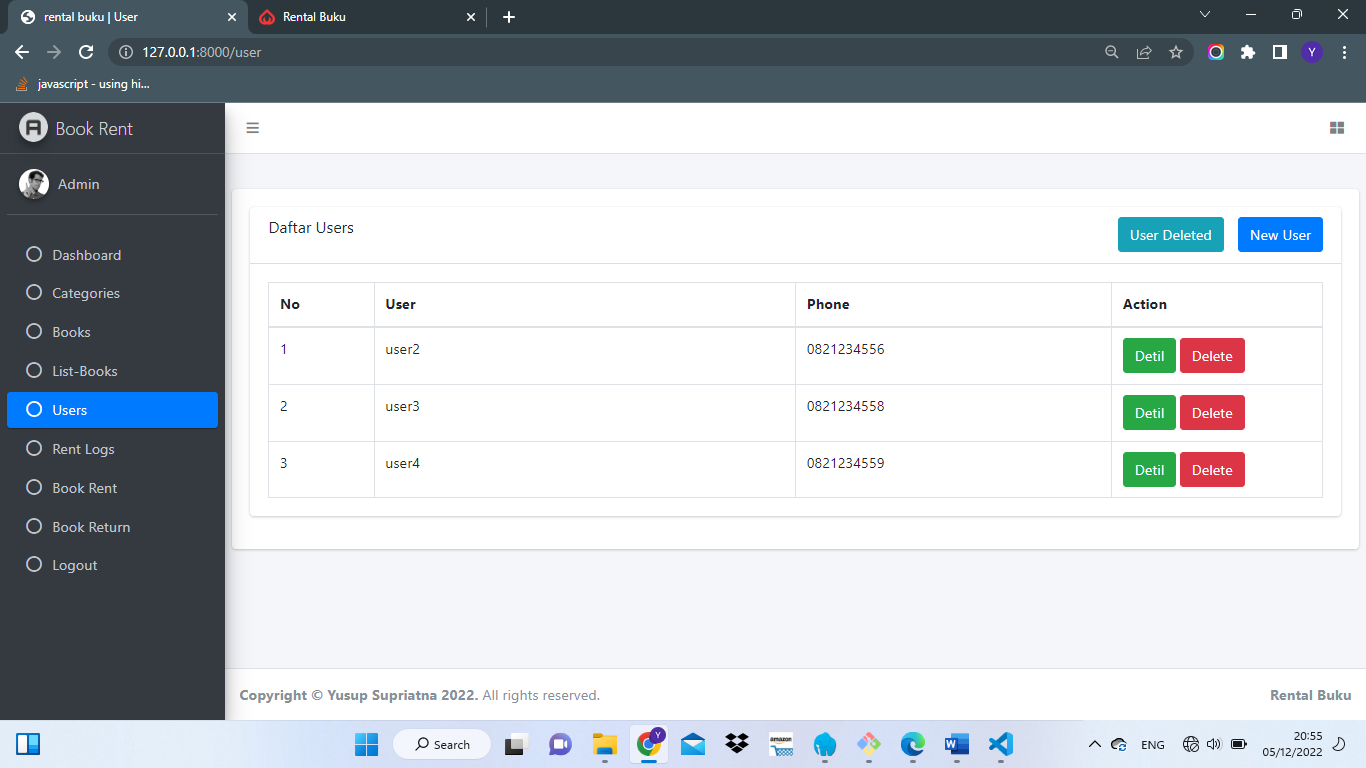
Gambar 3.5.15 Tampilan User Baru Aplikasi Rental Buku

