

GL03

DOKUMEN UJI PERANGKAT LUNAK

ReseaRead

untuk:

Pencari Buku Digital

Dipersiapkan oleh:

Arifki Ilham	(1301210405)
Rifqah Amaliyah	(1301213241)
Sofia Nafiu Nur Rohmah	(1301213112)
Fannisa Eimin Aurora	(1301213150)
Annisa Izzatul Latifa	(1301213328)

Prodi S1 Informatika - Universitas Telkom

Prodi S1 Informatika Universitas Telkom	Nomor Dokumen	Halaman
	DUPL-07 <ReseaRead>	50
	Revisi	0
		Tgl: 25 / 12 / 2023

Daftar Isi

Daftar Gambar	2
Daftar Tabel	4
1 Pendahuluan	4
1.1 Tujuan Pembuatan Dokumen	4
1.2 Ruang Lingkup Pengujian	5
1.3 Referensi	5
1.4 Overview Sistem & Fitur Utamanya	5
1.5 Overview Pengujian	6
1.5.1 Perangkat Keras Pengujian	6
1.5.2 Sumber Daya Manusia	6
1.5.3 Perangkat Lunak Pengujian	6
1.5.4 Material Pengujian	7
1.5.5 Strategi dan Metode Pengujian	7
1.5.6 Jadwal Pengujian	8
2 Pelaksanaan Pengujian	9
2.1 Pengujian UNIT	9
2.1.1 Pengujian White Box Method	9
2.1.2 Pengujian Class	14
2.2 Pengujian INTEGRASI	42
2.2.8 Pengujian Log out	47
2.3 USER ACCEPTANCE TEST (Berjuang mencari user potensial)	47
2.4 Summary Pengujian Yang Masih Gagal	49

Daftar Gambar

Gambar 1 Flowgraph	11
Gambar 2 Form Add Book	12
Gambar 3. Form Add Book	12
Gambar 4. Berhasil Add new Book	12
Gambar 5. Hasil tampilan setelah berhasil Add new Book	13
Gambar 6. Form Add Book	14
Gambar 7. Field Buku tidak terdapat di Database ReseaRead	14
Gambar 8 Field input manual Add new Book	14
Gambar 9 Berhasil menambahkan buku secara manual	15
Gambar 10 codingan method sign up	17
Gambar 11 Berhasil registered	19
Gambar 12 Registered gagal karena email tidak valid	19
Gambar 13 Register gagal karena password tidak lebih dari 8 karakter	19
Gambar 14 Register gagal karena email sudah pernah mendaftar	20
Gambar 15 Register gagal karena username sudah pernah mendaftar	20
Gambar 16 Codingan Sign in	21
Gambar 17 homepage login	22
Gambar 18 Berhasil login	22
Gambar 19 Login gagal karena username tidak ditemukan	23
Gambar 20 Login gagal karena password tidak valid	23
Gambar 21 Codingan Katalog	24
Gambar 22 Codingan Katalog	26
Gambar 23 homepage Katalog	26
Gambar 24 Homepage Katalog	26
Gambar 25 Codingan AddBook	29
Gambar 26 Field untuk mengisi judul buku	30
Gambar 27 User mengisi inputan yang diperlukan	31
Gambar 28 Berhasil AddBook	31
Gambar 29 Field untuk mengisi judul buku	32
Gambar 30 Gagal mencari judul buku karena tidak ditemukan di Database ReseaRead	32
Gambar 31 Codingan Delete Book	34
Gambar 32 Homepage Library	35
Gambar 33 Field apakah yakin menghapus buku dari library	35
Gambar 34 Berhasil menghapus buku	35
Gambar 35 Codingan addBook	37
Gambar 36 Berhasil Menambahkan buku	38
Gambar 37 Tidak berhasil menambahkan buku karena judul tidak dimasukkan	38
Gambar 38 Codingan getAllBookByStatus	39
Gambar 39 Homepage Library bagian Reading	39
Gambar 40 Homepage Library bagian Finished	39
Gambar 41 Codingan Logout	40
Gambar 42 Field LogOut berhasil	40
Gambar 43 Codingan Search	41
Gambar 44 Homepage Search	41
Gambar 45 Codingan BookDetails	42
Gambar 46 Tampilan BookDetails	42

Daftar Tabel

Tabel 1 Overview sistem dan fitur utama	9
Tabel 2 Jadwal Pengujian	10
Tabel 3 Hasil uji data normal method sign up	15
Tabel 4 Hasil uji data salah method sign up	16
Tabel 5 Hasil uji data normal method sign in	21
Tabel 6 Hasil uji data salah method sign in	21
Tabel 7 Hasil uji data normal halaman katalog	23
Tabel 8 Hasil uji data normal method ModalAddBook	27
Tabel 9 Hasil uji data salah method ModalAddBook	28
Tabel 10 Hasil uji data normal method DeleteBook	33
Tabel 11 Hasil uji data normal method addBoook	36
Tabel 13 Hasil uji data normal method getAllBookByStatus	38
Tabel 14 Hasil uji data normal method logout	40
Tabel 15 Hasil uji data normal method search	40
Tabel 16 Hasil uji data normal method bookDetails	41
Tabel 17 Hasil uji data normal usecase sign in	43
Tabel 18 Hasil uji data salah usecase sign in	43
Tabel 19 Hasil uji data normal usecase library	44
Tabel 20 Hasil uji data normal usecase catalog	45
Tabel 20 Hasil uji data normal use case search	46
Tabel 21 Hasil uji data normal usecase addbook	46
Tabel 22 Hasil uji data salahusecase addbook	47
Tabel 23 Hasil uji data normal use case profile setting	47
Tabel 24 Hasil uji data salah use case profile setting	47
Tabel 25 Hasil uji data salah normal case homepage	48
Tabel 26 Hasil uji data salah normal case logout	48
Tabel 27 User acceptance test	50
Tabel 28 Summary pengujian gagal	50

1 Pendahuluan

1.1 Tujuan Pembuatan Dokumen

Dokumen ini berisi Dokumen Uji Perangkat Lunak (DUPL) digunakan sebagai panduan untuk menguji perangkat lunak Books Explorer Application yang telah dibuat. Dokumen ini digunakan

untuk melihat hasil uji perangkat lunak Books Explorer Application apakah sudah sesuai rancangan yang telah dibuat atau tidak. Dokumen ini disusun untuk para programmer sebagai panduan dalam mempertimbangkan pengembangan perangkat lunak selanjutnya, dengan tujuan mencapai target perancangan yang telah ditetapkan.

1.2 Ruang Lingkup Pengujian

Lingkungan Pengujian Perangkat Lunak ini meliputi Perangkat Keras Pengujian, Material Pengujian, Perangkat Lunak Pengujian, Sumber Daya Manusia, Material Pengujian, Strategi dan Metode Pengujian, Tujuan Pengujian, Rencana Pengujian, Pelaksanaan Pengujian dan Kesimpulan Pengujian

1.3 Referensi

Menjelaskan referensi SKPL & DPPL yang akan digunakan sebagai rujukan pembuatan dokumen ini

- IEEE Std 830-1993, IEEE Recommended Practices for Software Requirement Specifications
- SKPL Diaspora yang telah dibuat sebelumnya

1.4 Overview Sistem & Fitur Utamanya

No	Nama Use Case	Deskripsi Fitur
1	Membaca data buku	User dapat mengakses data buku (deskripsi buku, review dan rating, penulis, jumlah halaman, dan tag buku)
2	Menambah buku baru ke dalam Sistem	User dapat menambahkan buku baru kedalam database setelah melakukan proses login
3	Menambahkan buku ke library	User dapat menambahkan buku yang terdaftar pada library
4	Menghapus buku dari library	User dapat menghapus buku yang terdaftar pada library
5	Sign in	User dapat masuk sebagai pengguna terdaftar
6	Sign up	User dapat mendaftar sebagai pengguna baru
7	Melihat katalog	User dapat melihat katalog buku berdasarkan hasil pencarian.
8	Menampilkan buku berdasarkan kategori status	User dapat melihat buku berdasarkan kategori status
9	Log out	User dapat keluar dari web

10	Pencarian buku	User dapat mencari buku dengan menuliskan keyword pada searchbar
----	----------------	--

Tabel 1 Overview sistem dan fitur utama

1.5 Overview Pengujian

1.5.1 Perangkat Keras Pengujian

Laptop digunakan sebagai perangkat keras umum dalam pengujian perangkat lunak, berperan dalam eksekusi skenario pengujian fungsional. Tugas ini mencakup konfirmasi fitur utama dan peneguhan bahwa perilaku aplikasi sesuai dengan standar spesifikasi. Selain itu, laptop digunakan untuk memverifikasi pengalaman pengguna, melibatkan evaluasi terhadap desain, responsivitas, dan konsistensi antarmuka guna memastikan penggunaan yang optimal.

Spesifikasi hardware :

- Processor : Intel ® Celeron ® CPU N2840 @2.16GHz.
- RAM : 2.00 GB
- Tipe Sistem : 32-bit Operating System
- Hard Disk : Minimal 100GB

1.5.2 Sumber Daya Manusia

Sumber daya manusia yang terlibat dalam pengujian perangkat lunak ini adalah para pengembang atau developer. Peran utama developer dalam pengujian adalah untuk mendesain dan menulis skrip pengujian otomatis serta menyusun unit test untuk memastikan fungsi-fungsi perangkat lunak yang mereka kembangkan berjalan sesuai dengan spesifikasi yang telah ditetapkan. Selain itu, developer juga berkontribusi dalam mengidentifikasi dan memperbaiki bug atau masalah yang terdeteksi selama pengujian, serta berpartisipasi dalam pengujian integrasi untuk memastikan bahwa komponen-komponen perangkat lunak berinteraksi dengan baik satu sama lain.

1.5.3 Perangkat Lunak Pengujian

- Sistem Operasi : Windows 10, 11
- Bahasa Pemrograman : JavaScript
- Visual Studio Code

Visual Studio Code (VS Code) adalah editor kode sumber yang populer dengan fungsi yang mendukung pengujian perangkat lunak. Dengan integrasi ekstensi untuk berbagai kerangka pengujian, debugger yang terintegrasi, dukungan Git, pengaturan proyek yang ringan, dan integrasi dengan alat pengujian terkenal seperti Selenium, VS Code memudahkan pengembang dan penguji untuk menjalankan, mengelola, dan mengoptimalkan pengujian perangkat lunak. Terminal terintegrasi, linting, dan dukungan untuk berbagai

bahasa pemrograman menjadikannya solusi yang fleksibel dan disukai untuk kebutuhan pengujian fungsional dan kualitas kode.

- XAMPP ver 8.2.12-0 terdiri dari http server dan MySQL database

XAMPP menyediakan lingkungan pengembangan web lokal dengan server Apache, MySQL, dan dukungan untuk bahasa pemrograman seperti PHP dan Perl. Berperan sebagai alat pengujian perangkat lunak, XAMPP memungkinkan pengembang untuk menguji dan mengembangkan aplikasi tanpa ketergantungan pada server produksi, mempercepat siklus pengembangan, dan mengurangi risiko. Dengan fitur konfigurasi yang mudah dan perlengkapan tambahan seperti PHPMyAdmin dan FileZilla, XAMPP menjadi pilihan efisien untuk pengujian aplikasi web sebelum diluncurkan ke lingkungan produksi.

- Browser : Google Chrome

1.5.4 Material Pengujian

Menjelaskan modul – modul yang akan diuji

- Login
- Register
- Melihat catalog
- Menambah buku ke dalam sistem
- Menambahkan buku ke library
- Menghapus buku dari library
- Menampilkan buku berdasarkan kategori status
- Logout
- Pencarian buku
- Membaca data buku
- Menampilkan profile settings

1.5.5 Strategi dan Metode Pengujian

a. Black-box

Pengujian Black Box merupakan pendekatan di mana evaluasi perangkat lunak dilakukan tanpa memperhatikan struktur internal atau rincian implementasi. Pendekatan ini efektif untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi dan kebutuhan pengguna tanpa memperhatikan rincian internal.

b. White-box

Pengujian struktural, melibatkan pemeriksaan terhadap struktur internal dan logika kode sumber perangkat lunak. Tujuan utama dari pengujian white box adalah memastikan bahwa setiap komponen internal beroperasi dengan benar dan efisien.

Dengan fokus pada logika internal, pengujian white box membantu mengidentifikasi cacat atau kesalahan logika yang mungkin tidak terdeteksi oleh pendekatan black box.

c. Pengujian Integrasi

Pengujian integrasi adalah fase pengujian perangkat lunak yang fokus pada pengujian interaksi antara komponen atau modul sistem. Tujuannya adalah memastikan bahwa berbagai bagian sistem berfungsi secara harmonis saat diintegrasikan dan mendeteksi potensi kesalahan atau ketidakcocokan. Dalam pengujian ini, komponen-komponen diuji bersama untuk memverifikasi fungsi dan aliran data di antara mereka.

d. Pengujian User Acceptance

Pengujian User Acceptance (UAT) adalah fase akhir dalam siklus pengujian perangkat lunak, di mana pengguna akhir menguji aplikasi untuk memastikan bahwa itu memenuhi harapan dan kebutuhan mereka. Tujuannya adalah untuk menilai kesiapan dan kelayakan aplikasi sebelum dirilis secara resmi. Selama UAT, pengguna melakukan pengujian fungsional dan verifikasi apakah aplikasi memenuhi spesifikasi, bekerja sesuai dengan skenario penggunaan nyata, serta memberikan pengalaman pengguna yang memuaskan.

