

DPPL-01

DESKRIPSI PERANCANGAN 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)

Program Studi Informatika
Fakultas Informatika
Jl. Telekomunikasi 1, Dayeuhkolot Bandung

	Prodi S1 Informatika - Fakultas Informatika	Nomor Dokumen		Halaman
		DPPL-01		<i>63/63</i>
		Revisi	0	<i>Tgl: 03/01/24</i>

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

Daftar Isi	5
1. Pendahuluan	7
1.1 Tujuan Penulisan Dokumen	7
1.2 Lingkup Masalah	7
1.3 Referensi	7
2 Perancangan Global	8
2.1 Rencana Lingkungan Implementasi	8
2.2 Deskripsi Arsitektur Perangkat Lunak	8
3 Perancangan Rinci	9
3.1 Realisasi Use Case	9
3.1.1 Melihat Katalog	11
3.1.2 Membaca Rating dan Review	14
3.1.3 Mencari Buku	17
3.1.4 UI Design dan Deskripsi Objek UI #4 Library	21
3.1.5 Membaca Data Buku	25
3.1.6 Menambah Buku Baru ke dalam Sistem	28
3.1.7 Konfigurasi Akun	34
3.1.8 Authentication	37
3.1.9 Featured Book	41
3.1.10 Continue Reading	44
3.1.11 TopPicks	46
3.1.12 Logout	49
3.2 Diagram Kelas Keseluruhan	52
3.3 Perancangan Detail Kelas	53
3.3.1 Tabel Kelas : UserModel	53
3.3.2 Tabel Kelas : LibraryModel	53
3.3.3 Tabel Kelas : BookModel	53
3.3.4 Tabel Kelas : UserPostModel	54
3.3.5 Tabel Kelas : ReviewsModel	54
3.4 Perancangan Algoritma dan/atau Query	55
3.4.1 Perancangan Kelas UserModel	55
3.4.2 Perancangan Kelas LibraryModel	56
3.4.3 Perancangan Kelas BookModel	57
3.4.4 Perancangan Kelas UserPostModel	58
3.4.5 Perancangan Kelas ReviewsModel	59
3.4.6 Perancangan Kelas User	60
3.5 Matriks Keruntutan (Requirement Traceability Matrix)	61

1. Pendahuluan

Dokumen ini dibuat dalam rangka menjelaskan Deskripsi Perancangan Perangkat Lunak (DPPL) pada sebuah aplikasi ReseaRead. Untuk selanjutnya, dokumen ini akan disesuaikan berdasarkan nama nya menjadi DPPL ReseaRead yang dapat dijadikan panduan bagi pengembang dalam membangun sebuah aplikasi yang terorganisir dengan baik

1.1 Tujuan Penulisan Dokumen

Dokumen ini berisi Deskripsi Perancangan Perangkat Lunak (DPPL) atau Software Requirement (SRS) untuk aplikasi ReseaRead. Tujuan dari penulisan dokumen ini adalah untuk memberikan panduan mengenai Perangkat Lunak yang akan dibangun baik berupa gambaran umum maupun penjelasan detail menyeluruh terhadap perangkat lunak pada tahap selanjutnya.

1.2 Lingkup Masalah

DPPL ini bertujuan untuk menjadi acuan terhadap pembuatan aplikasi Books Explorer Application yang bertujuan untuk menyelesaikan masalah pencarian *review* buku atau lainnya yang sulit di internet. nama aplikasi ini adalah ReseaRead.

1.3 Referensi

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

Prodi S1 Informatika Tel-U	DPPL-Diaspora	Halaman 6 dari 61
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

2 Perancangan Global

Pada bagian ini akan dipaparkan terkait deskripsi perancangan global yang akan meliputi penjelasan tentang rancangan lingkungan implementasi, deskripsi komponen, dan deskripsi data yang membangun perangkat lunak.

2.1 Rencana Lingkungan Implementasi

Aplikasi ini diimplementasikan di lingkungan web.

System Side	Technology
Front-End	NextJS
Back-End	ExpressJS + Sequelize (ORM)
DBMS	MariaDB (Development)
Development Tools	PHPMysqlAdmin(XAMPP), VS Code

2.2 Deskripsi Arsitektur Perangkat Lunak

No	Nama Komponen	Keterangan
1	User Interface	Tampilan dari halaman, tombol, elemen UI, dan lainnya.
2	Authentication	Proses user untuk mengakses fitur login, logout, record, hapus akun.
3	Search/pencarian	Halaman untuk mengelola pencarian dan tampilan hasil pencarian buku berdasarkan kriteria yang diberikan pengguna.

4	Book Details	Halaman untuk menampilkan informasi rinci tentang buku tertentu berdasarkan permintaan pengguna.
5	Review	Halaman tempat menyimpan, menampilkan, dan mengelola ulasan dan peringkat buku yang diberikan pengguna.
6	Library	Halaman untuk menyimpan <i>tracking</i> bacaan buku.
7	Featured Book	Menampilkan Buku secara acak di homepage.
8	Continue Reading	Menampilkan buku Library user <i>up to 3</i> di homepage.
9	TopPicks	Menampilkan buku di homepage dengan rating diatas 4.
10	Database	Menyimpan data terkait, user, buku, library user, reviews, postingan user.

3 Perancangan Rinci

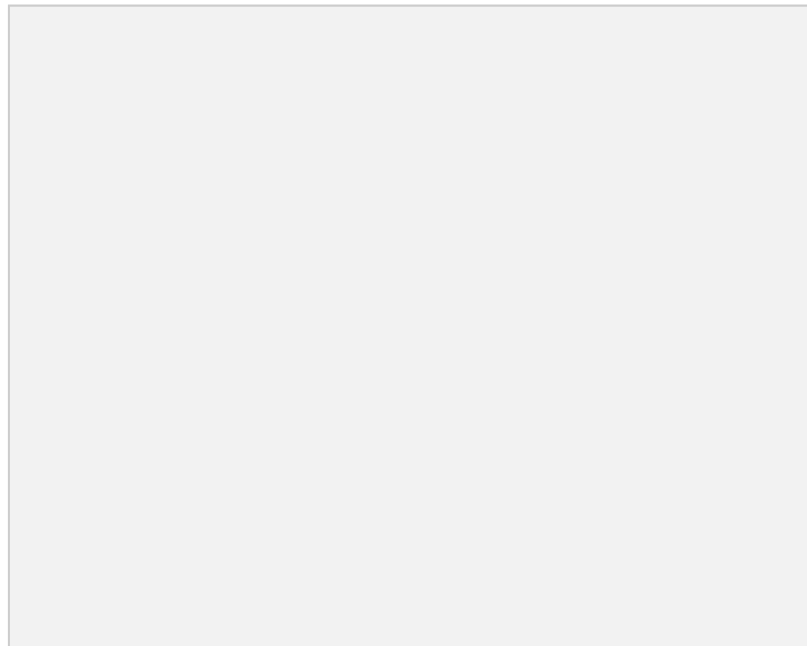
3.1 Realisasi Use Case

No	Nama Use Case	Deskripsi Use Case
1	Catalog	User dapat mengakses katalog untuk melihat buku-buku.
2	Rating & Review	User dapat memberikan rating dan review terkait buku.
3	Book Search	User dapat mencari buku sesuai yang diinputkan.

4	Library	User dapat menggunakan fitur ini sebagai buku <i>tracker</i> personal. Tiap buku juga bisa di set status nya sesuai dengan yang ada di aplikasi.
5	Book Detail	User dapat membaca review dan melihat rating yang telah diberikan oleh user lainnya.
6	Add Book	User bisa menambahkan buku ke aplikasi ReseaRead jika buku yang dicari oleh User tidak dimiliki oleh aplikasi.
7	Setting	User dapat mengkonfigurasi akun nya.
8	Authentication	User dapat register dan login di aplikasi ReseaRead serta set token ketika User sudah login.
9	Featured Book	User dapat menerima buku <i>featured</i> yang berada homepage.
10	Continue Reading	User dapat melihat buku Library nya di homepage <i>up to</i> 3 buku.
11	TopPicks	User dapat menerima 4 buku Top Picks dengan rating > 4.
12	Logout	User dapat Logout dari aplikasi ReseaRead.

3.1.1 Melihat Katalog

3.1.1.1 Use Case Scenario #1 : Melihat Katalog

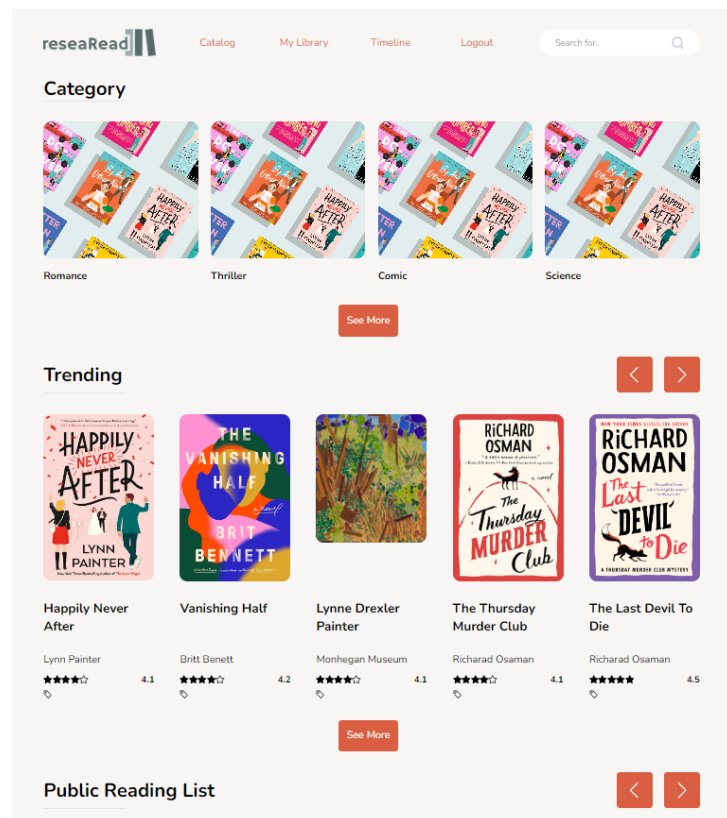


Skenario Use Case Melihat Katalog :

Pre-Condition:	<ul style="list-style-type: none">• Sistem menampilkan halaman katalog / browse pada website.• Halaman katalog / browse dapat diakses jika aktor sudah login.
Post-Condition:	<ul style="list-style-type: none">• Aktor dapat melihat katalog buku berdasarkan hasil pencarian / hasil rekomendasi pencarian / pada berbagai halaman website yang menampilkan katalog buku.
Use Case Description	
a. Primary Flow	<ol style="list-style-type: none">1. Aktor mengunjungi Homepage.2. Aktor menekan Catalog.3. Aktor melihat buku yang tertampil pada

	Katalog.
b. Alternative Flow	<ul style="list-style-type: none"> • Aktor dapat melihat katalog buku pada halaman browse. • Aktor dapat melihat katalog buku pada hasil pencarian. • Aktor dapat melihat katalog buku berdasarkan hasil rekomendasi ketika melakukan pencarian buku pada halaman pencarian.

3.1.1.2 UI Design dan Deskripsi Objek UI #1 Melihat Katalog



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
CTL	CatalogPage	Halaman Catalog

Catalog Page

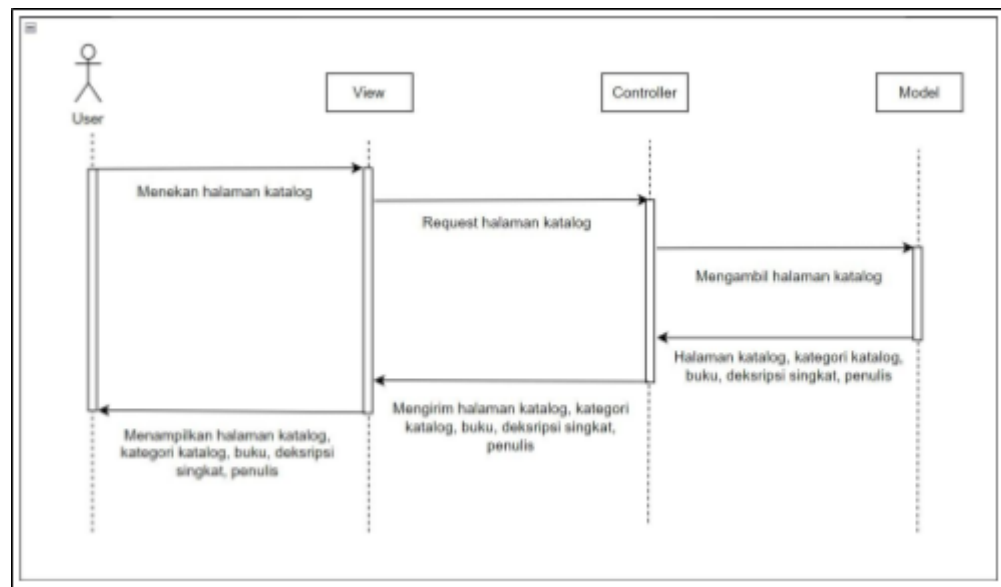
Id_Objek	JENIS	LABEL	Keterangan
Category	Component	See More	Fungsi untuk menampilkan buku -buku berdasarkan kategori.
Trending	Component	See More	Menampilkan buku yang yang trending