### Tampilan Approve User Baru



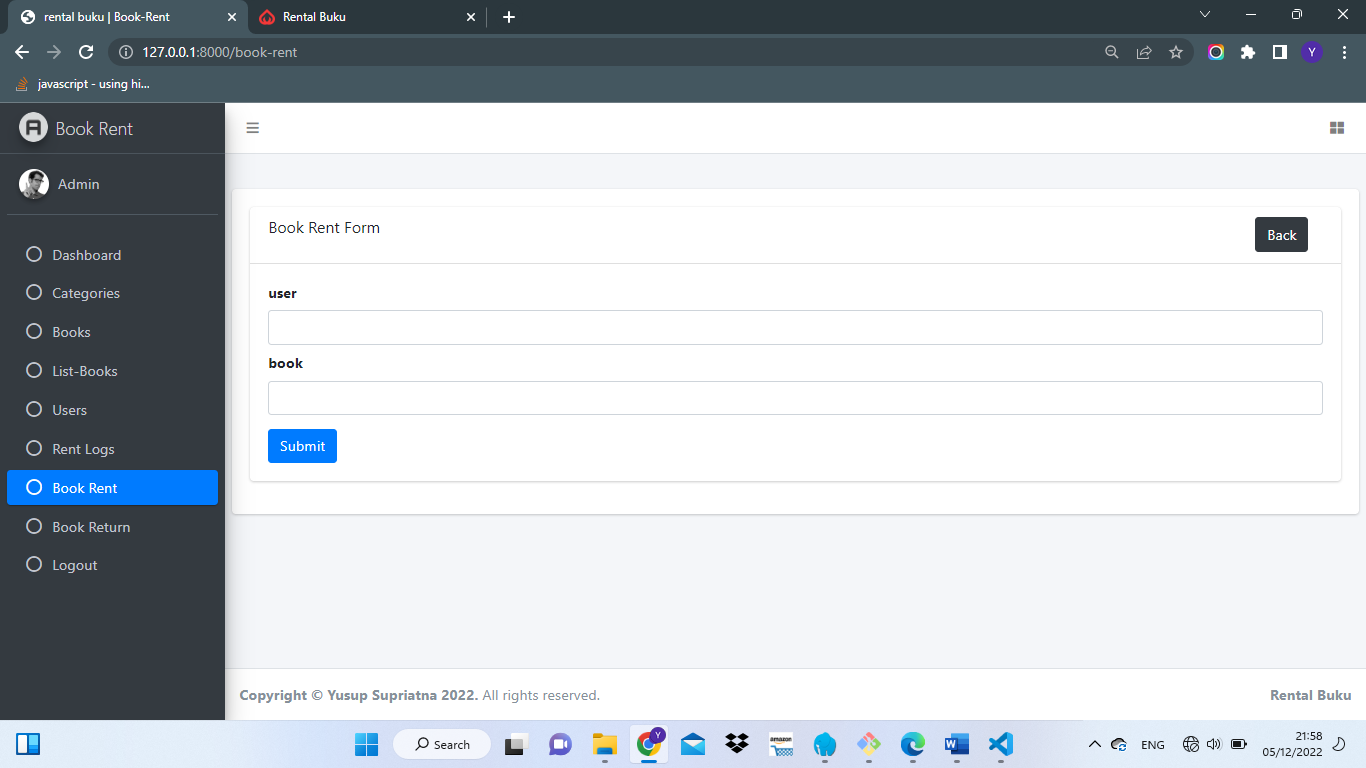
Gambar 3.5.16 Tampilan Approve User

### Tampilan User Approved



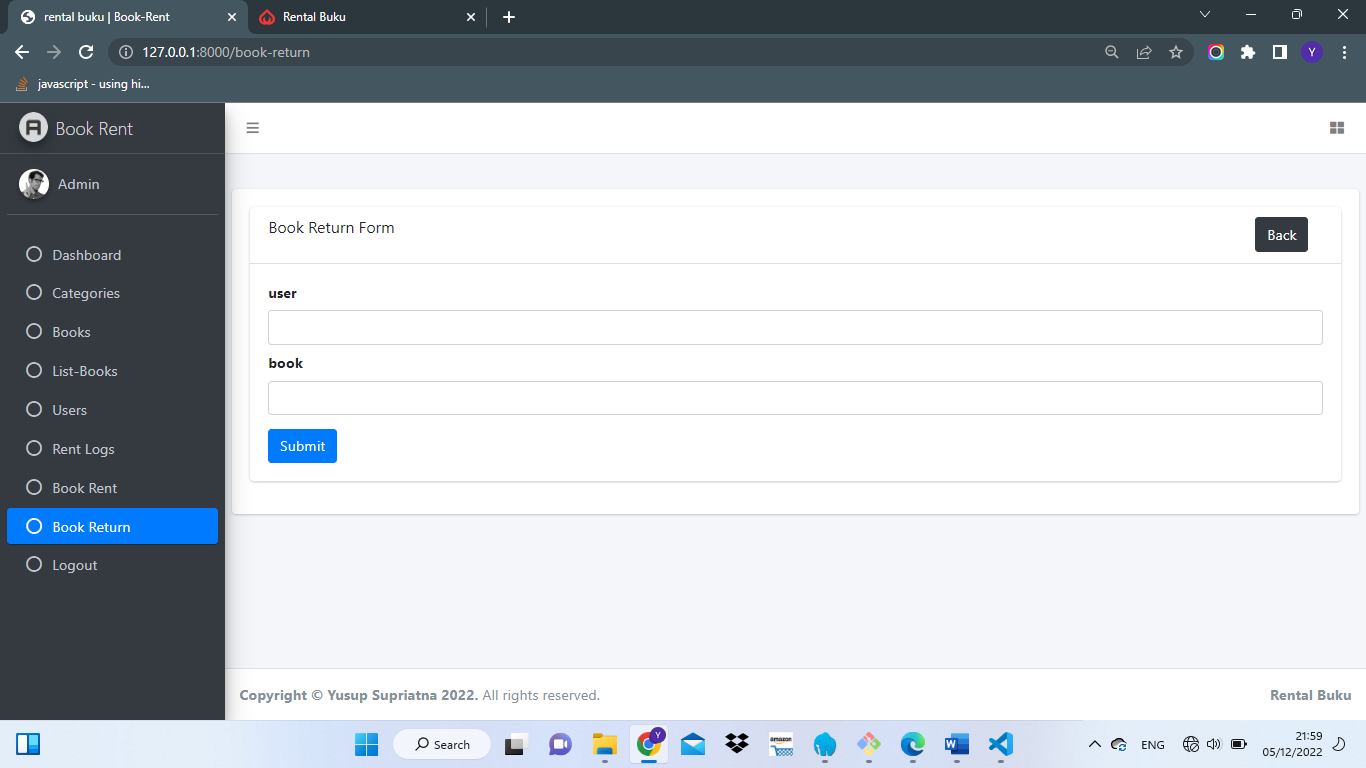
Gambar 3.5.17 Tampilan User Approved

### Tampilan Form Peminjaman Buku



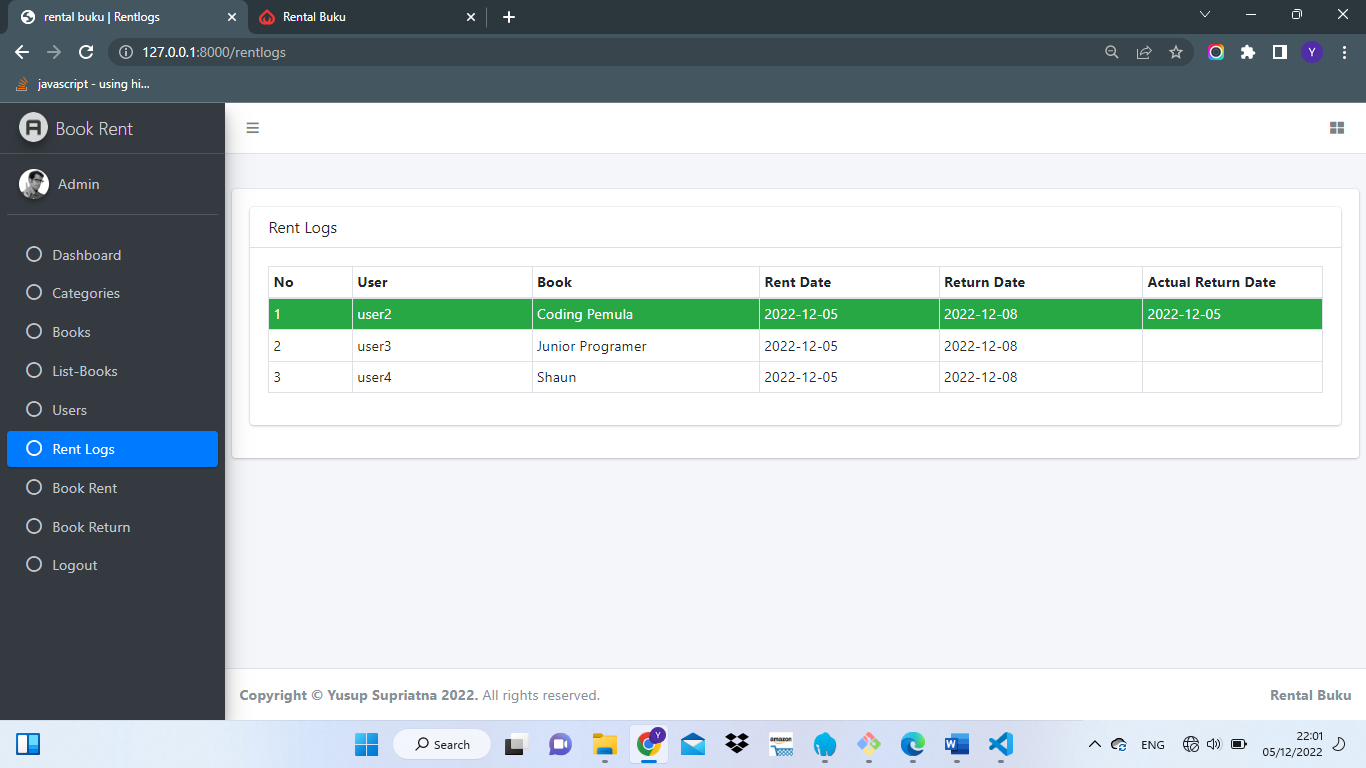
Gambar 3.5.18 Tampilan Form Peminjaman Buku

### Tampilan Form Pengembalian Buku



Gambar 3.5.19 Tampilan Form Pengembalia Buku

### Tampilan Peminjaman dan Pengembalian Buku



Gambar 3.5.20 Tampilan Peminjaman dan Pengembalian Buku

## Kendala Implementasi

Kendala yang dihadapi penulis saat pembuatan Aplikasi Rental Buku sebagai berikut:

1. Kurangnya wawasan dalam menganalisis *project* yang akan

dikerjakan sehingga kesulitan ketika merancang alur *project*.

2. Kurangnya wawasan dengan *framework* Laravel, sehingga penulis kesulitan dalam menyusun *code* program*.*

## Pemecahan Atas Kendala

Pemecahan atas kedala yang dihadapi penulis dalam pembuatan Aplikasi Rental Buku sebagai berikut:

1. Bertanya kepada senior yang sudah lama membuat sistem.

2. Bertanya kepada forum dan komunitas Laravel Indonesia secara otodidak.