

DPPL-001

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

iLOOK

untuk:

Masyarakat yang membutuhkan liburan

Dipersiapkan oleh:


Azhar Nugraha Darajat	1301194064
Chara Maria E.Y	1301194348
Ilo Raditio Wiriawan	1301194198
Muhammad Fachri Habibi	1301194110

Program Studi S1 Teknik Informatika – Fakultas Informatika

Universitas Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

 UNIVERSITAS Telkom	Program Studi S1 Teknik Informatika - Fakultas Informatika	Nomor Dokumen		Halaman
		<i>DPPL-0001</i>		
		Revisi	-	<i>16 Juni 2021</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
Tujuan Penulisan Dokumen	7
Lingkup Masalah	7
Definisi dan Istilah	7
Referensi	8
Ikhtisar Dokumen	8
Deskripsi Perancangan Global	10
Rancangan Lingkungan Implementasi	10
Deskripsi Arsitektural	10
Deskripsi Komponen	12
Perancangan Rinci	14
Realisasi Use Case	14
3.1.1 Use Case Registrasi	15
3.1.1.1 Perancangan Antarmuka Use Case 1	17
3.1.1.2 Sequence Diagram	18
3.1.2 Use Case Login	19
3.1.2.1 Perancangan Antarmuka Use Case 2	20
3.1.2.2 Sequence Diagram	21
3.1.3 Use Case Show Wisata	22
3.1.3.1 Perancangan Antarmuka Use Case 3	23
3.1.3.2 Sequence Diagram	23
3.1.4 Use Case Search	23
3.1.4.1 Perancangan Antarmuka Use Case 4	24
3.1.4.2 Sequence Diagram	25
3.1.5 Use Case Tambah planner	25
3.1.5.1 Perancangan Antarmuka Use Case 5	26
3.1.5.2 Sequence Diagram	26
3.1.6 Use Case Hapus planner	27
3.1.6.1 Perancangan Antarmuka Use Case 6	28
3.1.6.2 Sequence Diagram	29
3.1.7 Use Case Memberi Rating	29
3.1.7.1 Perancangan Antarmuka Use Case 7	31
3.1.7.2 Sequence Diagram	32
3.1.8 Use Case Tambah Pariwisata	32
3.1.8.1 Perancangan Antarmuka Use Case 8	33
3.1.8.2 Sequence Diagram	34
3.1.9 Use Case Edit Pariwisata	35
3.1.9.1 Perancangan Antarmuka Use Case 9	36
3.1.9.2 Sequence Diagram	37

3.1.10 Use Case Hapus Pariwisata	37
3.1.10.1 Perancangan Antarmuka Use Case 10	38
3.1.10.2 Sequence Diagram	38
3.1.11 Use Case Tambah Paket	39
3.1.11.1 Sequence Diagram	40
3.1.11.2 Sequence Diagram	41
3.1.12 Use Case Edit Paket	41
3.1.12.1 Perancangan Antarmuka Case 12	42
3.1.11.2 Sequence Diagram	43
3.1.12 Use Case Hapus Paket	43
3.1.12.1 Perancangan Antarmuka Case 12	44
3.1.12.2 Sequence Diagram	45
3.1.13 Use Case Logout	45
3.1.13.1 Perancangan Antarmuka Use Case 11	46
3.1.13.2 Sequence Diagram	46
3.2 Perancangan Detail Kelas	47
3.3 Diagram Kelas Keseluruhan	48
3.4 Algoritma/Query	48
3.4.1 Algoritma #1	48
3.4.2 Algoritma #2	49
3.4.3 Algoritma #3	50
3.4.4 Algoritma #4	50
3.4.5 Algoritma #5	51
3.4.6 Algoritma #6	51
3.4.7 Algoritma #7	52
3.4.8 Algoritma #8	53
3.4.9 Algoritma #9	53
3.4.10 Algoritma #10	55
3.5 Diagram Statechart	56
3.5.1 Register	56
3.5.2 Login	57
3.5.3 Show Wisata	58
3.5.4 Tambah planner	59
3.5.5 Hapus planner	60
3.5.6 Tambah Pariwisata	61
3.5.7 Edit Pariwisata	62
3.5.8 Search	63
3.5.9 Memberi Rating	64
3.5.10 Logout	65
4 Matriks Keruntutan	65

1. Pendahuluan

1.1 Tujuan Penulisan Dokumen

Tujuan pembuatan DPPL ini ditujukan untuk memenuhi tugas besar mata kuliah RPL: Desain dan Implementasi serta mendokumentasikan segala kegiatan atau aktivitas yang dilakukan selama perkembangan proyek perangkat lunak mobile iLook dimulai dari tahap *user requirements*, analisis dan desain, implementasi hingga *testing*. Selain itu penulisan dokumen ini akan digunakan sebagai salah satu acuan dalam pengimplementasian. Tujuan dari perangkat lunak iLook ini yaitu Pengguna dapat melihat daftar pariwisata yang terdapat di Jawa Barat, dan juga pelanggan dapat membuat sebuah planner pariwisata yang ingin dikunjungi.

1.2 Lingkup Masalah

iLook merupakan sebuah website dan juga perangkat lunak yang berbasis mobile yang dapat memudahkan bagi pengguna khususnya bagi masyarakat yang sedang mencari tempat-tempat pariwisata yang terdapat di Jawa Barat. Dengan adanya iLook pengguna dapat melihat tempat pariwisata yang berada di Jawa Barat dengan mudah melalui perangkat lunak ini. Selain itu mereka dapat menambahkan pariwisata yang ingin dikunjungi ke dalam planner. Dan juga mereka dapat melihat paket pariwisata yang telah disediakan oleh tour agency.

1.3 Definisi dan Istilah

Berikut ini adalah beberapa definisi, singkatan, dan akronim yang terdapat di dalam dokumen ini:

Table 1 Daftar definisi dan Akronim

Kata Kunci atau frase	Definisi dan atau akronim
DPPL	Dokumen yang mendeskripsikan dan menjabarkan secara terperinci mengenai perancangan perangkat lunak yang akan dibangun.
SDP	<i>Software Development Plan</i>
DPPL	Dokumen yang mendeskripsikan dan menjabarkan secara terperinci

	mengenai perancangan perangkat lunak yang akan dibangun.
RPPL	Rencana Pengembangan Perangkat Lunak Dokumen yang berisi rencana pengembangan perangkat lunak dalam suatu proyek, yang mencakup gambaran umum proyek, gambaran manajerial proyek, dan gambaran teknik proyek.
SRS	<i>Software Requirement Specification</i>
SKPL	Spesifikasi Kebutuhan Perangkat Lunak Dokumen hasil analisis yang berisi spesifikasi kebutuhan <i>user</i>
RPL	Rekayasa Perangkat Lunak Kegiatan pengembangan perangkat lunak
IEEE	<i>Institute of Electrical and Electronics Engineers</i> Standar internasional untuk pengembangan dan rancangan produk
ANSI	<i>American Standard Institute</i> Lembaga standarisasi Amerika

1.4 Referensi

Dokumen ini merujuk pada hasil wawancara dan observasi pada masyarakat tentang informasi yang berkaitan dengan berbagai kebutuhan yang mencakup data secara umum. diperlukannya dan penulisan dokumen berdasarkan pada :

1. IEEE Std 830-1993, *IEEE Recommended Practice for Software Requirement Specifications*.
2. Pressman, R.S., *"Software Engineering, a Practitioner's Approach"* Seventh Edition, McGraw Hill, 2009.
3. Spesifikasi Kebutuhan Perangkat Lunak (SKPL) Aplikasi Rekomendasi Pariwisata Jawa Barat

1.5 Ikhtisar Dokumen

Dokumen DPPL ini berisikan deskripsi rancangan perangkat lunak sistem informasi perpustakaan yang akan dikembangkan berdasarkan dokumen SKPL. Pada dokumen DPPL ini akan dijelaskan rincian dari rancangan perangkat lunak sehingga

dapat diimplementasikan. Dokumen ini secara garis besar terdiri dari empat bab dengan perincian sebagai berikut:

A. Pendahuluan

Pendahuluan berisi penjelasan tentang dokumen DPPL yang mencakup tujuan pembuatan dokumen ini, lingkup masalah yang diselesaikan oleh perangkat lunak yang dikembangkan, definisi, aturan penamaan dan penomoran, referensi, dan ikhtisar dokumen.

B. Deskripsi Perancangan Global

Deskripsi perancangan global berisi tentang rancangan dari perangkat lunak yang akan dibangun meliputi, rancangan lingkungan implementasi, deskripsi arsitektural, dan deskripsi komponen.

C. Perancangan Rinci

Perancangan rinci pada dokumen ini berisi tentang realisasi *use case*, perancangan detil kelas, deskripsi diagram kelas, algoritma/*query*, diagram *statechart*, perancangan antarmuka, dan perancangan representasi persistensi kelas.

D. Matriks Keterunutan

Matriks keterunutan berisi tentang hal fungsional yang terdapat pada dokumen DPPL.

2 Deskripsi Perancangan Global

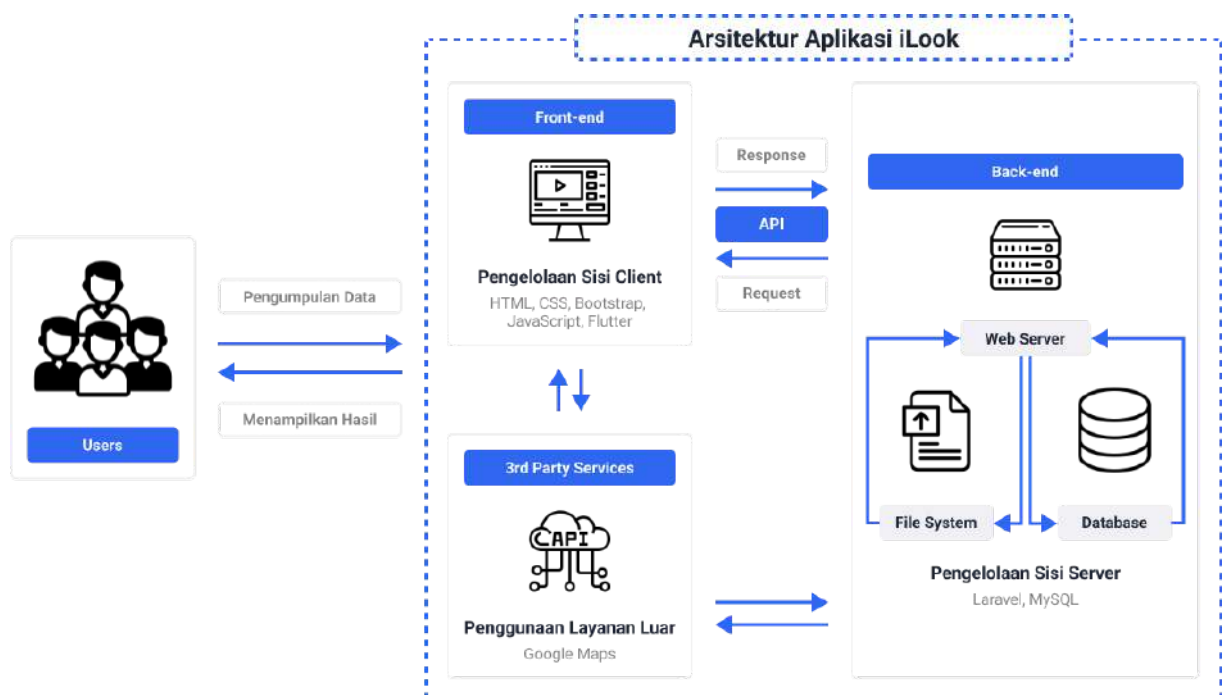
Deskripsi perancangan global meliputi penjelasan tentang rancangan lingkungan implementasi, deskripsi arsitektural, dan deskripsi komponen yang membangun perangkat lunak.

2.1 Rancangan Lingkungan Implementasi

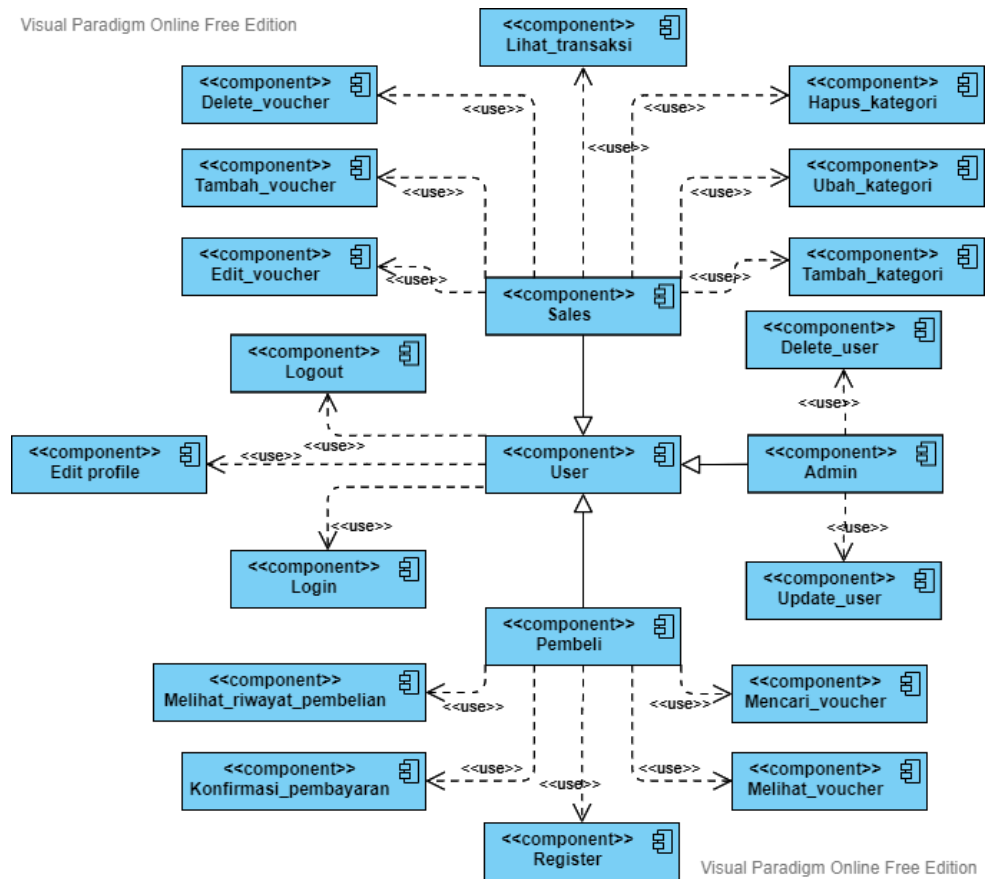
Sistem ini diimplementasikan dalam lingkungan sebagai berikut :

1. Sistem operasi : Windows, Android, Mac, Linux
2. Bahasa Pemrograman : HTML, Java Script, CSS, PHP, Dart
3. DBMS : MySQL
4. Development Tool: PHPmyAdmin(XAMPP), Code Igniter 3, Visual Studio Code, Android Studio

2.2 Deskripsi Arsitektural



Merupakan gambaran arsitektur atau komponen yang akan diterapkan pada perangkat lunak “iLook” untuk mempermudah pengembang dalam mengembangkan atau mengimplementasikan perangkat lunak ini.



Gambar 1 Component Diagram

2.3 Deskripsi Komponen

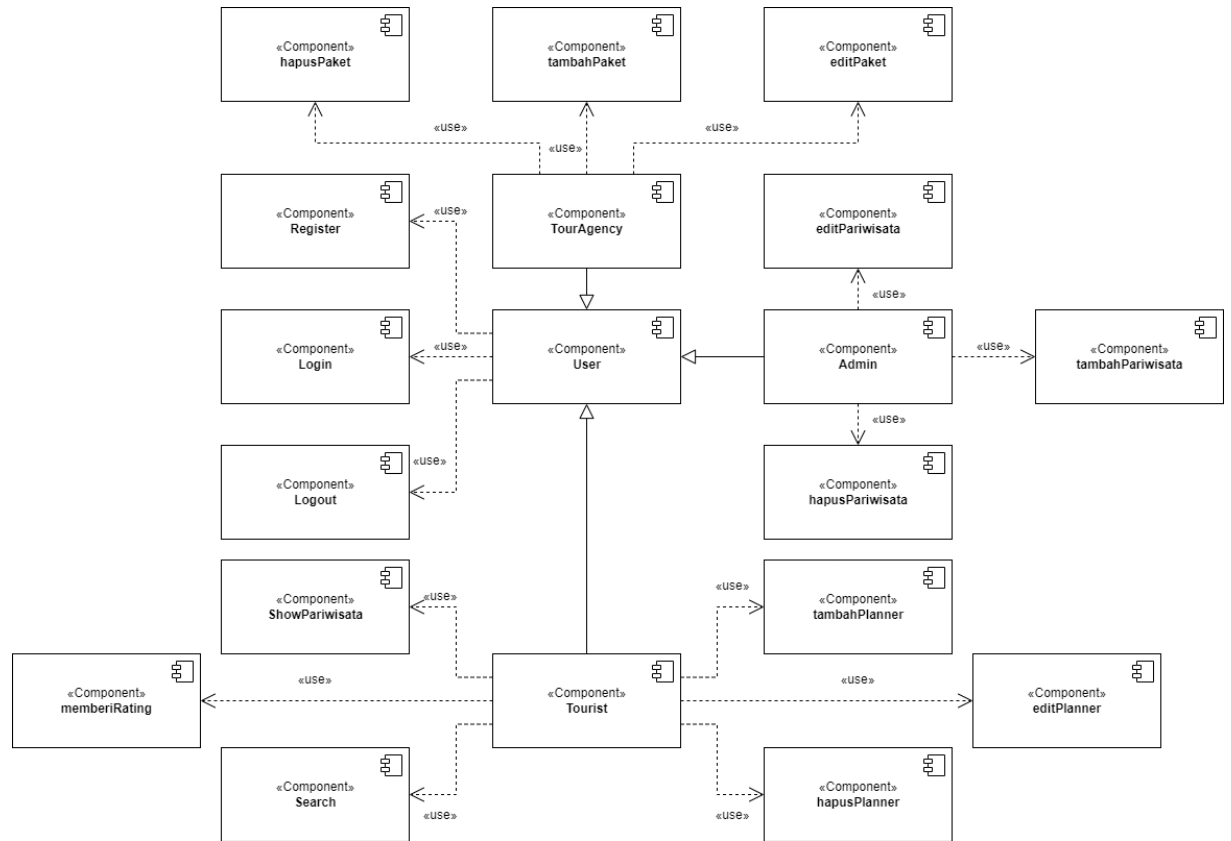


Table 4 Deskripsi Komponen

No	Nama Komponen	Keterangan
1.	Tourist	Pengguna dalam sistem iLook
2.	Admin	Pengguna dalam sistem iLook
3.	Tour Agency	Pengguna dalam sistem iLook
4.	Register	Menu untuk melakukan registrasi akun
5.	Login	Menu untuk masuk agar dapat mengakses menu lain
6.	Logout	Menu untuk keluar dari website dan perangkat lunak iLook
7.	Show Wisata	Menu untuk melihat daftar pariwisata
8.	Search	Menu untuk mencari pariwisata yang diinginkan
9.	Tambah Planner	Menu untuk membuat planner
10.	Hapus Planner	Menu untuk menghapus planner
11.	Edit Planner	Menu untuk mengedit planner
12.	Memberi Rating	Menu untuk memberi rating pada pariwisata
13.	Tambah Pariwisata	Menu untuk menambahkan pariwisata ke dalam daftar pariwisata
14.	Edit Pariwisata	Menu untuk mengubah informasi pariwisata

