BANCEUY SMART HERITAGE : SISTEM INFORMASI DIGITAL KAMPUNG ADAT BANCEUY

***SOFTWARE DESIGN DOCUMENTATION***

***Version:* <1.0>**

**Riski Fitriani – 10110054**

**Rendi Perdiansah - 10110052**

**PROGRAM STUDI SISTEM INFORMASI**

**JURUSAN TEKNOLOGI INFORMASI DAN KOMPUTER**

**POLITEKNIK NEGERI SUBANG**

**2025**

**DAFTAR ISI**

[**1** **PENDAHULUAN** 3](#_Toc197256867)

[**1.1** **Latar Belakang** 3](#_Toc197256868)

[**1.2** **Tujuan** 5](#_Toc197256869)

[**1.3 Ruang Lingkup** 6](#_Toc197256870)

[**1.4 Definisi, Istilah dan Singkatan** 7](#_Toc197256871)

[**1.5 Referensi** 8](#_Toc197256872)

[**2** **DESKRIPSI UMUM** 9](#_Toc197256873)

[**3** **ARSITEKTUR SISTEM** 9](#_Toc197256874)

[**3.1 Kebutuhan Fungsional** 9](#_Toc197256875)

[**1. Pendahuluan** 9](#_Toc197256876)

[**2. Input** 10](#_Toc197256877)

[**3.** **Proses** 10](#_Toc197256878)

[**4.** **Output** 11](#_Toc197256879)

[**3.2 Kebutuhan Non Fungsional** 11](#_Toc197256880)

[**4** **PEMODELAN PERANGKAT LUNAK** 13](#_Toc197256881)

[**4.1** **Use Case Diagram** 13](#_Toc197256882)

[**4.1.1** **Deskripsi Aktor** 14](#_Toc197256883)

[**4.1.2** **Deskripsi Use Case Diagram** 15](#_Toc197256884)

[**4.1.3 Scenario Use Case Diagram** 18](#_Toc197256885)

[**4.2 Activity Diagram** 55](#_Toc197256909)

# **PENDAHULUAN**

## **Latar Belakang**

Sistem informasi yang terpusat sangat dibutuhkan untuk penunjang kebutuhan calon pengujung dalam pariwisata. oleh karena itu sistem informasi yang terpusat dengan baik dapat membantu calon penjung dan masyarakat bisa mengakses informasi wisata dengan mudah. [1] Banceuy *Smart Heritage* dianggap sebagai solusi atas permasalahan dalam menunjang kebutuhan calon wisatawan dan masyarakat. [2] Banceuy *Smart Heritage* juga menjadi salah satu cara untuk meningkatkan pengunjung Kampung Adat Banceuy serta mempermudah pengelolaan kunjungan. Pada penelitian yang dilakukan sebelumya menyoroti bahwa pembuatan sistem informasi berbasis *website* untuk wisatawan lebih cepat dan mudah sebagai panduan wisatawan, selain itu untuk memudahkan pencarian informasi wisata. [3]

Pada tahun 2017 Kampung Adat Banceuy diresmikan oleh Bupati Subang menjadi desa wisata, namun sebelum diresmikan pada tahun 2017 Kampung Adat Banceuy sudah memiliki organisasi pengelola pariwisata yaitu Kompepar dan berganti nama menjadi Pokdarwis pada tahun 2020. Pokdarwis ini bertugas mengelola kegiatan wisata, rekap pengunjung, strategi pengenalan Banceuy secara luas dan pelestarian budaya di Banceuy. Namun masih terdapat beberapa masalah yang yang ada, dimulai dari tidak adanya sistem informasi yang terpusat untuk memudahkan calon pengunjung ke Banceuy karena calon pengunjung seringkali bingung harus menghubungi siapa ketika ingin berkunjung ke Banceuy. Permasalahan kedua mengenai perluasan informasi Banceuy sebagai wisata kampung adat edukasi yang ada di Kabupaten Subang. Permasalahan ketiga tidak adanya gambaran kegiatan Budaya apa saja yang ada di Banceuy sehingga membuat calon pengunjung tidak memiliki plan pilihan kegiatan untuk berkunjung ke sana. Permasalahan keempat, pada saat ini krisis terkait kurang dikenalinya budaya lokal, dengan mengenal kebudayaan, maka kebudayaan dapat dilestarikan agar tidak punah, kebudayaan Indonesia begitu beragam dan kaya namun, keberadaannya belum mampu dinikmati secara optimal oleh masyarakat.