1.5.6 Jadwal Pengujian

(Sebutkan kasus data pengujinya)

No	Use Case	PIC	Jadwal Pengujian
1	Sign in	Annisa Izzatul L.	25/12/2023
2	Sign up	Annisa Izzatul L.	25/12/2023
3	Melihat katalog	Annisa Izzatul L.	25/12/2023
4	Menambah buku baru ke dalam Sistem	Rifqah Amaliyah	25/12/2023
5	Menambahkan buku ke library	Sofia Nafiu N.	25/12/2023
6	Menghapus buku dari library	Sofia Nafiu N.	25/12/2023
7	Menampilkan buku berdasarkan kategori status	Rifqah Amaliyah	25/12/2023
8	Log out	Rifqah Amaliyah	25/12/2023
9	Pencarian buku	Rifqah Amaliyah	25/12/2023
10	Membaca data buku	Rifqah Amaliyah	25/12/2023

11	Menampilkan profile settings	Arifki ilham	25/12/2023
----	------------------------------	--------------	------------

Tabel 2 Jadwal Pengujian

2 Pelaksanaan Pengujian

(Uraikan pengujian sesuai dengan butir uji yang sudah didefinisikan dalam rencana pengujian)
Pengujian yang dilakukan pada perangkat lunak Books Explorer Application yang telah dibuat. diantaranya adalah Sign in, Sign up, Melihat catalog, Menambah buku baru kedalam system, Menambah buku ke library, Menghapus buku dari Library, Menampilkan buku berdasarkan kategori status, Pencarian buku, Membaca data buku, Menampilkan Profile, Log out.

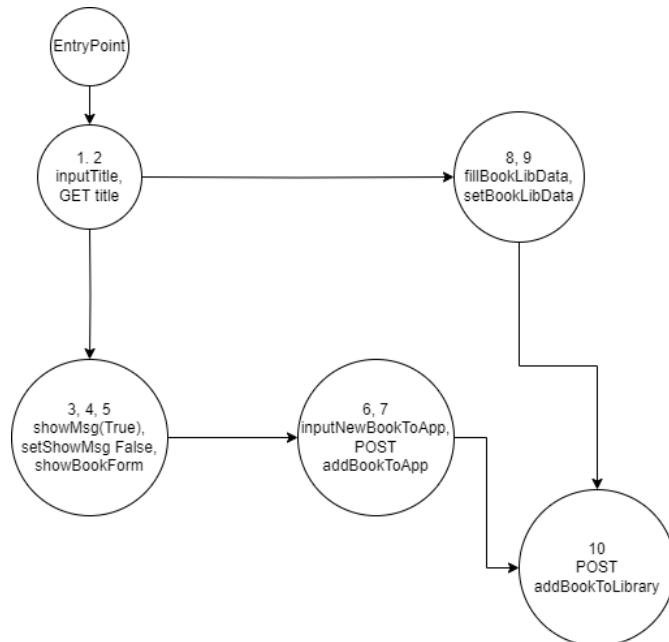
2.1 Pengujian UNIT

2.1.1 Pengujian White Box Method

a. Method

Pilihan metode yang paling kompleks adalah method AddBook dari class Component. Dalam kasus ini, User bisa memasukan buku ke library nya, dengan catatan buku tersebut bisa ada atau tidak ada di dalam aplikasi. Akan terjadi pemanggilan API untuk 2 kasus yang berbeda.

b. Flowgraph



Gambar 1 Flowgraph

https://drive.google.com/file/d/14MnO_LISfdZRB4xqaKMkODGWN0_OFrAK/view?usp=sharing

c. Hitung cyclomatic complexity nya

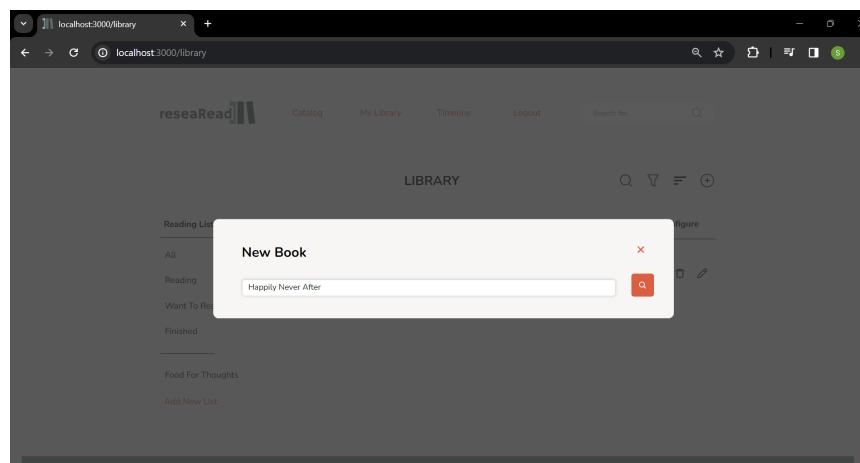
$$\text{Cyclomatic complexity} = E - N + 2$$

$$E = \text{Jumlah Edge}$$

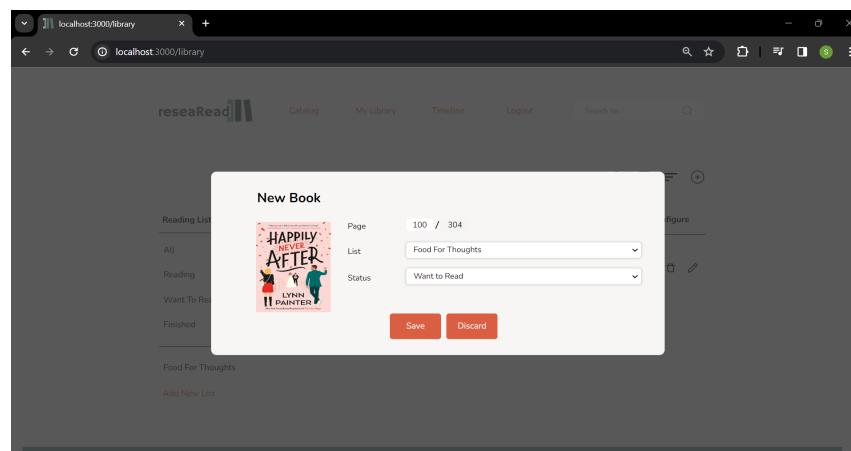
N = Node

$$\begin{aligned}\text{Cyclomatic complexity} \\ &= E - N + 2 \\ &= 5 - 5 + 2 = 2\end{aligned}$$

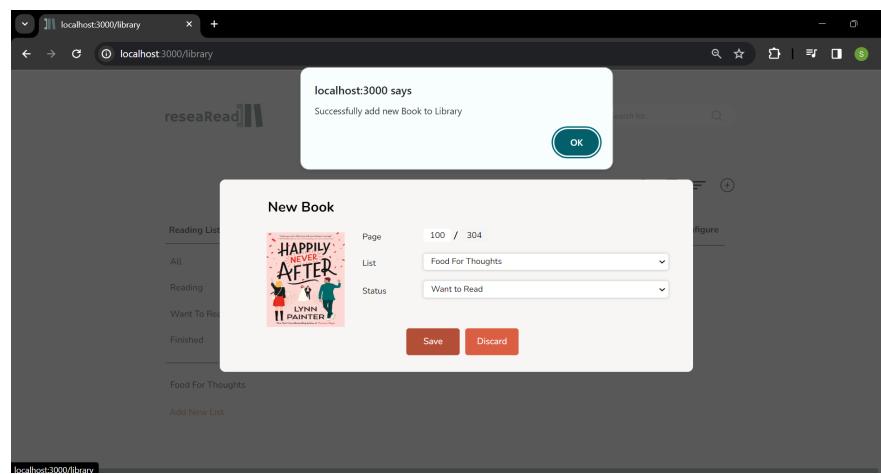
- d. Daftar Path yang perlu diuji
 - 1) 1-2-8-9-10
 - 2) 1-2-3-4-5-6-7-10
- e. Data uji path
Untuk mencapai skenario pada poin d, masing masing path memerlukan inputan sebagai berikut :
 - 1) bookTitle, bookStatus, list, currentPage.
 - 2) bookTitle, bookAuthor, currentPage, totalPages, coverUrl, bookStatus, bookList
- f. Tampilkan screenshot hasilnya.
 - 1) Path1



Gambar 2 Form Add Book



Gambar 3. Form Add Book



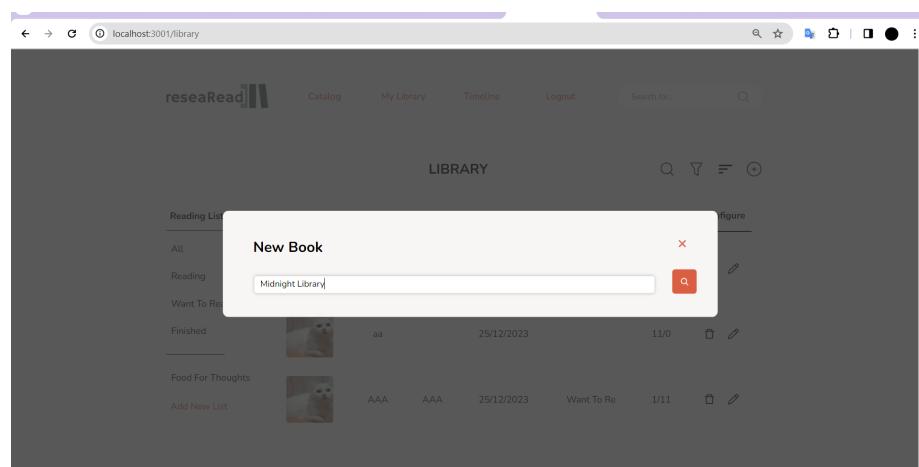
Gambar 4. Berhasil Add new Book

The screenshot shows a web-based library management system. At the top, there is a navigation bar with links for Catalog, My Library, Timeline, Logout, and a search bar. Below the navigation is a header with the title 'LIBRARY' and various search and filter icons. A table lists books in the library. The table has columns for Cover, Title, Author, Date Added, Status, Page, and Configure. There are four rows of data:

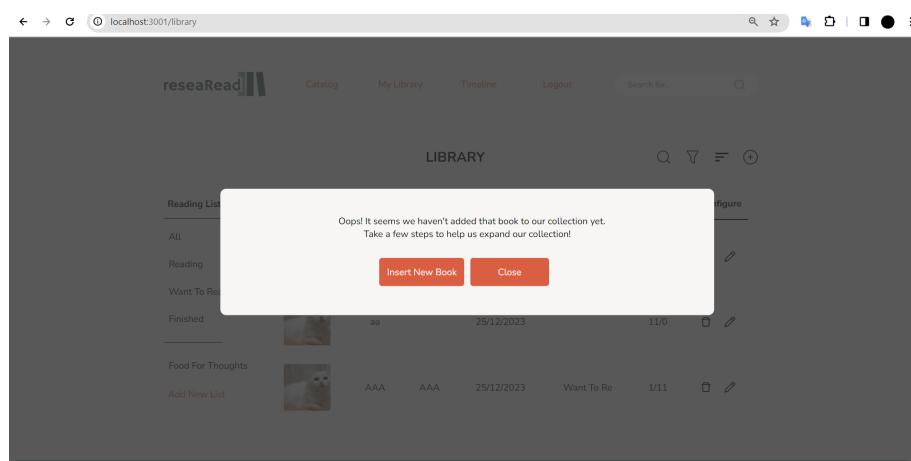
Reading List	Cover	Title	Author	Date Added	Status	Page	Configure
All		Lynne Dexter Painter	Monhegan Museum	25/12/2023	Want To Re	12/58	
Reading		Happily PPM After	noraief	25/12/2023	Reading	120/354	
Want To Read		Happily Never After	Lynn Painter	25/12/2023	Want To Re	120/304	
Finished							
Food For Thoughts							
Add New List							