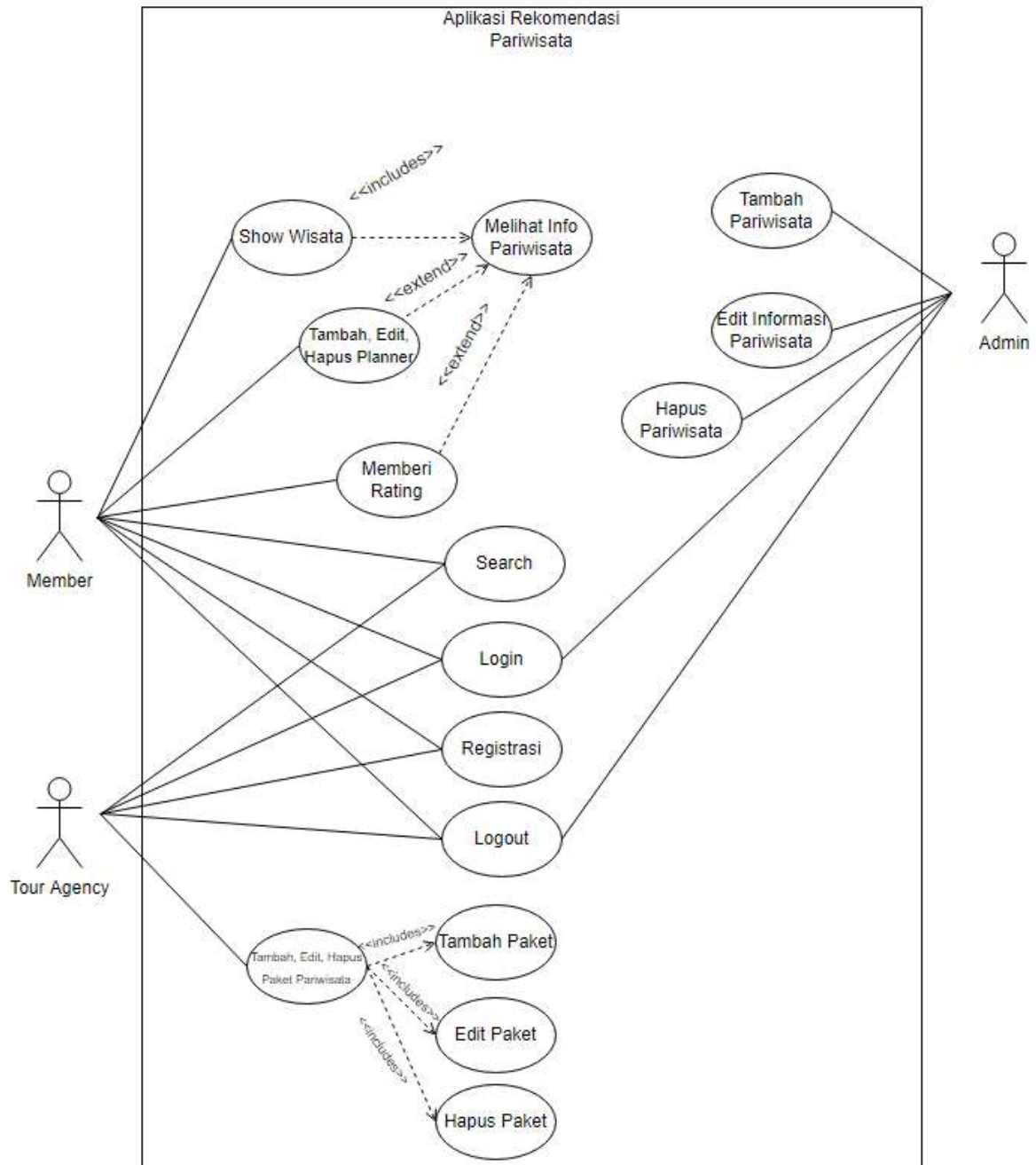
15.	Hapus Pariwisata	Menu untuk menghapus pariwisata dari daftar pariwisata
16.	Tambah Paket Pariwisata	Menu untuk menambahkan paket pariwisata
17.	Edit Paket Pariwisata	Menu untuk mengedit paket pariwisata
18.	Hapus Paket Pariwisata	Menu untuk menghapus paket pariwisata

3 Perancangan Rinci

Dalam bab perancangan rinci akan dijelaskan mengenai realisasi use case, perancangan detail kelas, diagram kelas keseluruhan, algoritma/query, diagram statechart, perancangan antarmuka, dan perancangan representasi persistensi kelas

3.1 Realisasi Use Case

Sub Bab ini menjelaskan tentang realisasi semua use case yang telah dirancang pada dokumen spesifikasi kebutuhan perangkat lunak.



No	Nama Use Case	Deskripsi Use Case
1	Registrasi	Fitur ini digunakan oleh semua aktor dan berfungsi untuk mendaftar dan membuat akun baru
2	Login	User menginputkan username dan password untuk dapat masuk ke akun dan halaman utama aplikasi
3	Show Wisata	Fitur ini digunakan oleh user untuk melihat daftar pariwisata
4	Search	User dapat mencari pariwisata yang diinginkan
5	Tambah planner	User dapat menambahkan pariwisata yang ingin dikunjungi ke dalam planner
6	Hapus planner	User dapat menghapus pariwisata yang sudah ada pada planner
7	Memberi Review	User dapat memberi ulasan terhadap sebuah pariwisata
8	Tambah Pariwisata	Admin menambahkan pariwisata ke dalam daftar pariwisata
9	Edit Pariwisata	Admin melakukan edit informasi terhadap pariwisata yang telah ada
10	Hapus Pariwisata	Admin menghapus pariwisata yang terdapat di dalam daftar pariwisata
11	Tambah Paket	Tour Agency dapat menambahkan paket ke dalam daftar paket
12	Edit Paket	Tour Agency dapat mengubah paket yang sudah ada dalam daftar paket
13	Hapus Paket	Tour Agency dapat menghapus paket yang sudah ada dalam daftar paket
14	Logout	User melakukan logout untuk mengeluarkan dari akun

3.1.1 Use Case Registrasi

Skenario Use Case 1: Registrasi

Skenario Use Case 1 : Registrasi

Nama Use Case	Registrasi
Deskripsi	Fitur ini digunakan oleh semua aktor dan berfungsi untuk mendaftar dan membuat akun baru
Pre-kondisi	User melakukan registrasi data dengan mengisi data yang diminta oleh sistem
Post-kondisi	User berhasil membuat akun untuk melakukan login pada aplikasi
Platform	Website dan Mobile

Table Typical Course of Event

Aktor	Sistem
1. User Membuka Aplikasi	
	2. Sistem menampilkan halaman registrasi
3. User mengisi nama, username, email, dan password	
4. User menekan tombol get started	
	5. Sistem melakukan pengecekan data, jika data username dan email telah terdaftar maka kembali ke langkah 2
	6. Jika belum tersedia maka lanjut ke langkah 7
	7. Sistem merekam data registrasi yang diinput oleh user
	8. Memproses dan menyimpan data registrasi ke dalam database
	9. Sistem menampilkan halaman login dan pop-up "Akun berhasil dibuat"

3.1.1.1 Perancangan Antarmuka Use Case 1

iLook Explore Places Privacy policy About Sign in **Get started**

Getting started with us

Please fill out the form below first, then you're good to go. Already have an account? [Sign in](#).

Pick your role

User

Tour guide

Next

SSS

iLook Explore Places Privacy policy About Sign in **Get started**

Getting started with us

Please fill out the form below first, then you're good to go. Already have an account? [Sign in](#).

Set up your account

Name
Enter your name

Username
Create your username

Phone number
Enter your phone

Password
Contain at least 8 characters

[Back](#) **Register**

By clicking Register, I have read and understood the [Privacy policy](#).

Getting started with us

Please fill out the form below first, then you're good to go. Already have an account? [Sign in](#).

Set up your account

Agency name

Username

Phone number

Password



Location

[Back](#)
[Register](#)

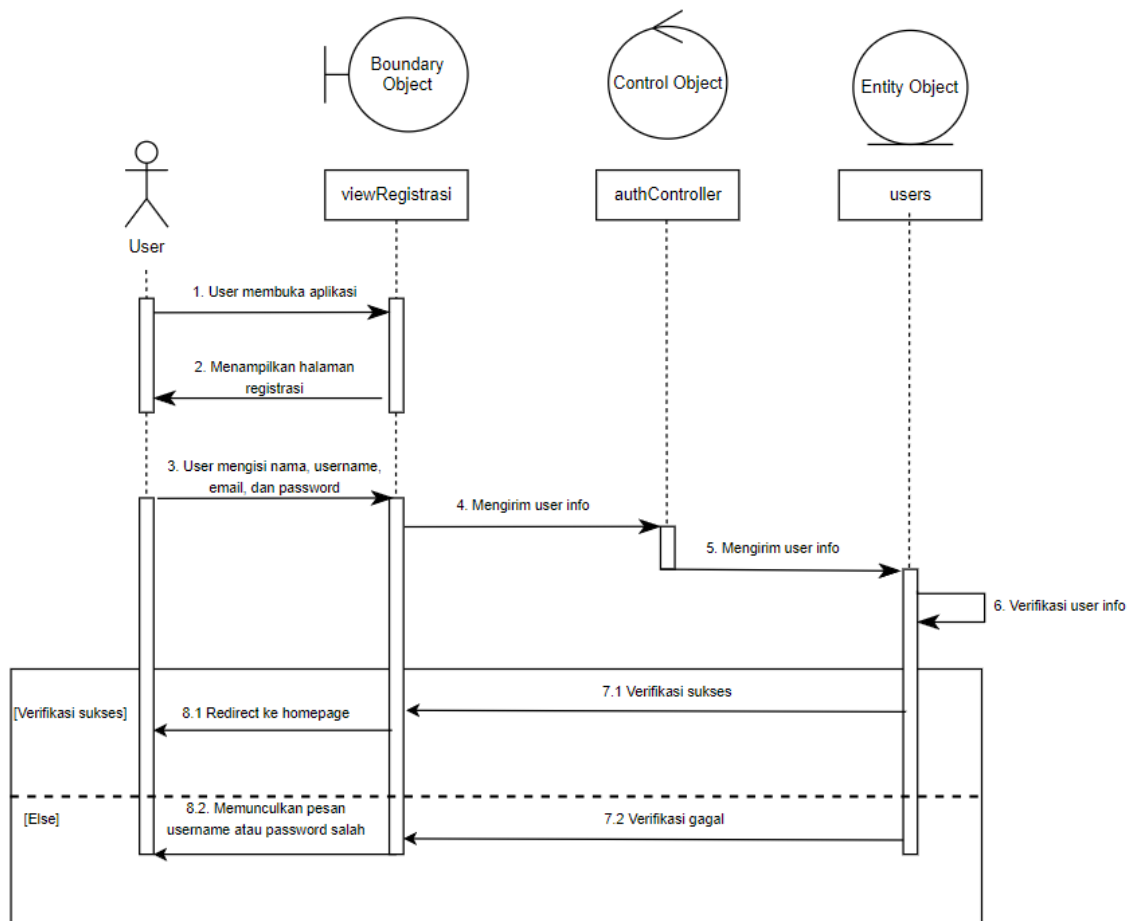

Lets Get Started

Create an account to continue

[Register](#)

Already have an account ? [Sign in](#)

3.1.1.2 Sequence Diagram



3.1.2 Use Case Login

Skenario Use Case 2: Login

Skenario Use Case 2 : Login

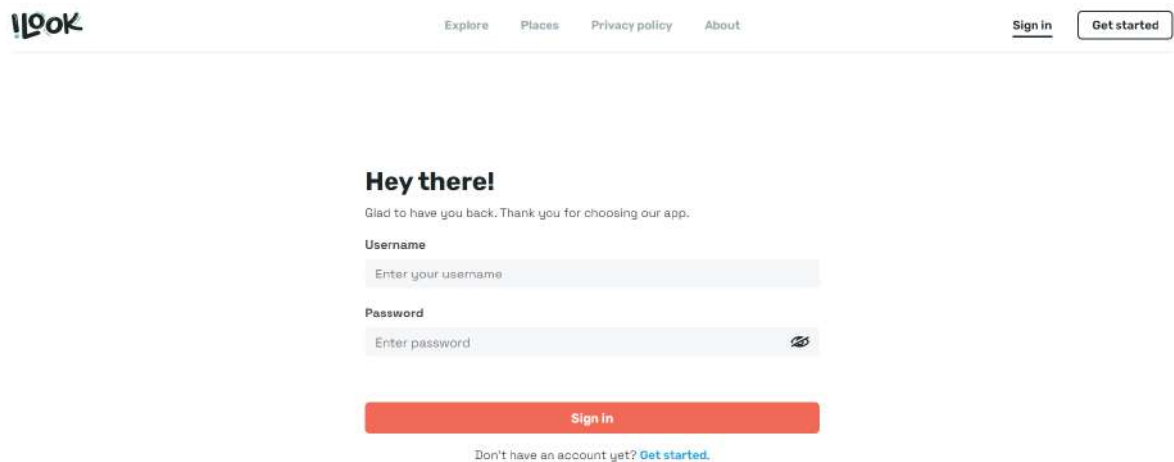
Nama Use Case	Login
Deskripsi	User menginputkan username dan password untuk dapat masuk ke akun dan halaman utama aplikasi
Pre-kondisi	User menginput username dan password
Post-kondisi	User berhasil masuk ke akun dan masuk ke halaman utama aplikasi
Platform	Website dan Mobile

Table Typical Course of Event

Aktor	Sistem
1. User Membuka Aplikasi	
	2. Sistem menampilkan halaman login

3. User mengisi username dan password akun	
	4. Sistem melakukan pengecekan data. Jika data akun yang dimasukkan salah maka akan kembali ke nomor 3, jika data sudah benar maka sistem akan menampilkan halaman utama
5. User berhasil masuk ke halaman utama	


3.1.2.1 Perancangan Antarmuka Use Case 2



iLook Explore Places Privacy policy About Sign in Get started

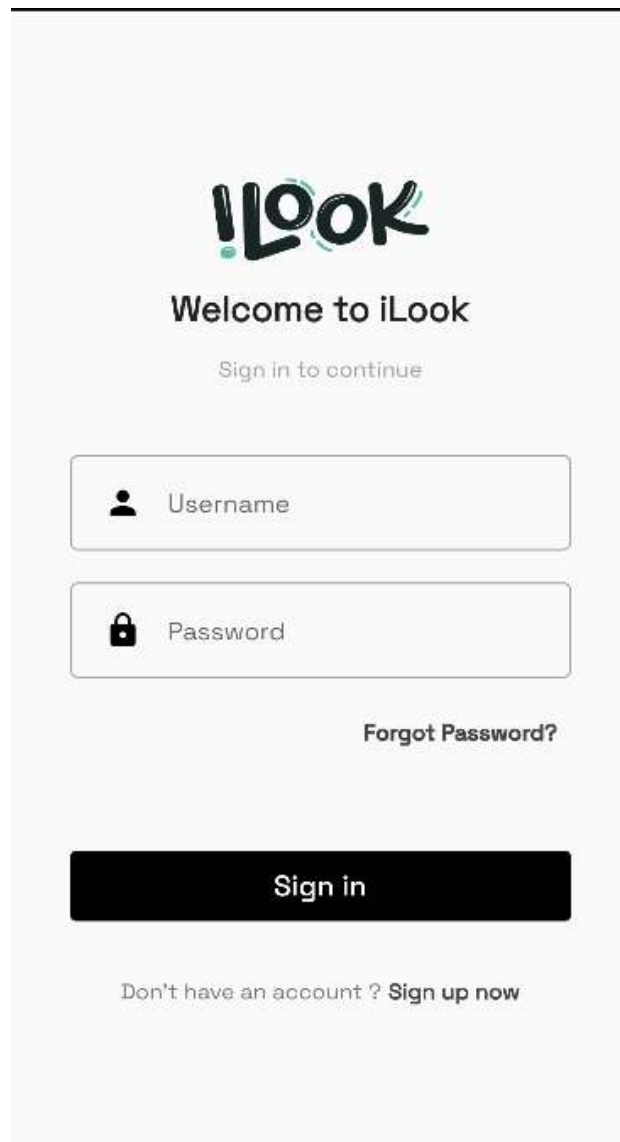
Hey there!
Glad to have you back. Thank you for choosing our app.

Username
Enter your username

Password
Enter password 

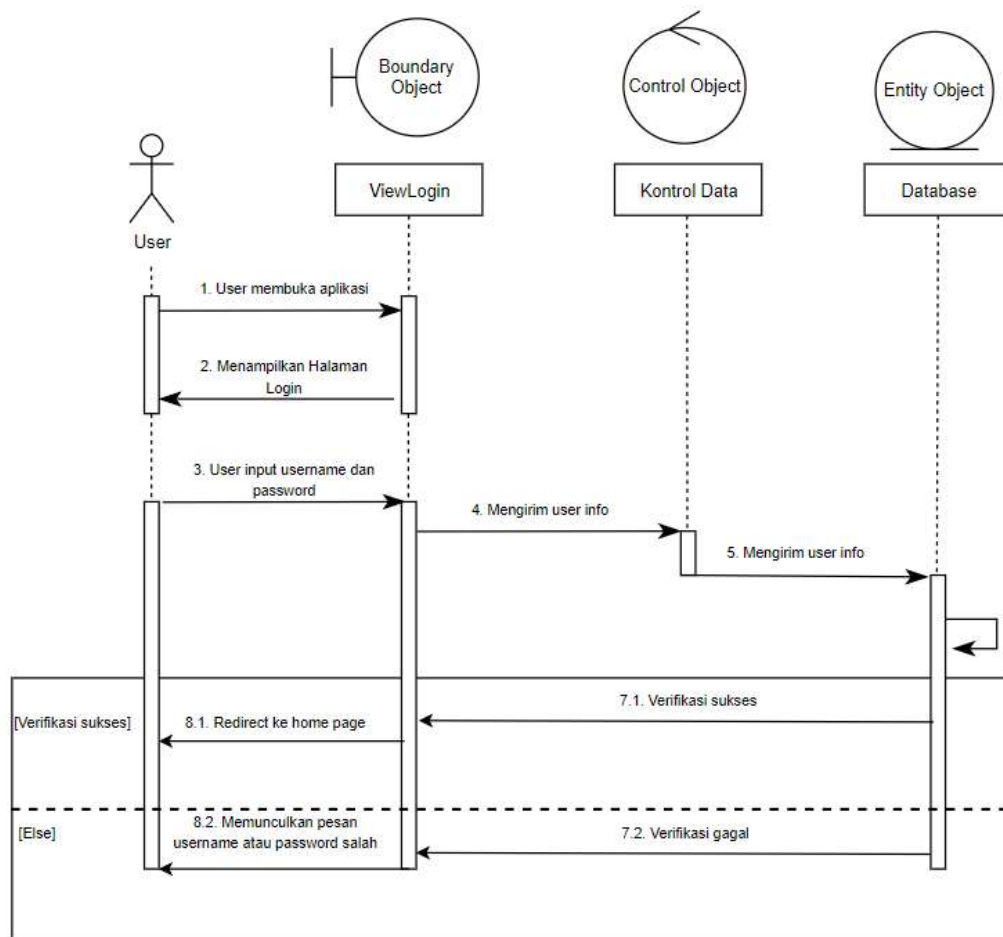
Sign in

Don't have an account yet? [Get started.](#)



The image shows a login interface for a service called 'iLook'. At the top, the 'iLook' logo is displayed in a stylized font. Below the logo, the text 'Welcome to iLook' is centered, followed by the instruction 'Sign in to continue'. There are two input fields: the first is for 'Username' with a person icon, and the second is for 'Password' with a lock icon. A link for 'Forgot Password?' is located below the password field. A large black button with the text 'Sign in' is positioned below the input fields. At the bottom, there is a link that says 'Don't have an account ? Sign up now'.