Pemanfaatan teknologi informasi menjadi solusi dalam menangani permasalahan yang sedang terjadi, melalui *website* Banceuy Smart Heritage pengunjung akan mudah mendapatkan informasi terkait *wisata* edukasi di Banceuy. Kemudian kebudayaan yang ada di indonesia dapat dilestarikan lebih mudah dan cepat, bahkan dapat membantu organisai terkait untuk memperkenalkan Kebudayaan yang ada di Indonesia. *Website* *Smart Heritage* juga menjadi penunjang untuk mempermudah pengelolaan wisata Kampung Adat Banceuy dan sistem informasi ini membuat proses bisnis menjadi otomatis. Untuk mewujudkan kebutuhan wisatawan yang sekarang yang menginginkan informasi dengan mudah dan praktis di dalam genggaman mengingat sekarang sudah banyak masyarakat yang menggunakan smartphone dan lebih memilih *searching* untuk menemukan informasi yang dicarinya. [4] Seiring dengan perkembangan zaman, tantangan terhadap pelestarian kearifan lokal semakin meningkat. Arus globalisasi membawa masuk berbagai budaya asing yang dapat mengancam keberadaan nila nilai tradisional. Oleh karena itu, penting untuk menggali dan memahami kearifan lokal dalam konteks historis dan kontemporer agar dapat menemukan cara untuk melestarikannya. [5]

## **Tujuan**

Tujuan dari penelitian ini, yang berkaitan dengan dokumen *requirement system*, adalah untuk memastikan bahwa sistem Banceuy *Smart Heritage* dapat dikembangkan sesuai dengan kebutuhan pengguna. Adapaun tujuan yang ingin dicapai dalam penelitian ini adlah sebagai berikut:

1. Mendefinisikan secara rinci kebutuhan sistem, baik dari segi fungsionalitas maupun non-fungsionalitas dari sistem Banceuy *Smart Heritage*.
2. Menggambarkan secara menyeluruh rancangan sistem yang akan digunakan untuk proses pemesanan berbagai layanan yang tersedia di Banceuy *Smart Heritage* termasuk pemesanan paket, *Homestay* dan custom kegiatan edukasi di Banceuy. Selain itu sisitem ini dirancang dengan adanya notifikasi otomatis yang dikirimkan kepada *Admin* dan pengelola *Homestay*, sehingga dapat memastikan kelancaran operasional dalam pengelolaan pemesanan aktivitas pengunjung
3. Menyediakan panduan yang jelas bagi tim pengembang dalam setiap tahapan proses perancangan dan implementasi sistem. Panduan ini bertujuan untuk memastikan bahwa seluruh aspek teknis dan fungsionalitas dari sistem dikembangkan sesuai dengan kebutuhan yang telah ditetapkan. Selain itu, dokumen ini berperan sebagai acuan utama bagi pemangku kepentingan, seperti pengelola wisata, pengelola *homestay*, dan pihak terkait lainya agar mereka memiliki pemahaman yang sama mengenai fitur yang akan dibangun dalam sistem.
4. Menyajikan perancangan sistem secara terperinci guna memastikan bahwa setiap elemen yang dikembangkan benar – benar selaras dengan kebutuhan pengguna dan tujuan utama dari sistem yaitu promosi wisata serta pelestarian budaya lokal di Banceuy.
5. Melalui perancangan yang sistematis dan terdokumentasi dengan baik, penelitian ini bertujuan untuk mengurangi kemungkinan ambiguitas yang terjadi dalam komunikasi antara berbagai pihak yang terlibat, seperti pengembang sisitem, pengguna akhir juga pemangku kepentingan lainya.