Gambar 5. Hasil tampilan setelah berhasil Add new Book

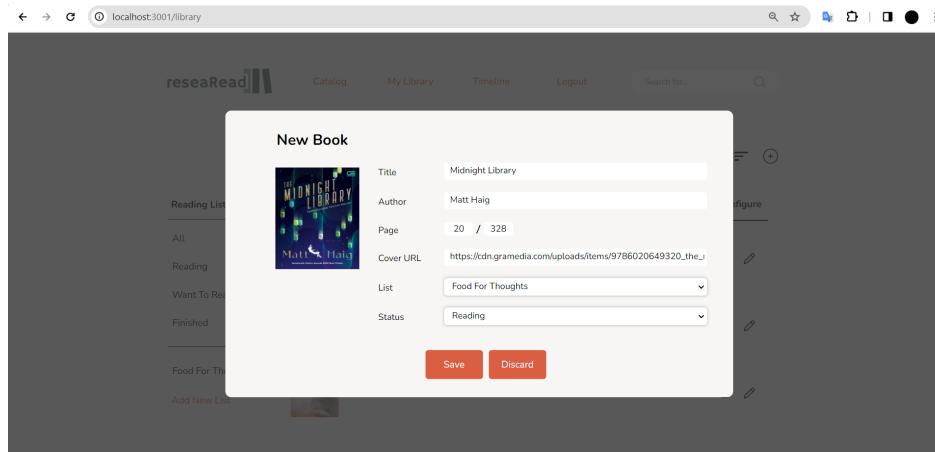
2) Path 2



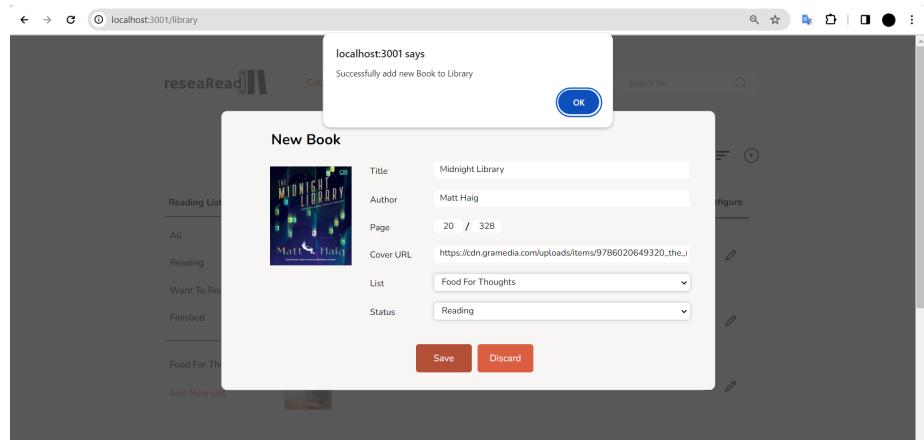
Gambar 6. Form Add Book



Gambar 7. Field Buku tidak terdapat di Database ReseaRead



Gambar 8 Field input manual Add new Book



Gambar 9 Berhasil menambahkan buku secara manual

2.1.2 Pengujian Class

2.1.2.1 Method signup

CLASS	Method	Kasus dan Hasil Uji (Data Normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
user	sign up	<ul style="list-style-type: none"> - username : peanutbutter - email : pbnj@resea.com - password : rifqah123 	Dapat menampilkan alert yang berisi pesan “Successfully Registered an Account” kemudian mengantar user ke halaman utama website	Berhasil melakukan register sebagai user ReseaRead	[<input type="checkbox"/>] diterima

Tabel 3 Hasil uji data normal method sign up

CLASS	Method	Kasus dan Hasil Uji (Data Salah)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
user	sign up	- username : jeruk - email : jeruk@resea - password : jeruk123	Dapat menampilkan alert yang berisi pesan “Invalid Email Address”	Berhasil mendeteksi format email yang tidak sesuai	[<input type="checkbox"/>] diterima
		- username : apel - email : butter@resea - password : rifqah	Dapat menampilkan alert yang berisi pesan “Password must be between 8 and 20 characters”	Berhasil mendeteksi panjang karakter password yang tidak sesuai requirement	[<input type="checkbox"/>] diterima
		- username : markisa - email : pbnj@resea.com - password : markisa123	Dapat menampilkan alert yang berisi pesan “Email Has Already Registered”	Berhasil mendeteksi email yang sudah terdaftar	[<input type="checkbox"/>] diterima
		- username : peanuttbutte - email : semangka@resea.com - password : semanga123	Dapat menampilkan alert yang berisi pesan “Username Has Already Registered”	Berhasil mendeteksi email yang sudah terdaftar	[<input type="checkbox"/>] diterima

Tabel 4 Hasil uji data salah method sign up

Code

```
const userRegister = async (req, res) => {
  try {
    const { email, password, userName } = req.body;

    const emailRegex = /^[^@\s]+@[^\s]+\.\[^@\s]+$/;
    const minPassLength = 8;
    const maxPassLength = 20;

    if (!emailRegex.test(email)) {
      return res.status(400).json({ message: "Invalid Email Address" });
    }

    if (
      !(password.length >= minPassLength && password.length <= maxPassLength)
    ) {
      return res.status(400).json({
        message: `Password must be between ${minPassLength} and ${maxPassLength} characters`,
      });
    }

    const isEmailExist = await User.findOne({ where: { email: email } });

    if (isEmailExist) {
      return res.status(409).json({ message: "Email has already registered" });
    }

    const isUsernameExist = await User.findOne({
      where: { userName: userName },
    });

    if (isUsernameExist) {
      return res
        .status(409)
        .json({ message: "Username has already registered" });
    }
  }

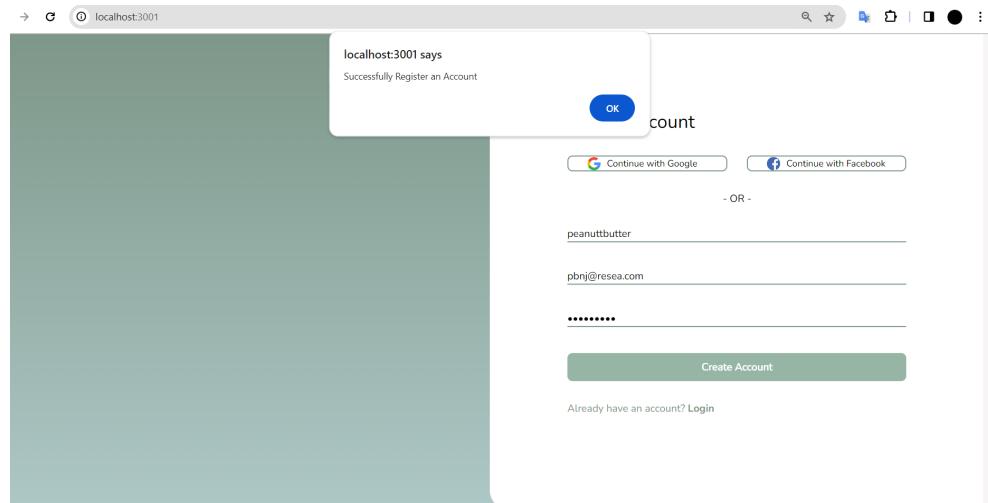
  const hashedPassword = await bcrypt.hash(password, 9);
  const newUser = await User.create({
    userId: uuidv4(),
    email,
    password: hashedPassword,
    userName,
  });

  const blobData = newUser.userImage;
  const decodeImageBuffer = buffer.Buffer.from(blobData, "base64");

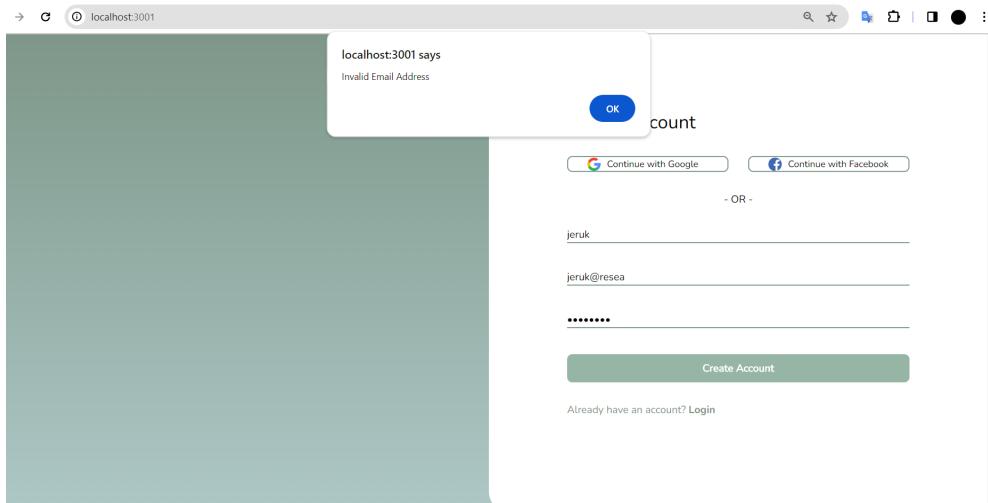
  return res.status(201).json({
    payload: { newUser },
    image: decodeImageBuffer,
    message: "Successfully Register an Account",
  });
} catch (err) {
  return errorCatch(res, err, 500, "Register an Account");
}
};
```

Gambar 10 codingan method sign up

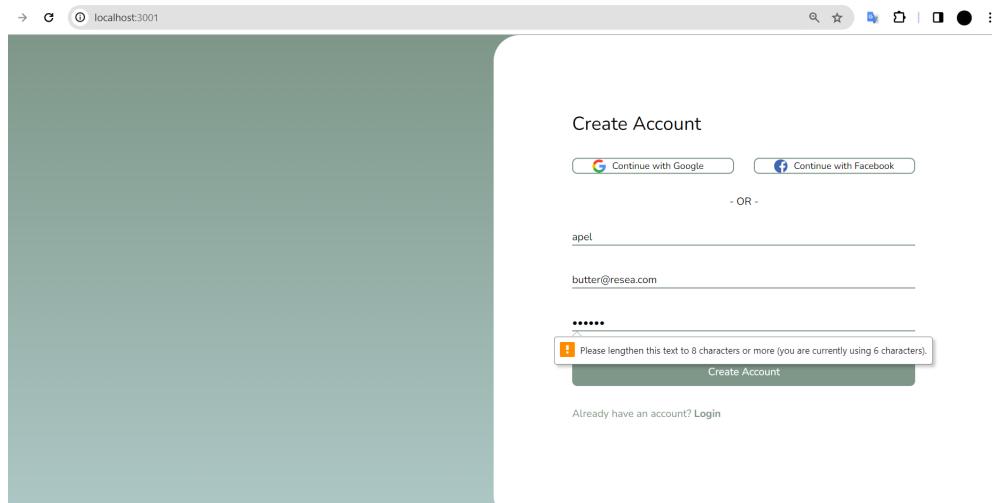
Screenshoot hasil pengujian



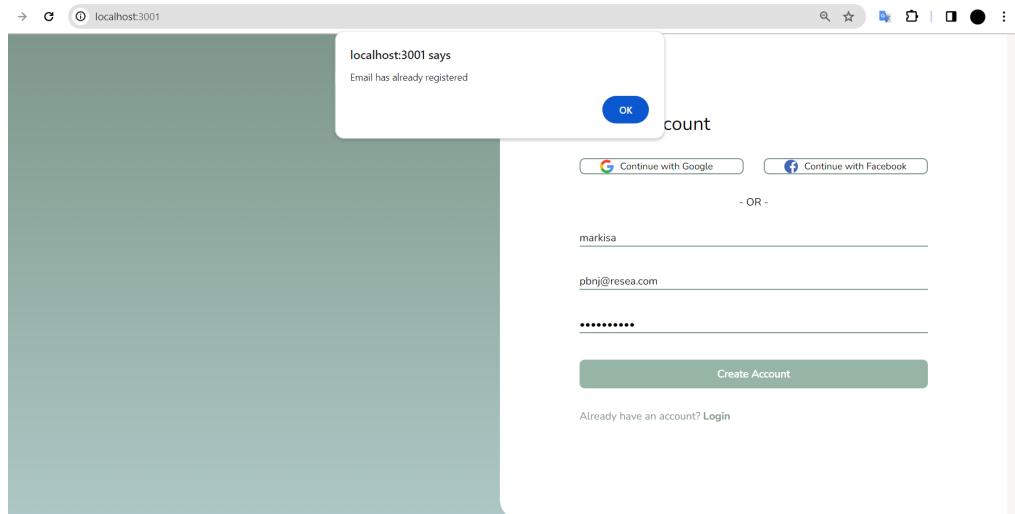
Gambar 11 Berhasil registered



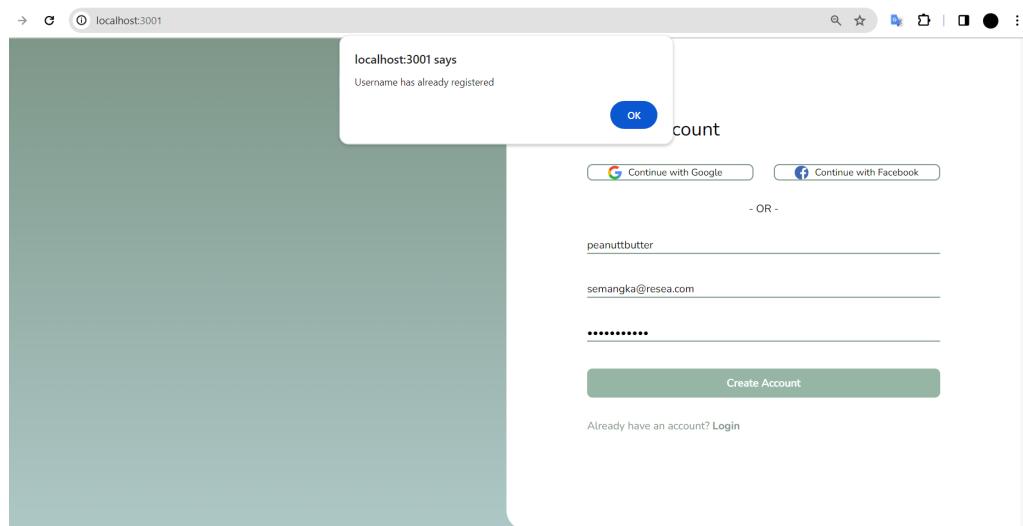
Gambar 12 Registered gagal karena email tidak valid



Gambar 13 Register gagal karena password tidak lebih dari 8 karakter



Gambar 14 Register gagal karena email sudah pernah mendaftar



Gambar 15 Register gagal karena username sudah pernah mendaftar

2.1.2.2 Method sign in

CLASS	Method	Kasus dan Hasil Uji (Data normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
user	sign	<ul style="list-style-type: none"> - email : annisa@gmail.com - password : 12345678 	Dapat menampilkan form data sign in	Berhasil menampilkan data inputan user pada form sign in	[<input type="checkbox"/>] diterima

CLASS	Method	Kasus dan Hasil Uji (Data normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
		Klik tombol Login	Mengarahkan user terdaftar ke homepage Researched	Berhasil masuk ke halaman homepage	[<input type="checkbox"/>] diterima