3.1.1.3 Identifikasi Object Baru & Tipe Kelas #1 Melihat Katalog

Tabel Object perancangan

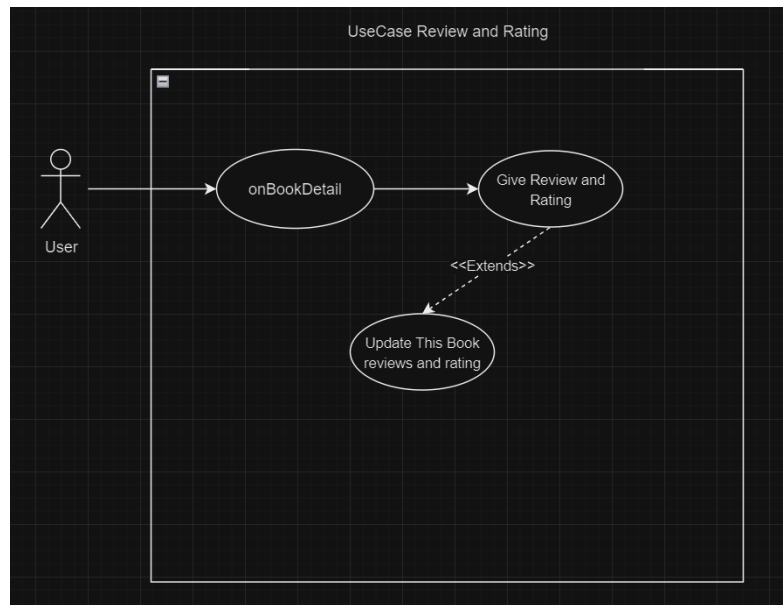
No	Nama Object Baru	Jenis / Tipe Kelas
1.	new Book	Book

3.1.1.4 Sequence Diagram #1 : Melihat Katalog



3.1.2 Membaca Rating dan Review

3.1.2.1 Use Case Scenario #2 : Membaca Rating dan Review

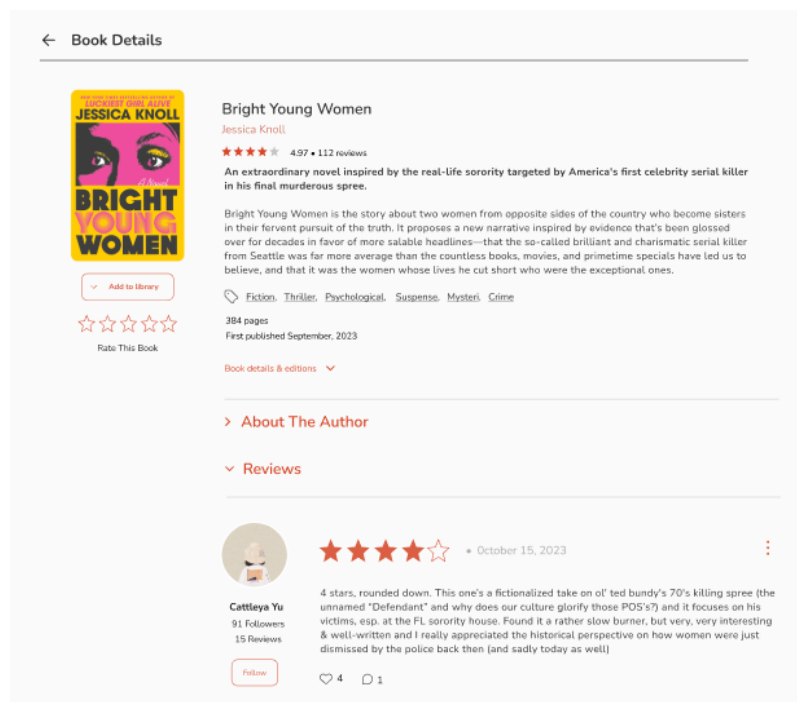


Skenario Use Case Membaca review dan rating :

Pre-Condition:	Sistem sudah menampilkan halaman book details
Post-Condition:	Sistem telah memastikan bahwa review dan rating yang ditampilkan adalah data yang valid tanpa ada manipulasi sehingga sistem mendapatkan informasi yang akurat tentang buku-buku yang ada.
Use Case Description	
Primary flow	<ol style="list-style-type: none">1. Aktor membuka sistem library kemudian mencari buku yang diinginkan.2. Aktor berhasil menemukan hasil pencarian yang diinginkan3. Sistem menampilkan halaman detail buku

	<p>4. Aktor melihat bagian review dan rating yang tersedia pada halaman detail</p> <p>5. Aktor membaca review yang tersedia guna mendapatkan informasi tentang pengalaman aktor lain dengan buku yang diinginkan.</p>
Secondary flow	-

3.1.2.2 UI Design dan Deskripsi Objek UI #2 Membaca Rating dan Review



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
BDETAIL	BookDetailPage	Halaman detail buku

Prodi S1 Informatika Tel-U	DPPL-Diaspora	Halaman 14 dari 61
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

Catalog Page

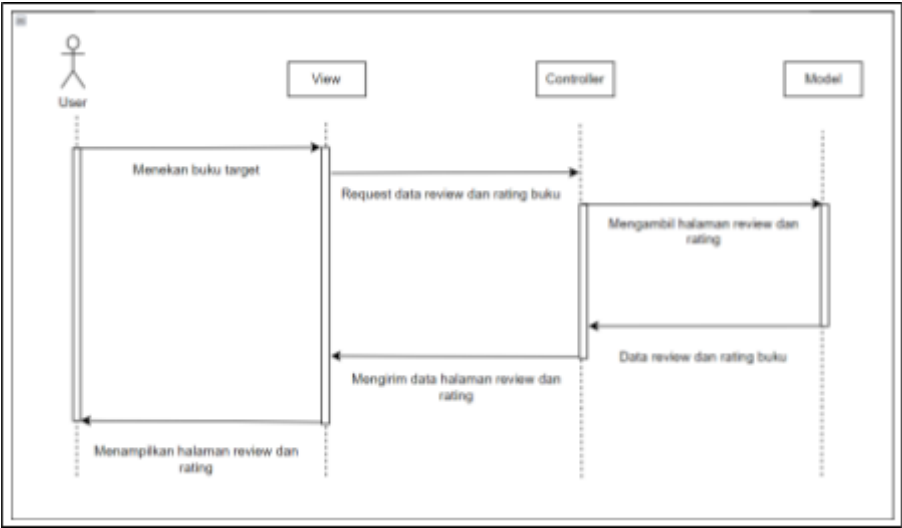
Id_Objek	JENIS	LABEL	Keterangan
Btn_BDetail01	Button	Add to Library	Menambahkan buku ke perpustakaan user
Btn_BDetail02	Rating	Rate this Book	Memberikan rating ke buku
Btn_BDetail03	Button	Edit	Mengedit pesan review pada detail buku
Btn_BDetail04	Button	Hapus	Menghapus pesan pada detail buku
Drop_BDetail01	Dropdown	Book details & editions	Menampilkan detail tambahan
Drop_BDetail02	Dropdown	About The Author	Menampilkan profil penulis
Drop_BDetail03	Dropdown	Reviews	Menampilkan review buku tersebut

3.1.2.3 Identifikasi Object Baru & Tipe Kelas #2 Membaca Rating dan Review

Tabel Object Perancangan

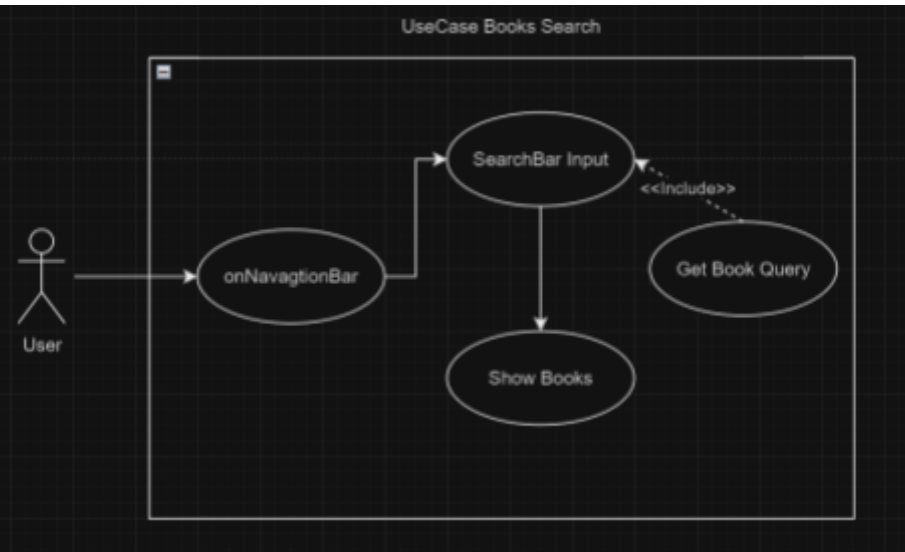
No	Nama Object Baru	Jenis / Tipe Kelas
1.	New	Book

3.1.2.4 Sequence Diagram #2 : Membaca Rating dan Review



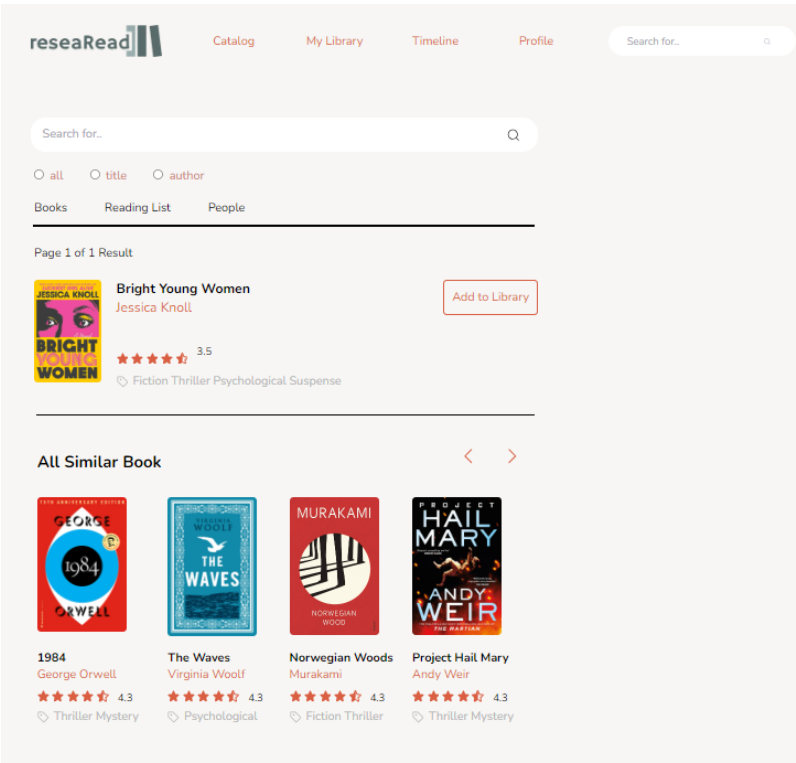
3.1.3 Mencari Buku

3.1.3.1 Use Case Scenario #3 : Mencari Buku



Pre Condition	<ul style="list-style-type: none"> • Sistem menampilkan halaman pencarian pada website. • Halaman pencarian dapat diakses jika User sudah login (mempunyai token)
Use Case Description	
a. Primary Flow	<ol style="list-style-type: none"> 1. Aktor menginputkan buku di Navigation Bar. 2. Aktor Menekan tombol "cari". 3. Aktor Menerima rekomendasi buku sesuai input dari Aktor.
b. Alternative Flow	<p>Aktor berada pada catalog</p> <ol style="list-style-type: none"> 1. Aktor dapat melihat katalog buku pada halaman browse. 2. Aktor dapat melihat katalog buku pada hasil pencarian. 3. Aktor dapat melihat katalog buku berdasarkan hasil rekomendasi ketika melakukan pencarian buku pada halaman pencarian <p>Aktor berada pada Search Result</p> <ol style="list-style-type: none"> 1. Aktor dapat menginputkan keyword buku yang diinginkan. 2. Sistem akan menampilkan buku sesuai keyword yang diberikan.