3.1.2.2 Sequence Diagram



3.1.3 Use Case Show Wisata

Skenario Use Case 3: Show Wisata

Skenario Use Case 3 : Show Wisata

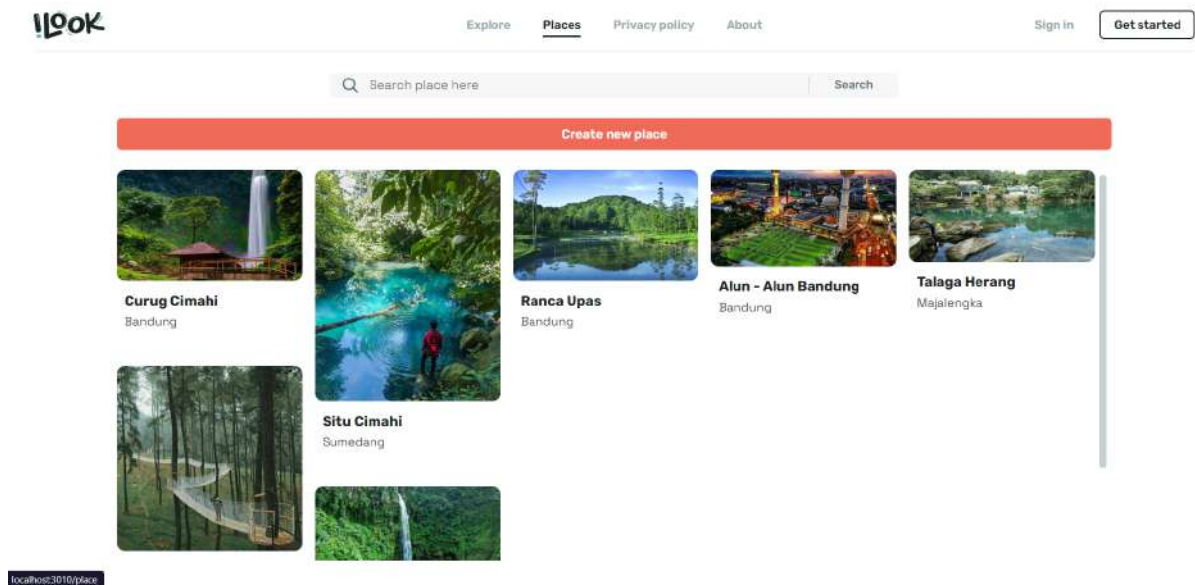
Nama Use Case	Show Wisata
Deskripsi	User dapat melihat rekomendasi wisata
Pre-kondisi	User menekan tombol explore
Post-kondisi	User berhasil mencari wisata yang populer
Platform	Website dan Mobile

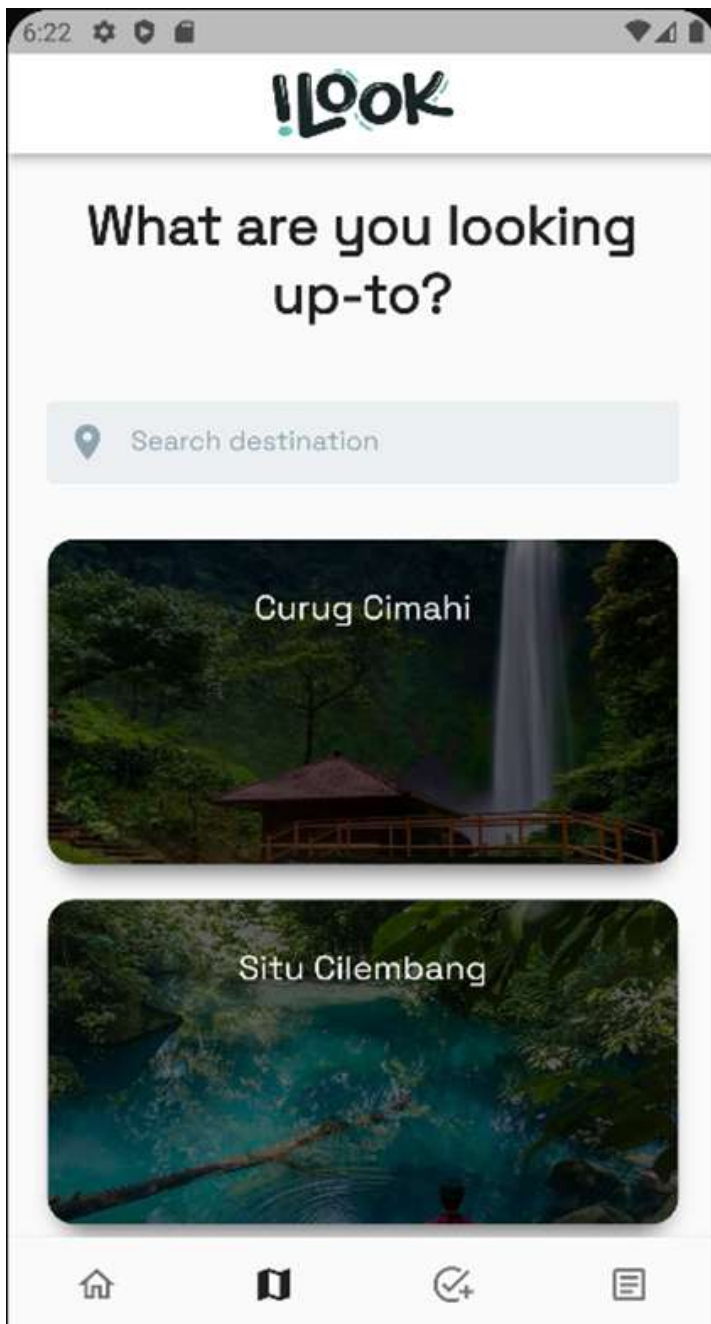
Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. User menekan tombol explore	
	3. Sistem menampilkan halaman

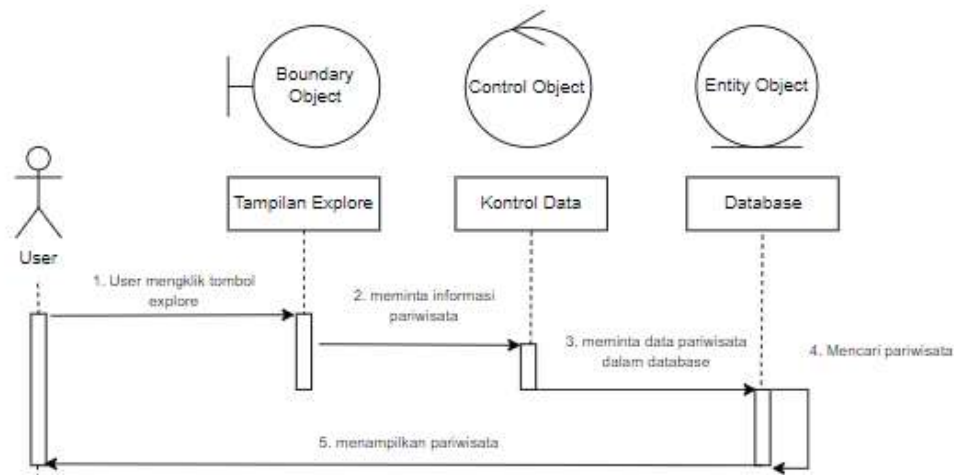
	explore
4. User memilih wisata	
	5. Sistem menampilkan info wisata

3.1.3.1 Perancangan Antarmuka Use Case 3





3.1.3.2 Sequence Diagram



3.1.4 Use Case Search

Skenario Use Case 4: Filter

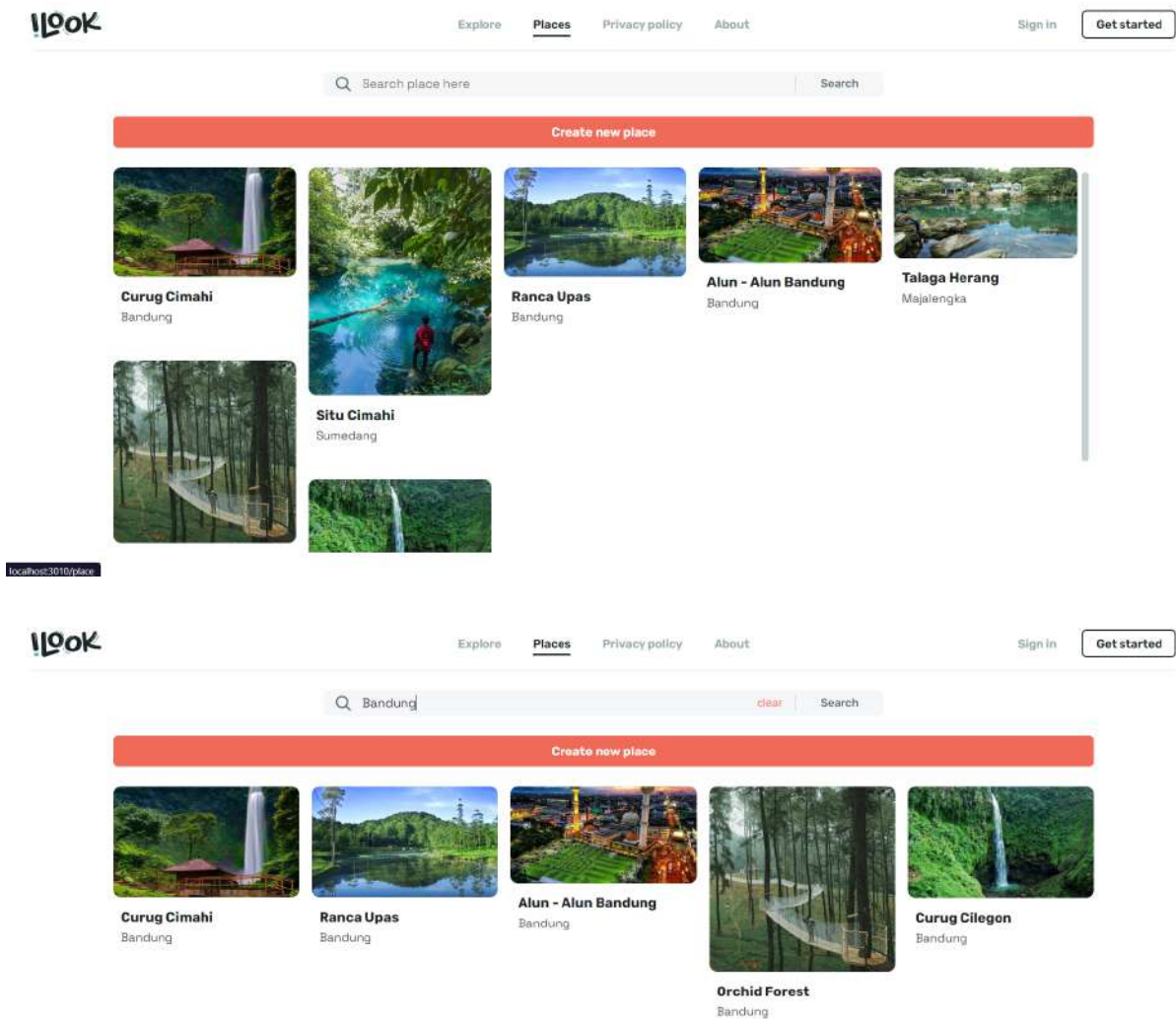
Skenario Use Case 4 : Filter

Nama Use Case	Search
Deskripsi	User dapat mencari pariwisata berdasarkan data yang dimasukan
Pre-kondisi	User menekan tombol explore
Post-kondisi	User berhasil memfilter daftar pariwisata
Platform	Website dan Mobile
Primary Flow	

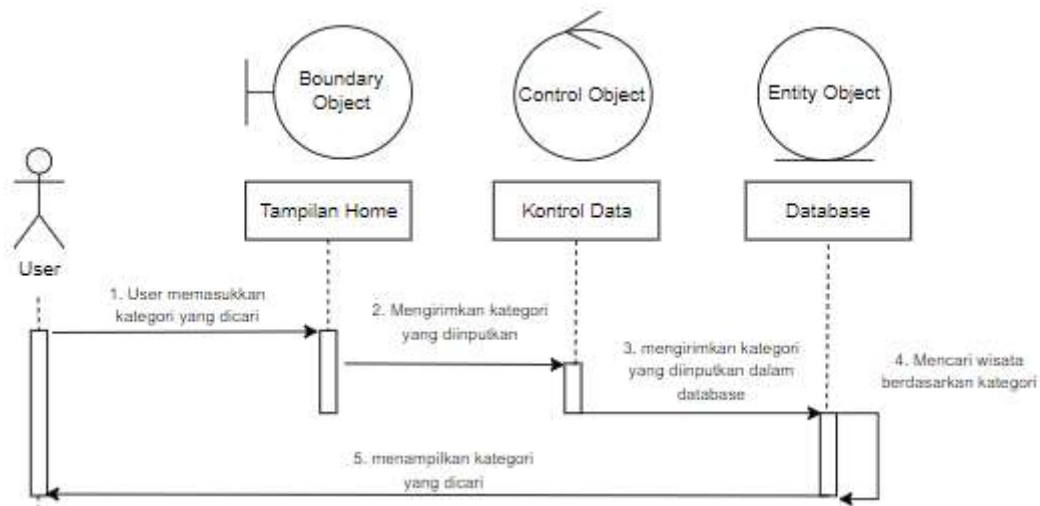
Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. User menekan tombol explore	
	3. Sistem menampilkan menu explore
4. User memasukkan pariwisata yang ingin dicari	
	5. Sistem menampilkan daftar pariwisata berdasarkan data yang dimasukkan

3.1.4.1 Perancangan Antarmuka Use Case 4



3.1.4.2 Sequence Diagram



3.1.5 Use Case Tambah planner

Skenario Use Case 5: Tambah planner

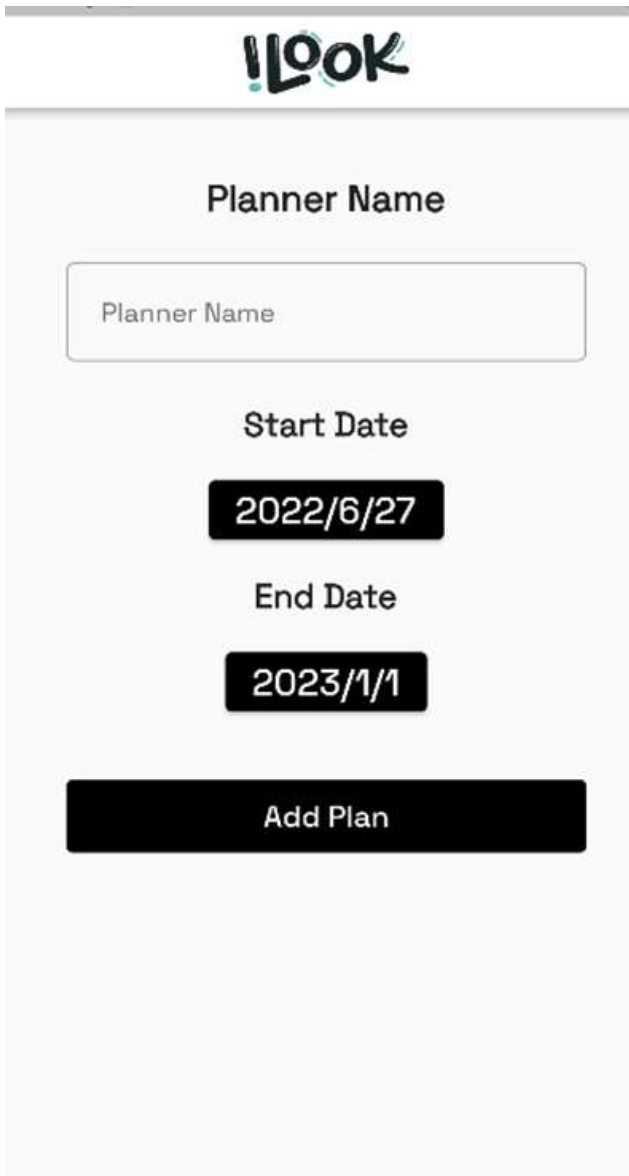
Skenario Use Case 5 : Tambah Planner

Nama Use Case	Tambah planner
Deskripsi	User dapat menambahkan tempat wisata yang diinginkan dalam planner.
Pre-kondisi	User memilih memilih wisata.
Post-kondisi	User berhasil menambahkan wisata yang diinginkan ke dalam planner
Platform	Mobile

Table Typical Course of Event

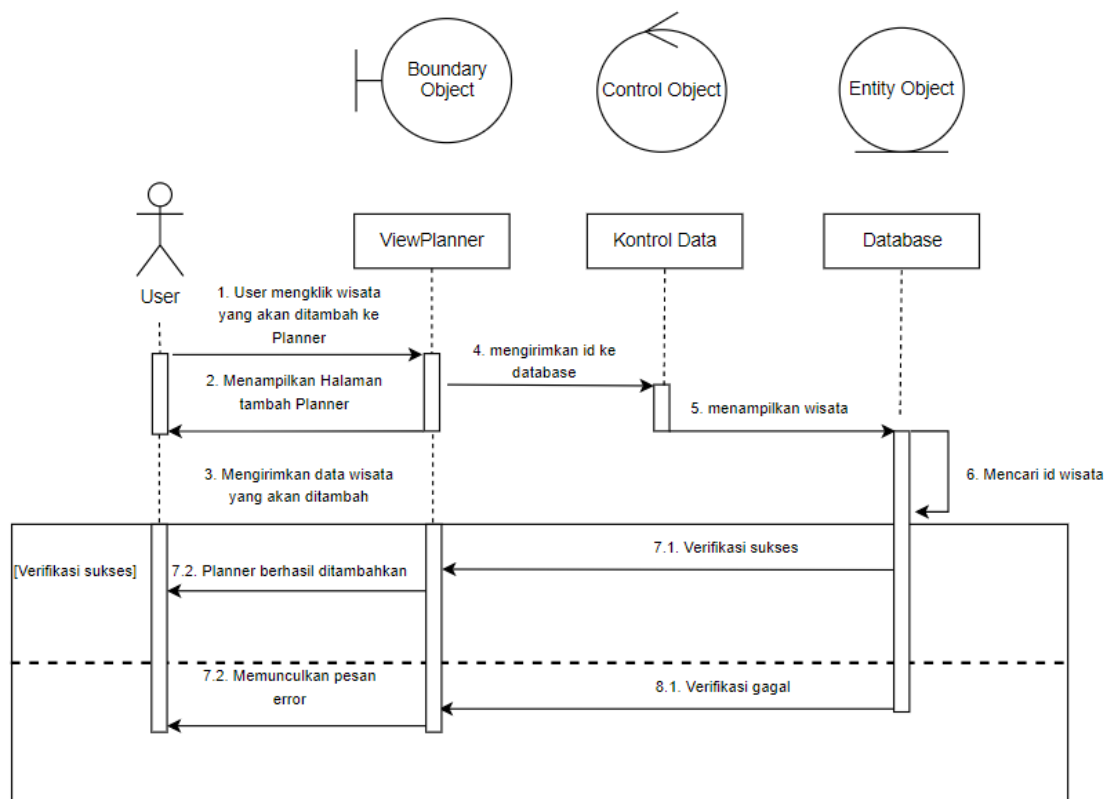
Aktor	Sistem
	6. Sistem menampilkan halaman utama
7. User memilih wisata	
	8. Sistem menampilkan informasi wisata
9. User menekan tambah planner	
	10. Sistem menyimpan wisata dalam planner
11. User menekan menu planner	

3.1.5.1 Perancangan Antarmuka Use Case 5



The screenshot displays the '!look' application interface for creating a plan. At the top is the '!look' logo. Below it, the title 'Planner Name' is centered. A text input field with the placeholder 'Planner Name' is provided. Further down, the 'Start Date' is set to '2022/6/27' and the 'End Date' is set to '2023/1/1', both displayed in black boxes. At the bottom, there is a large black button labeled 'Add Plan'.