Tabel 5 Hasil uji data normal method sign in

CLASS	Method	Kasus dan Hasil Uji (Data Salah)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
user	sign in	- email : izzatul@gmail.com - password : 12345678	Menampilkan alert “User not found” ketika data sign in belum terdaftar dalam sistem	Berhasil mendeteksi data akun yang tidak terdaftar serta menampilkan pesan “User not found”	[<input type="checkbox"/>] diterima
		- email : annisa@gmail.com - password : 87654321	Menampilkan alert “Invalid password” ketika password yang diinputkan tidak sesuai dengan data password yang terdaftar dalam database sistem	Berhasil mendeteksi ketidak sesuaian inputan password serta menampilkan pesan “Invalid password”	[<input type="checkbox"/>] diterima

Tabel 6 Hasil uji data salah method sign in

Code

```
const userLogin = async (req, res) => {
  try {
    const { email, password } = req.body;

    const userData = await User.findOne({ where: { email: email } });

    if (!userData) {
      return res.status(404).json({ message: "User not found" });
    }

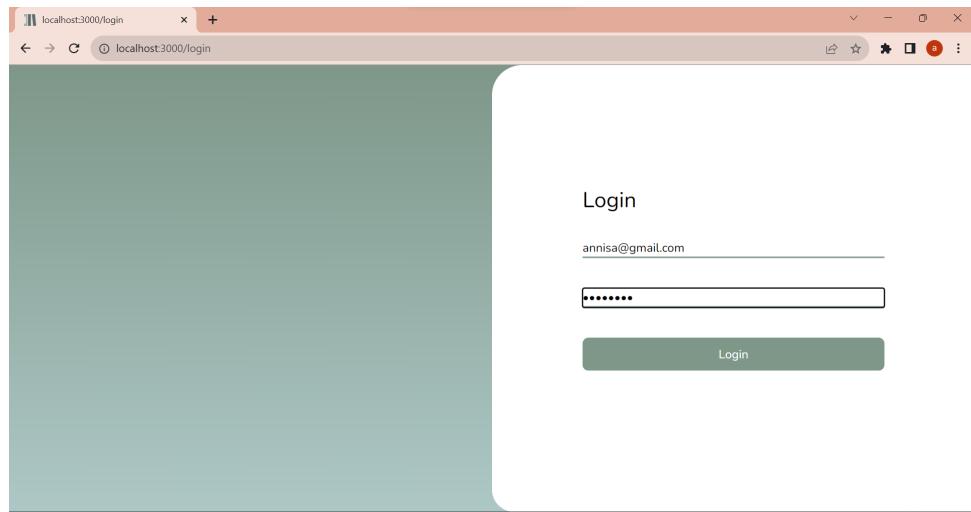
    const isPasswordValid = await bcrypt.compare(password, userData.password);

    if (!isPasswordValid) {
      return res.status(401).json({ message: "Invalid Password" });
    }

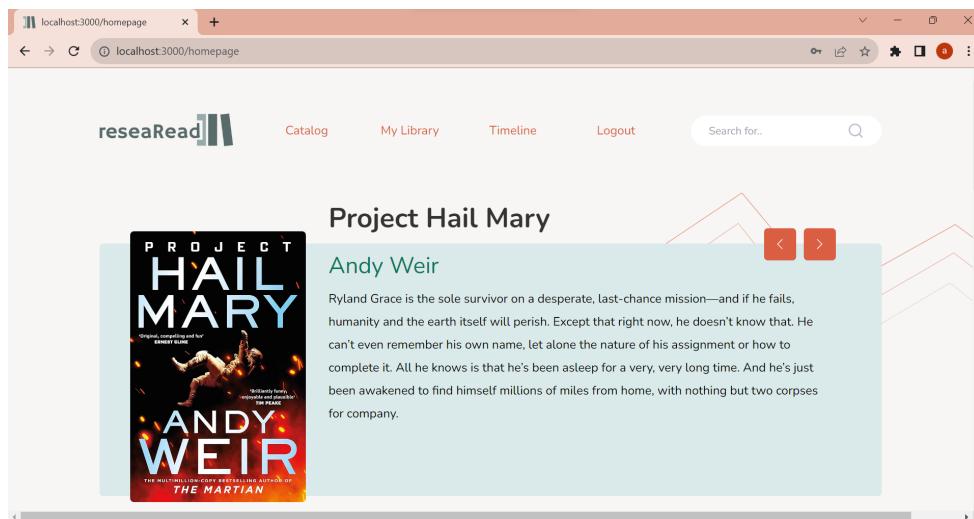
    const token = jwt.sign(
      { userId: userData.userId, email: userData.email },
      process.env.JWT_SECRET,
      { expiresIn: process.env.JWT_DURATION }
    );
    return res.status(200).json({ message: "Login Success", userData, token });
  } catch (err) {
    return errorCatch(res, err, 400, "Login");
  }
};
```

Gambar 16 Codingan Sign in

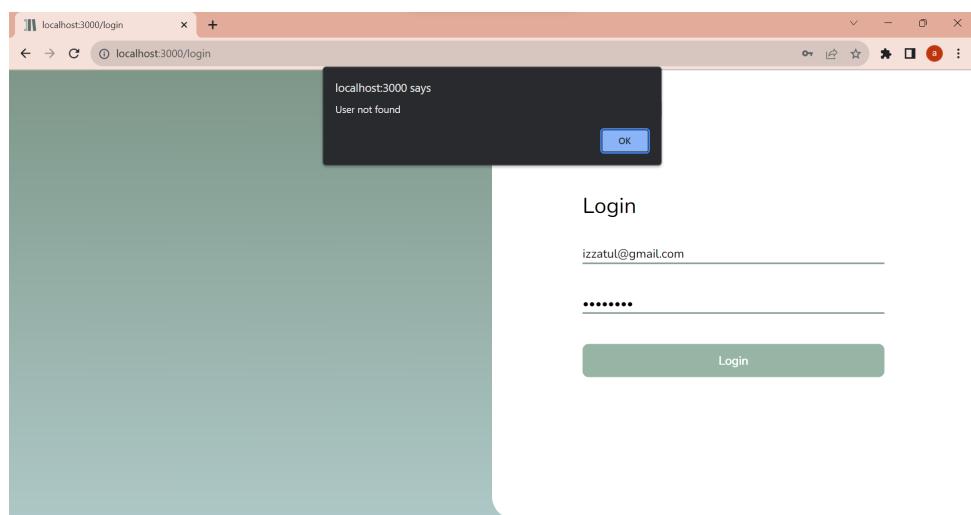
Screenshot hasil



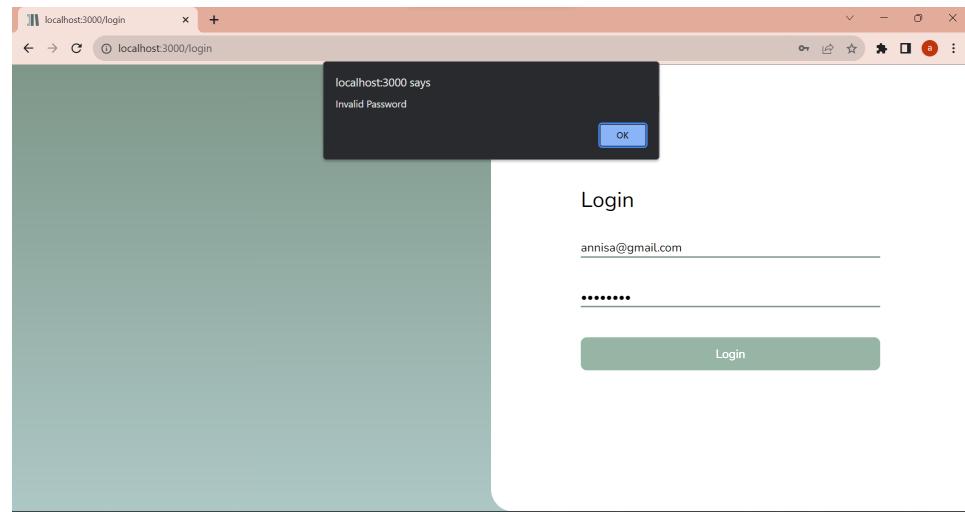
Gambar 17 homepage login



Gambar 18 Berhasil login



Gambar 19 Login gagal karena username tidak ditemukan



Gambar 20 Login gagal karena password tidak valid

2.1.2.3 Halaman Katalog

PROSES (DFD-1)/ USE CASE	Kasus dan Hasil Uji (Data normal)			
	Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Melihat Katalog	Klik 'Catalog' pada navbar	Menampilkan section book category dan section book trending	Berhasil menampilkan section book category dan book trending Tidak berhasil menampilkan keterangan tag book yang seharusnya ada pada buku yang tertampil di section book trending	[X] ditolak
	Klik 'See more' pada section book category	Sistem dapat menampilkan lebih banyak lagi kategori buku	Berhasil menampilkan semua list dalam kategori buku	[] diterima
	Klik 'See more' pada section book trending	Sistem dapat menampilkan lebih banyak lagi daftar buku trending	tombol see more tidak menavigasi ke page manapun	[X] ditolak

Tabel 7 Hasil uji data normal halaman katalog

Code

```
src > components > Category > index.jsx > ...
1  import Image from "next/image";
2  import { BsBell, BsGear } from "react-icons/bs";
3  import { CiViewTimeline, CiMail } from "react-icons/ci";
4  import { BsThreeDotsVertical } from "react-icons/bs";
5  import Link from "next/link";
6  import Button from "../Button";
7  import { useRouter } from "next/navigation";
8
9  const Category = ({ data, detail = null }) => {
10    const router = useRouter();
11    return (
12      <div>
13        <div className="grid grid-cols-4 grid-flow-row-dense gap-5">
14          {data.map((dt, index) => (
15            <div
16              key={"category_" + index}
17              className="cursor-pointer"
18              onClick={() => {
19                router.push("/discovery/category?id=" + dt.id);
20              }}
21            >
22              <div className="mt-4 relative h-[214px] w-full rounded-xl">
23                <Image
24                  className="rounded-xl "
25                  src={"/category.png"}
26                  alt={dt.name}
27                  fill
28                  objectFit="cover"
29                />
30              </div>
31              <div className="mt-4 font-bold items-start">{dt.name}</div>
32            </div>
33          )));
34        </div>
35        {detail !== null && (
36          <div className="flex justify-center m-8">
```

Gambar 21 Codingan Katalog

```

src > components > Trending > index.jsx > Trending > useEffect() callback > fetchTrendingBook
  1 "use client";
  2 import "./style.css";
  3 import { BsTag } from "react-icons/bs";
  4 import { BsStar } from "react-icons/bs";
  5 import { BsStarFill } from "react-icons/bs";
  6 import Button from "../Button";
  7 import { useEffect, useState } from "react";
  8 import axios from "axios";
  9 import ModalLoading from "@/components/LoadingSpinner";
10
11 const Trending = ({ API_URL }) => {
12   const [trendingBook, setTrendingBook] = useState([]);
13
14   useEffect(() => {
15     const fetchTrendingBook = async () => {
16       try {
17         const response = await axios.get(`${API_URL}/api/book/books/sized/`, {
18           params: {
19             page: 1,
20             pageSize: 5,
21           },
22         });
23         const bookData = response.data.payload.bookData;
24
25         setTrendingBook(bookData);
26       } catch (err) {
27         console.error(err);
28       }
29     };
29     fetchTrendingBook();
30   }, [API_URL]);
31   return (
32     <div>
33       <div className="grid grid-cols-5 gap-10 grid-flow-row-dense">
34         {trendingBook ? (
35           trendingBook.map((book) => (

```

```

36             trendingBook.map((book) => (
37               <div key={book.bookId} className="flex flex-col">
38                 <div className="mt-4 relative h-[292px] rounded-xl">
39                   <img
40                     className="rounded-xl"
41                     src={book.coverImage}
42                     alt={book.title}
43                     fill="true"
44                     objectfit="cover"
45                   />
46                 </div>
47                 <div className="mt-4 h-1/6 text-xl font-bold">{book.title}</div>
48                 <div className="mt-2 text-base text-customColorOrange">
49                   {book.author}
50                 </div>
51                 <div className="mt-2 flex items-center justify-between text-sm font-bold">
52                   <div className="flex">
53                     {Array.from({ length: 5 }, (_, idx) =>
54                       idx < Math.round(book.rating) ? (
55                         <span key={idx} className="text-customColorOrange">
56                           <BsStarFill />
57                         </span>
58                       ) : (
59                         <span key={idx} className="text-customColorGray">
60                           <BsStar />
61                         </span>
62                       )
63                     )}
64                   </div>
65                   <div>{book.rating}</div>
66                 </div>
67                 <div className="mt-2 text-sm flex items-center space-x-4 text-customColorGray">
68                   <div>
69                     <BsTag />
70                   </div>
71                   <div>{book.category}</div>