3.1.3.2 UI Design dan Deskripsi Objek UI #3 Mencari Buku



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
SR	SearchResult	Browsing Page Book

SearchResult Page

Id_Objek	JENIS	LABEL	Keterangan
SearchBar	Text Input	Search	Tempat menuliskan keyword pencarian
Button_SR1	Button	Add to Library	Menambahkan buku ke Library
RadioButton_SR1	Radio Button	All	Menampilkan hasil pencarian keseluruhan
RadioButton_SR2	Radio Button	Author	Menampilkan hasil pencarian dengan

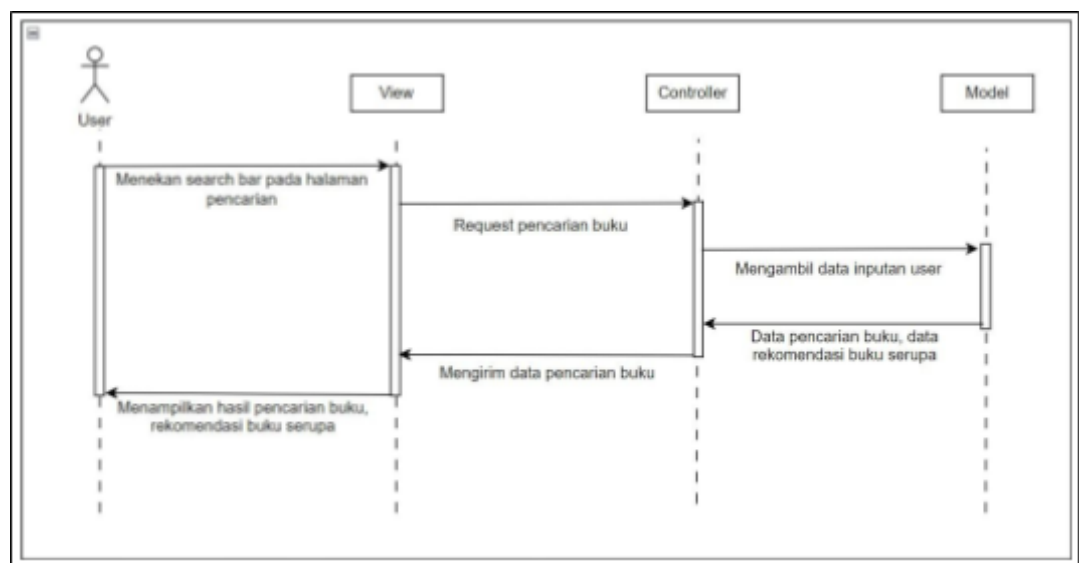
			parameter author
RadioButton_SR3	Radio Button	Title	Menampilkan hasil pencarian dengan parameter title

3.1.3.3 Identifikasi Object Baru & Tipe Kelas #3 Mencari Buku

Tabel Object Perancangan

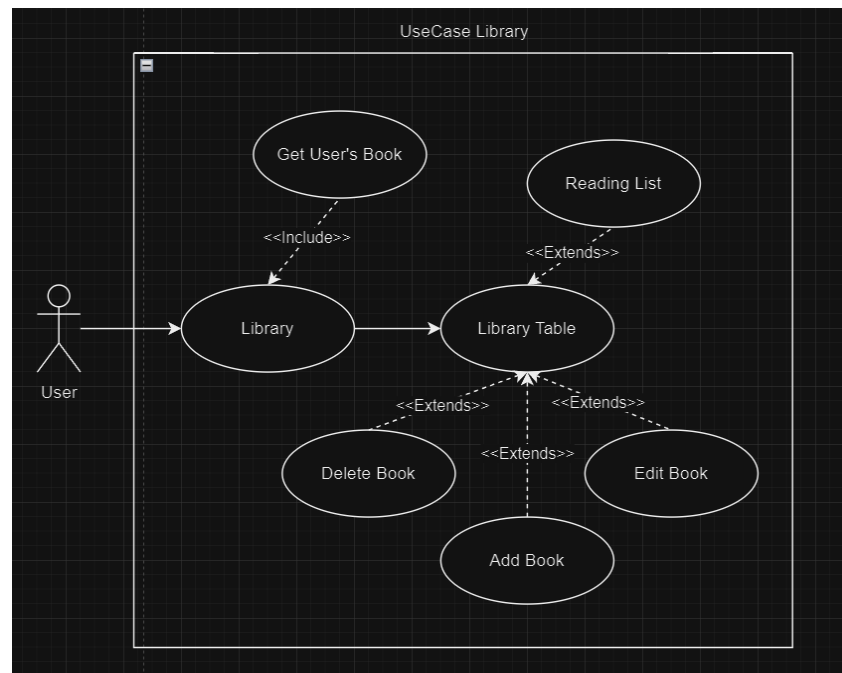
No	Nama Object Baru	Jenis / Tipe Kelas
1.	new User	User
2.	new Book	Book

3.1.3.4 Sequence Diagram #3 : Mencari Buku



3.1.4 UI Design dan Deskripsi Objek UI #4 Library

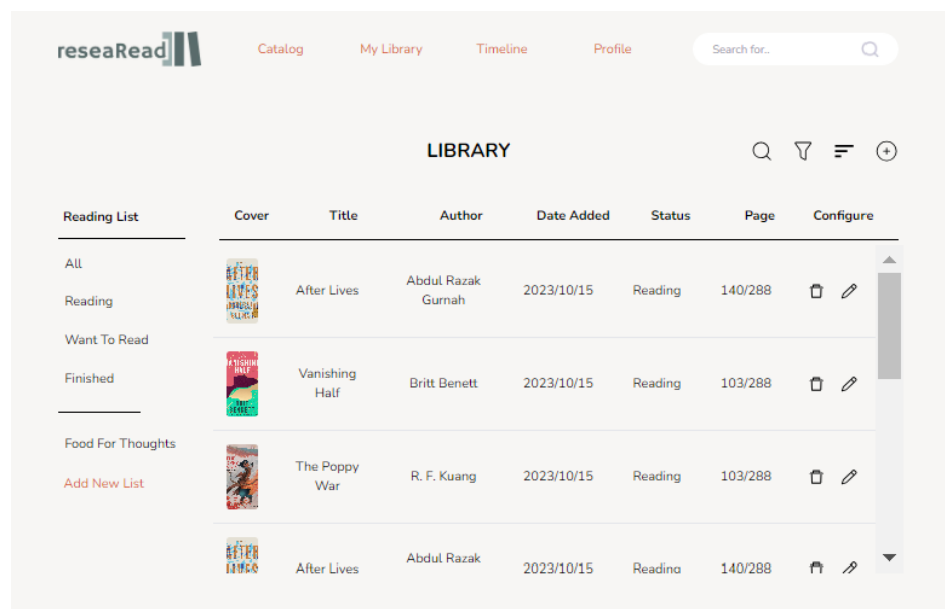
3.1.4.1 Use Case Scenario #4 : Library



1.	Kode User	-
2.	Kode Kebutuhan Fungsional	F-004
3.	Aktor	User
4.	Pre-Condition:	<ul style="list-style-type: none"> Aktor sudah terdaftar dan login (mempunyai token). Aktor ingin membuka Library.
5.	Post-Condition:	<ul style="list-style-type: none"> Sistem berhasil menampilkan Library personal Aktor ke layar.
6.	Use Case Description	
	a. Primary Flow	Buku Terdaftar pada Aplikasi

		<ol style="list-style-type: none"> 1. Aktor masuk ke halaman Library. 2. Sistem Menampilkan Library personal Aktor. 3. Aktor bisa menambah buku, edit koleksi buku tertentu, delete koleksi buku tertentu, dan sort berdasarkan status ataupun tipe reading list.
	b. Alternative Flow	-

3.1.4.2 UI Design dan Deskripsi Objek UI #4 Library



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
BLibrary	LibraryPage	Halaman library

LibraryPage

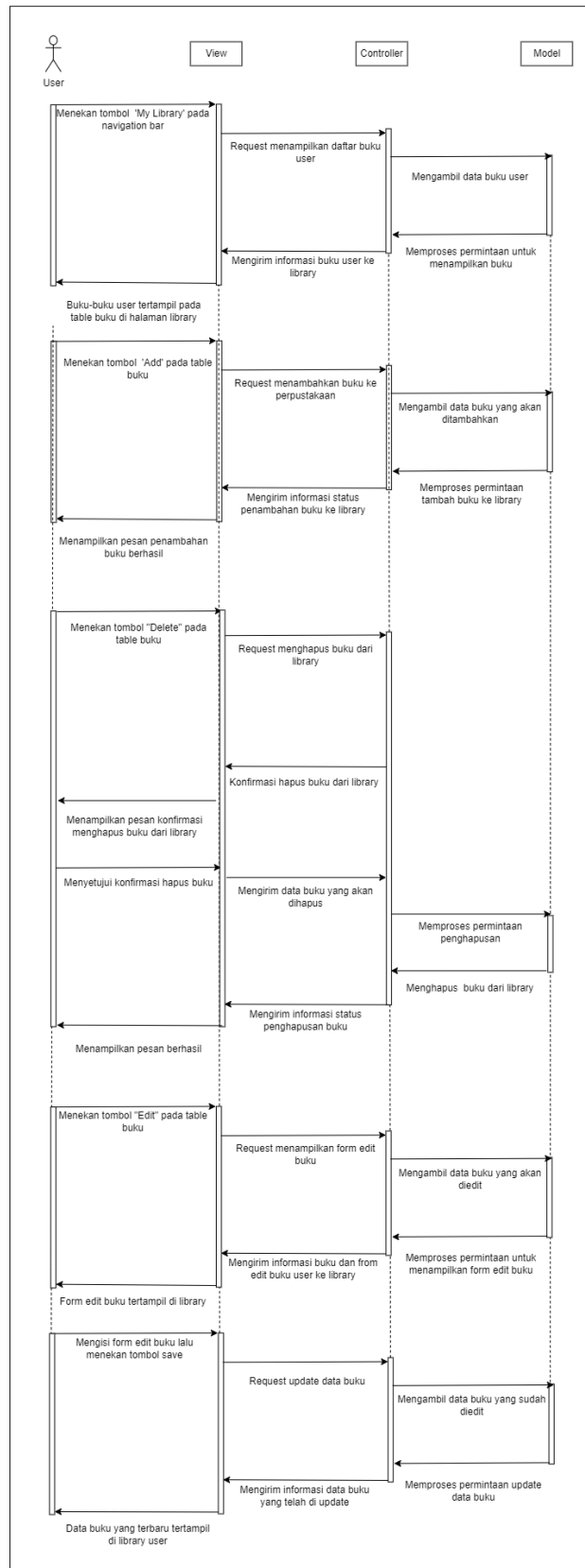
Id_Objek	JENIS	LABEL	Keterangan
Btn_BLibrary01	Button	All	Menampilkan seluruh buku di dalam library
Btn_BLibrary02	Button	Reading	Menampilkan buku dengan kategori reading
Btn_BLibrary03	Button	Want To Read	Menampilkan buku dengan kategori want to read
Btn_BLibrary04	Button	Finished	Daftar buku yang telah selesai dibaca
Btn_BLibrary05	Button	Hapus	Menghapus buku dalam library
Btn_BLibrary06	Button	Food For Thoughts	Menampilkan detail tambahan
Btn_BLibrary07	Button	Add New List	Menambahkan buku baru kedalam library
Btn_BLibrary08	Button	Edit	Mengedit data buku

3.1.4.3 Identifikasi Object Baru & Tipe Kelas #4 Library

Tabel Object Perancangan

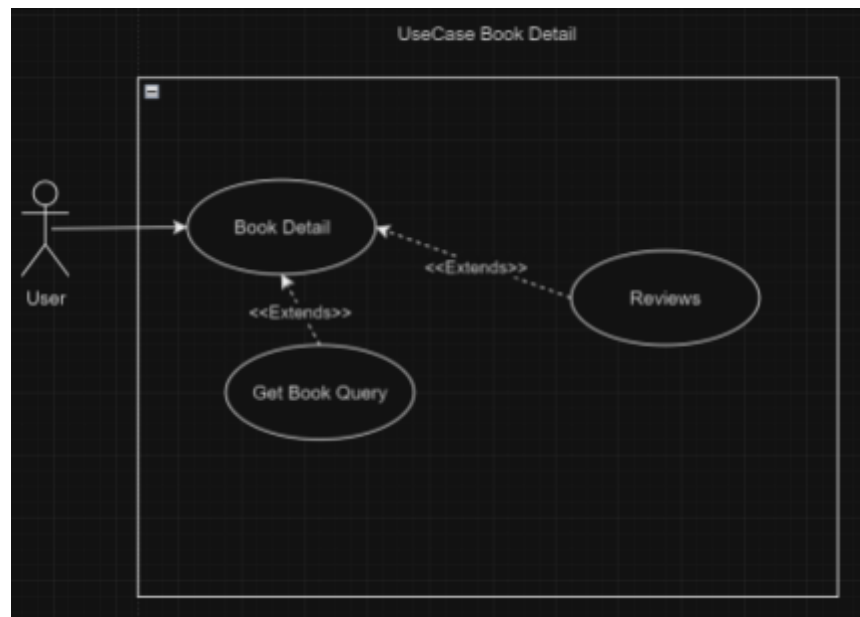
No	Nama Object Baru	Jenis / Tipe Kelas
1.	new Book	Book
2.	new readinglist	Readinglist

3.1.4.4 Sequence Diagram #4 : Library



3.1.5 Membaca Data Buku

3.1.5.1 Use Case Scenario #5 Membaca Data Buku

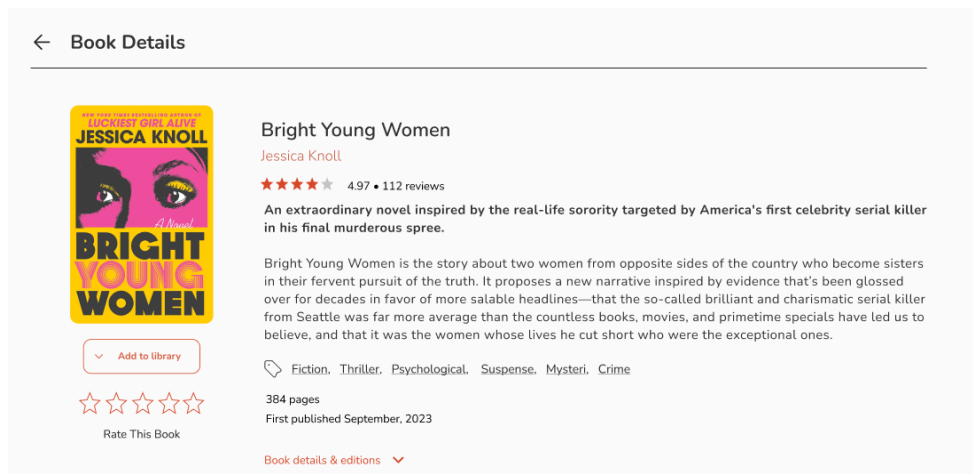


Skenario Use Case Membaca Data Buku :

Pre-Condition:	<ul style="list-style-type: none">• Aktor sudah login (mempunyai token).• Aktor ingin mengakses halaman web buku tertentu.• Sistem menampilkan Featured Book di homepage.• Sistem menampilkan Continue Reading (buku Library Aktor)• Sistem menampilkan Top Picks di homepage.
Post-Condition:	<ul style="list-style-type: none">• Sistem Menampilkan data buku ke di halaman web buku.• Sistem menampilkan reviews.
Use Case Description	
a. Primary Flow	1. Aktor menekan judul buku yang ada di tampilan pada page tertentu.