3.1.5.2 Sequence Diagram



3.1.6 Use Case Hapus planner

Skenario Use Case 6: Hapus planner

Skenario Use Case 6 : Hapus Planner

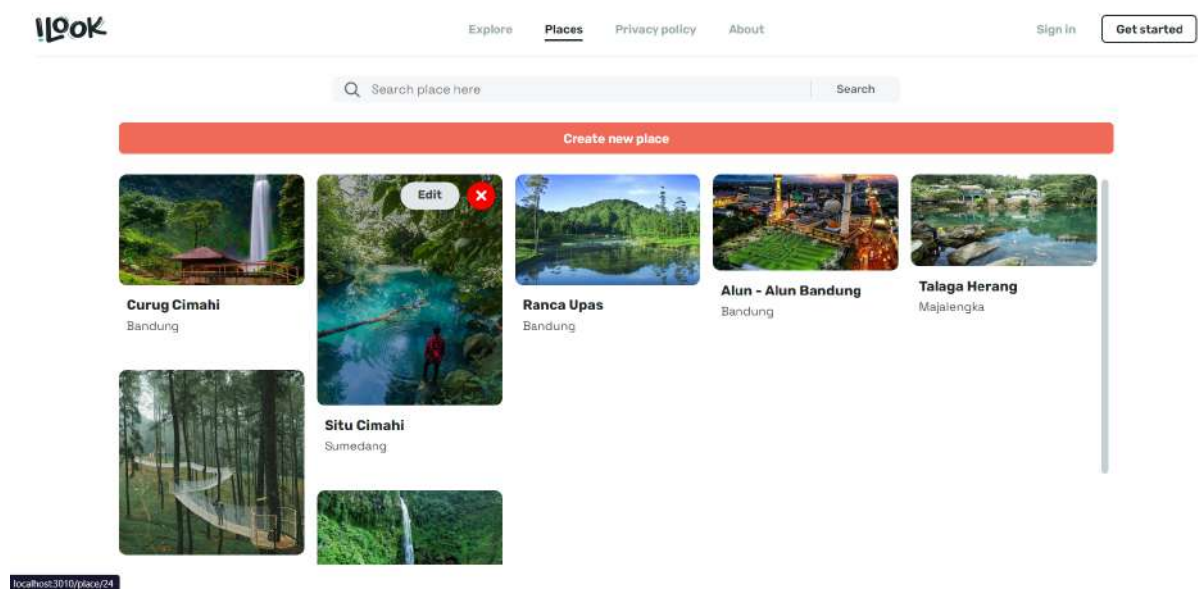
Nama Use Case	Hapus planner
Deskripsi	User dapat menghapus wisata yang sudah ada pada planner
Pre-kondisi	User memilih wisata
Post-kondisi	User berhasil menghapus wisata yang diinginkan dalam planner
Platform	Mobile
Primary Flow	

Table Typical Course of Event

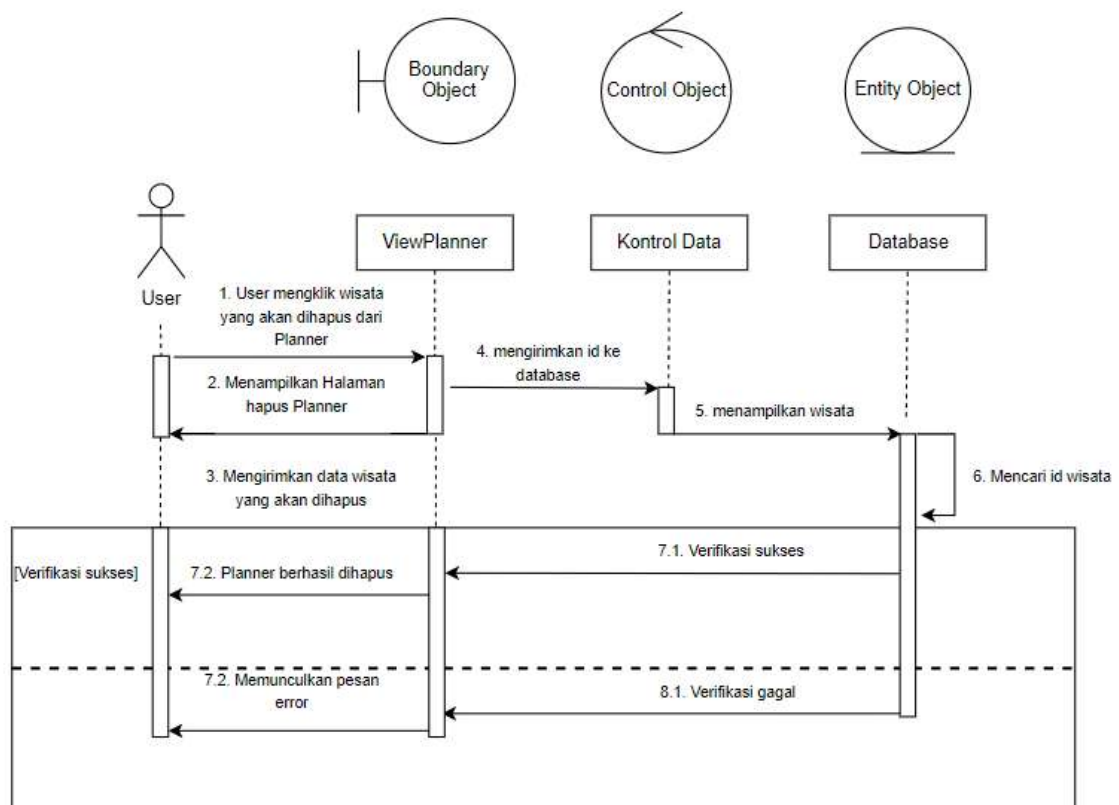
Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. User memilih menu planner	

	3. Sistem menampilkan halaman planner
4. User memilih wisata yang akan dihapus	
5. User memilih hapus	
	6. Sistem menghapus data

3.1.6.1 Perancangan Antarmuka Use Case 6



3.1.6.2 Sequence Diagram



3.1.7 Use Case Memberi Rating

Skenario Use Case 7: Memberi Rating

Skenario Use Case 7 : Memberi Rating

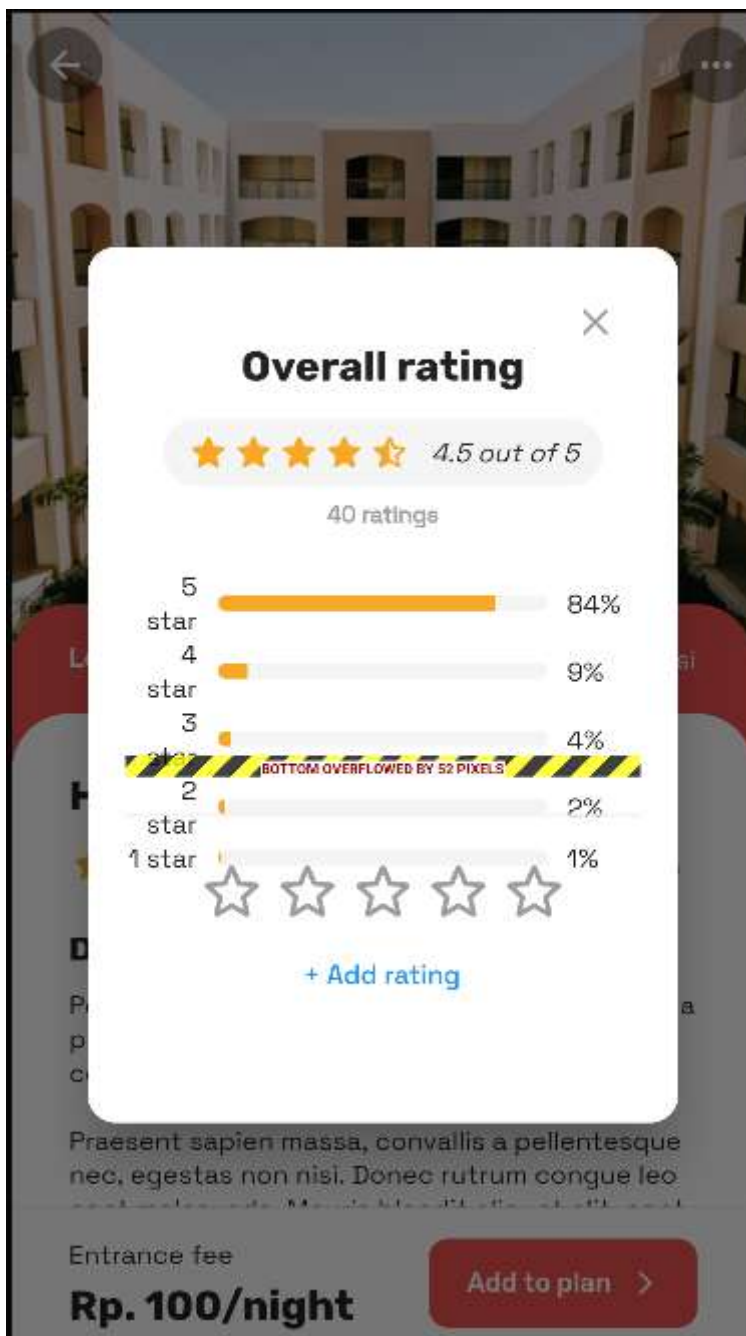
Nama Use Case	Memberi Review
Deskripsi	User dapat memberi rating terhadap sebuah pariwisata
Pre-kondisi	User memilih pariwisata
Post-kondisi	User berhasil memberikan rating terhadap pariwisata
Platform	Mobile

Table Typical Course of Event

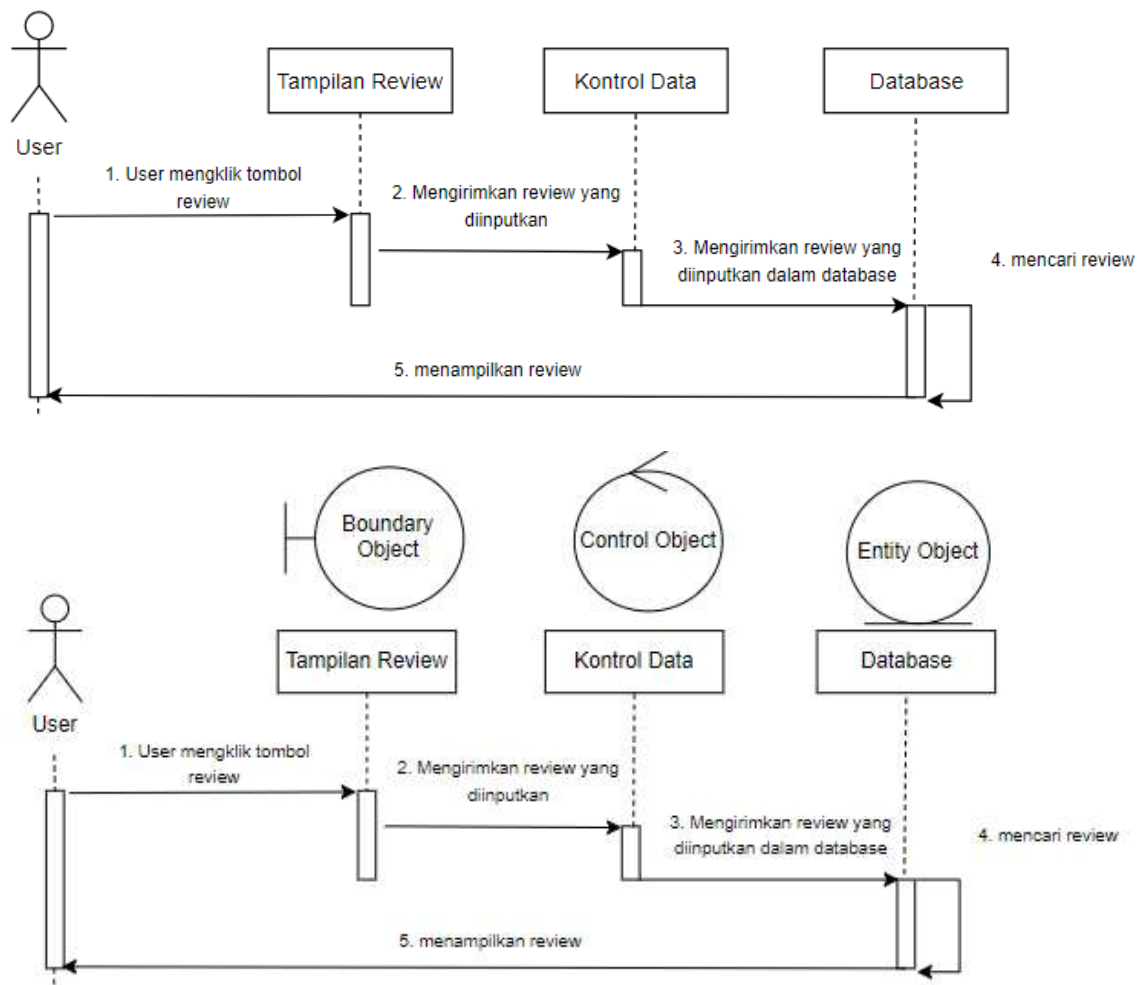
Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. User memilih menu explore	

	3. Sistem menampilkan daftar pariwisata
4. User memilih pariwisata	
	5. Sistem menampilkan informasi pariwisata
6. User memencet tombol rating	
	7. Sistem menampilkan form rating
8. User mengisi form rating	
9. User menekan tombol submit	
	10. Sistem menyimpan data review pariwisata

3.1.7.1 Perancangan Antarmuka Use Case 7



3.1.7.2 Sequence Diagram



3.1.8 Use Case Tambah Pariwisata

Skenario Use Case 8: Tambah Pariwisata

Skenario Use Case 8 : Tambah Pariwisata

Nama Use Case	Tambahkan Pariwisata
Deskripsi	Admin dapat menambahkan tempat wisata yang diinginkan dalam planner.
Pre-kondisi	Admin memilih memilih wisata.
Post-kondisi	Admin berhasil menambahkan wisata yang diinginkan ke dalam planner
Platform	Website

Table Typical Course of Event

Prodi S1 Teknik Informatika - Universitas Telkom	DPPL-001	Halaman 34 dari 68
Dokumen ini dan informasi yang ada di dalamnya adalah milik Prodi S1 Teknik Informatika-Universitas Telkom dan bersifat rahasia. Dilarang untuk mereproduksi dokumen ini tanpa diketahui oleh Program Studi S1 Teknik Informatika, Universitas Telkom		

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. Admin memilih tombol tambah pariwisata	
	3. Sistem menampilkan form tambah pariwisata
4. Admin mengisi informasi pariwisata	
5. Admin memencet tombol submit	
	6. Sistem menyimpan data pariwisata kedalam database

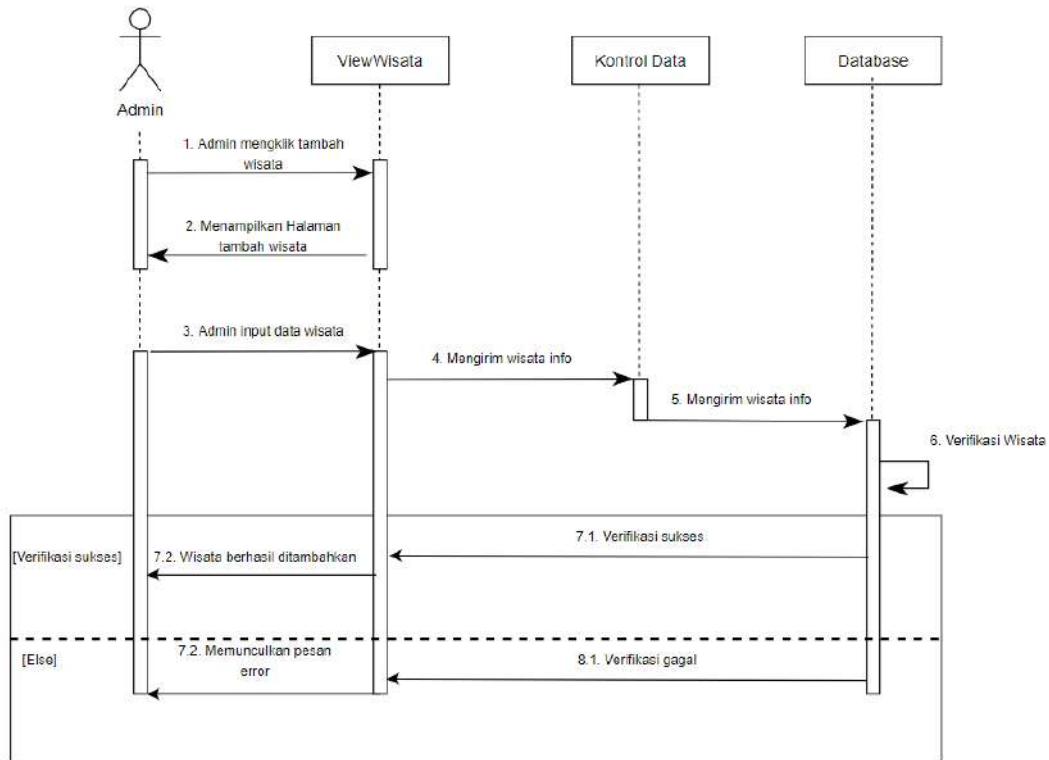
3.1.8.1 Perancangan Antarmuka Use Case 8

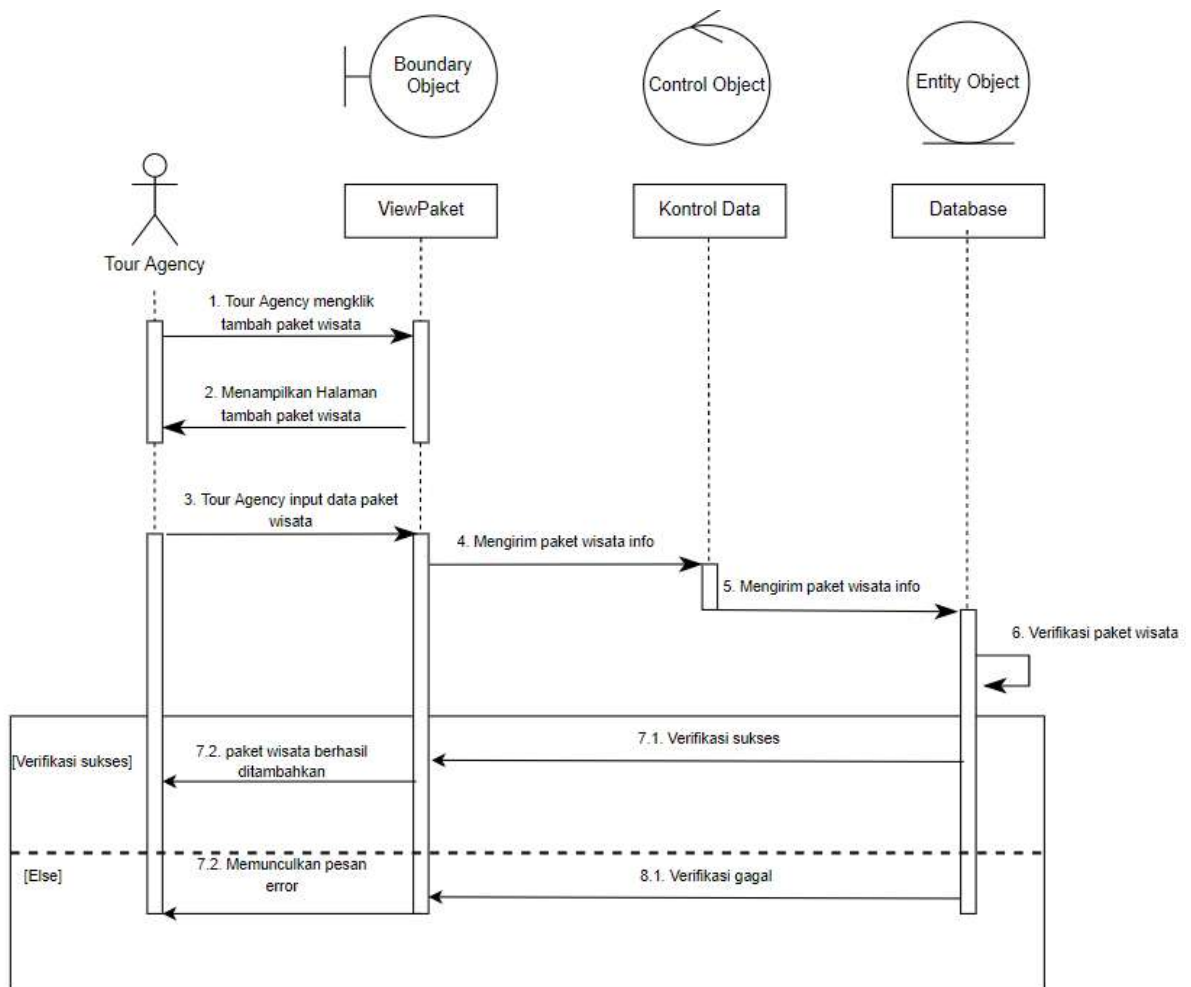
The screenshot shows the 'Create place' form in the iLook application. The form is titled 'Create place' and includes the following fields and elements:

- Thumbnail image:** A large text area with the placeholder text 'Drop your image file here or [Browse from your computer](#)'.
- Place name:** A text input field with the placeholder text 'Enter place name'.
- Category:** A dropdown menu with the selected value 'Taman Hiburan' and a downward arrow.
- Price:** A text input field with the placeholder text 'Enter ticket price'.
- Description:** A text input field with the placeholder text 'Enter place description'.

At the top of the page, there are navigation links: 'Explore', 'Places' (which is underlined), 'Privacy policy', and 'About'. On the right side, there are buttons for 'Sign in' and 'Get started'.