```

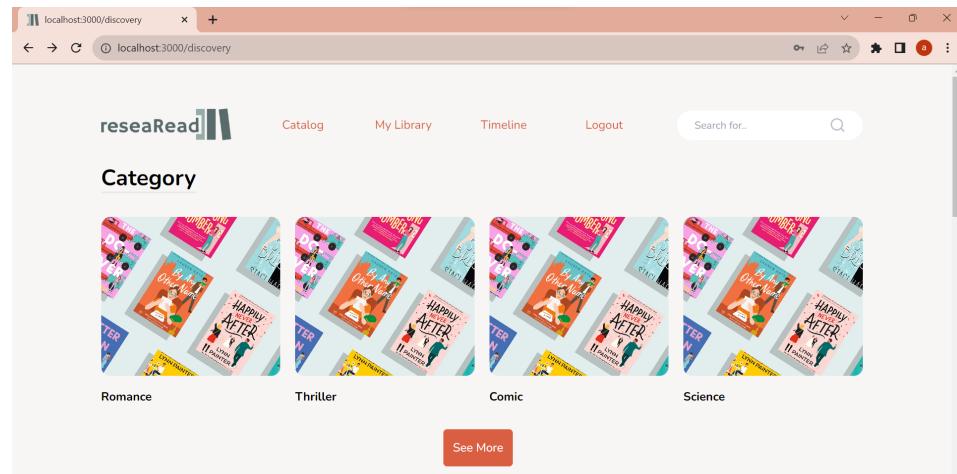
```

72         |     </div>
73         |   </div>
74       );
75     ) : (
76     |   <ModalLoading />
77   )
78   </div>
79   <div className="flex justify-center m-8">
80     |   <button buttonName="See More" size="20" color="white" targetPage="#" />
81   </div>
82 </div>
83   );
84 }
85
86 export default Trending;
87

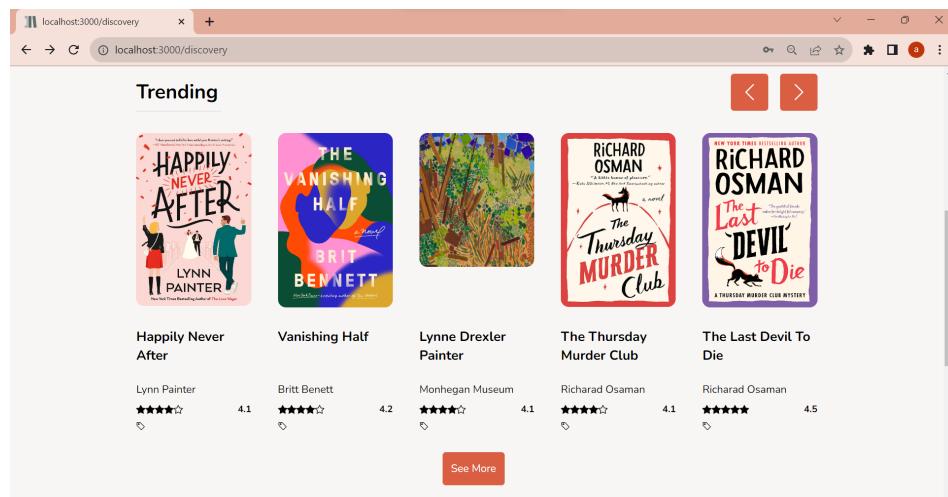
```

Gambar 22 Codingan Katalog

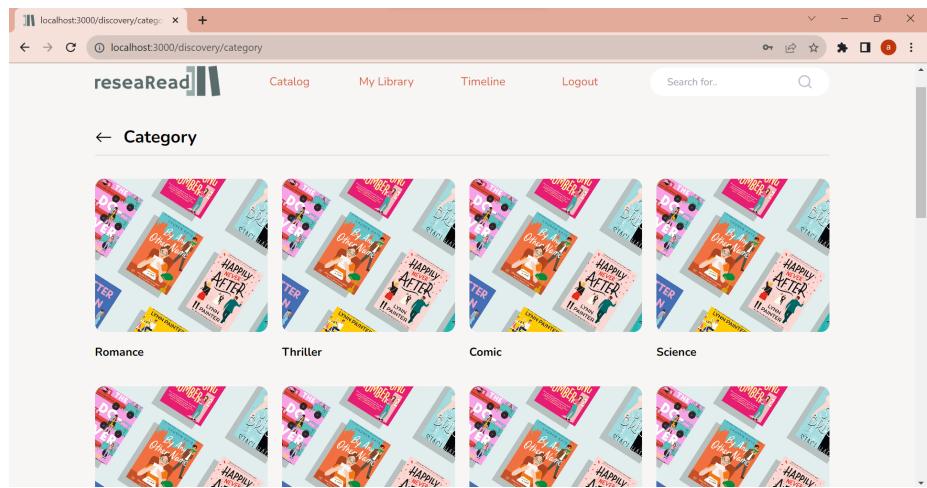
Screenshot hasil pengujian



Gambar 23 homepage Katalog



Gambar 24 Homepage Katalog



Gambar 25 Homepage Category in Catalog

2.1.2.4 Method ModalAddBook

CLAS S	Method	Kasus dan Hasil Uji (Data normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
Librar y	ModalAddBo ok	<ul style="list-style-type: none"> - input title : Happily Never After - Page : 100 - Status : reading - List : Food For Thoughts 	<ul style="list-style-type: none"> - Sistem melakukan pengecekan apakah judul buku yang diinputkan sudah ada dalam database sistem atau belum - Sistem menampilkan form add book - Sistem menampilkan alert ketika proses tambah buku selesai 	<ul style="list-style-type: none"> - Berhasil mendeteksi buku yang sudah terdaftar dalam database sistem. - Berhasil menampilkan form add book - Berhasil menampilkan alert ketika proses tambah buku selesai 	[<input type="checkbox"/>] diterima

Tabel 8 Hasil uji data normal method ModalAddBook

CLASS	Method	Kasus dan Hasil Uji (Data Salah)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
		- input title: Happily Nevet After	<ul style="list-style-type: none"> - Sistem melakukan pengecekan apakah judul buku yang diinputkan sudah ada dalam database sistem atau belum - Memberikan opsi lanjutan kepada user jika ingin menambahkan data buku baru ke dalam sistem 	<ul style="list-style-type: none"> - Berhasil mendeteksi judul yang tidak terdaftar dalam database sistem - Berhasil menampilkan modal untuk megarahkan user jika ingin menambahkan data buku baru kedalam sistem 	[<input type="checkbox"/>] diterima

Tabel 9 Hasil uji data salah method ModalAddBook

Code

```

1  "use client";
2  import { useEffect, useState } from "react";
3  import "./modal.css";
4  import axios from "axios";
5  import ModalAddBook_SearchTitle from "@/components/ModalAddBook_SearchTitle";
6
7  import ModalAddNewBook_Message from "@/components/ModalAddNewBook_Message";
8  import ModalAddBook from "@/components/ModalAddBook";
9  import ModalAddNewBookForm from "/.../.../.../.../components/ModalAddNewBookForm";
10
11 const Modal = ({ closeModal, userId, API_URL }) => {
12   const [title, setTitle] = useState("");
13   const [bookData, setBookData] = useState({});
14   const [showSearchTitleModal, setShowSearchTitleModal] = useState(true);
15   const [showMessageModal, setShowMessageModal] = useState(false);
16   const [showNewBookToLibModal, setShowNewBookToLibModal] = useState(false);
17   const [showNewBookFormAndLibModal, setShowNewBookFormAndLibModal] =
18     useState(false);
19
20   const handleSearchTitle = async (e) => {
21     e.preventDefault();
22     setShowSearchTitleModal(false);
23     try {
24       const bookTitleResponse = await axios.get(
25         `${API_URL}/api/book/books/one`,
26         {
27           params: {
28             bookTitle: title,
29           },
30         }
31       );
32
33       if (bookTitleResponse.data.message === "Book Not Found") {
34         setShowMessageModal(true);
35       } else {
36         setShowNewBookToLibModal(true);
37         setBookData(bookTitleResponse.data.payload);
38       }
39     } catch (err) {
40       console.error("There's an error during API Request : ", err.message);
41     }
42   };

```

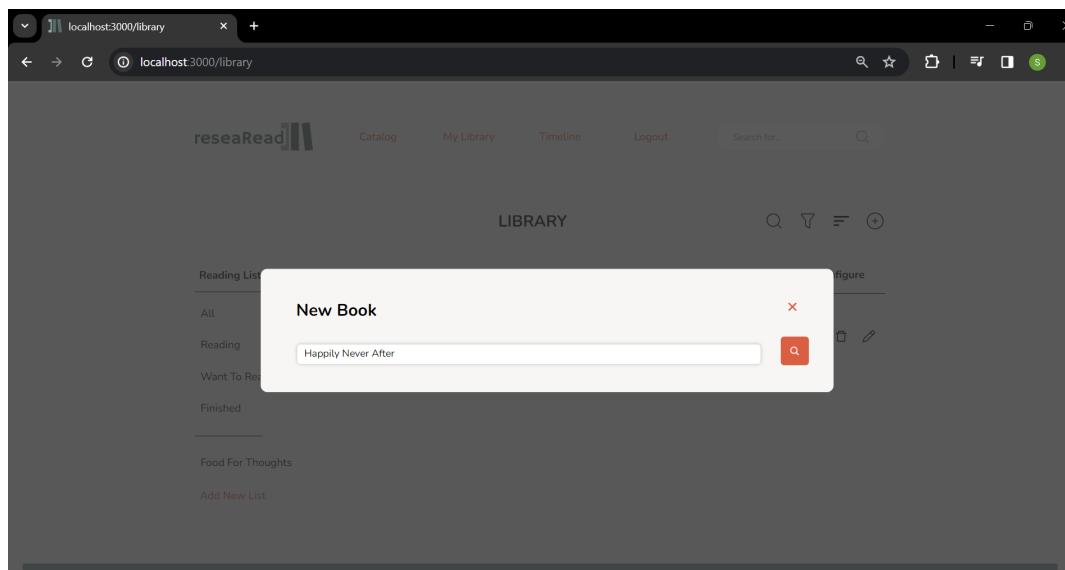
```

44  const onInsertNewBook = () => {
45    setShowMessageModal(false);
46    setShowNewBookFormAndLibModal(true);
47  };
48
49  return (
50    <div className="modal">
51      <div className="overlay"></div>
52      <div className="modal-content">
53        {showSearchTitleModal && (
54          <ModalAddBook_SearchTitle
55            closeModal={closeModal}
56            updateTitleValue={setTitle}
57            handleSearchTitle={handleSearchTitle}
58          />
59        )}
60        {showMessageModal && (
61          <ModalAddNewBook_Message
62            closeModal={closeModal}
63            onInsertNewBook={onInsertNewBook}
64          />
65        )}
66        {showNewBookToLibModal && (
67          <ModalAddBook
68            addedTitle={bookData.title}
69            addedPage={bookData.page}
70            addedCoverImage={bookData.coverImage}
71            closeModal={closeModal}
72            userId={userId}
73            API_URL={API_URL}
74          />
75        )}
76        {showNewBookFormAndLibModal && (
77          <ModalAddNewBookForm
78            addedTitle={title}
79            closeModal={closeModal}
80            userId={userId}
81            API_URL={API_URL}
82          />
83        )}
84      </div>
85    </div>
86  );
87};

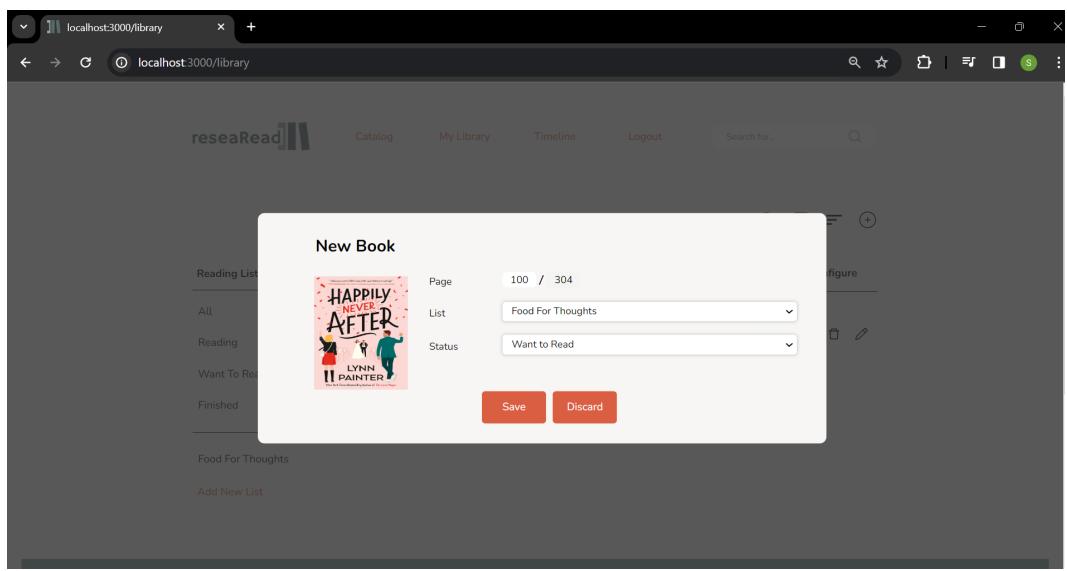
```