	2. Sistem akan mengarahkan Aktor ke halaman Book Detail sesuai dengan judul apa yang di click.
b. Alternative	<p>Di Halaman Search Book</p> <ul style="list-style-type: none"> • Aktor mencari buku yang diinginkan. • Aktor menekan salah satu buku yang muncul. • Sistem akan mengarahkan Aktor ke halaman Book Detail sesuai dengan judul apa yang di click.

3.1.5.2 UI Design dan Deskripsi Objek UI #5 Membaca Data Buku



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
BDETAIL	BookDetailPage	Halaman detail buku

Catalog Page

Id_Objek	JENIS	LABEL	Keterangan
Btn_BDetail01	Button	Add to Library	Menambahkan buku ke perpustakaan

Prodi S1 Informatika Tel-U	DPPL-Diaspora	Halaman 25 dari 61
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

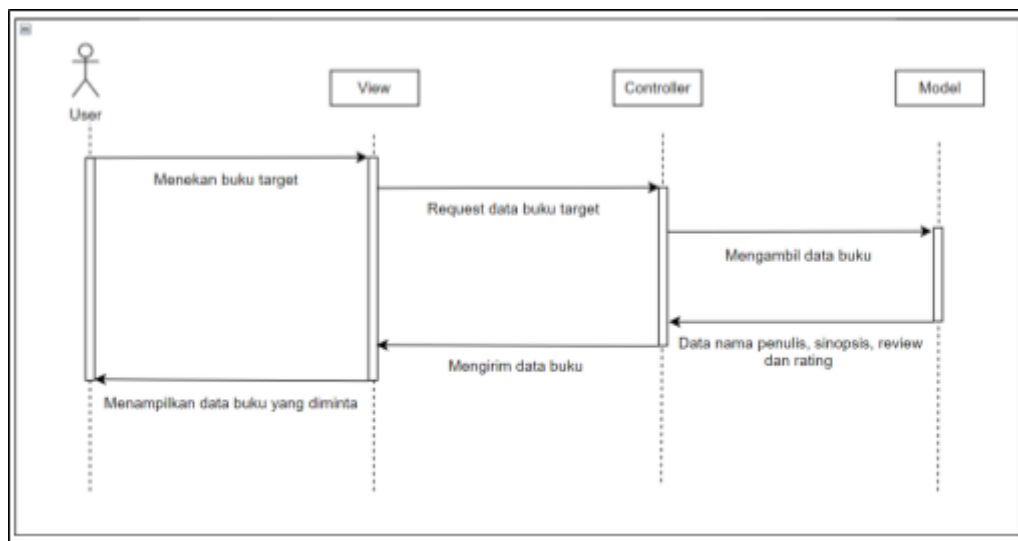
			user
Btn_BDetail02	Rating	Rate this Book	Memberikan rating ke buku
Btn_BDetail03	Button	Edit	Mengedit pesan review pada detail buku
Btn_BDetail04	Button	Hapus	Menghapus pesan pada detail buku
Drop_BDetail01	Dropdown	Book detail & editions	Menampilkan detail tambahan
Drop_BDetail02	Dropdown	About The Author	Menampilkan profil penulis
Drop_BDetail03	Dropdown	Reviews	menampilkan review buku tersebut

3.1.5.3 Identifikasi Object Baru & Tipe Kelas #5 Membaca Data Buku

Tabel Object Perancangan

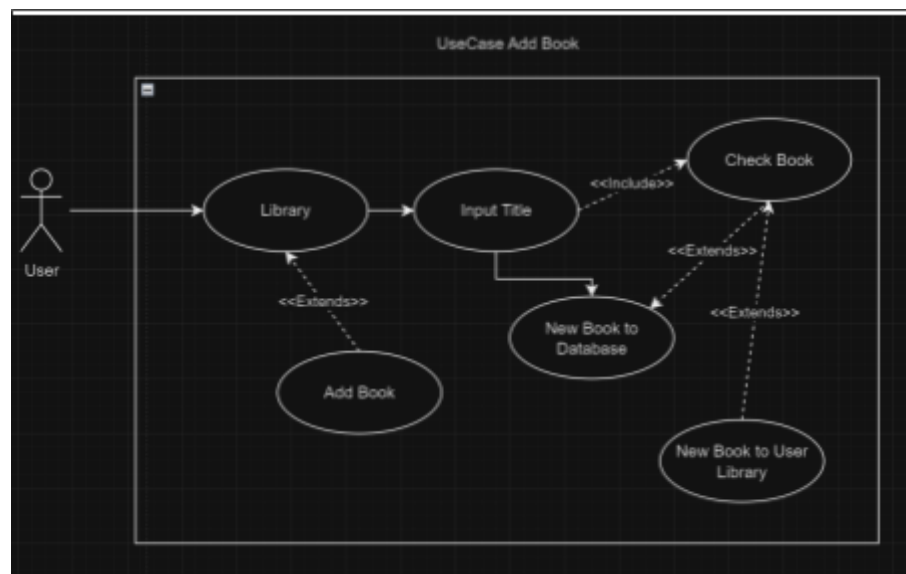
No	Nama Object Baru	Jenis / Tipe Kelas
1.	new Book	Book
2.	new Library	Library

3.1.5.4 Sequence Diagram #5 Membaca Data Buku



3.1.6 Menambah Buku Baru ke dalam Sistem

3.1.6.1 Use Case Scenario #6 : Menambah Buku Baru ke dalam Sistem



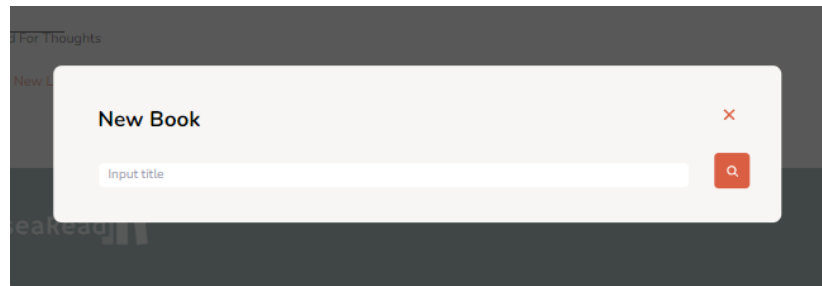
Skenario Use Case Add Book:

Pre-Condition:

- Aktor telah login (mempunyai token).
- Aktor ingin menambahkan buku baru ke aplikasi atau dan buku ke Library

	personal Aktor.
Post-Condition:	<ul style="list-style-type: none"> • Aktor berhasil menambahkan buku ke aplikasi ReseaRead (jika dilakukan). • Aktor berhasil menambahkan buku ke Library personal Aktor.
Use Case Description	
a. Primary Flow	<p>Buku Terdaftar pada Aplikasi</p> <ol style="list-style-type: none"> 1. Aktor masuk ke halaman Library. 2. Aktor menekan tombol "tambah". 3. Aktor memasukkan judul buku. 4. Judul buku yang diminta terdaftar. 5. Aktor memasukkan nomor halaman untuk page break, status buku, tipe dan reading list buku. 6. Buku berhasil ditambahkan ke Library.
b. Alternative Flow	<p>Buku Tidak Terdaftar pada Aplikasi</p> <p>Aktor masuk ke halaman Library.</p> <p>Aktor menekan tombol "tambah".</p> <p>Aktor memasukkan judul buku.</p> <p>Judul buku yang diminta tidak terdaftar.</p> <p>Aktor memasukkan nama buku, nama penulis, nomor halaman untuk page break, total halaman buku, tipe dan reading list buku.</p> <p>Buku ditambahkan ke aplikasi ReseaRead.</p> <p>Buku berhasil ditambahkan ke Library.</p>

3.1.6.2 UI Design dan Deskripsi Objek UI #6 Menambah Buku Baru ke dalam Sistem

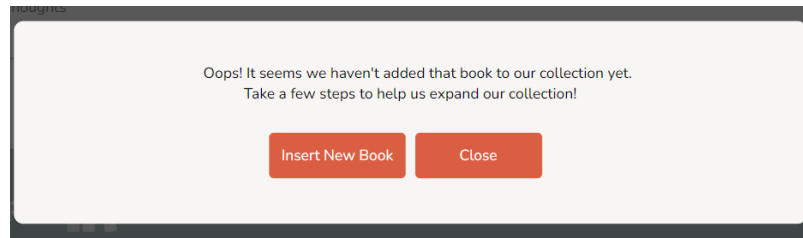


Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
BookAddSys	BookAddPage	Halaman tambah buku ke sistem

Catalog Page

Id_Objek	JENIS	LABEL	Keterangan
TextInput01	TextInput	Input title	Menginputkan judul buku yang akan ditambah
Btn_BookAddSys02	Button	Search	Mencari judul buku yang telah diinputkan oleh user
Btn_BookAddSys03	Button	Close	Keluar dari fitur New Book



Id_Objek	JENIS	LABEL	Keterangan
Btn_BookAddSys01	Button	Insert New Book	Menambahkan buku baru kedalam sistem
Btn_BookAddSys02	Button	Close	Keluar dari halaman tambah buku

Id_Objek	JENIS	LABEL	Keterangan
TextInput01	TextInput	Title	Menginputkan judul buku yang akan ditambahkan
TextInput01	TextInput	Author	Menginputkan penulis buku

Prodi S1 Informatika Tel-U	DPPL-Diaspora	Halaman 30 dari 61
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

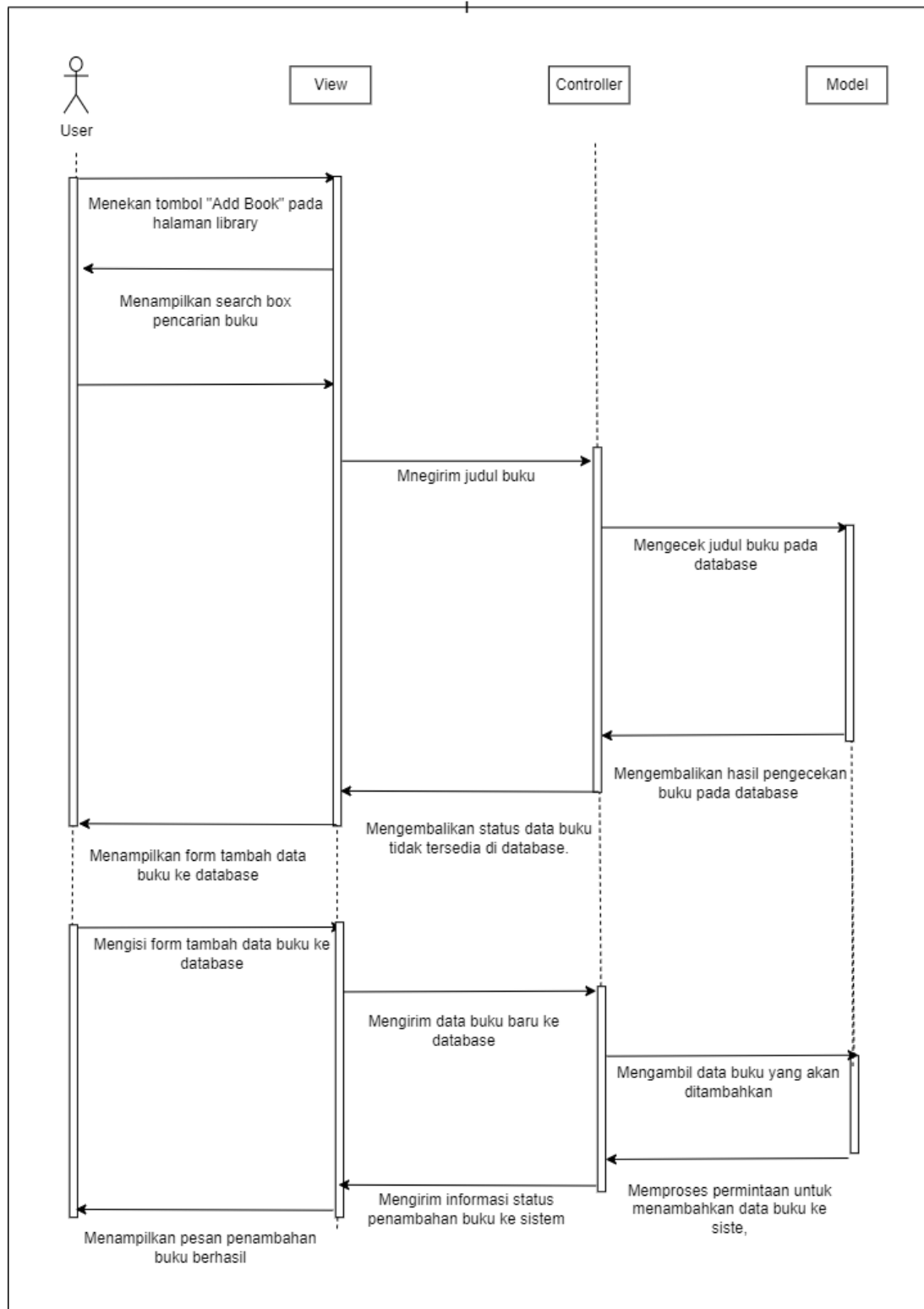
TextInput01	TextInput	Page	Menginputkan jumlah halaman buku
TextInput01	TextInput	Cover URL	Menginputkan cover URL buku
Drop_BBookAddsys01	Dropdown	List	Reading list
Drop_BBookAddsys02	Dropdown	Status	Status
Btn_BookAddSys01	Button	Save	Menyimpan buku yang baru ditambahkan
Btn_BookAddSys02	Button	Discard	Membuat data yang diinputkan