3.1.8.2 Sequence Diagram





3.1.9 Use Case Edit Pariwisata

Skenario Use Case 9: Edit Pariwisata

Skenario Use Case 9 : Edit Pariwisata

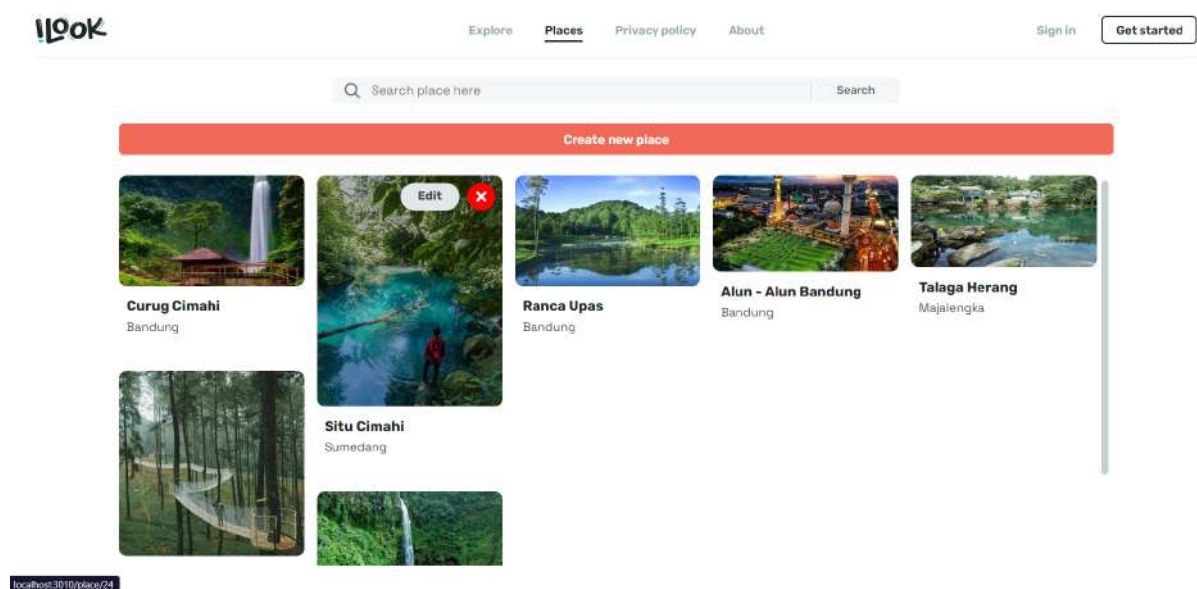
Nama Use Case	Edit Pariwisata
Deskripsi	Admin melakukan edit informasi terhadap pariwisata yang sudah ada
Pre-kondisi	Admin telah login akun
Post-kondisi	Admin berhasil mengubah informasi pariwisata
Platform	Website

Table Typical Course of Event

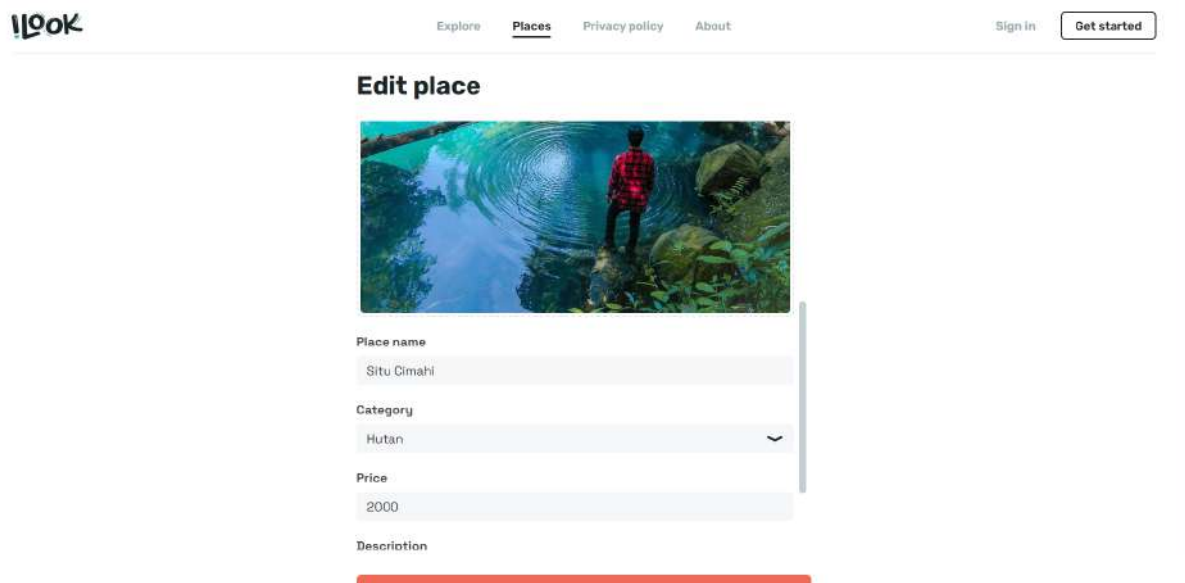
Prodi S1 Teknik Informatika - Universitas Telkom	DPPL-001	Halaman 37 dari 68
Dokumen ini dan informasi yang ada di dalamnya adalah milik Prodi S1 Teknik Informatika-Universitas Telkom dan bersifat rahasia. Dilarang untuk mereproduksi dokumen ini tanpa diketahui oleh Program Studi S1 Teknik Informatika, Universitas Telkom		

User	System
	1. Sistem menampilkan halaman utama
2. Admin memilih wisata yang ingin diedit	
	3. Sistem menampilkan info wisata
4. Admin menekan tombol edit	
	5. Sistem menampilkan halaman edit
6. Admin mengedit data yang ingin diubah	
	7. Sistem menyimpan data yang telah diubah

3.1.9.1 Perancangan Antarmuka Use Case 9




3.1.9.2 Sequence Diagram



iLook Explore **Places** Privacy policy About Sign in Get started

Edit place



Place name
Situ Cimahi

Category
Hutan

Price
2000

Description

3.1.10 Use Case Hapus Pariwisata

Skenario Use Case 10: Hapus Pariwisata

Skenario Use Case 10 : Hapus Pariwisata

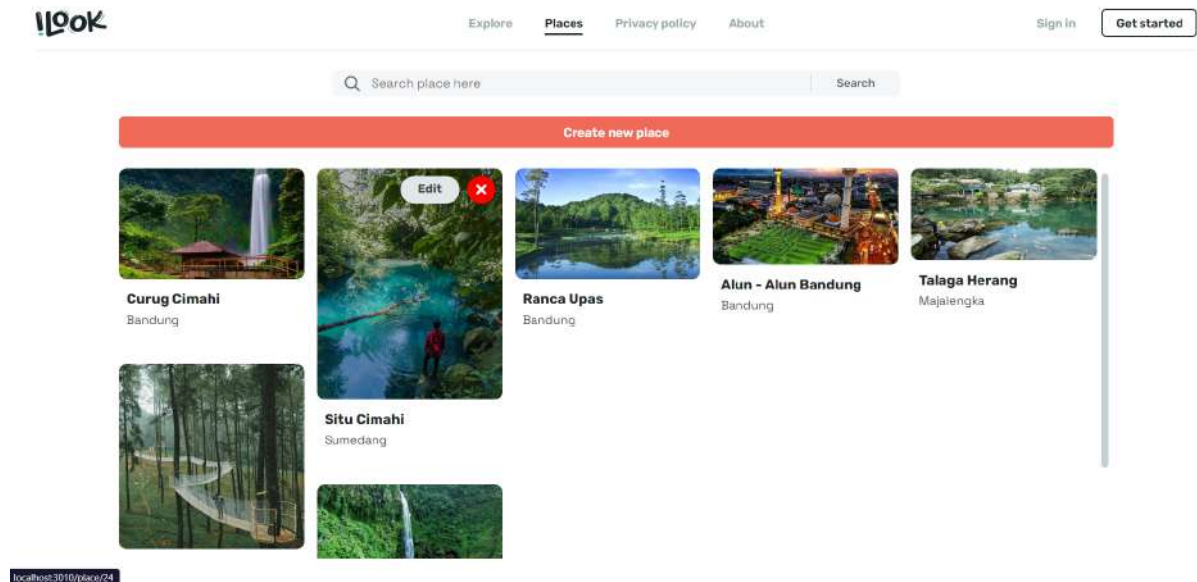
Nama Use Case	Hapus Pariwisata
Deskripsi	Admin menghapus info pariwisata yang sudah ada
Pre-kondisi	Admin telah login akun
Post-kondisi	Admin berhasil menghapus pariwisata yang ada.
Platform	Website

Table Typical Course of Event

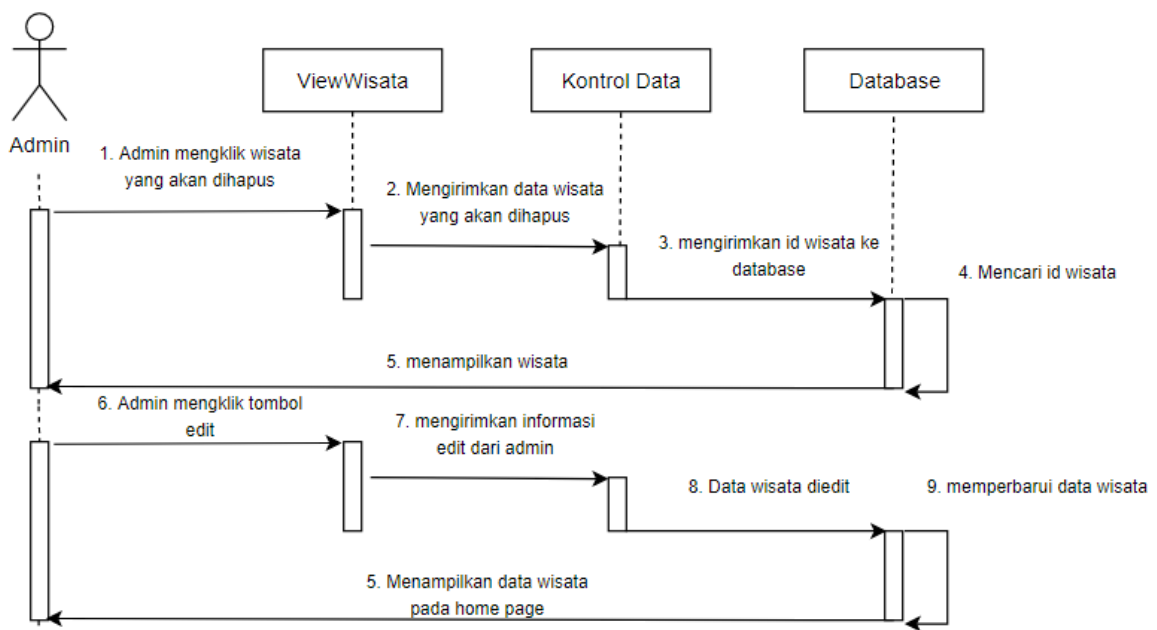
User	System
	1. Sistem menampilkan halaman utama
2. Admin memilih wisata yang ingin dihapus	
	3. Sistem menampilkan info wisata
4. Admin menekan tombol hapus	
	5. Sistem menampilkan halaman

	hapus
6. Admin mengedit data yang ingin dihapus	
	7. Sistem menyimpan data yang telah diubah

3.1.10.1 Perancangan Antarmuka Use Case 10



3.1.10.2 Sequence Diagram



3.1.11 Use Case Tambah Paket

Skenario Use Case 8: Tambah Pariwisata

Skenario Use Case 8 : Tambah Pariwisata

Nama Use Case	Tambahkan Paket
Deskripsi	Tour Agency dapat menambahkan paket yang diinginkan dalam.
Pre-kondisi	Tour Agency memilih memilih wisata.
Post-kondisi	Tour Agency berhasil menambahkan paket wisata yang diinginkan ke dalam daftar paket
Platform	Website

Table Typical Course of Event

Aktor	Sistem
	1. Sistem menampilkan halaman utama
2. Tour Agency memilih tombol tambah Paket	
	3. Sistem menampilkan form tambah paket
4. Tour Agency mengisi informasi paket	
5. Tour Agency memencet tombol submit	
	7. Sistem menyimpan data paket pariwisata kedalam database

3.1.11.1 Sequence Diagram

look

[Explore](#)[Places](#)[Plans](#)[Packages](#)[Privacy policy](#)[About](#)

[Sign in](#)[Get started](#)

Create Package

Package title

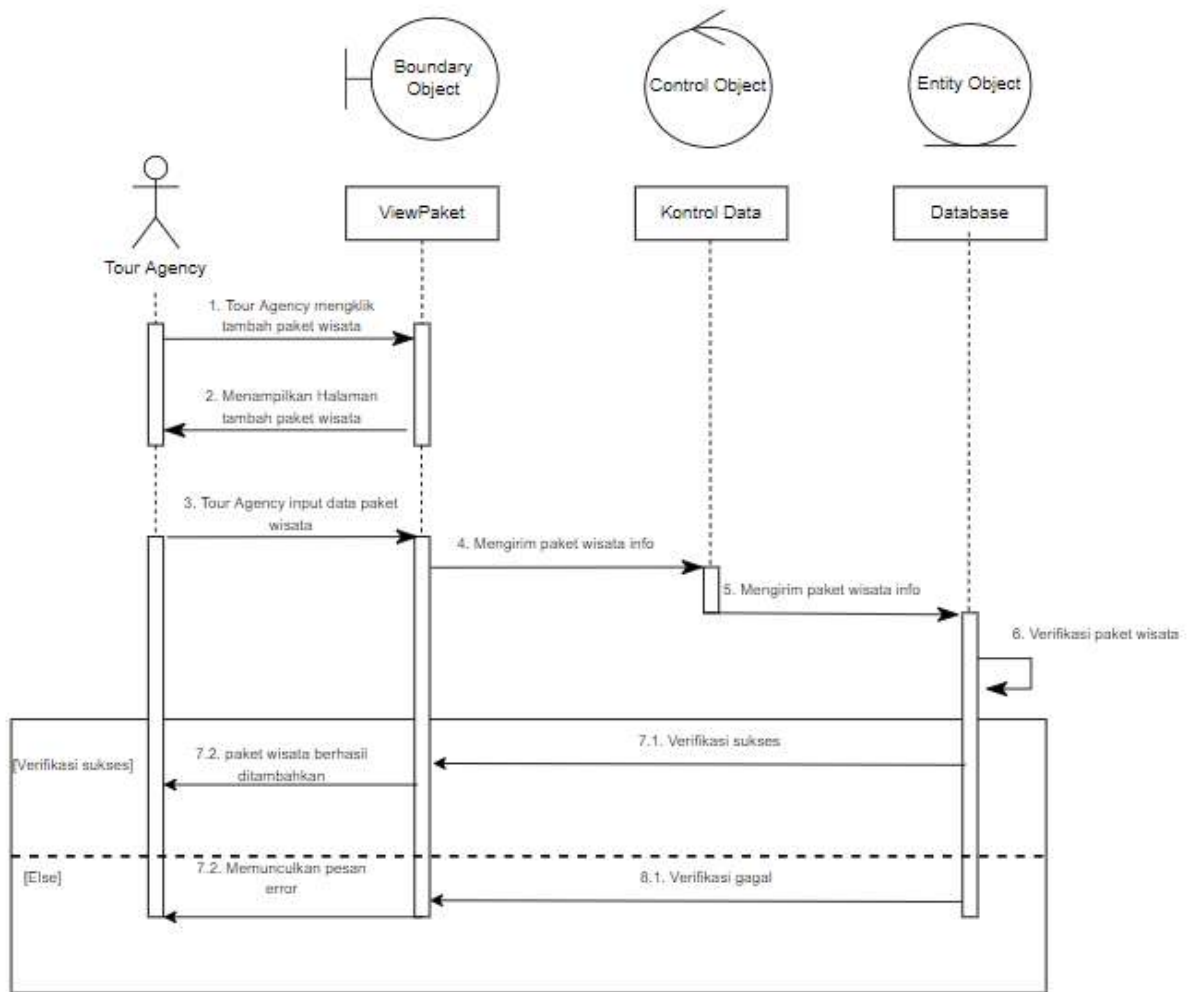
Enter package title

Description

Enter description

Create package

3.1.11.2 Sequence Diagram



3.1.12 Use Case Edit Paket

Skenario Use Case 9: Edit Pariwisata

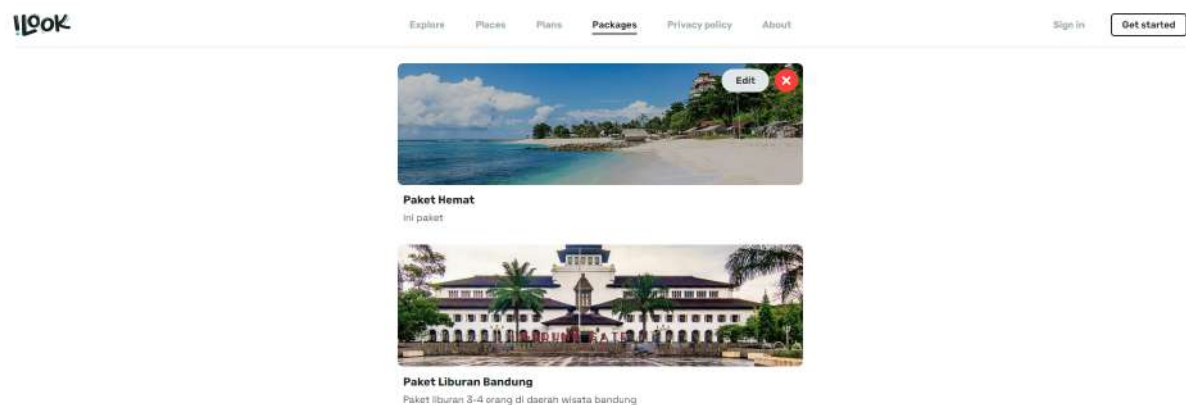
Nama Use Case	Edit Paket
Deskripsi	Tour Agency melakukan edit informasi terhadap paket pariwisata yang sudah ada
Pre-kondisi	Tour Agency telah login akun
Post-kondisi	Tour Agency berhasil mengubah informasi paket pariwisata
Platform	Website

Table Typical Course of Event

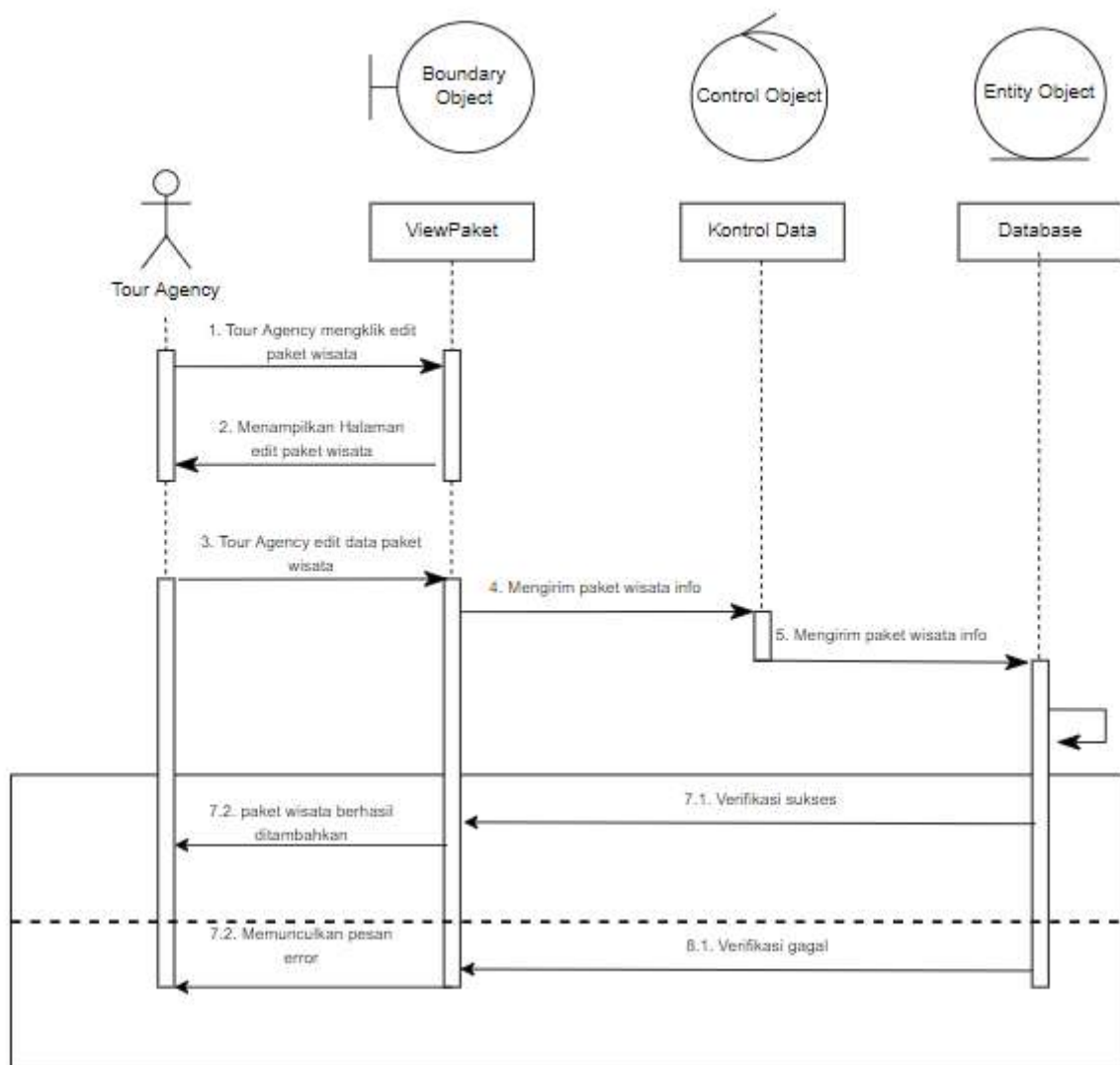
Prodi S1 Teknik Informatika - Universitas Telkom	DPPL-001	Halaman 43 dari 68
Dokumen ini dan informasi yang ada di dalamnya adalah milik Prodi S1 Teknik Informatika-Universitas Telkom dan bersifat rahasia. Dilarang untuk mereproduksi dokumen ini tanpa diketahui oleh Program Studi S1 Teknik Informatika, Universitas Telkom		