## **Ruang Lingkup**

Ruang lingkup dari sistem mencakup beberapa aspek utama yang bertujuan untuk meningkatkan pengelolaan dan promosi wisata Banceuy. Adapun ruang lingkup sistem ini adalah sebagai berikut:

1. Mendukung efisiensi divisi marketing Pokdarwis banceuy, dalam meningkatkan efisiensi dan efektifitas pengelolaan wisata. Dengan adanya sistem ini proses pencatatan kunjungan, pengelolaan data wisata serta kordinasi antara pengelola dapat dilakukan dengan lebih cepat dan terorganisir.
2. Penyedia sistem interaktif untuk informasi dan pemesanan wisata, menangani pemesanan paket wisata serta menyediakan informasi lengkap mengenal sejarah, tradisi adat, atraks rakyat, kuliner khas, permainan tradisional, alat musik daerah, wisata alam, serta berbagai *event* di Banceuy. Dengan fitur interaktif ini wisatawan dapat dengan mudah mengakses informasi dan merencanakan kunjungan mereka.
3. Fokus pada implementasi dan pengembangan selanjutnya, menitikberatkan pada perancangan sisitem yang siap dikembangkan lebih lanjut.

## **Defenisi, Istilah, dan Singkatan**

1. SRS : *Software Requirements Specification* (Spesifikasi Kebutuhan Perangkat Lunak).
2. Pokdarwis : Pokdarwis adalah singkatan dari Kelompok Sadar Wisata. Pokdarwis adalah organisasi masyarakat yang dibentuk untuk melestarikan tempat wisata di Indonesia. (Wikipedia)
3. Kampung Adat : Adalah wilayah yang dikelola oleh masyarakat adat yang memiliki hubungan erat dengan lingkungan hidup. Kampung adat juga merupakan komunitas tradisional yang berfokus pada adat dan tradisi. (Wikipedia)
4. *Heritage*  : *Heritage* atau warisan adalah sesuatu yang diwariskan dari generasi sebelumnya, dipelihara saat ini, dan diwariskan kepada generasi selanjutnya. (Wikipedia)
5. *Website* : Adalah sekumpulan halaman *web* yang saling berhubungan yang umumnya berada pada peladen yang sama berisikan kumpulan informasi yang disediakan secara perorangan, kelompok, atau organisasi. Sebuah situs *web* biasanya ditempatkan setidaknya pada sebuah *server* *web* yang dapat diakses melalui jaringan seperti Internet, ataupun jaringan area lokal (LAN) melalui

## **Referensi**

1. Usop, T. B. (n.d.). PELESTARIAN ARSITEKTUR TRADISIONAL DAYAK PADA PENGENALAN RAGAM BENTUK KONSTRUKSI DAN TEKNOLOGI TRADISIONAL DAYAK DI KALIMANTAN TENGAH.
2. Rajagukguk, I. S., & Malak, A. (2022). PERANCANGAN SISTEM INFORMASI PENGENALAN KEBUDAYAAN SUKU MOI BERBASIS WEB PADA LEMBAGA ADAT MALA MOI KOTA SORONG PROVINSI PAPUA BARAT DESIGN OF INFORMATION SYSTEM FOR THE INTRODUCTION OF WEB-BASED MOI CULTURE AT THE MALA MOI INDIGENOUS INSTITUTION OF SORONG CITY WEST PAPUA PROVINCE. In Jurnal Elektro Luceat (Vol. 8, Issue 1).
3. Administrator,+SISTEM+INFORMASI+PROMOSI+PARIWISATA+BERBASIS+WEBSITE+DI+KABUPATEN+MALAKA. (n.d.).
4. Nur Annisa, E., Hafifah Matondang, N., Afrizal, S., Studi Sistem Informasi, P., Ilmu Komputer, F., Veteran Jakarta Jl Rs Fatmawati, U., Labu, P., Selatan, J., & Jakarta, D. (n.d.). SISTEM INFORMASI PARIWISATA BERBASIS WEB PADA KABUPATEN NUNUKAN.
5. Lawalata, I. F., Wibowo, A., & Setiawan, A. (n.d.). PERANCANGAN DAN PEMBUATAN WEBSITE PADA KOMUNITAS DISCERNING UNIVERSITAS KRISTEN PETRA.