3.1.6.3 Identifikasi Object Baru & Tipe Kelas #6 Menambah Buku Baru ke dalam Sistem

Tabel Object Perancangan

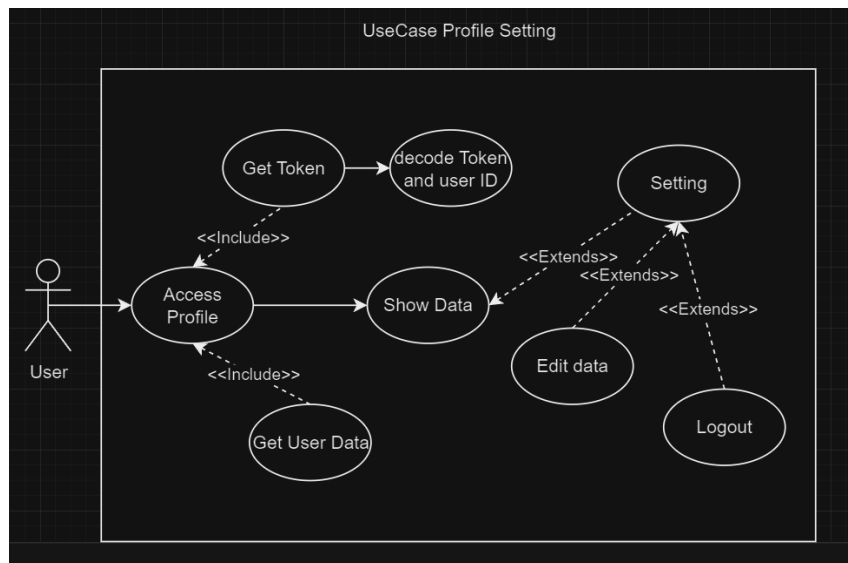
No	Nama Object Baru	Jenis / Tipe Kelas
1.	requestedBook	Book

3.1.6.4 Sequence Diagram #6 : Menambah Buku Baru ke dalam Sistem



3.1.7 Konfigurasi Akun

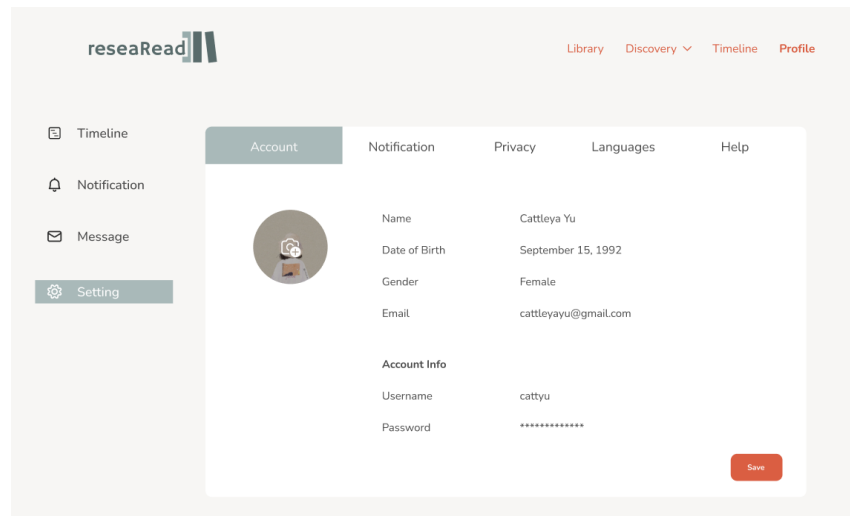
3.1.7.1 Use Case Scenario #7 : Konfigurasi Akun



Skenario Use Case Konfigurasi akun :

Pre-Condition:	<ul style="list-style-type: none">• Aktor harus login (mempunyai token).• Aktor ingin melihat data tertentu.
Post-Condition:	<ul style="list-style-type: none">• Sistem Menampilkan profil yang layak ditampilkan di layar.• Aktor bisa merubah data sesuai yang ada ditampilkan (Jika iya).
Use Case Description	
a. Primary Flow	<ol style="list-style-type: none">1. Aktor menekan tombol Profil.2. Akan menampilkan halaman profil user sesuai dengan user id dari token yang di set dari autentikasi.3. Aktor menekan Setting.4. Aktor bisa mengedit data yang diinginkan.

3.1.7.2 UI Design dan Deskripsi Objek UI #7 Konfigurasi Akun



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
Setting	AccountSetting	Halaman

Account Setting Page

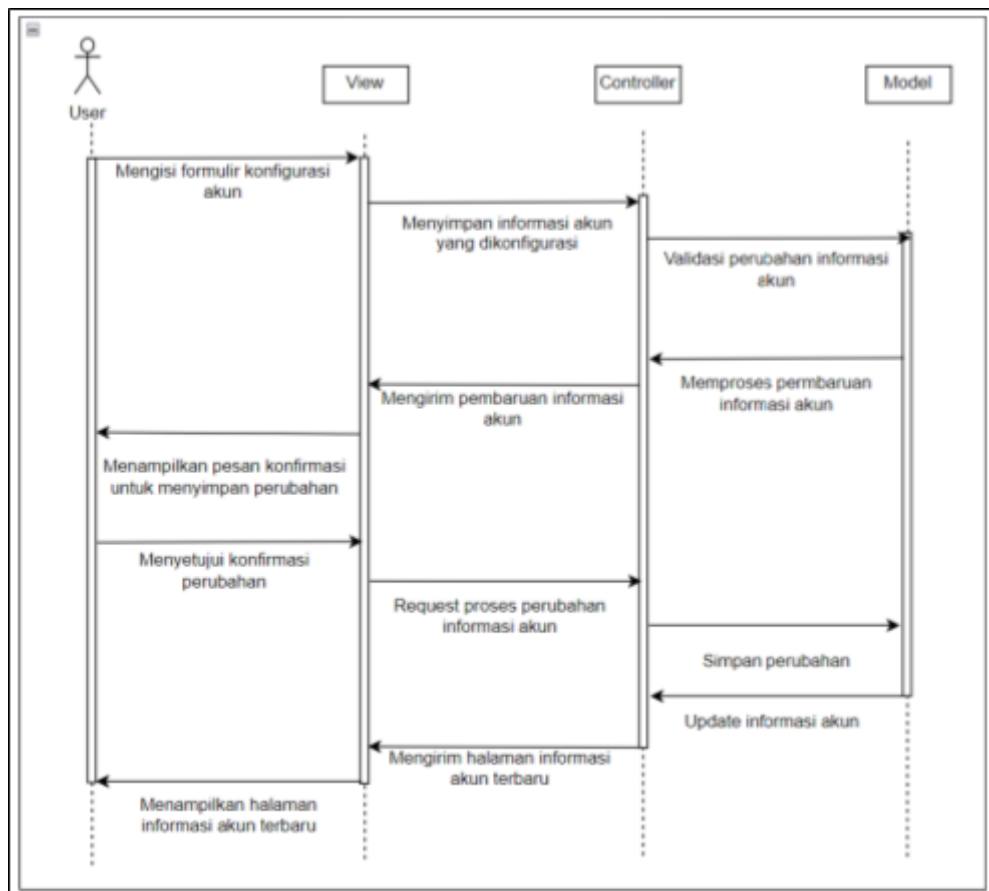
Id_Objek	JENIS	LABEL	Keterangan
Setting_Account	button	Account	Halaman pengaturan akun
Setting_Notification	button	Notificatio n	Halaman pengaturan notifikasi
Setting_Privacy	button	Privacy	Halaman pengaturan privacy
Setting_Language	button	Language	Halaman pengaturan bahasa
Setting_Help	button	Help	Halaman pengaturan ..
Setting_Button01	button	Button	Menyimpan perubahan
Setting_Button02	button	Add picture	Bytton untuk mengganti foto profil user

3.1.7.3 Identifikasi Object Baru & Tipe Kelas #7 Konfigurasi Akun

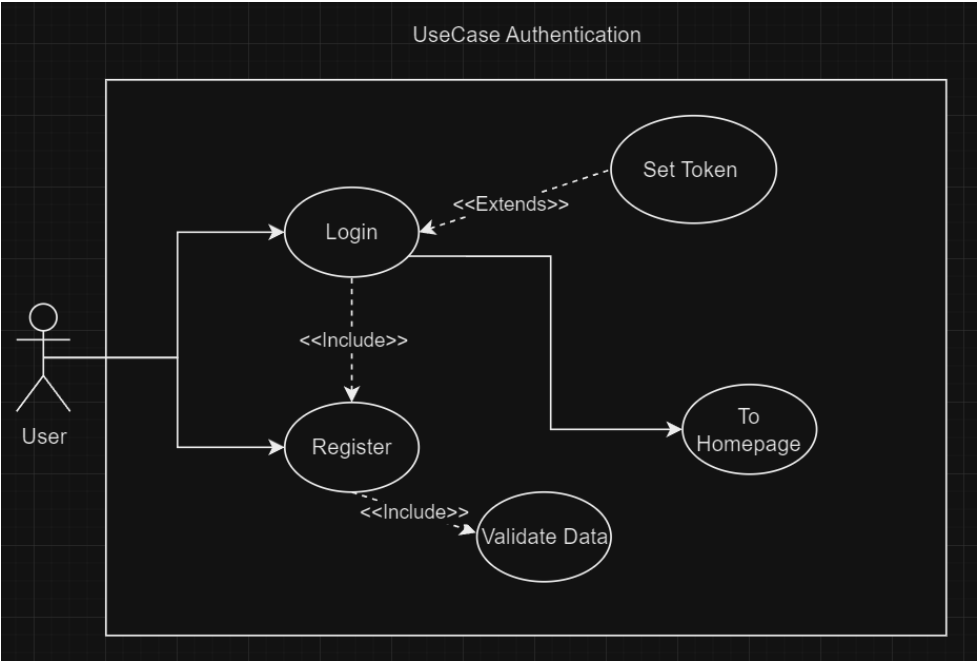
Tabel Object Perancangan

No	Nama Object Baru	Jenis / Tipe Kelas
1.	thisUser	User

3.1.7.4 Sequence Diagram #7 : Konfigurasi Akun



3.1.8 Authentication

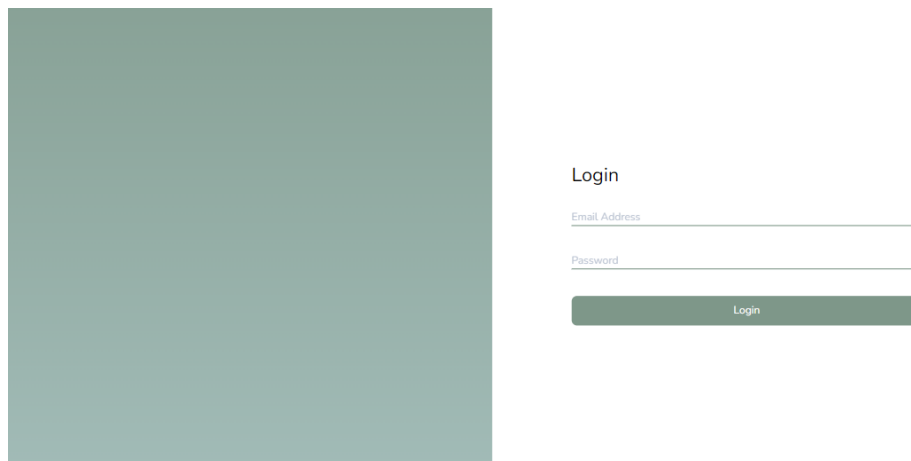


3.1.8.1 Use Case Scenario #8 : Authentication

Pre-Condition:	<ul style="list-style-type: none"> Aktor belum mempunyai akun / sudah mempunyai akun. Aktor mengakses situs ReseRead.
Post-Condition:	<ul style="list-style-type: none"> Aktor berhasil membuat akun dan diteruskan ke homepage dengan token yang di set, atau Aktor berhasil login dan diteruskan ke homepage dengan token yang di set
Use Case Description	
a. Primary Flow	Aktor belum mempunyai akun 1. Aktor mengakses laman register. 2. Aktor menginputkan username,