Gambar 25 Codingan AddBook

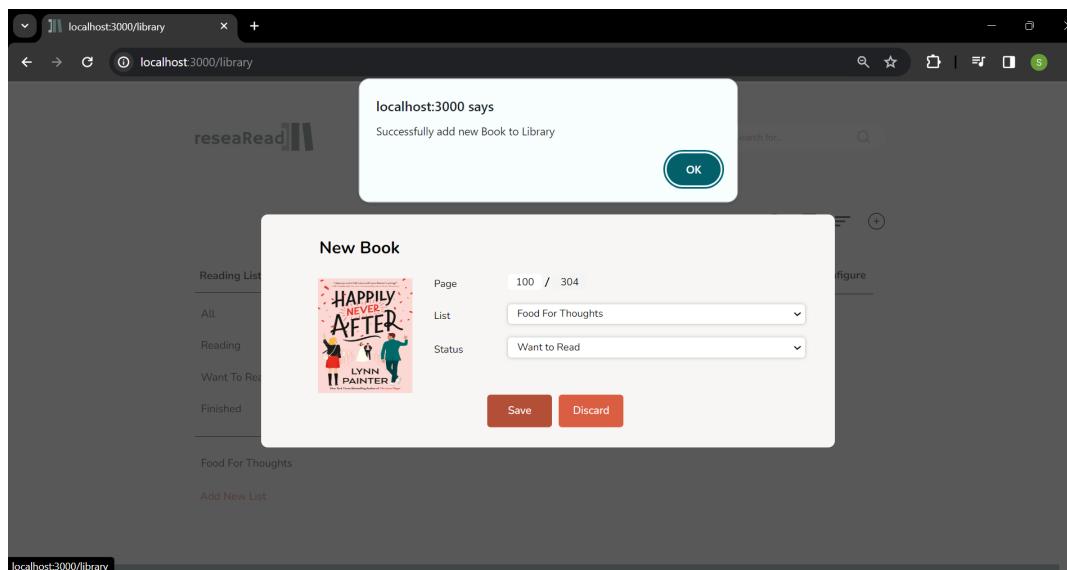
Screenshot hasil pengujian



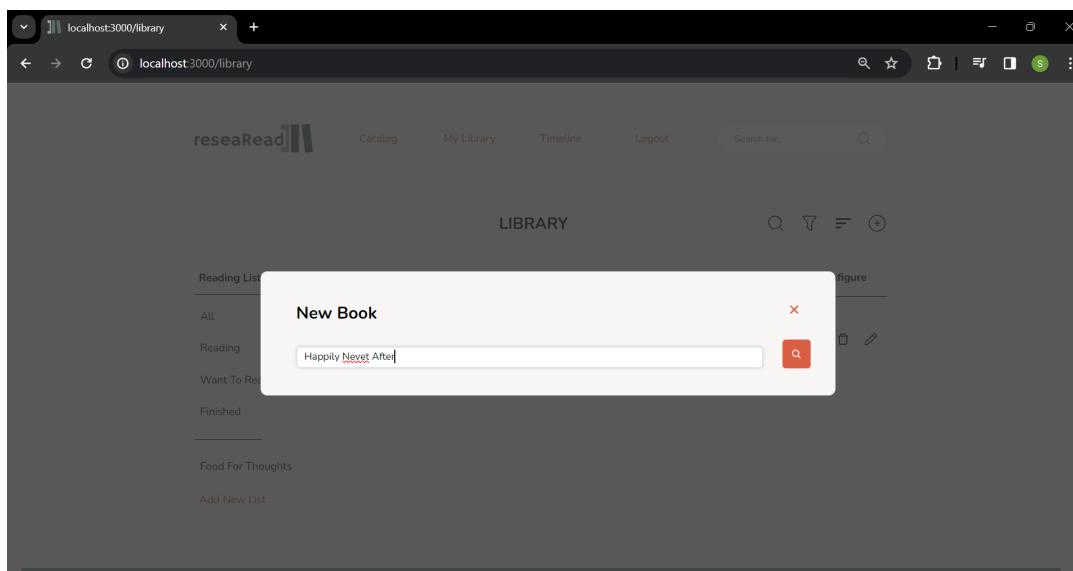
Gambar 26 Field untuk mengisi judul buku



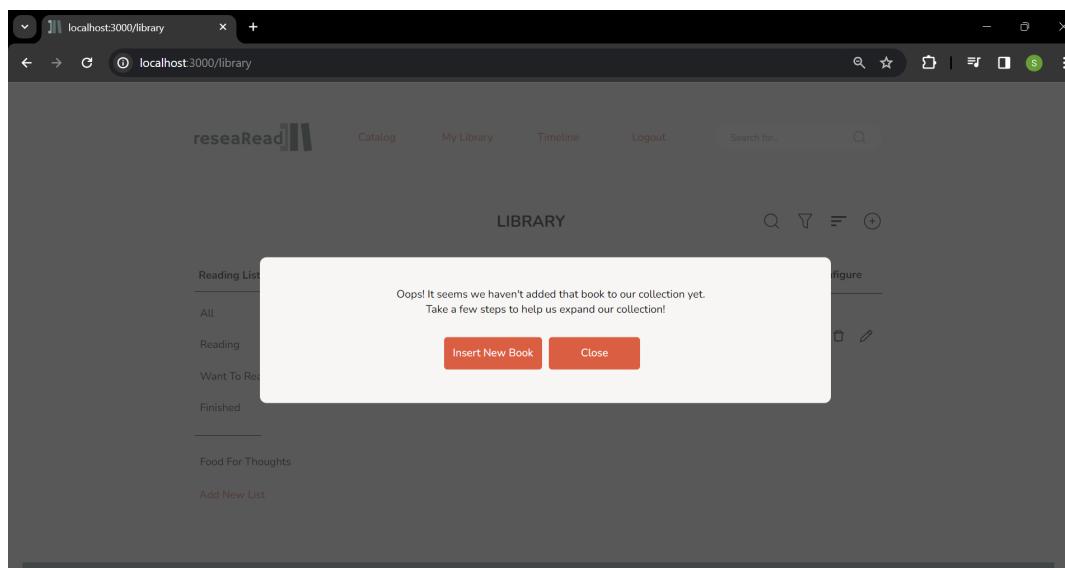
Gambar 27 User mengisi inputan yang diperlukan



Gambar 28 Berhasil AddBook



Gambar 29 Field untuk mengisi judul buku



Gambar 30 Gagal mencari judul buku karena tidak ditemukan di Database ReseaRead

2.1.2.5 Delete Book

CLASS	Method	Kasus dan Hasil Uji (Data normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
Library	DeleteBook	-	<ul style="list-style-type: none"> - Menampilkan modal yang berisi konfirmasi sebelum user menghapus buku dari library - Menghapus buku dari library user 	<ul style="list-style-type: none"> - Berhasil menampilkan model konfirmasi sebelum proses penghapusan dilakukan - Berhasil menampilkan alert yang berisi pesan "Successfully remove a Book from your Library" setelah 	[<input type="checkbox"/>] diterima

CLASS	Method	Kasus dan Hasil Uji (Data normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
				<p>proses penghapusan dilakukan</p> <ul style="list-style-type: none"> - Berhasil menghapus buku dari list library user. 	

Tabel 10 Hasil uji data normal method DeleteBook

Code

```

1  "use client";
2  import { useRouter } from "next/navigation";
3  import Button from "../../components/Button";
4  import "./modal-deletebook.css";
5  import axios from "axios";
6
7  const API_URL = process.env.NEXT_PUBLIC_API_URL;
8
9  const ModalEditBook = ({ closeModalDelete, libraryId }) => {
10    const removeBook = async () => {
11      try {
12        console.log("Test");
13        const removeBookInLibResponse = await axios.delete(
14          `${API_URL}/api/library/remove/${libraryId}`
15        );
16
17        if (removeBookInLibResponse.status === 200) {
18          alert("Successfully remove a book from your Library");
19          window.location.reload();
20        }
21      } catch (err) {
22        console.log("There's is a a problem when removing: ", err);
23      }
24    };

```

```

26     return (
27       <div className="modal">
28         <div className="overlay">
29           <div className="modalcontent">
30             <div className="mb-3">
31               |   <p>Are you sure you want to remove this book from your Library?</p>
32             </div>
33
34             <div className="mt-2 ml-2">
35               <div className="mt-3 mb-5 modal-button">
36                 <Button
37                   buttonName="Yes"
38                   color="white"
39                   targetPage="/library"
40                   width={"50px"}
41                   handleButton={removeBook}
42                 />
43                 <button onClick={closeModalDelete}>
44                   <Button
45                     buttonName="No"
46                     color="white"
47                     targetPage="/library"
48                     width={"50px"}
49                   />
50                 </button>
51               </div>
52             </div>
53           </div>
54         </div>
55       </div>
56     );

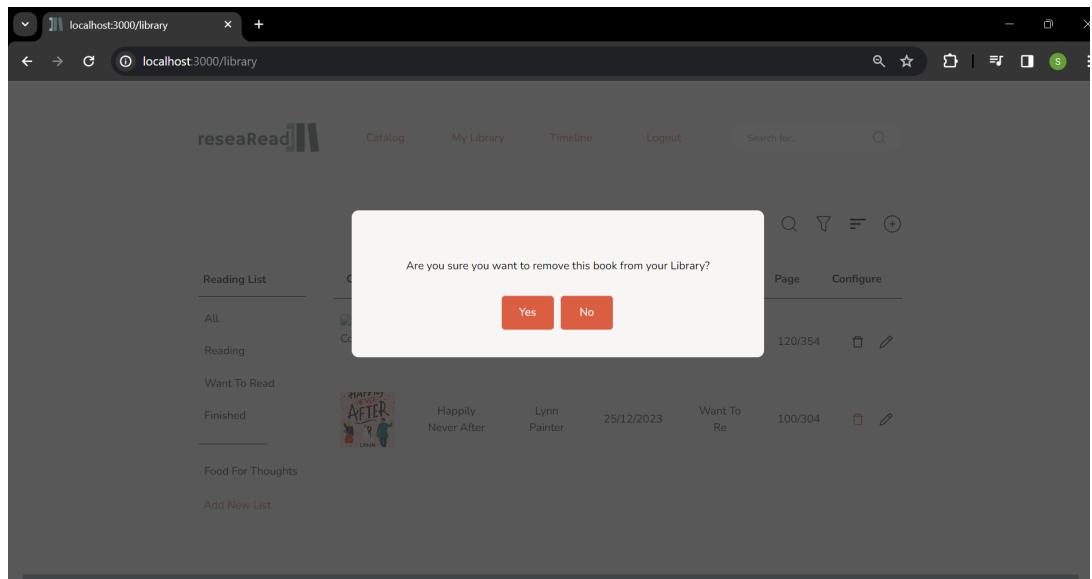
```

Gambar 31 Codingan Delete Book

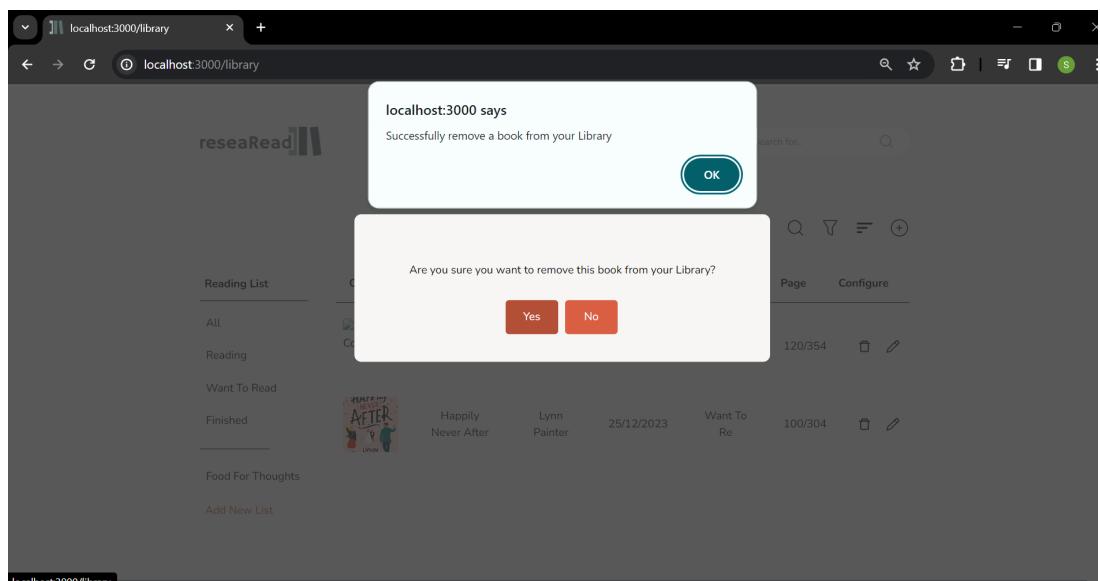
Screenshot hasil pengujian method

Reading List	Cover	Title	Author	Date Added	Status	Page	Configure
All Reading		Happily PPM After	noraeif	25/12/2023	Reading	120/354	
Want To Read		Happily Never After	Lynn Painter	25/12/2023	Want To Re	100/304	

Gambar 32 Homepage Library



Gambar 33 Field apakah yakin menghapus buku dari library



Gambar 34 Berhasil menghapus buku

2.1.2.6 addBook (menambahkan data buku baru ke database sistem)

CLASS	Method	Kasus dan Hasil Uji (Data normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
library	addBook	<ul style="list-style-type: none"> - bookTitle : Midnight Library - author : Matt Haig - page : 20/328 - coverUrl : https://cdn.gramedia.com/uploads/ite 	Dapat menampilkan alert yang berisi pesan "Successfully add new book to library"	Berhasil menambahkan buku baru kedalam database sistem dan library user	[<input type="checkbox"/>] diterima

CLASS	Method	Kasus dan Hasil Uji (Data normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
		ms/978602064932 0_the_midiight_li brary_cov.jpg - list : Food For Thoughts - status : Reading			

Tabel 11 Hasil uji data normal method addBoook

CLASS	Method	Kasus dan Hasil Uji (Data salah)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
		- bookTitle : - author : Matt Haig - page : 20/328 - coverUrl : https://cdn.gramedia.com/uploads/items/9786020649320_the_midiight_library_cov.jpg - list : Food For Thoughts - status : Reading	Dapat menampilkan alert yang berisi pesan “Book data must be filled”	Berhasil mendeteksi data buku yang belum diinputkan user	<input type="checkbox"/> diterima