User	System
	1. Sistem menampilkan halaman utama
2. Tour Agency memilih paket wisata yang ingin diedit	
	3. Sistem menampilkan info paket wisata
4. Tour Agency menekan tombol edit	
	5. Sistem menampilkan halaman edit paket wisata
6. Tour Agency mengedit data yang ingin diubah	
	7. Sistem menyimpan data yang telah diubah

3.1.12.1 Perancangan Antarmuka Case 12



3.1.11.2 Sequence Diagram



3.1.12 Use Case Hapus Paket

Skenario Use Case 10: Hapus Pariwisata

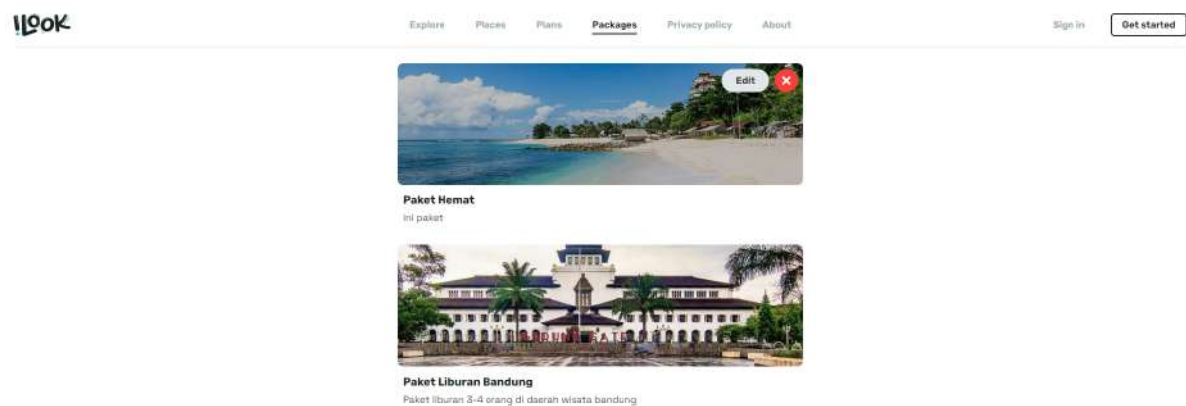
Nama Use Case	Hapus Paket
Deskripsi	Tour Agency menghapus paket pariwisata yang sudah ada
Pre-kondisi	Tour Agency telah login akun
Post-kondisi	Tour Agency berhasil menghapus pariwisata yang ada.
Platform	Website

Table Typical Course of Event

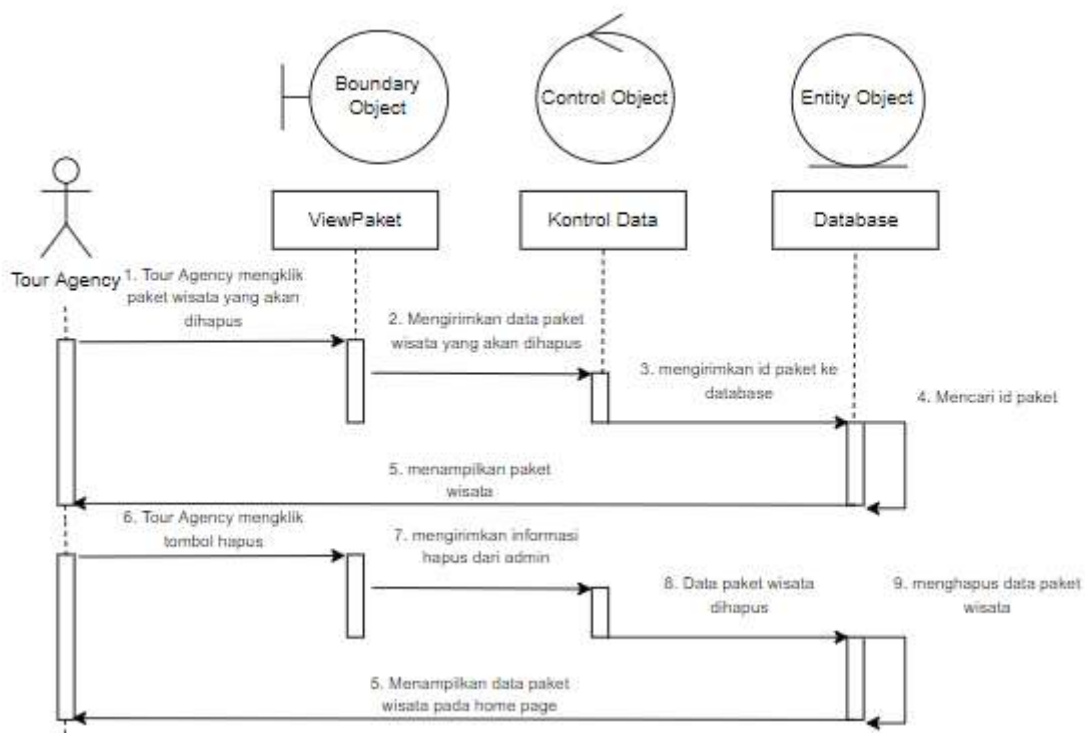
Prodi S1 Teknik Informatika - Universitas Telkom	DPPL-001	Halaman 45 dari 68
Dokumen ini dan informasi yang ada di dalamnya adalah milik Prodi S1 Teknik Informatika-Universitas Telkom dan bersifat rahasia. Dilarang untuk mereproduksi dokumen ini tanpa diketahui oleh Program Studi S1 Teknik Informatika, Universitas Telkom		

User	System
	1. Sistem menampilkan halaman utama
2. Tour Agency memilih paket wisata yang ingin dihapus	
	3. Sistem menampilkan info wisata
4. Tour Agency menekan tombol hapus	
	5. Sistem menampilkan halaman hapus
6. Tour Agency mengedit data yang ingin dihapus	
	7. Sistem menyimpan data yang telah diubah

3.1.12.1 Perancangan Antarmuka Case 12



3.1.12.2 Sequence Diagram



3.1.13 Use Case Logout

Skenario Use Case 11: Logout

Nama Use Case	Edit Logout
Deskripsi	User melakukan logout akun untuk mengeluarkan akun jika tidak dipakai pada aplikasi
Pre-kondisi	User telah login akun
Post-kondisi	User berhasil melakukan logout akun
Platform	Website dan Mobile

Table Typical Course of Event

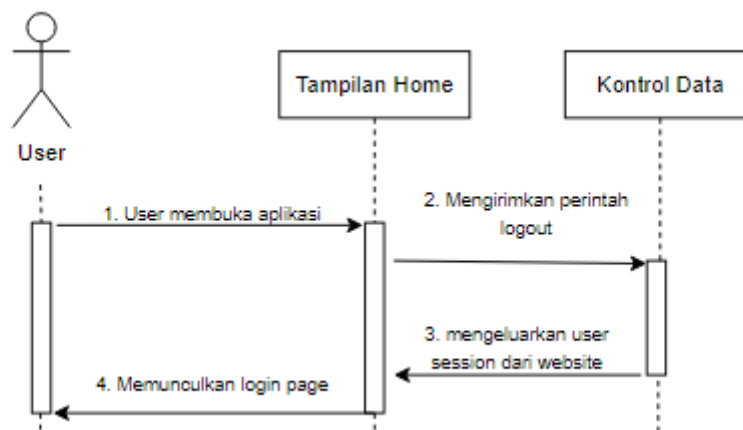
User	System
	1. Sistem menampilkan halaman utama
2. User menekan tombol profil	

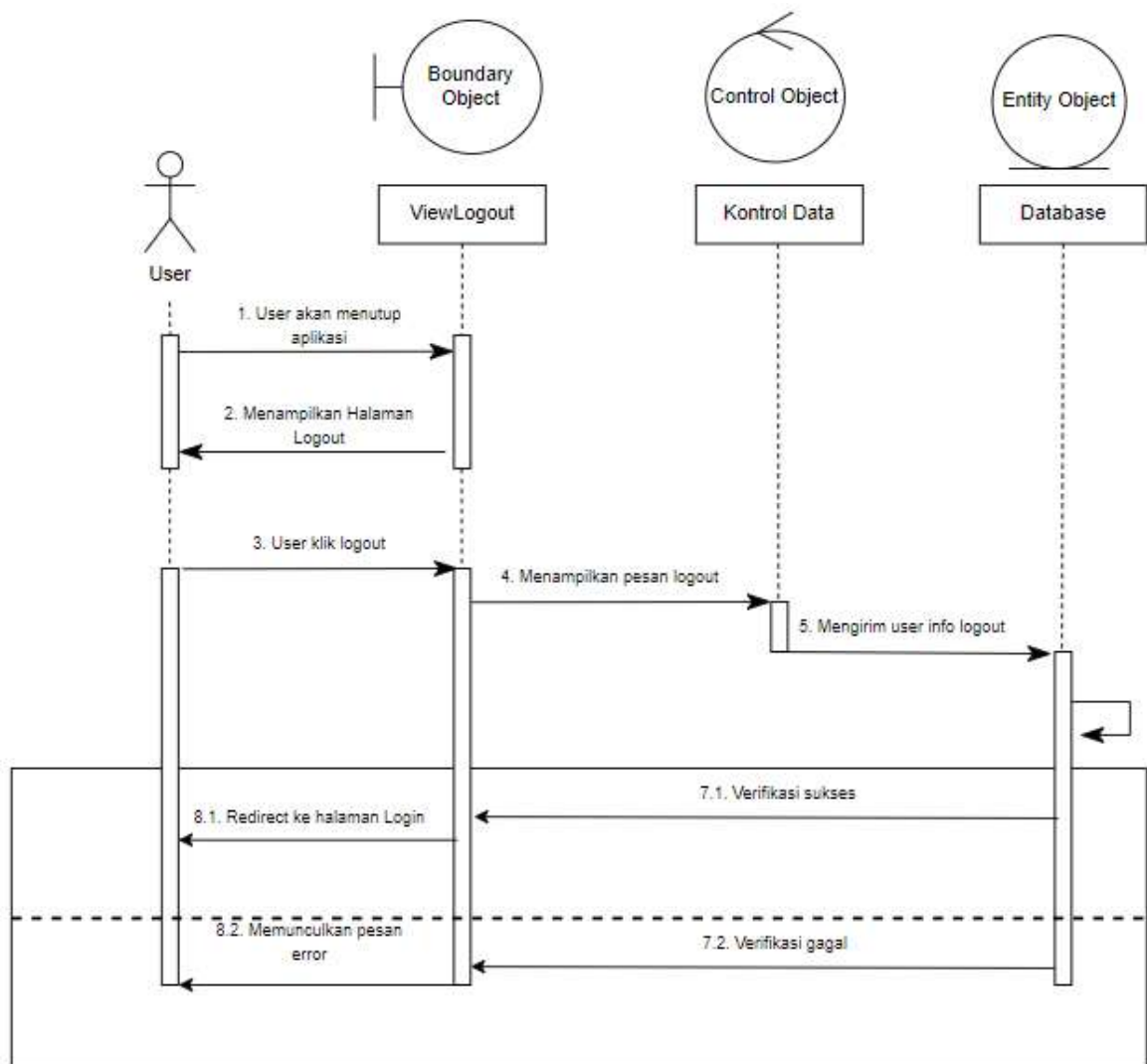
	3. Sistem menampilkan halaman profil
4. User menekan tombol pengaturan	
	5. Sistem menampilkan halaman pengaturan akun
6. User menekan tombol logout	
	7. Sistem mengeluarkan akun
8. User berhasil melakukan logout akun	

3.1.13.1 Perancangan Antarmuka Use Case 11


[Explore](#)
[Places](#)
[Privacy policy](#)
[About](#)
[Sign out](#)

3.1.13.2 Sequence Diagram

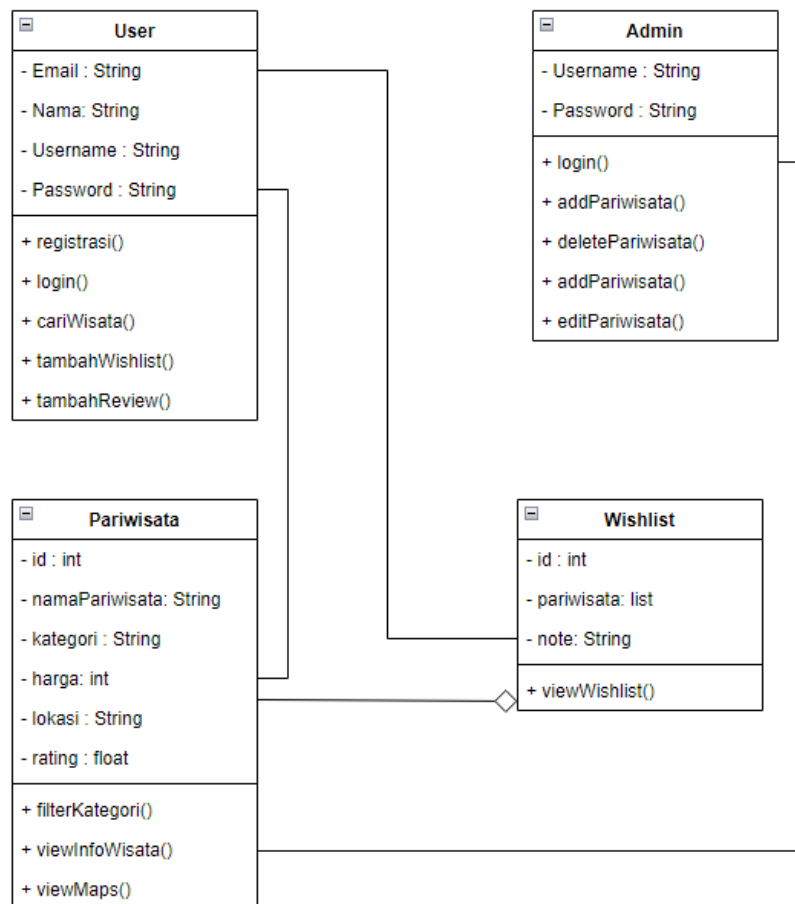




3.2 Perancangan Detail Kelas

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	User	Pariwisata, planner
2	Admin	Pariwisata
3	Pariwisata	User, planner, Admin
4	Planner	User, Pariwisata
5	Paket	User, Pariwisata
6	Rating	Pariwisata

3.3 Diagram Kelas Keseluruhan



3.4 Algoritma/Query

3.4.1 Algoritma #1

Nama Kelas : AuthController
Nama Operasi : Register
Algoritma : register

```
func register(Request request) {
    role = request->role
    if (role == "user") {
        validator = Validator::make(request->details, [
            'username'=> 'required|unique:users,username',
            'password'=>'required|min:8',
            'name'=>'required',
            'phoneNumber'=>'required',
        ])
    } else {
        validator = Validator::make(request->details, [
            'username'=> 'required|unique:users,username',
            'password'=>'required|min:8',
        ])
    }
}
```

```

        'name'=>'required',
        'phoneNumber'=>'required',
        'location'=>'required',
    ])
}
if (validator->fails()){
    return response()->json([
        'validation_errors' => validator->messages(),
    ]);
}
else {
    if ($role === 'user'){
        user = User::create([
            'username'=>request->details['username'],
            'password'=>Hash::make(request->details['password']),
            'name'=>request->details['name'],
            'phoneNumber'=>request->details['phoneNumber'],
            'roleType'=>1
        ]);
    }
    else {
        user = User::create([
            'username'=>request->details['username'],
            'password'=>Hash::make(request->details['password']),
            'name'=>request->details['name'],
            'phoneNumber'=>request->details['phoneNumber'],
            'location'=>request->details['location'],
            'roleType'=>2
        ]);
    }
    // Membuat token untuk registrasi
    token = user->createToken(user->username.'_Token')->plainTextToken;

    return response()->json([
        'status'=>200,
        'username'=>user->name,
        'token'=>token,
        'message'=>'Registered Successfully',
    ]);
}
}
}

```

3.4.2 Algoritma #2

Nama Kelas : AuthController
 Nama Operasi : Login
 Algoritma : login

```

func login(Request request) {
    $validator = Validator::make($request->all(), [
        'username' =>'required',
        'password' =>'required',
    ]);
    if($validator->fails()){

```

```

        return response()->json([
            'validation_errors' => $validator->messages(),
        ]);
    }else {
        $user = User::where('username', $request->username)->first();
        if (! $user || ! Hash::check($request->password, $user->password)){
            return response()->json([
                'status'=>401,
                'message'=>'Invalid credentials',
            ]);
        }else{
            if($user->roleType == 0){
                $token =
                $user->createToken($user->username.'_AdminToken')->plainTextToken;
            }else if ($user->roleType == 2){
                $token =
                $user->createToken($user->username.'_AgencyToken')->plainTextToken;
            }else {
                $token =
                $user->createToken($user->username.'_Token')->plainTextToken;
            }
            return response()->json([
                'status'=>200,
                'username'=>$user->name,
                'token'=>$token,
                'message'=>'Signed in Successfully'
            ]);
        }
    }
}
}

```

3.4.3 Algoritma #3

Nama Kelas : ContentController
 Nama Operasi : Explore
 Algoritma : showPariwisata

```

function showPariwisata(){
    $pariwisata = Pariwisata::all();
    return response()->json([
        'status'=>200,
        'pariwisata' => $pariwisata,
    ]);
}

```

3.4.4 Algoritma #4

Nama Kelas : ContentController
 Nama Operasi : Search
 Algoritma : searchPariwisata

```

func searchPariwisata(String lokasi){
    $pariwisata = Pariwisata::whereRaw("lokasi REGEXP '" . $lokasi . "'")
        ->orWhereRaw("nama REGEXP '" . $lokasi . "'")->get();
    return response()->json([
        'status'=>200,
        'pariwisata' => $pariwisata,
    ]);
}

```

3.4.5 Algoritma #5

Nama Kelas : PlannerController
 Nama Operasi : Tambah planner
 Algoritma : createPlanner

```

public function createPlanner(Request $request){
    $validator = Validator::make($request->all(), [
        'nama'=> 'required',
        'start_at' => 'required|date',
        'end_at' => 'required|date|after_or_equal:start_date',
    ]);

    if($validator->fails()){
        return response()->json([
            'status'=>400,
            'validation_errors' => $validator->messages(),
        ]);
    } else {
        if(auth('sanctum')->check()){
            $planner = new Planner;
            // $planner->userID = auth('sanctum')->user()->id;
            $planner->nama = $request->nama;
            $planner->start_at = $request->start_at;
            $planner->end_at = $request->end_at;
            $planner->save();
            return response()->json([
                'status' => 200,
                'message'=> 'Planner created successfully'
            ]);
        } else {
            return response()->json([
                'status'=>401,
                'message'=> 'Please login first'
            ]);
        }
    }
}

```

3.4.6 Algoritma #6

Nama Kelas : PlannerController
 Nama Operasi : Hapus planner

Algoritma : delete

```
public function delete($id){
    if(auth('sanctum')->check()){
        $planner = Planner::find($id);
        if($planner){
            $planner->delete();
            // if (PlannerContainer::where('planID', $id)->exists()){
            //     PlannerContainer::where('planID', $id)->delete();
            // }
            return response()->json([
                'status'=>200,
                'message'=>'Data pariwisata has been deleted',
            ]);
        } else {
            return response()->json([
                'status'=>404,
                'message'=>'No Plan ID Found',
            ]);
        }
    } else {
        return response()->json([
            'status'=>401,
            'message'=>'Please login first'
        ]);
    }
}
```

3.4.7 Algoritma #7

Nama Kelas : RatingController
Nama Operasi : Memberi Rating
Algoritma : sendRating

```
public function sendRating(Request $request, $id){
    $validator = Validator::make($request->all(), [
        'rating' =>'required',
    ]);
    if ($validator->fails()){
        return response()->json([
            'validation_errors' => $validator->messages(),
        ], 401);
    } else {
        $rating = Rating::create([
            'rating' => $request->rating,
            'wisataID' => $id,
        ]);
        return response()->json($rating, 200);
    }
}
```

3.4.8 Algoritma #8

Nama Kelas : AdminController
Nama Operasi : Tambah Pariwisata
Algoritma : createPariwisata

```
public function createPariwisata(Request $request){
    $validator = Validator::make($request->all(), [
        'nama'=> 'required',
        'kategoriID'=>'required',
        'harga'=>'required',
        'deskripsi'=>'required',
        'lokasi'=>'required',
        'urlGambar' => 'required|image|mimes:jpg,png,jpeg'
    ]);

    if($validator->fails()){
        return response()->json([
            'status'=>400,
            'validation_errors' => $validator->messages(),
        ]);
    }else{
        $pariwisata = new Pariwisata;
        $pariwisata->nama = $request->nama;
        $pariwisata->kategoriID = $request->kategoriID;
        $pariwisata->harga = $request->harga;
        $pariwisata->deskripsi = $request->deskripsi;
        $pariwisata->lokasi = $request->lokasi;

        $gambar = $request->file('urlGambar');
        $gambar_uploaded_path = $gambar->store('gambar', 'public');

        $pariwisata->urlGambar = Storage::url($gambar_uploaded_path);
        $pariwisata->rating = NULL;
        $pariwisata->save();
        return response()->json([
            'status'=>200,
            'message'=>'Data has been added successfully',
        ]);
    }
}
```