	<p>email, dan password.</p> <ol style="list-style-type: none"> email harus valid dan panjang password dalam rentang 8 sampai 20. Aktor menekan Create Account. Aktor selanjutnya masuk ke fase login (untuk proses set token dan tidak perlu input data untuk login) dan diteruskan ke homepage.
b. Alternative Flow	<p>Aktor sudah mempunyai akun</p> <ol style="list-style-type: none"> Aktor mengakses halaman login. (Jika Aktor mengakses lama register, ada tombol untuk <i>redirect</i> ke halaman login. Aktor menginputkan data yang sesuai. Proses validasi terkait data yang diinput. Set token dan diteruskan ke homepage

3.1.8.2 UI Design dan Deskripsi Objek UI #8 Authentication



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
Login	AccountLogin	Halaman

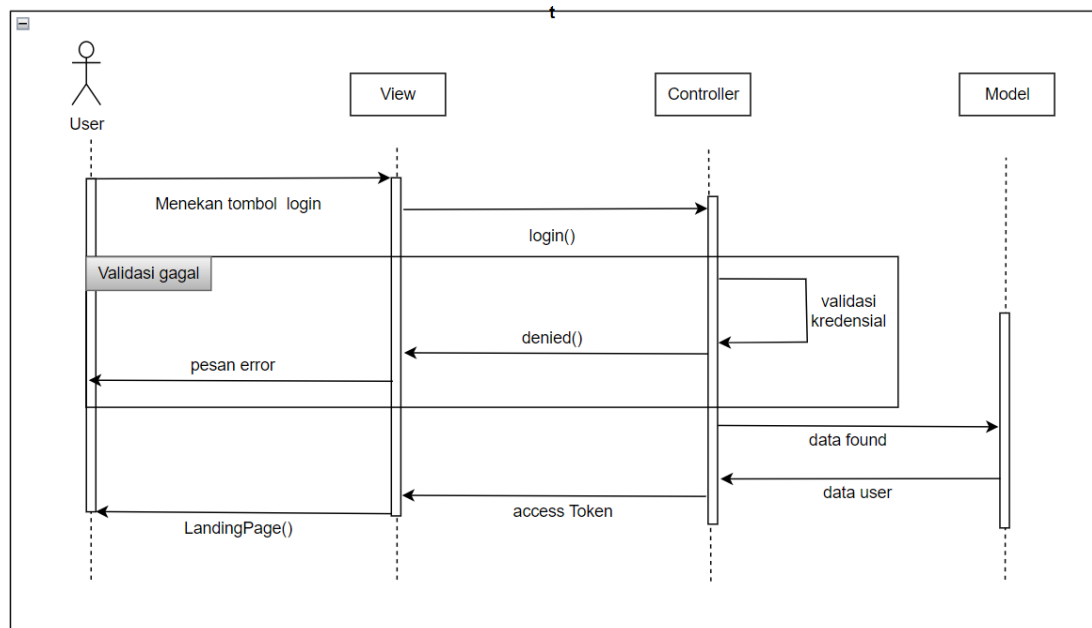
Login Page

Id_Objek	JENIS	LABEL	Keterangan
TextInput01	TextInput	Email/Address	Menginputkan Email/Addres user
TextInput02	TextInput	Password	Menginputkan password user
Btn_Login01	Button	Login	Masuk ke dalam aplikasi ReseaRead

3.1.8.3 Identifikasi Object Baru & Tipe Kelas #8 Login

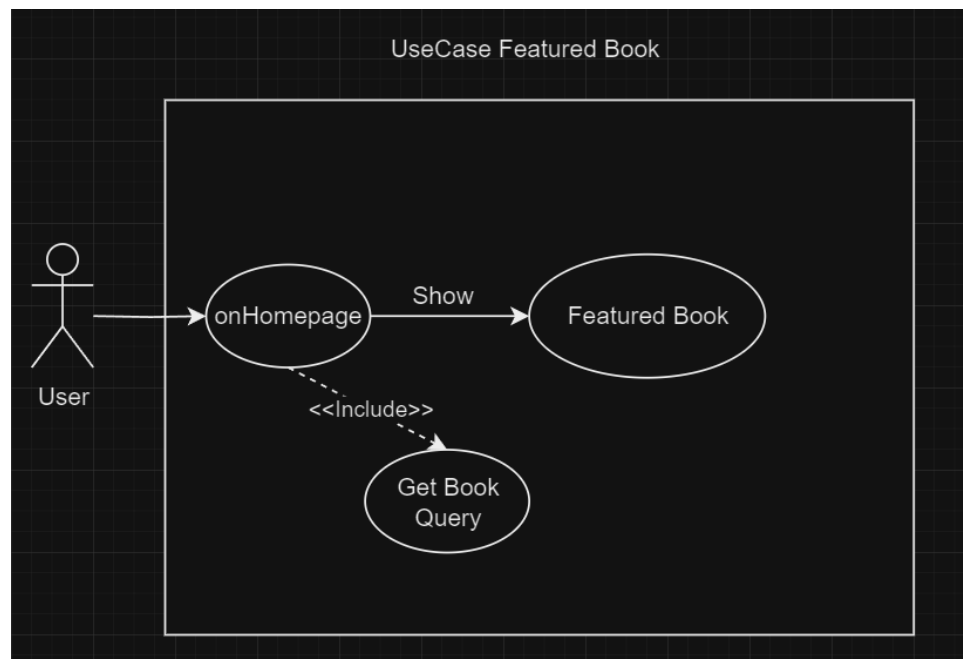
No	Nama Object Baru	Jenis / Tipe Kelas
1.	newUser	User

3.1.8.4 Sequence Diagram #8 : Login



3.1.9 Featured Book

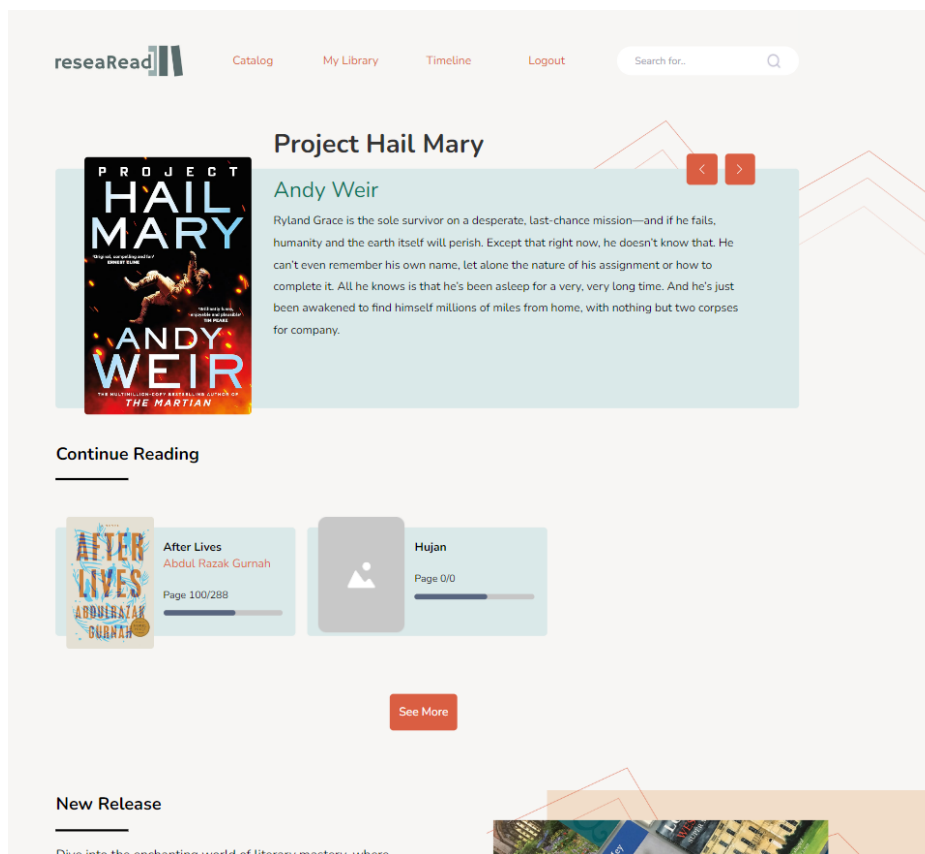
3.1.9.1 Use Case Scenario #9 : Featured Book



Pre-Condition:	<ul style="list-style-type: none">• Aktor telah login (memiliki token).• Aktor berada pada homepage.
Post-Condition:	<ul style="list-style-type: none">• Sistem menampilkan Featured Book ke homepage user.
Use Case Description	
a. Primary Flow	<ol style="list-style-type: none">1. Aktor mengakses homepage.2. Sistem menampilkan buku Featured ke homepage user (setiap saat user mengakses homepage).
b. Alternative Flow	-

3.1.9.2 UI Design dan Deskripsi Objek UI #9 Featured Book

Prodi S1 Informatika Tel-U	DPPL-Diaspora	Halaman 40 dari 61
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
FtrBook	FeaturedBook	Halaman

Featured Book Page

Id_Objek	JENIS	LABEL	Keterangan
Btn_FtrBook01	Button	Back	Kembali ke tampilan buku sebelumnya
Btn_FtrBook02	Button	Next	Pindah ke tampilan buku selanjutnya
Btn_FtrBook03	Button	See More	Menampilkan buku lebih banyak

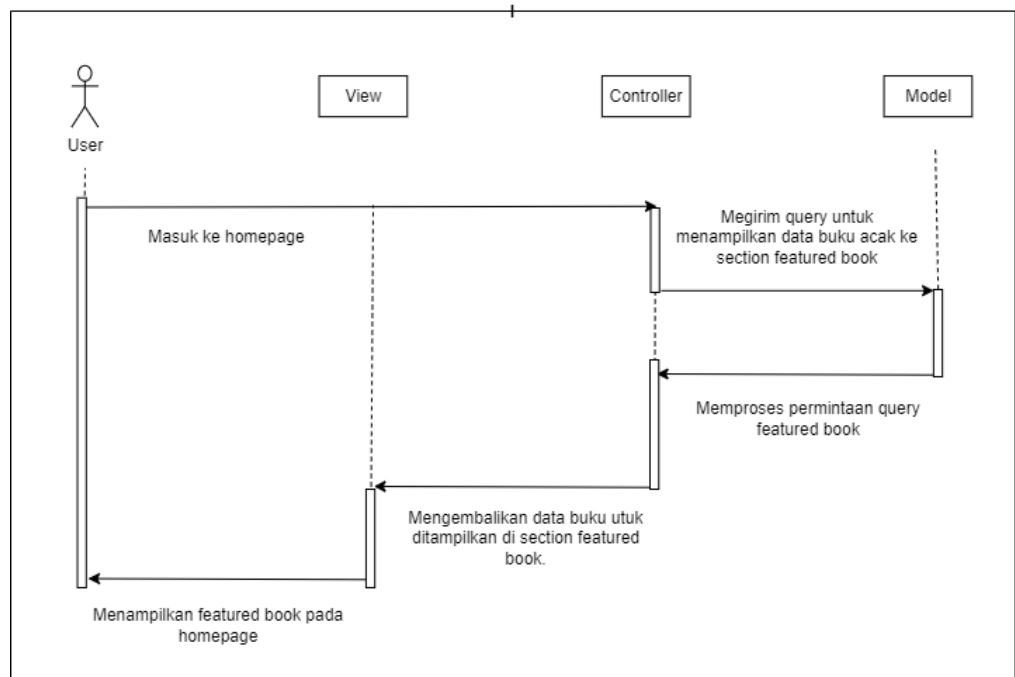
3.1.9.3 Identifikasi Object Baru & Tipe Kelas #9 Featured Book

Prodi S1 Informatika Tel-U	DPPL-Diaspora	Halaman 41 dari 61
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

Tabel Object Perancangan

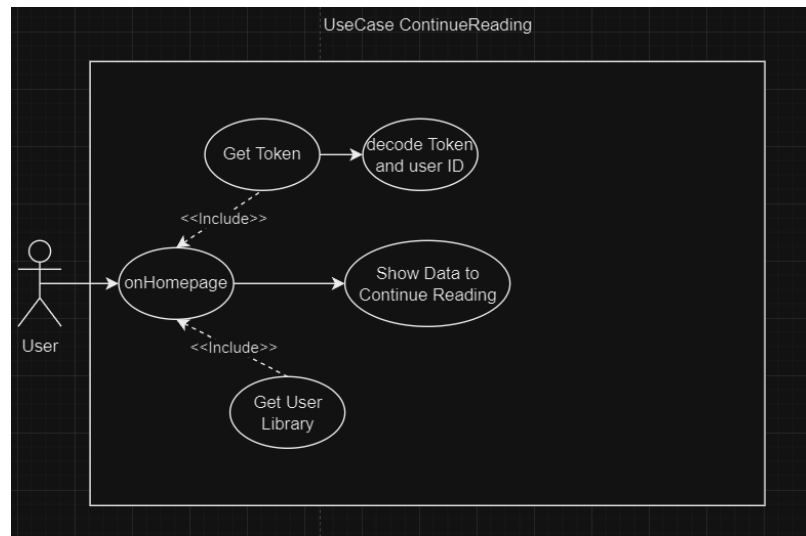
No	Nama Object Baru	Jenis / Tipe Kelas
1.	new Book	Book