Tabel 12 Hasil uji data salah method addBook

Code

```
const handleAddingNewBook = async (e) => {
  e.preventDefault();

  try {
    const addBookToSystem = await axios.post(` ${API_URL}/api/book/addBook` , {
      author,
      description: "There is no description yet.",
      rating: 0,
      genre: "Other",
      coverImage: coverBook,
      page: totalPages,
      title,
    });
  }

  //alert(addBookToSystem.data.message);

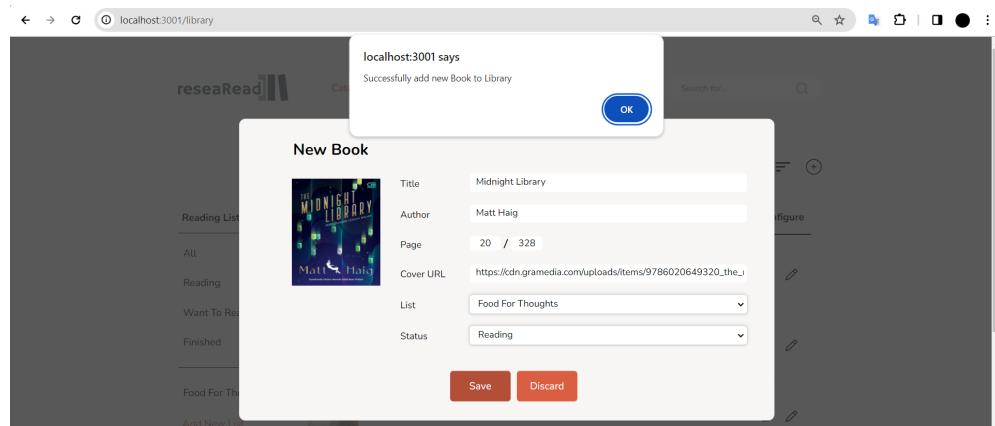
  if (addBookToSystem.status === 200) {
    await axios.post(` ${API_URL}/api/library/addBook` , {
      userId,
      title,
      readStatus: status,
      pageProgress: currentPage,
      listType,
    });

    alert("Successfully add new Book to Library");
    window.location.reload();
  }
}

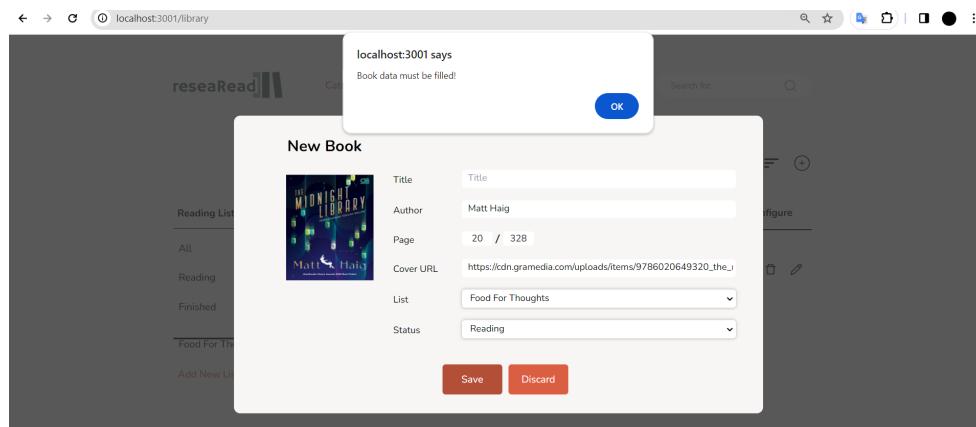
} catch (err) {
  console.error("Error when add new book", err.message);
  alert("Book data must be filled!");
}
```

Gambar 35 Codingan addBook

Screenshot hasil pengujian method



Gambar 36 Berhasil Menambahkan buku



Gambar 37 Tidak berhasil menambahkan buku karena judul tidak dimasukkan

2.1.2.7 getAllBookByStatus

CLASS	Method	Kasus dan Hasil Uji (Data normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
library	getAllBookByStatus	Klik section reading pada reading list di halaman library	Menampilkan buku yang statusnya reading	Berhasil menampilkan list buku dengan status reading	[<input type="checkbox"/>] diterima
		Klik section finished pada reading list di halaman library	Menampilkan buku yang statusnya finished	Berhasil menampilkan list buku dengan status finished	[<input type="checkbox"/>] diterima

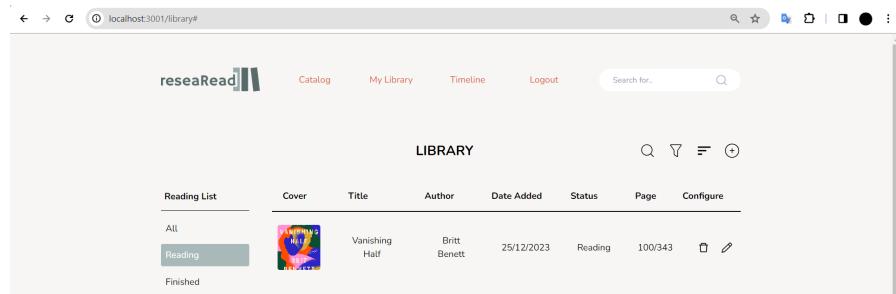
Tabel 13 Hasil uji data normal method getAllBookByStatus

Code

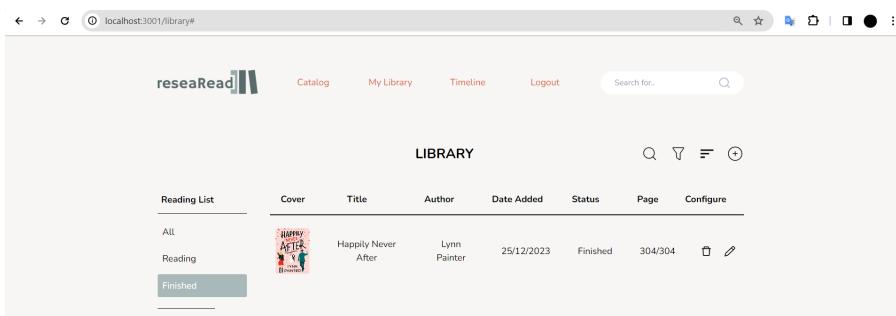
```
useEffect(() => {
  const fetchUserLibrary = async () => {
    try {
      const response = await axios.get(`${API_URL}/api/library/${userId}`);
      const thisUserBook = response.data.libraryBooks;
      const filteredLibByStatus =
        statusType === "All"
          ? thisUserBook
          : thisUserBook.filter(
              (bookInLib) => bookInLib.readStatus === statusType
            );
      setUserLibrary(filteredLibByStatus);
    } catch (err) {
      console.log(err);
    }
  };
  fetchUserLibrary();
}, [userId, API_URL, statusType]);
```

Gambar 38 Codingan getAllBookByStatus

Screenshoot hasil pengujian method



Gambar 39 Homepage Library bagian Reading



Gambar 40 Homepage Library bagian Finished

2.1.2.8 Method Logout

CLASS	Method	Kasus dan Hasil Uji (Data normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
user	logout	klik menu logout pada navbar	Dapat menampilkan alert yang berisi pesan "Successfully"	Dapat menampilkan alert yang berisi pesan "Successfully"	[<input type="checkbox"/>] diterima

CLASS	Method	Kasus dan Hasil Uji (Data normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
			logout” kemudian beralih ke menu login	logout”	

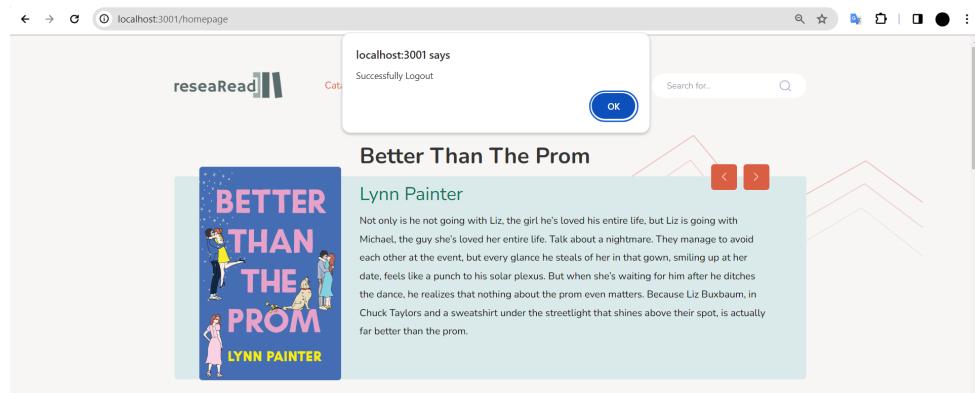
Tabel 14 Hasil uji data normal method logout

Code

```
const handleLogout = async () => {
  await AsyncStorage.removeItem("token");
  alert("Successfully Logout");
  router.push("/login");
};
```

Gambar 41 Codingan Logout

Screenshot hasil pengujian method



Gambar 42 Field LogOut berhasil

2.1.2.9 Method Search

CLASS	Method	Kasus dan Hasil Uji (Data normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
user	search	“Normal People”	<ul style="list-style-type: none"> - Searchbar dapat diisi oleh user untuk menuliskan inputan pencarian - Dapat menampilkan hasil pencarian yang sesuai dengan inputan 	<ul style="list-style-type: none"> - User dapat menuliskan keyword pencarian pada searchbar - 	[X] ditolak

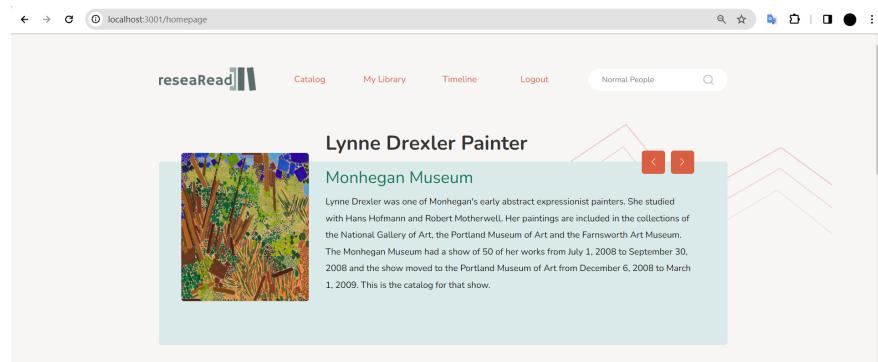
Tabel 15 Hasil uji data normal method search

Code

```
<Form className="search-bar w-full m-0">
  <input type="text" placeholder="Search for tags... " />
  <button type="submit">
    <Image
      src="/icon-search.png"
      alt="icon-search"
      width={20}
      height={20}
    />
  </button>
</Form>
```

Gambar 43 Codingan Search

Screenshot hasil pengujian



Gambar 44 Homepage Search

2.1.2.10 Method bookDetails

CLASS	Method	Kasus dan Hasil Uji (Data normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
book	bookDetails	-	- Menampilkan detail buku saat user klik judul buku	- Book details ditampilkan	[<input type="checkbox"/> diterima]

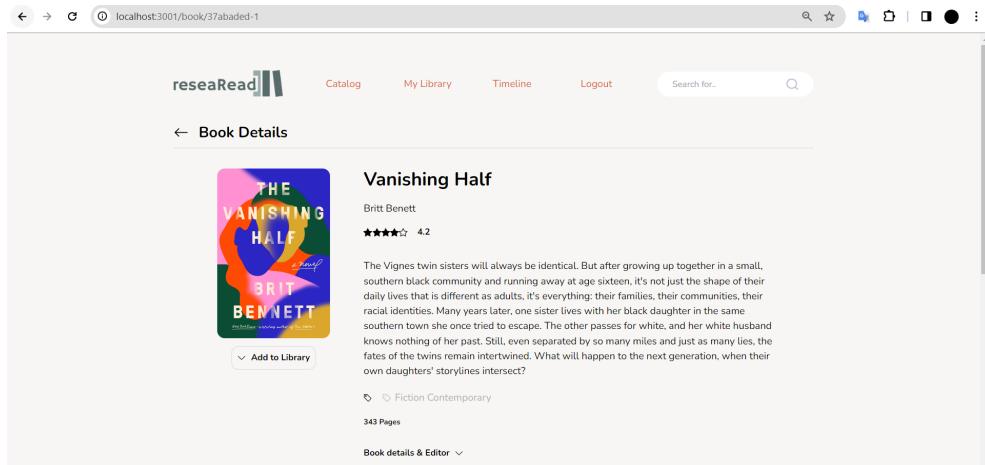
Tabel 16 Hasil uji data normal method bookDetails

Code

```
return (
  <main className="mt-8 container mx-auto max-w-screen-lg">
    <div className="w-full">
      <div className="font-bold flex items-center space-x-4 pt-2 pb-2 text-customDefaultText text-2xl w-full border-b-2">
        <button
          onClick={() => {
            router.push('/homepage');
          }}
        >
          <BsArrowLeft />
        </button>
        <div className="cursor-pointer">Book Details</div>
      </div>
    </div>
    <div className="mt-4">
      {bookPage && userId && (
        <BookDetail data={bookPage} userId={userId} API_URL={API_URL} />
      )}
    </div>
  </main>
);
```