3.4.9 Algoritma #9

Nama Kelas : AdminController
Nama Operasi : Edit Pariwisata
Algoritma : edit & update

```
public function edit($id){
    $pariwisata = Pariwisata::find($id);
    if($pariwisata){
```

```

        return response()->json([
            'status' => 200,
            'pariwisata'=>$pariwisata,
        ]);
    } else {
        return response()->json([
            'status' => 404,
            'message'=>'No Pariwisata ID found',
        ]);
    }
}

public function update(Request $request, $id){
    $validator = Validator::make($request->all(), [
        'nama'=> 'required',
        'kategoriID'=>'required',
        'harga'=>'required',
        'deskripsi'=>'required',
        'lokasi'=>'required',
        'urlGambar' => 'image|mimes:jpg,png,jpeg'
    ]);

    if($validator->fails()){
        return response()->json([
            'status'=>422,
            'validation_errors' => $validator->messages(),
        ]);
    }else{
        $pariwisata = Pariwisata::find($id);

        if($pariwisata){
            $pariwisata->nama = $request->nama;
            $pariwisata->kategoriID = $request->kategoriID;
            $pariwisata->harga = $request->harga;
            $pariwisata->deskripsi = $request->deskripsi;
            $pariwisata->lokasi = $request->lokasi;

            if($request->hasFile('urlGambar')){
                $gambarURL = $pariwisata->urlGambar;
                $path = substr($gambarURL, strpos($gambarURL, 'gambar/') +
7);

                if(Storage::disk('gambar')->exists($path)){
                    Storage::disk('gambar')->delete($path);
                }
                $gambar = $request->file('urlGambar');
                $gambar_uploaded_path = $gambar->store('gambar', 'public');
                $pariwisata->urlGambar =
Storage::url($gambar_uploaded_path);
            }
            $pariwisata->save();
            return response()->json([
                'status'=>200,
                'message'=>'Pariwisata data has been updated successfully',
            ]);
        }
    }
}

```



```

        else{
            return response()->json([
                'status'=>404,
                'message'=>'No Pariwisata ID Found',
            ]);
        }
    }
}

```

3.4.10 Algoritma #10

Nama Kelas : AdminController
 Nama Operasi : Hapus Pariwisata
 Algoritma : delete

```

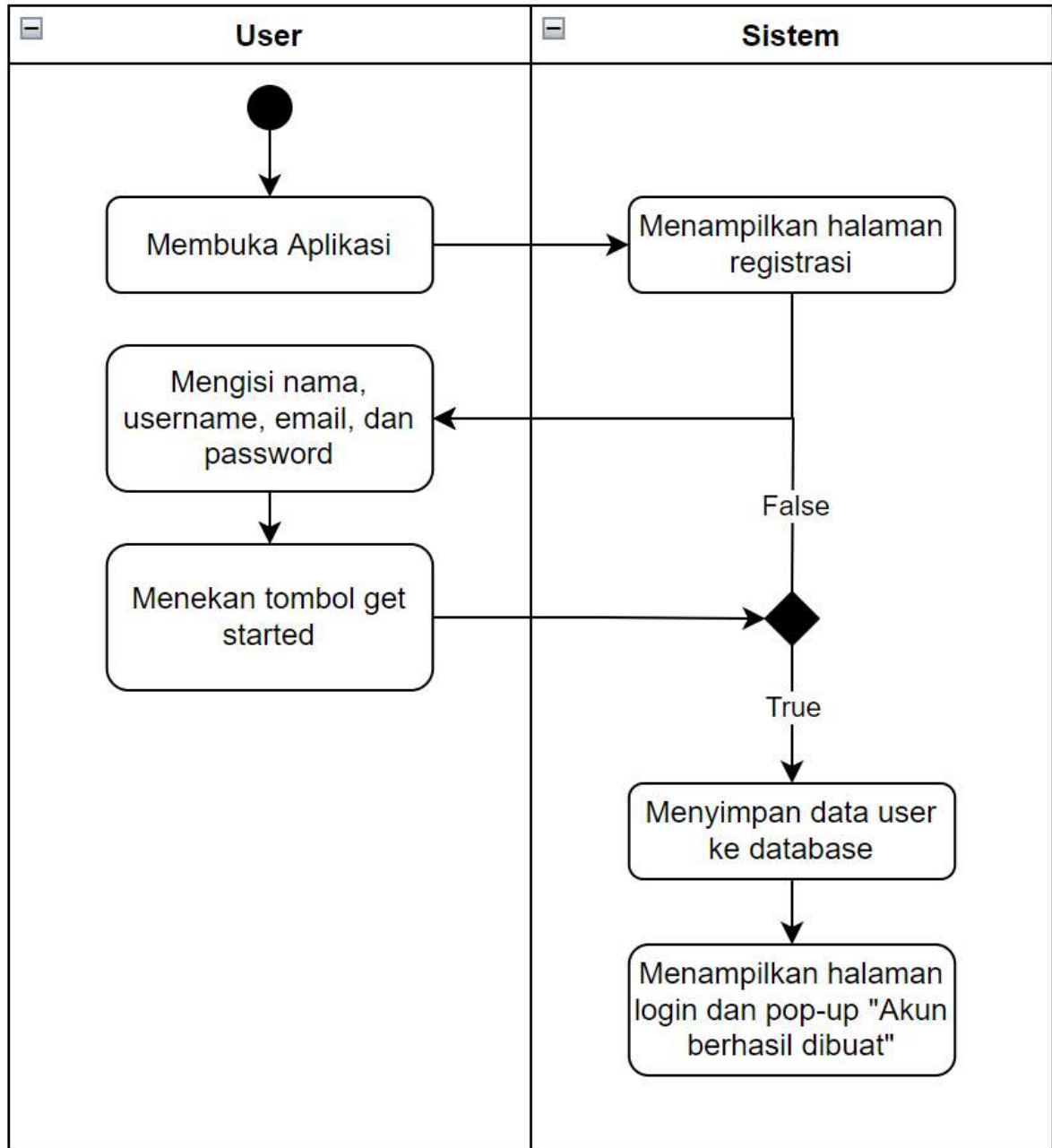
func delete(String id){
    $pariwisata = Pariwisata::find($id);
    if($pariwisata){
        $pariwisata->delete();
        return response()->json([
            'status'=>200,
            'message'=>'Data pariwisata has been deleted',
        ]);
    }else {
        return response()->json([
            'status'=>404,
            'message'=>'No Pariwisata ID Found',
        ]);
    }
}