3.1.9.4 Sequence Diagram #9 : Featured Book



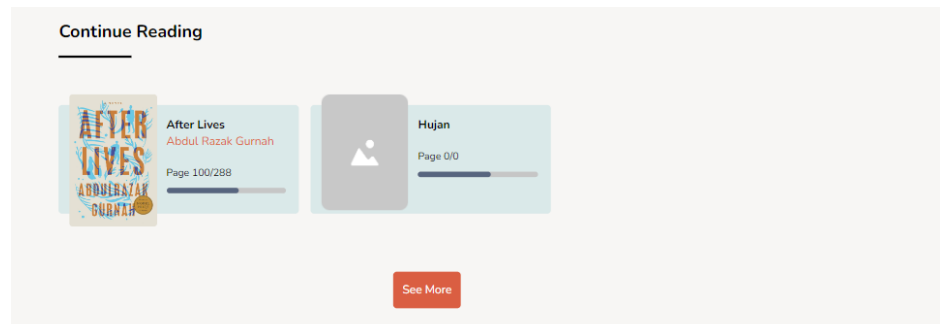
3.1.10 Continue Reading

3.1.10.1 Use Case Scenario #10 : Continue Reading



Pre-Condition:	<ul style="list-style-type: none">• Aktor harus login (mempunyai token).• Aktor memiliki buku di Library minimal 1 (agar bisa melihat komponen Continue Reading).
Post-Condition:	<ul style="list-style-type: none">• Sistem menampilkan 3 buku di Library saat di halaman homepage.
Use Case Description	
a. Primary Flow	<ol style="list-style-type: none">1. Aktor mengakses homepage.2. Sistem menampilkan <i>up to</i> 3 buku Library pada homepage Aktor.
b. Alternative Flow	-

3.1.10.2 UI Design dan Deskripsi Objek UI #10 Continue Reading



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
HP	HomePage	Halaman awal web

Continue Reading Page

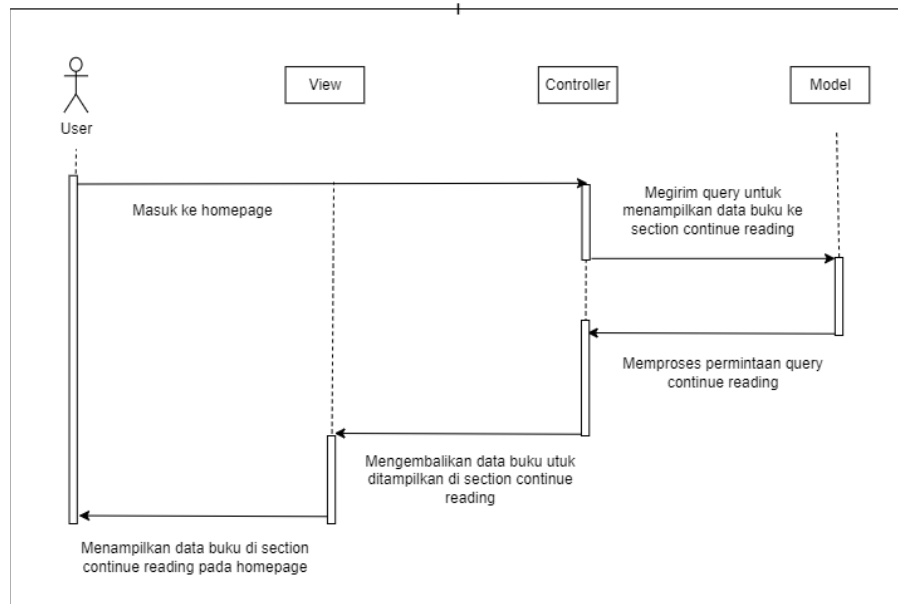
Id_Objek	JENIS	LABEL	Keterangan
HP_Button	Button	See More	Menampilkan buku lebih banyak

3.1.10.3 Identifikasi Object Baru & Tipe Kelas #10 Continue Reading

Tabel Object Perancangan

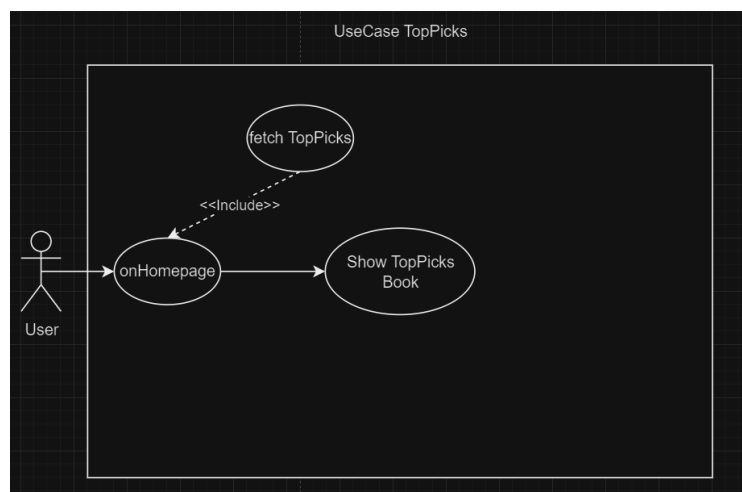
No	Nama Object Baru	Jenis / Tipe Kelas
1.	new Book	Book

3.1.10.4 Sequence Diagram #10 : Continue Reading



3.1.11 TopPicks

3.1.11.1 Use Case Scenario #11 : TopPicks

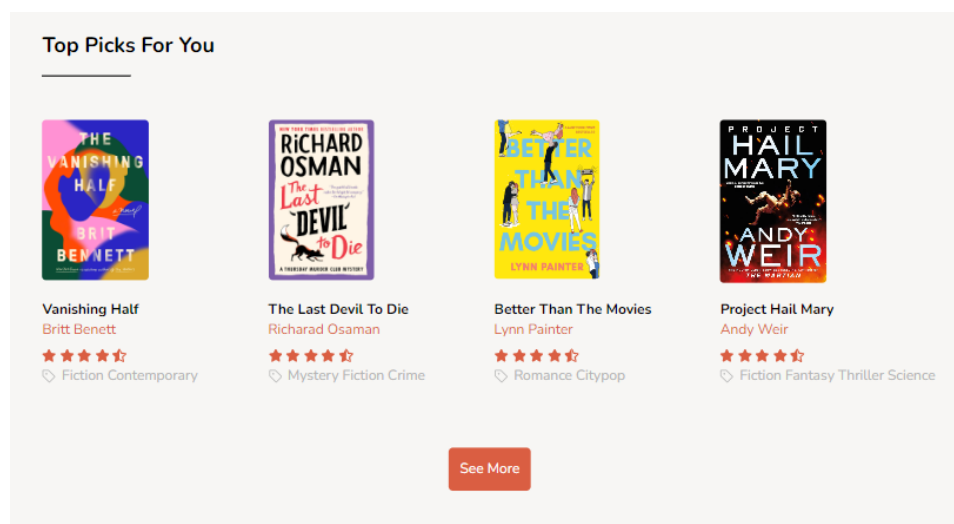


Pre-Condition:

- Aktor harus login (mempunyai token).

	<ul style="list-style-type: none"> Aktor berada di homepage.
Post-Condition:	<ul style="list-style-type: none"> Sistem menampilkan 4 buku di homepage user dengan rating > 4.
Use Case Description	
a. Primary Flow	<ol style="list-style-type: none"> Aktor mengakses homepage. Sistem menampilkan <i>up to</i> 4 buku pada homepage Aktor dengan rating > 4.
b. Alternative Flow	-

3.1.11.2 UI Design dan Deskripsi Objek UI #11 TopPicks



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
HP	HomePage	Halaman awal web

Catalog Page

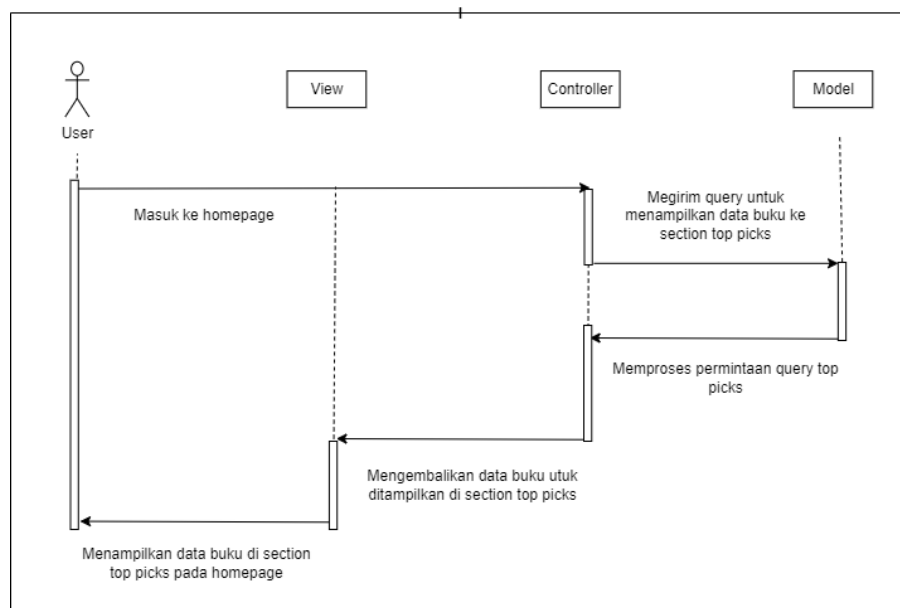
Id_Objek	JENIS	LABEL	Keterangan
HP_Button	Button	See More	Fungsi untuk menampilkan lebih banyak lagi buku-buku pilihan/rekomendasi

3.1.11.3 Identifikasi Object Baru & Tipe Kelas #11 TopPicks

Tabel Object perancangan

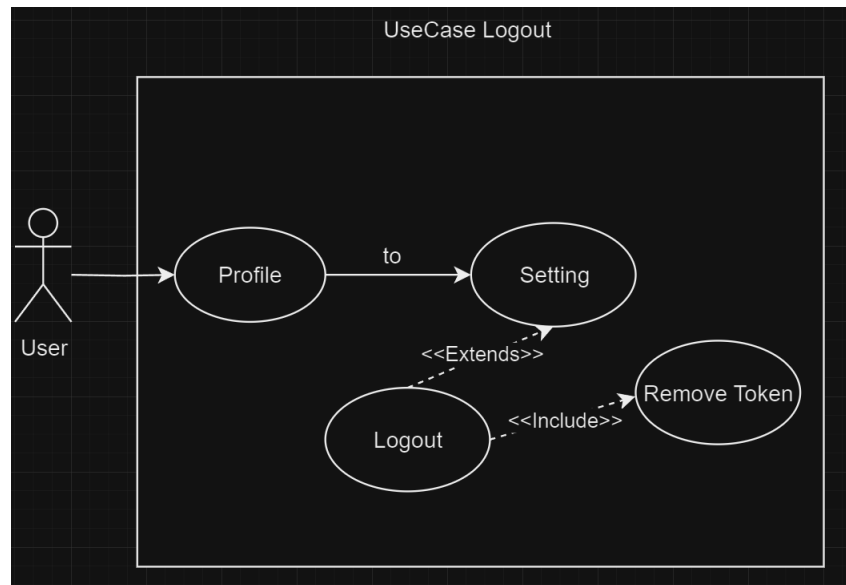
No	Nama Object Baru	Jenis / Tipe Kelas
1.	new Book	Book

3.1.11.4 Sequence Diagram #11 : TopPicks



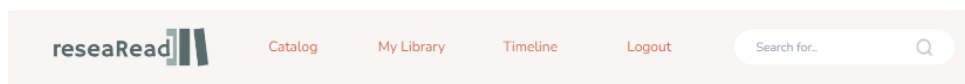
3.1.12 Logout

3.1.12.1 Use Case Scenario #12 : Logout



Pre-Condition:	<ul style="list-style-type: none">• Aktor sudah login (mempunyai token).
Post-Condition:	<ul style="list-style-type: none">• Aktor berhasil logout dari aplikasi ReseaRead (token <i>removed</i>)
Use Case Description	
a. Primary Flow	<ul style="list-style-type: none">• Aktor Logout Manual<ol style="list-style-type: none">1. Aktor mengakses profile.2. kemudian Aktor mengakses setting.3. Aktor menekan tombol logout.
b. Alternative Flow	-

3.1.12.2 UI Design dan Deskripsi Objek UI #12 Logout



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
HP	HomePage	Halaman awal web

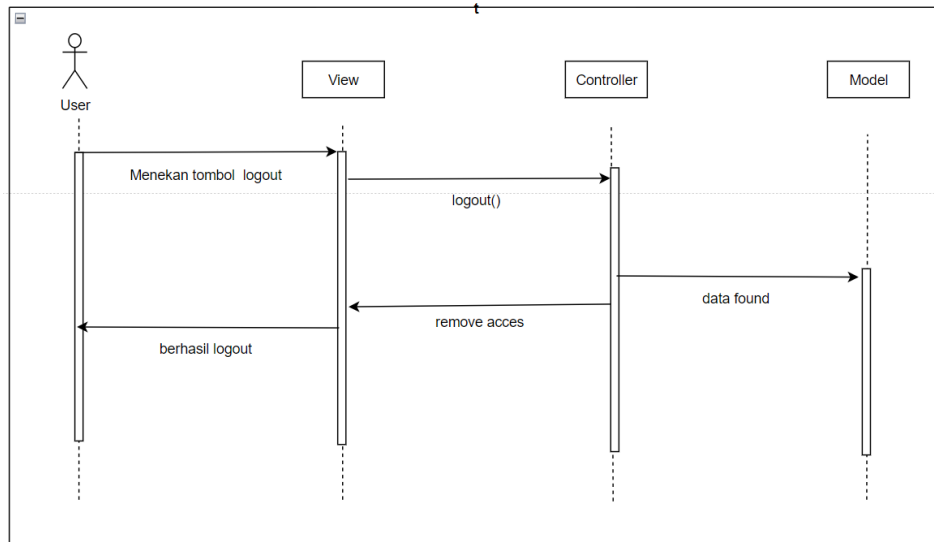
Catalog Page

Id_Objek	JENIS	LABEL	Keterangan
TPCS_Nav01	Button	Catalog	Berpindah ke halaman katalog
TPCS_Nav02	Button	My Library	Berpindah ke halaman my library
TPCS_Nav03	Button	Timeline	Berpindah ke halaman timeline
TPCS_Nav04	Button	Logout	Keluar dari halaman website