Gambar 45 Codingan BookDetails

Screenshot Pengujian



Gambar 46 Tampilan BookDetails

2.2 Pengujian INTEGRASI

2.2.1 Usecase sign in

Use case	Kasus dan Hasil Uji (Data normal)			
	Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Sign in	username : Veroes email : Veroes@resea.com password : 123456789	<ol style="list-style-type: none"> 1. Fungsi handleRegister melakukan validasi data 2. User yang inputan datanya sudah divalidasi akan dialihkan ke proses login 3. AsyncLocalStorage.setItem 4. Setelah token berhasil di set, user dialihkan ke homepage 	<ul style="list-style-type: none"> - Berhasil menampilkan register page - Field text pada register page dapat diisi oleh user - Sistem menampilkan alert message “Register Successfull” untuk merespon data inputan yang sudah sesuai - User dialihkan ke homepage 	[<input type="checkbox"/>] diterima

Tabel 17 Hasil uji data normal usecase sign in

Use case	Kasus dan Hasil Uji (Data Salah)			
	Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Sign in	username : Veroes email : Veroes@resea.com password : 123	<ol style="list-style-type: none"> 1. Fungsi handleRegister melakukan validasi data 2. User yang inputan datanya tidak sesuai akan mendapatkan alert message 	<ul style="list-style-type: none"> - Berhasil menampilkan register page - Field text pada register page dapat diisi oleh user - Sistem menampilkan alert message “Please lengthen this text to 8 characters or more” untuk merespon data inputan password yang tidak sesuai 	[<input type="checkbox"/>] diterima

Tabel 18 Hasil uji data salah usecase sign in

2.2.2 UseCase Library

Use case	Kasus dan Hasil Uji (Data normal)			
	Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
library	-	<ol style="list-style-type: none"> 1. Fungsi handleAddnewBook akan memvalidasi buku yang akan ditambahkan 2. User yang sudah menambahkan buku akan diarahkan ke halaman library baru dimana terdapat buku yang telah ditambahkan 3. Fungsi ModalEditBook akan memvalidasi buku yang ada di dalam library ketika akan diedit 4. User yang sudah mengedit buku akan diarahkan ke halaman library dengan buku yang sudah teredit. 5. Fungsi removeBook akan memvalidasi buku yang dihapus 6. User yang sudah menghapus buku akan diarahkan ke halaman library buku yang baru 	<ul style="list-style-type: none"> - Berhasil menampilkan library page - button addnewbook dapat diklik dan user dapat menginputkan judul di field inputan - Sistem menampilkan alert message “Successfully add new book to Library” - Berhasil menampilkan field yang berisi (input title, input author, input page, input cover url, input list, input status) yang dapat diisi oleh user - Berhasil menampilkan field pesan apakah yakin atau dihapus atau tidak. - User dapat menekan tombol yes - Dan akan muncul alert pesan bahwa buku telah di remove 	[<input type="checkbox"/>] diterima

Tabel 19 Hasil uji data normal usecase library

2.2.3 Usecase Catalog

Use case	Kasus dan Hasil Uji (Data normal)			
	Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Catalog	-	<ol style="list-style-type: none"> 1. Komponen Category akan dipanggil di halaman ini untuk menampilkan section buku berdasarkan kategori 2. Pada komponen Trending, fungsi fetchTrendingBook akan mengambil data buku yang sedang trend dari database melalui API. Data tersebut kemudian disimpan dalam variabel bookData. Setelah itu, fungsi setTrendingBook dipanggil untuk menyimpan data buku yang sedang trend. 3. Komponen Trending dipanggil di halaman ini untuk menampilkan daftar buku pada section trending 	<ul style="list-style-type: none"> - Halaman catalog berhasil menampilkan book category - Halaman catalog berhasil menampilkan daftar buku trending 	[<input type="checkbox"/>] diterima

Tabel 20 Hasil uji data normal usecase catalog

2.2.4 Usecase Book Search

Use Case	Kasus dan Hasil Uji (Data normal)			
	Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
	<ul style="list-style-type: none"> - Search: Bring Young Women 	<ol style="list-style-type: none"> 1. SearchBar diisi dengan judul buku yang ingin dicari. 2. SearchResultSelecton akan menampilkan hasil pencarian sesuai yang diinputkan 3. User melihat daftar buku yang relevan dengan pencarian user 	<ul style="list-style-type: none"> - SearchBar dapat diisi oleh user dan dapat digunakan untuk mencari buku - Hasil pencarian buku berfungsi dengan baik. - Menampilkan hasil sesuai dengan judul yang dicari user 	[<input type="checkbox"/>] diterima

Tabel 20 Hasil uji data normal use case search

2.2.5 Usecase addBook

Use Case	Kasus dan Hasil Uji (Data normal)			
	Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
AddBook	book title : Happily Never After	<ol style="list-style-type: none"> 1. fungsi icon button tambah agar user dapat dengan cepat mencari buku yang ingin ditambahkan sesuai judul buku tersebut 2. button search akan menampilkan buku yang dicari 3. user dapat menambahkan buku sesuai list dan status buku tersebut, jika jadi ditambahkan fungsi button “save”, jika user ingin menyimpan buku tersebut untuk ditambahkan ke dalam library 4. fungsi button “discard”, jika user tidak jadi menambahkan buku tersebut 5. jika buku berhasil ditambahkan akan muncul modal dimana pesan “Successfully add new book to library” 	<ul style="list-style-type: none"> - icon button tambah berhasil menampilkan modal input add new book - button search berhasil menampilkan buku yang sedang dicari - list dan status buku dapat dipilih sesuai dengan kebutuhan user - button save berhasil menyimpan buku yang ingin ditambahkan ke dalam library - button discard berhasil kembali ke menu my library jika tidak ingin ditambahkan ke dalam library 	[<input type="checkbox"/>] diterima

Tabel 21 Hasil uji data normal usecase addbook

Use Case	Kasus dan Hasil Uji (Data Salah)			
	Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
	Buku tidak ada dalam database title Buku : “Aku”	akan muncul pop up window yang akan memberitahukan pesan jika buku tidak terdapat di dalam collection database. jika ingin tetap menambahkan buku terdapat button “Insert new book” yang akan menambahkan buku ke dalam sistem di database dan ada button “cancel” jika tidak ingin menambahkan buku di collection database	modal berhasil ditampilkan	[<input type="checkbox"/>] diterima

Tabel 22 Hasil uji data salahusecase addbook

2.2.6 Usecase profile setting

Use Case	Kasus dan Hasil Uji (Data normal)			
	Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
profile setting	Pada account masukkan nama : Tanggal lahir : Jenis Kelamin : Email : username : password :	Data dalam account bisa berubah sesuai data yang diinputkan	Dapat menampilkan data account sesuai	[<input type="checkbox"/>] diterima
	Klik button “save” data account user bisa update sesuai inputan terbaru	Data account bisa tersimpan di sistem account	Dapat menampilkan data account sesuai perubahan terakhir	[<input type="checkbox"/>] diterima

Tabel 23 Hasil uji data normal use case profile setting

Use Case	Kasus dan Hasil Uji (Salah)			
	Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
profile setting	masukkan data salah password tidak sesuai karakter	menampilkan modal dimana memberikan pesan jika password tidak sesuai dengan karakter	dapat menampilkan modal pesan dimana memberikan pesan jika password tidak sesuai dengan karakter	

Tabel 24 Hasil uji data salah use case profile setting

2.2.7 Usecase homepage

Use Case	Kasus dan Hasil Uji (Data normal)			
	Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
homepage	<ul style="list-style-type: none"> • API_URL • Token => userId 	<p>Dengan menggunakan API_URL, Featured Book bisa ditampilkan di layar dengan salah satu fungsi book.controller => getOneBook</p> <p>Dengan menggunakan API_URL dan userId buku library user (<i>up to 3</i>) ditampilkan ke komponen Continue Reading</p> <p>Dengan menggunakan API_URL, TopPicks ditampilkan ke layar</p>	Featured book, Continue Reading, dan TopPicks bisa dimunculkan	[<input type="checkbox"/>] diterima

Tabel 25 Hasil uji data salah normal case homepage

2.2.8 Pengujian Log out

USE CASE	Kasus dan Hasil Uji (Data normal)			
	Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
logout	Token User dari decodeTokenAndSEt UserId()	<p>Token pada browser untuk situs reseaRead di remove.</p> <p>Redirect User ke login page.</p>	Sistem menampilkan ke layar User bahwa berhasil logout.	[<input type="checkbox"/>] diterima

Tabel 26 Hasil uji data salah normal case logout

2.3 USER ACCEPTANCE TEST (Berjuang mencari user potensial)

USER ACCEPTANCE TEST OLEH : PETUGAS PELAKSANA PENGUJIAN
TANGGAL : 25 Desember 2023

Use Case Yang Diuji	Rencana Pengujian	Hasil Yang Diharapkan	Hasil Aktual	Kesimpulan
Sign up	<ol style="list-style-type: none"> 1. Mendaftarkan akun baru kedalam sistem dengan data normal 2. Mendaftarkan akun baru kedalam sistem dengan data yang kurang sesuai 	<ol style="list-style-type: none"> 1. Registrasi akun berhasil 2. Sistem dapat mendeteksi jika terdapat kesalahan dalam inputan data registrasi 	<ol style="list-style-type: none"> 1. User berhasil melakukan registrasi akun 2. Kesalahan inputan dapat dideteksi 	Modul registrasi normal

Sign in	<ol style="list-style-type: none"> Melakukan sign in menggunakan data akun terdaftar Melakukan sign in dengan data akun yang belum terdaftar Melakukan sign in dengan inputan yang tidak sesuai 	<ol style="list-style-type: none"> Sign in berhasil Sistem mendeteksi data akun yang belum terdaftar Sistem mendeteksi kesalahan inputan data sign in 	<ol style="list-style-type: none"> User berhasil melakukan sign in Sistem berhasil mendeteksi akun yang belum terdaftar Kesalahan inputan dapat dideteksi 	Modul login normal
Melihat katalog	<ol style="list-style-type: none"> Menampilkan kategori buku Menampilkan daftar buku trending pada section trending 	<ol style="list-style-type: none"> Sistem menampilkan section book category Sistem menampilkan daftar buku pada section trending 	<ol style="list-style-type: none"> Section book category berhasil ditampilkan Section trending berhasil ditampilkan Preview data buku pada section trending belum tertampil sempurna 	Modul catalog perlu perbaikan
Menambah buku baru ke dalam Sistem	<ol style="list-style-type: none"> Memasukkan data buku dengan lengkap kedalam form addbook ke database Memasukkan data buku yang tidak lengkap kedalam form addbook ke database 	<ol style="list-style-type: none"> Sistem memberikan alert atau pesan error jika terdapat kesalahan inputan oleh user saat mengisi form add book Sistem menambahkan data buku baru kedalam database sistem 	<ol style="list-style-type: none"> alert atau pesan error muncul jika terdapat kesalahan inputan oleh user saat mengisi form add book Buku berhasil ditambahkan kedalam database sistem 	Modul addBook ke dalam sistem normal
Menambahkan buku ke library	<ol style="list-style-type: none"> Memasukkan judul buku yang ingin ditambahkan Memasukkan data buku yang ingin ditambahkan 	<ol style="list-style-type: none"> Sistem melakukan pengecekan judul buku Sistem menampilkan form addbook Sistem menampilkan modal untuk mengarahkan user melakukan proses tambah data buku ke library jika buku belum terdaftar Sistem menambahkan buku ke dalam library 	<ol style="list-style-type: none"> Sistem berhasil mendeteksi judul buku yang sudah/belum terdaftar dalam database Sistem berhasil menampilkan form add book Sistem berhasil memunculkan modal Sistem berhasil menambahkan buku ke dalam library 	modul tambah buku ke library normal
Menghapus buku dari library	<ol style="list-style-type: none"> User menuju halaman library User menekan icon delete yang berada pada halaman library 	<ol style="list-style-type: none"> Sistem Menampilkan modal yang berisi konfirmasi sebelum user menghapus buku dari library Sistem mendeteksi user mengkonfirmasi untuk menghapus buku atau tidak. 	<ol style="list-style-type: none"> Sistem berhasil menampilkan model konfirmasi sebelum proses penghapusan dilakukan Sistem berhasil menampilkan alert yang berisi pesan "Successfully" 	modul hapus buku dari library normal

		3. Sistem menghapus buku dari library user	remove a Book from your Library” setelah proses penghapusan dilakukan 3. Sistem berhasil menghapus buku dari list library user.	
Menampilkan buku berdasarkan kategori status	1. User klik ‘reading’ pada menu list	1. Sistem menampilkan buku yang statusnya sesuai	Sistem berhasil menampilkan buku yang statusnya sesuai	Modul Reading list normal
Log out	1. User keluar dari akun web Researread	2. User dialihkan ke halaman login	1. User berhasil log out dan dialihkan ke halaman login	Modul logout normal

Tabel 27 User acceptance test

2.4 Summary Pengujian Yang Masih Gagal

(Berisi laporan dari pengujian yang telah dilakukan, dengan menyampaikan informasi status dari setiap fungsional yang masih gagal)

Kelas Uji	Butir Uji	Kesimpulan pengujian
Melihat Katalog	Tidak berhasil menampilkan keterangan tag book yang seharusnya ada pada buku yang tertampil di section book trending	Ditolak
Melihat Katalog	Tombol see more tidak navigasi ke page manapun	Ditolak
Search	Setelah ikon search di klik, sistem tidak menampilkan hasil pencarian	Ditolak

Tabel 28 Summary pengujian gagal