```

3.5 Diagram Statechart

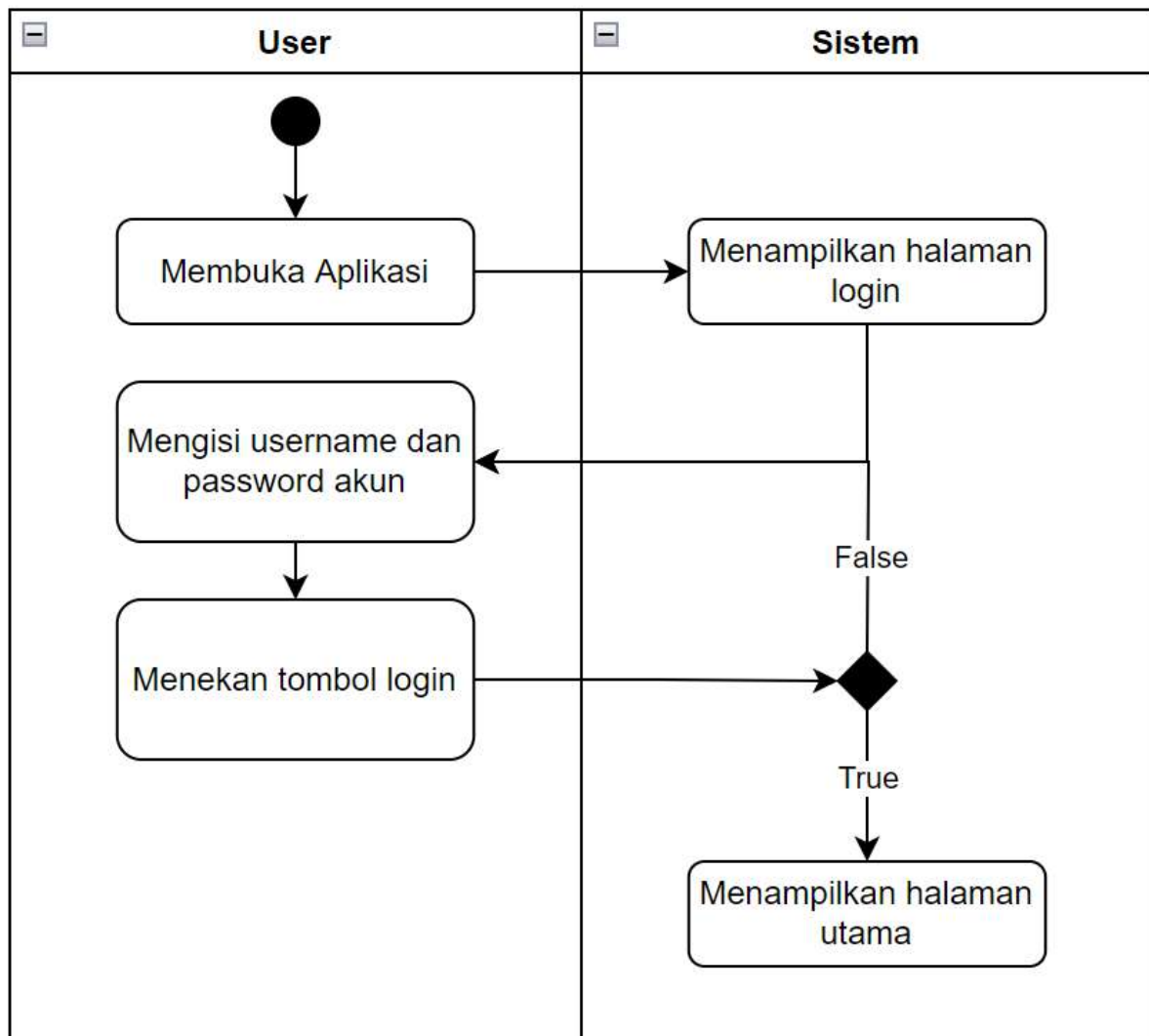
3.5.1 Register

Registrasi



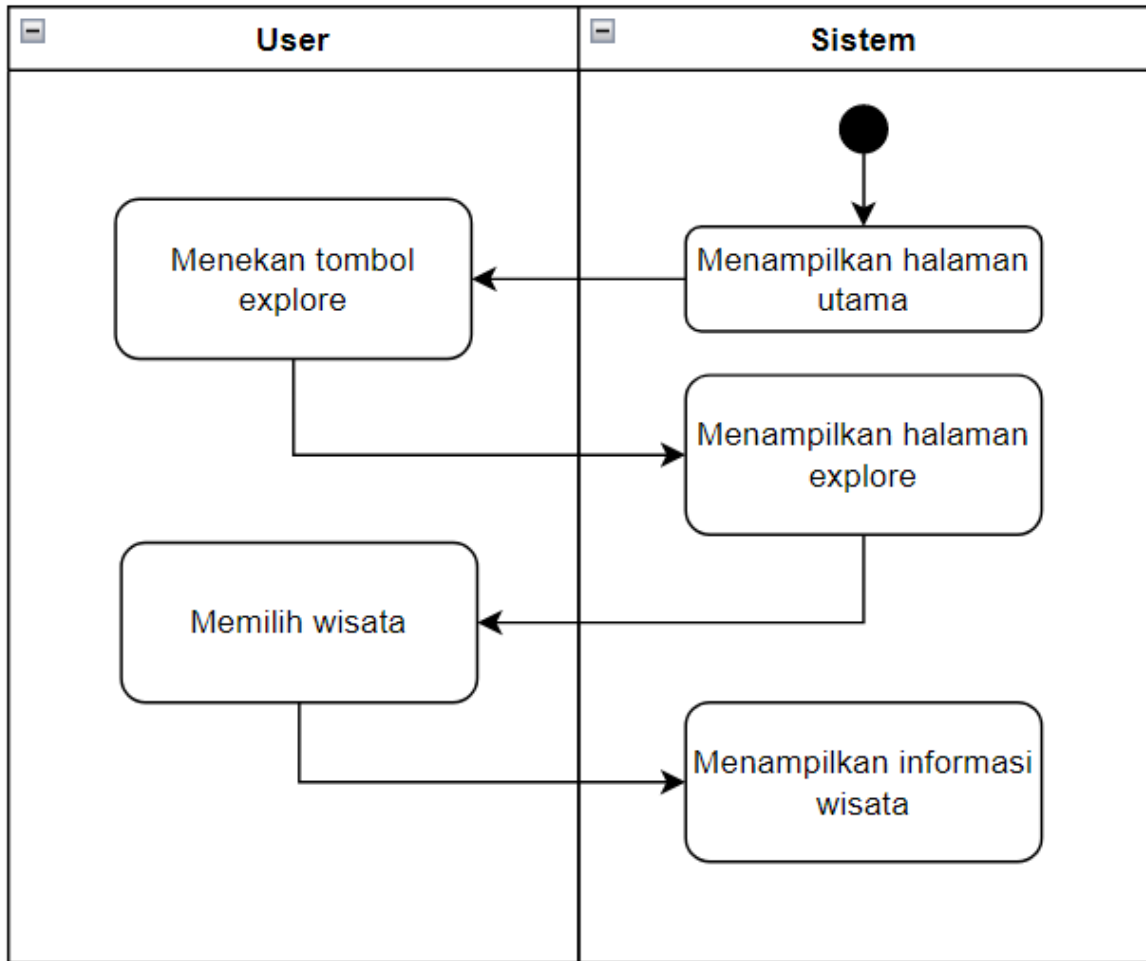
3.5.2 Login

Login



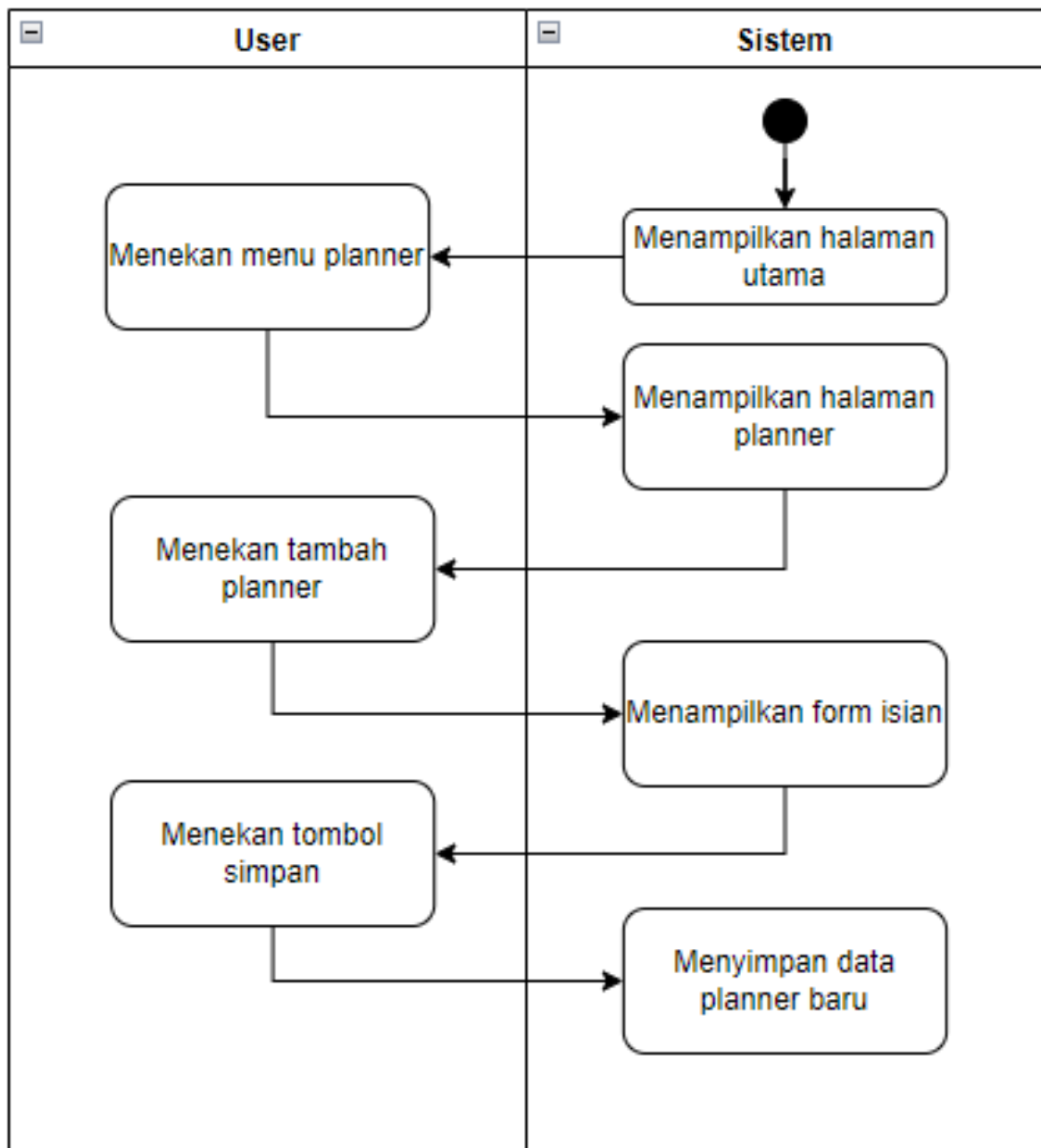
3.5.3 Show Wisata

Show Wisata

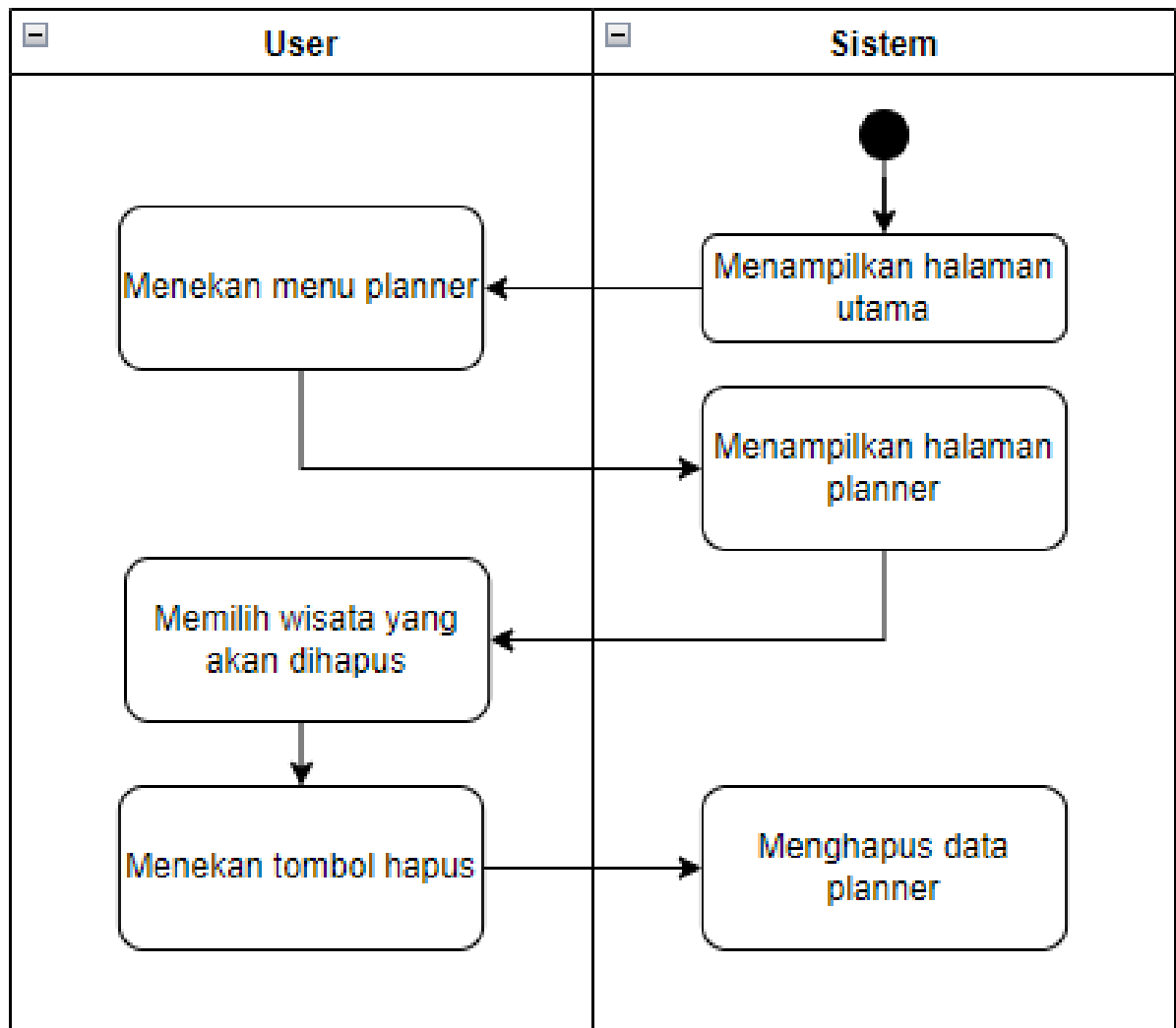


3.5.4 Tambah planner

Tambah Planner

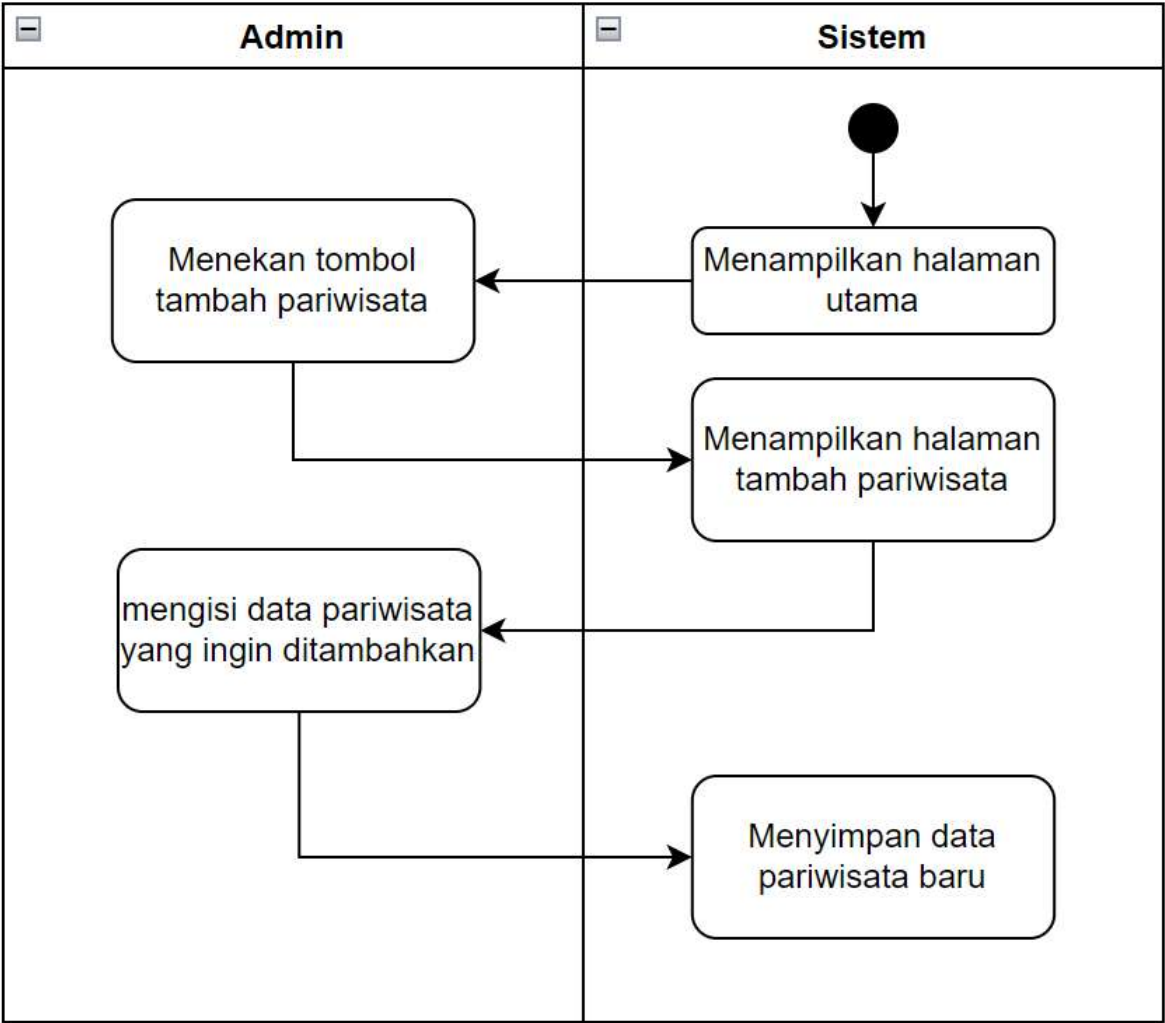


3.5.5 Hapus planner



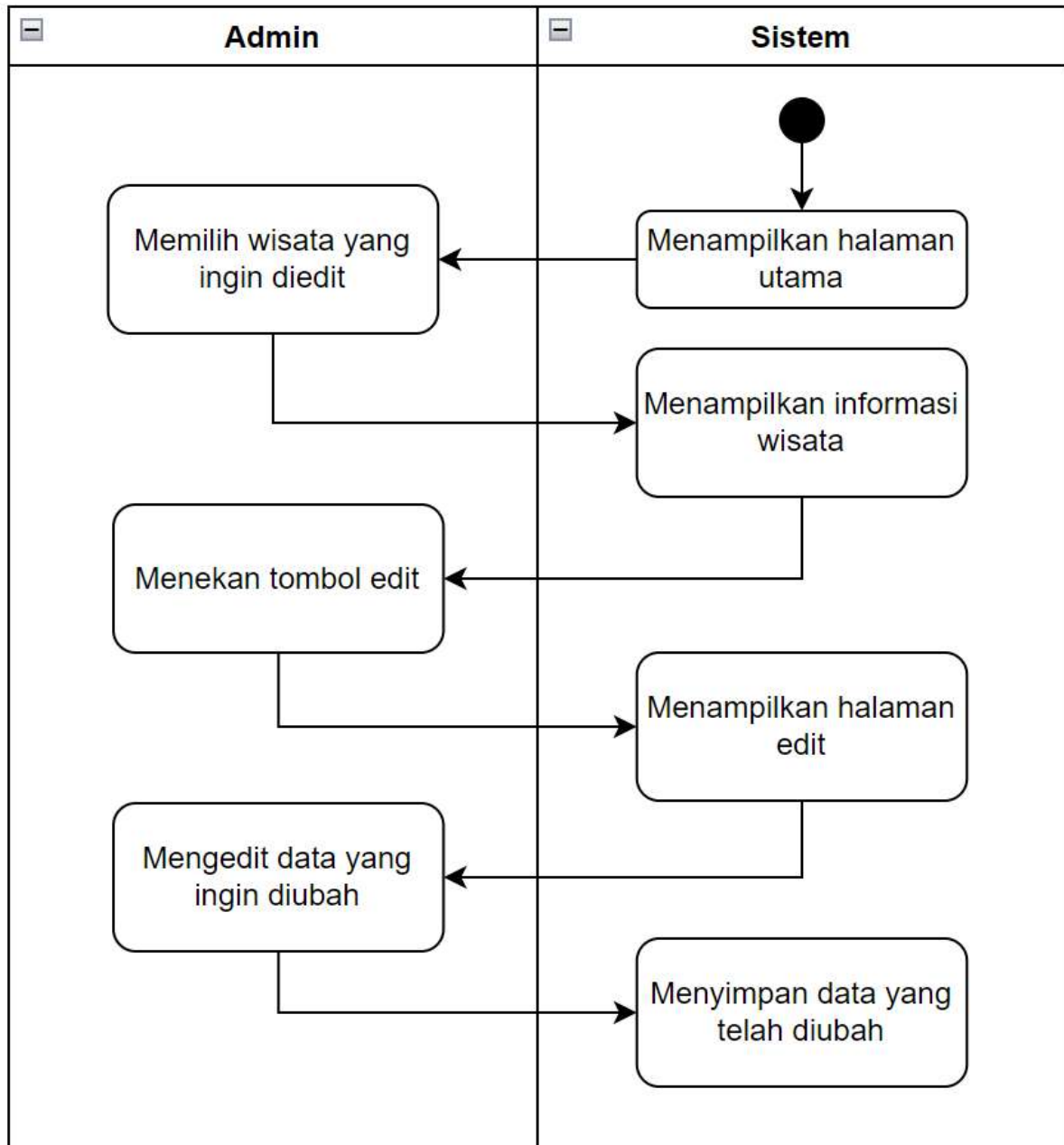
3.5.6 Tambah Pariwisata

Tambah Pariwisata



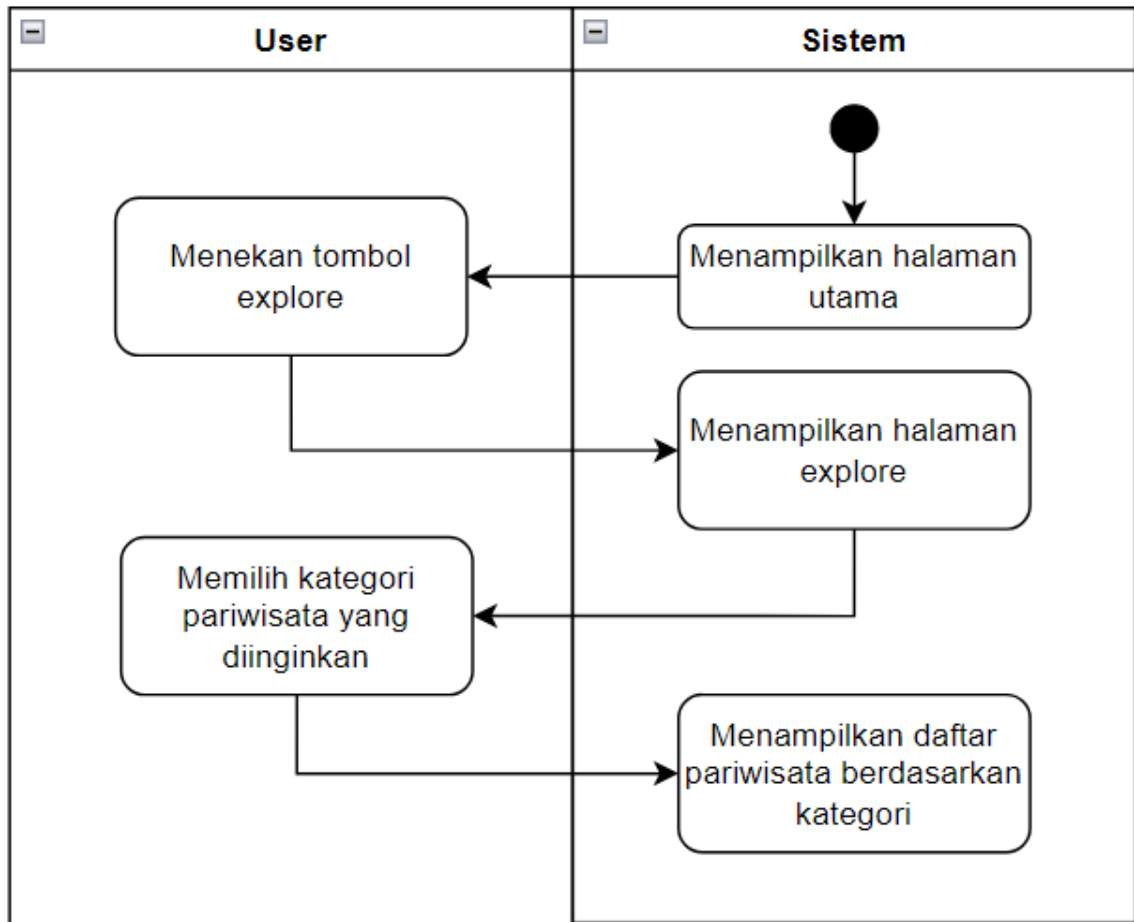
3.5.7 Edit Pariwisata

Edit Pariwisata



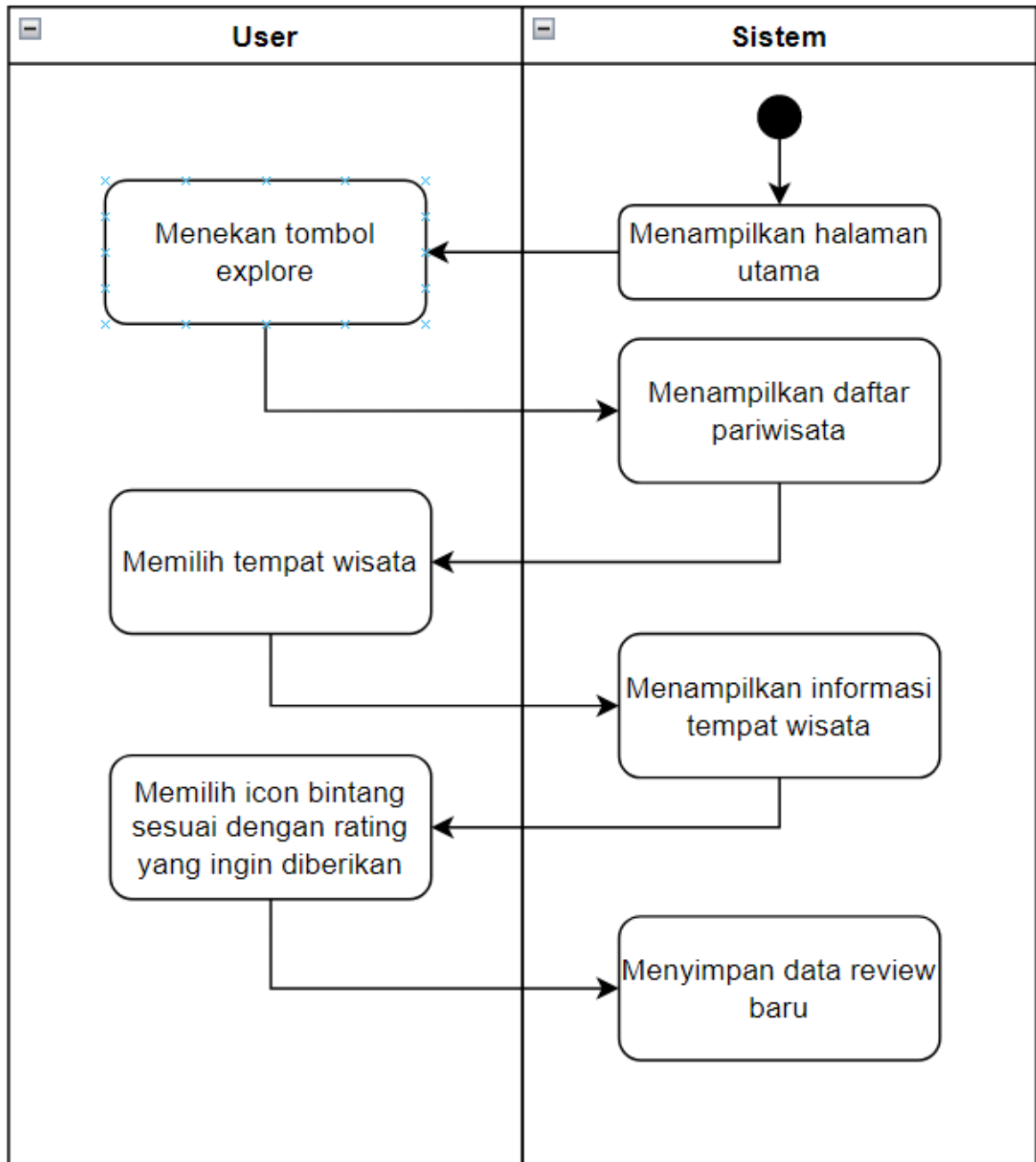
3.5.8 Search

Search



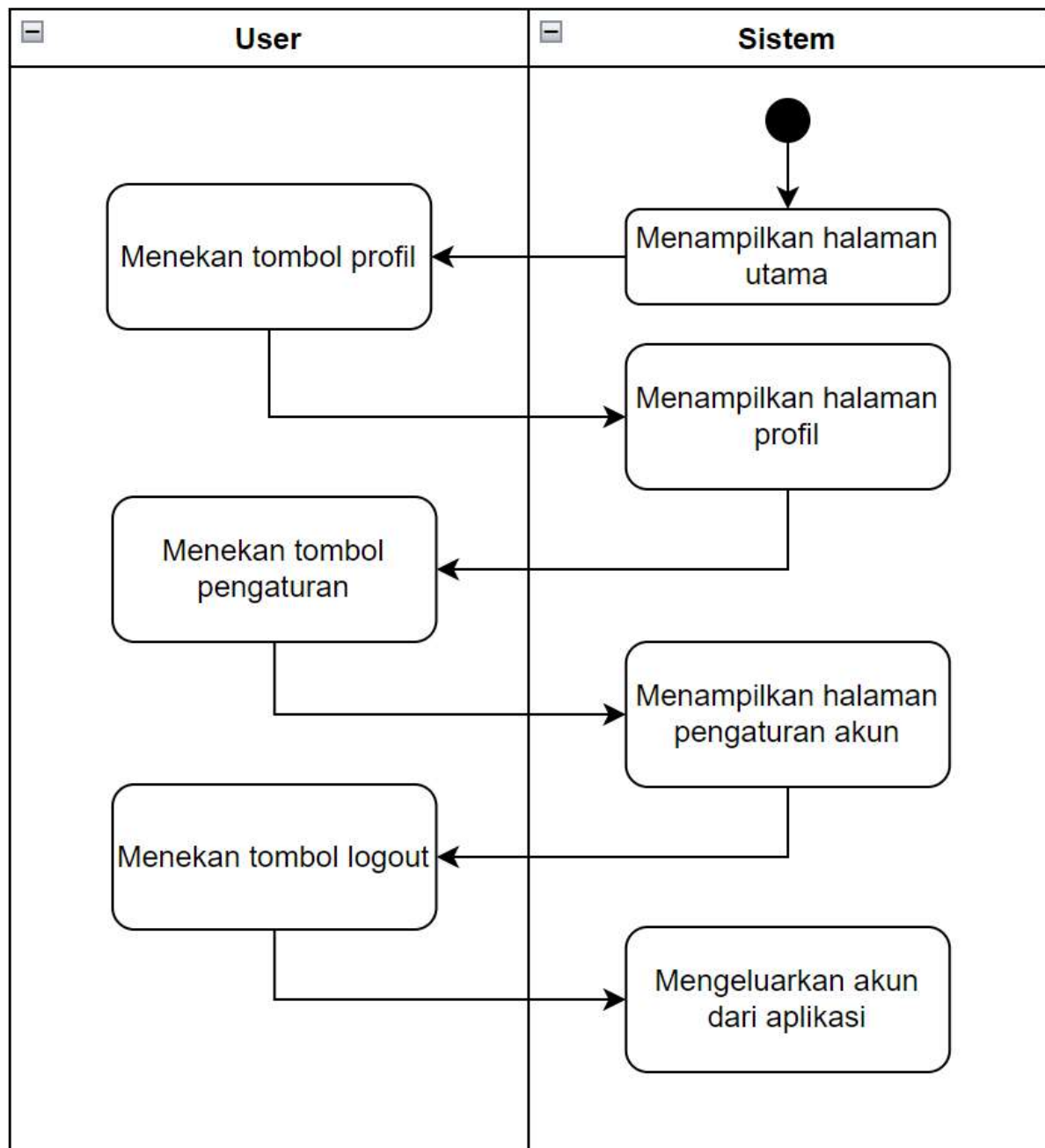
3.5.9 Memberi Rating

Memberi Rating



3.5.10 Logout

Logout



4 Matriks Kerunutan

Requirement	Use Case Terkait	Kelas
FR-01	Registrasi	User

FR-02	Login	User, Admin
FR-03	Explore	User, Pariwisata
FR-04	Filter	User, Pariwisata
FR-05	Tambah planner	User, Pariwisata, planner
FR-06	Hapus planner	User, Pariwisata, planner
FR-07	Memberi Rating	User, Pariwisata
FR-08	Tambah Pariwisata	Admin, Pariwisata
FR-09	Edit Pariwisata	Admin Pariwisata
FR-10	Hapus Pariwisata	Admin, Pariwisata
FR-11	Tambah Paket Pariwisata	TourAgency
FR-12	Hapus Paket Pariwisata	TourAgency
FR-13	Edit Paket Pariwisata	TourAgency
FR-14	Logout	User, Admin