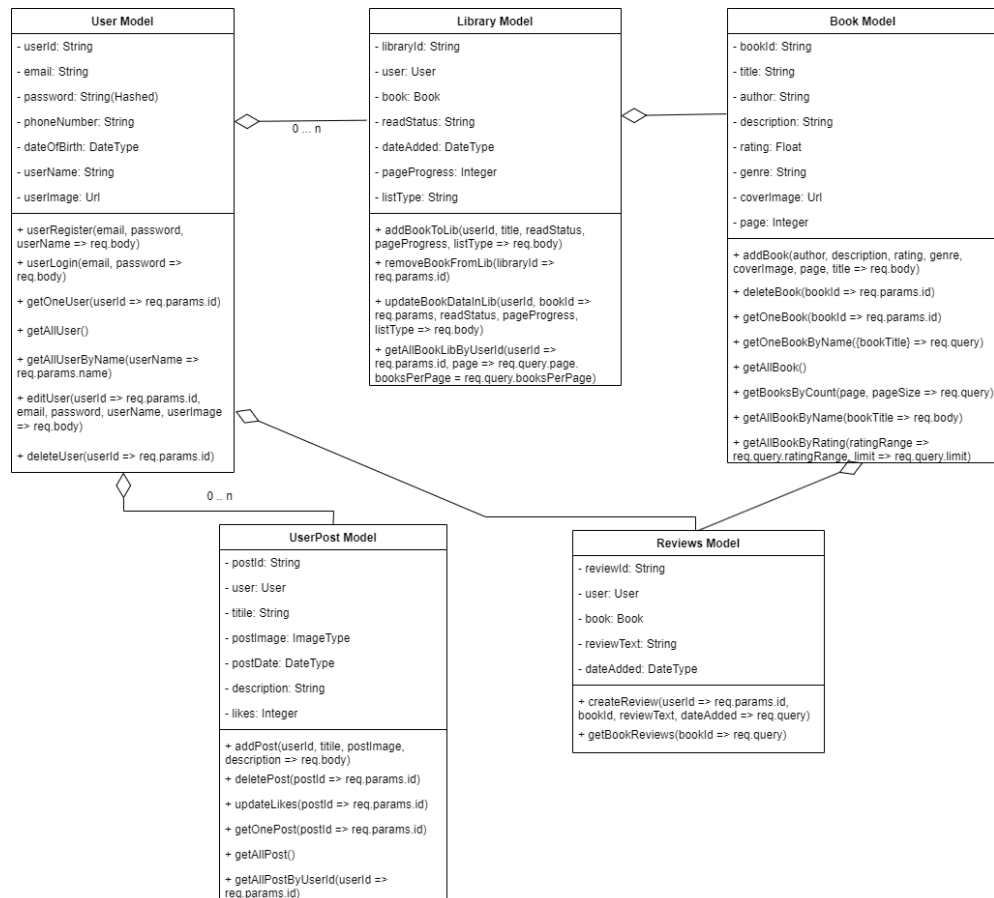
3.1.12.3 Identifikasi Object Baru & Tipe Kelas #12 Logout

-

3.1.12.4 Sequence Diagram #12 : Logout



3.2 Diagram Kelas Keseluruhan



3.3 Perancangan Detail Kelas

3.3.1 Tabel Kelas : UserModel

ID Kelas	Nama Kelas	Atribut	Method / Operation
Class01	UserModel	(-) userId (-) email (-) password (-) phoneNumber (-) dateOfBirth (-) userName (-) userImage	(+) userRegister (+) userLogin (+) getOneUser (+) getAllUser (+) getAllUserByName (+) editUser (+) deleteUser

3.3.2 Tabel Kelas : LibraryModel

ID Kelas	Nama Kelas	Atribut	Method / Operation
Class02	LibraryModel	(-) libraryId (-) user (-) book (-) readStatus (-) dateAdded (-) pageProgress (-) listType	(+) addBookToLib (+) removeBookFromLib (+) updateBookDataInLib (+) getAllBookLibByUserId

3.3.3 Tabel Kelas : BookModel

ID Kelas	Nama Kelas	Atribut	Method / Operation
Class03	BookModel	(-) bookId (-) title (-) author (-) description	(+) addBook (+) deleteBook (+) getOneBook (+) getOneBookByName

		(-) rating (-) genre (-) coverImage (-) page	(+) getAllBook (+) getBooksByCount (+) getAllBookByName (+) getAllBookByRating
--	--	---	---

3.3.4 Tabel Kelas : UserPostModel

ID Kelas	Nama Kelas	Atribut	Method / Operation
Class04	UserPostModel	(-) postId (-) user (-) title (-) postImage (-) postDate (-) description (-) likes	(+) addPost (+) deletePost (+) updateLikes (+) getOnePost (+) getAllPost (+) getAllPostByUserId

3.3.5 Tabel Kelas : ReviewsModel

ID Kelas	Nama Kelas	Atribut	Method / Operation
Class04	ReviewsModel	(-) reviewId (-) user (-) book (-) reviewText (-) dateAdded	(+) createReview (+) getBookReviews

3.4 Perancangan Algoritma dan/atau Query

3.4.1 Perancangan Kelas UserModel

```
"use strict";
const { Model } = require("sequelize");

module.exports = (sequelize, DataTypes) => {
  class UserPost extends Model {
    static associate(models) {
      UserPost.belongsTo(models.User, {
        foreignKey: "userId",
        onDelete: "CASCADE",
      });
    }
  }
  UserPost.init(
    {
      postId: {
        type: DataTypes.STRING(15),
        allowNull: false,
        primaryKey: true,
      },
      title: {
        type: DataTypes.STRING(50),
        allowNull: false,
      },
      postImage: {
        type: DataTypes.BLOB,
        allowNull: true,
      },
      postDate: {
        type: DataTypes.DATEONLY,
        allowNull: false,
      },
      description: {
        type: DataTypes.TEXT,
        allowNull: true,
      },
      likes: {
        type: DataTypes.INTEGER,
        allowNull: false,
      },
    },
    sequelize
  );
};
```

```

        defaultValue: 0,
      },
    },
    { sequelize, modelName: "UserPost", tableName: "Tbl_UserPost" }
  );

  return UserPost;
};

```

3.4.2 Perancangan Kelas LibraryModel

```

"use strict";
const { Model } = require("sequelize");

module.exports = (sequelize, DataTypes) => {
  class Library extends Model {
    static associate(models) {
      Library.belongsTo(models.User, {
        foreignKey: "userId",
        onDelete: "CASCADE",
      });

      Library.belongsTo(models.Book, {
        foreignKey: "bookId",
        onDelete: "CASCADE",
      });
    }
  }
  Library.init(
    {
      libraryId: {
        type: DataTypes.STRING(15),
        allowNull: false,
        primaryKey: true,
      },

      readStatus: {
        type: DataTypes.STRING(20),
        allowNull: true,
      },
      dateAdded: {
        type: DataTypes.DATEONLY,
        allowNull: true,
      },
      pageProgress: {
        type: DataTypes.INTEGER,

```

```

        allowNull: true,
      },
      listType: {
        type: DataTypes.STRING,
        allowNull: true,
      },
    },
    { sequelize, modelName: "Library", tableName: "Tbl_Library" }
  );

  return Library;
};

```

3.4.3 Perancangan Kelas BookModel

```

"use strict";
const { Model } = require("sequelize");

module.exports = (sequelize, DataTypes) => {
  class Book extends Model {
    static associate(models) {
      Book.hasMany(models.Library, {
        foreignKey: "bookId",
        onDelete: "CASCADE",
      });

      Book.hasMany(models.Review, {
        foreignKey: "bookId",
        onDelete: "CASCADE",
      });
    }
  }
  Book.init(
    {
      bookId: {
        type: DataTypes.STRING(10),
        allowNull: false,
        primaryKey: true,
      },
      title: {
        type: DataTypes.CHAR(50),
        allowNull: false,
      },
      author: {
        type: DataTypes.STRING(50),
        allowNull: true,

```



```

    },
    description: {
      type: DataTypes.TEXT,
      allowNull: true,
    },
    rating: {
      type: DataTypes.FLOAT,
      allowNull: false,
    },
    genre: {
      type: DataTypes.STRING(40),
      allowNull: false,
    },
    coverImage: {
      type: DataTypes.STRING,
      allowNull: true,
    },
    page: {
      type: DataTypes.INTEGER,
      allowNull: false,
    },
  },
  { sequelize, modelName: "Book", tableName: "Tbl_Book" }
);

return Book;
};

```

3.4.4 Perancangan Kelas UserPostModel

```

"use strict";
const { Model } = require("sequelize");

module.exports = (sequelize, DataTypes) => {
  class UserPost extends Model {
    static associate(models) {
      UserPost.belongsTo(models.User, {
        foreignKey: "userId",
        onDelete: "CASCADE",
      });
    }
  }
  UserPost.init(
    {

```

```

    postId: {
      type: DataTypes.STRING(15),
      allowNull: false,
      primaryKey: true,
    },
    title: {
      type: DataTypes.STRING(50),
      allowNull: false,
    },
    postImage: {
      type: DataTypes.BLOB,
      allowNull: true,
    },
    postDate: {
      type: DataTypes.DATEONLY,
      allowNull: false,
    },
    description: {
      type: DataTypes.TEXT,
      allowNull: true,
    },
    likes: {
      type: DataTypes.INTEGER,
      allowNull: false,
      defaultValue: 0,
    },
  },
  { sequelize, modelName: "UserPost", tableName: "Tbl_UserPost" }
);

return UserPost;
};

```

3.4.5 Perancangan Kelas ReviewsModel

```

"use strict";
const { Model } = require("sequelize");

module.exports = (sequelize, DataTypes) => {
  class Review extends Model {
    static associate(models) {
      Review.belongsTo(models.User, {
        foreignKey: "userId",
        onDelete: "CASCADE",
      });

      Review.belongsTo(models.Book, {
        foreignKey: "bookId",

```

```

        onDelete: "CASCADE",
      });
    }
  }
  Review.init(
    {
      reviewId: {
        type: DataTypes.STRING(20),
        allowNull: false,
        primaryKey: true,
      },

      reviewText: {
        type: DataTypes.TEXT, // Correct data type for a text field
        allowNull: false,
      },

      dateAdded: {
        type: DataTypes.DATEONLY,
        allowNull: false,
      },
    },
    { sequelize, modelName: "Review", tableName: "Tbl_Review" }
  );

  return Review;
};

```

3.4.6 Perancangan Kelas User

```

"use strict";
const { Model } = require("sequelize");

module.exports = (sequelize, DataTypes) => {
  class User extends Model {
    static associate(models) {
      User.hasMany(models.Library, {
        foreignKey: "userId",
        onDelete: "CASCADE",
      });

      User.hasMany(models.UserPost, {
        foreignKey: "userId",
        onDelete: "CASCADE",
      });

      User.hasMany(models.Review, {

```

```

        foreignKey: "userId",
        onDelete: "CASCADE",
    });
}
}
User.init(
{
    userId: {
        type: DataTypes.STRING(10),
        allowNull: false,
        primaryKey: true,
    },
    email: {
        type: DataTypes.STRING(50),
        allowNull: false,
    },
    password: {
        type: DataTypes.CHAR(128),
        allowNull: false,
    },
    phoneNumber: {
        type: DataTypes.STRING(15),
        allowNull: true,
    },
    dateOfBirth: {
        type: DataTypes.DATEONLY,
        allowNull: true,
    },
    userName: {
        type: DataTypes.STRING(15),
        allowNull: false,
    },
    userImage: {
        type: DataTypes.BLOB,
        allowNull: false,
        defaultValue: "./images/userImgPlaceholder.png",
    },
},
{ sequelize, modelName: "User", tableName: "Tbl_User" }
);

return User;
};

```

3.5 Matriks Keruntutan (Requirement Traceability Matrix)

Kode	Nama Functional	Nama Use Case
------	-----------------	---------------

Prodi S1 Informatika Tel-U	DPPL-Diaspora	Halaman 60 dari 61
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

FR	Requirement	
FR-01	Cari Buku	Mencari Buku
FR-02	Data Buku	Membaca Data Buku
FR-03	Katalog	Melihat Katalog
FR-04	Menambah buku	Menambah Buku Baru ke dalam Sistem
FR-05	Review	Membaca review dan rating
FR-06	Akses Profile	Mengakses profile user lain
FR-07	Sign in/Regist	Login / Logout
FR-08	Setting Akun	Konfigurasi Akun
FR-09	Postingan	Membuat atau menghapus postingan
FR-10	Library	Menambahkan / Menghapus buku dari Library