## **Gambaran Umum Dokumen**

### **Pendahuluan**

Pada bagian pendahuluan menjelaskan alasan dikembangkanya Banceuy *Smart Heritage* sebagai solusi digital untuk mendukung pelestarian budaya dan promosi wisata di Kampung Adta Banceuy. Selain itu, bagian ini mencakup tujuan utama sisitem, ruang lingkup penggunaanya, serta pihak pihak yang terlibat dalam operasionalnya.

### **Deskripsi Umum**

Dalam deskripsi umum ini menjelaskan spesifikasi Banceuy *Smart Heritage* sebagai sistem informasi digital berbasis *web*. Dokumen ini mencakup detail mengenai kebutuhan manfaat yang ditawarkan hingga perangkat lunak. Sistem ini dirancang agar dapat dimanfaatkan oleh berbagai pihak, termasuk masyarakat lokal, wisatawan, serta organisasi Pokdarwis. Pengguna dalam sistem ini terbagi ke dalam tiga kategori utama, yaitu *Admin*, Kontributor Budaya, dan Pengguna Umum, yang masing-*masing* memiliki peran serta tingkat akses berbeda dalam pengelolaan dan penggunaan informasi yang tersedia.

# **DESKRIPSI UMUM**

## **Perspektif Produk**

Banceuy *Smart Heritage*: Sisitem Informasi Digital Kampung Adat Banceuy berbasis *web* merupakan sebuah sistem Informasi berbasis *website* yang dibangun untuk mempermudah pengelolaan pengunjung, promosi wisata, dan pelestarian budaya lokal. Sistem ini nantinya akan dikelola oleh *Admin* yang bertugas mengelola seluruh tampilan, kemudian pengelola *Homestay* yang akan mengelola terkait pemesanan *Homestay* dan pengguna yaitu wisatawan untuk melakukan kunjungan serta mengatehui informasi terkait Banceuy. Banceuy *Smart Heritage* ini memiliki beberapa fitur dan menu terkait pengelolaan kunjungan, pemesanan paket wisata, *custome* kegiatan edukasi, pemesanan *Homestay*, informasi terkait sub budaya di Banceuy, Galeri kegiatan, dan informasi *contact person* hingga media sosial Kampung Adat Banceuy.

## ***Software Interface***

Dalam pembuatan sistem ini, dibutuhkan *software* untuk mendukung proses Pengembangan Banceuy *Smart Heritage* : Sistem Informasi Digital Kampung Adat Banceuy. *Software* tersebut antara lain:

* + 1. Sistem Operasi

Sistem Operasi (*User)*: Semua tipe *platform*

Sistem Operasi (Pembangunan *Website*): Windows 11

* + 1. Data

Database: MySQL

*Thesaurus*: Bahasa Indonesia

* + 1. Bahasa pemrograman *Web*

Bahasa pemrograman: PHP, HTML, CSS dan Javascript

*Framework*: Laravel

* + 1. *Browser*

Semua *browser*

* + 1. Desain Antarmuka

Alat desain: Figma dan Canva

* + 1. Software Pendukung Lain

Software pendukung lainnya sebagai berikut:

1. Visual Studio Code
2. Midtrans
3. Mailtrap
4. Microsoft Word
5. Microsoft Excel
6. StarUML

## ***Hardware Interface***

*Hardware interface* yang dibutuhkan untuk membantu proses perancangan dari Sistem Informasi ini yaitu:

* + - 1. *Keyboard* merupakan salah satu alat untuk proses meng*input*kan informasi yang dibutuhkan oleh sistem. Baik berupa karakter, angka maupun simbol-simbol.
      2. *Mouse* dapat membantu sistem untuk dapat mengenali *input* dari pengguna dengan melakukan klik.
      3. Monitor dapat membantu pengguna untuk mengetahui dan menampilkan informasi.
      4. Printer dapat membantu *admin* dan pimpinan dalam mencetak surat bukti peminjaman, pengembalian dan laporan.
      5. Smartphone berfungsi untuk memindai QR code dengan memanfaatkan kamera bawaan yang dilengkapi teknologi pemrosesan gambar.

## **Manfaat Produk**

Manfaat dari produk yang dapat diperoleh dengan menggunakan sistem Banceuy *Smart Heritage* berbasis *website* ini sebagai berikut:

1. Memudahkan Divisi *Marketing* terhadap pengelolaan pengunjung yang akan datang ke Banceuy, dengan sistem ini secara otomatis akan tercatat seperti pengunjung, jadwal kunjungan, kegiatan dan lama waktu yang akan dihabiskan oleh wisatawan. Hal ini akan membantu pengelolaan kunjungan lebih efektif, sehingga pihak pengelola wisata yaitu Pokadarwis akan terbantu menangani kunjungan dengan sistem berbasis *website* ini.
2. Menyediakan informasi terpusat bagi calon wisatawan, sehingga mereka tidak akan kebingungan jika ingin berkunjung ke Kampung Adat Banceuy.
3. Membantu informasi cepat tersampaikan terkait pemesanan *Homestay* langsung kepada pengelola *Homestay* dengan adanya notifikasi pemesanan *Homestay* pada sistem *website* ini.
4. Membantu melestarikan budaya dengan lebih luas, karna sistem ini akan mudah diakses maka dari itu pengunjung domestik maupun mancanegara akan mudah mendapat informasi terkait kebudayaan lokal yang ada di Kampung Adat Banceuy.

## **Karakteristik *User***

1. *Admin*

*Admin* memiliki tanggung jawab penuh dalam mengelola seluruh sistem. Dengan hak akses tanpa batas, *Admin* dapat mengelola data kunjungan ataupun pesanan paket dan lainya.

1. Pengelola *Homestay*

Pengelola *Homestay* bertanggung jawab atas pengelolaan pemesanan *Homestay* yang dilakukan oleh pengunjung termasuk konfirmasi, dan pembatalan.

1. Pengguna / Masyarakat Umum

Pengguna atau masyarakat umum adalah individu yang dapat mengakses sistem untuk mencari informasi terkait layanan yang tersedia. Mereka dapat melakukan pemesanan *Homestay*, paket wisata, serta mengakses galeri kegiatan untuk mendapatkan gambaran tentang pengalaman yang ditawarkan.

## **Batasan-Batasan**

Sistem yang dikembangkan hanya mencakup hingga tahap pengujian dan penggunaan program. Penulis membatasi ruang lingkup permasalahan dan berfokus pada aspek yang berkaitan dengan Banceuy *Smart Heritage*, yaitu sebagai berikut:

1. Banceuy *Smart Heritage* berfokus pada kebutuhan Divisi *Marketing* untuk menyediakan informasi terpusat bagi pengunjung, mengelola kunjungan lebih efektif, promosi wisata yang lebih luas dan pelestarian budaya.
2. Banceuy *Smart Heritage* menangani data kunjungan Kampung Adat Banceuy, dimulai dari identitas pengunjung, paket yang dipilih, lama berkunjung dan jadwal kunjungan.

## **Asumsi dan Ketergantungan**

* 1. Pengguna sistem ini perlu memiliki pemahaman dalam menggunakan serta mengoperasikan teknologi dan internet, khususnya yang berbasis *website.*
  2. *Website* ini berfungsi dengan dukungan akses internet.

# **SPESIFIKASI KEBUTUHAN**

## **Kebutuhan Antarmuka Eksternal**

### **Antarmuka Pemakai (*User Interface)***

Antarmuka pengguna dalam sistem Banceuy *Smart Heritage* dirancang agar mudah digunakan dan responsif di berbagai perangkat. Antarmuka ini mencakup:

1. Desain Responsif
2. Sistem dapat diakses melalui berbagai perangkat, termasuk komputer, tablet, dan ponsel.
3. Menggunakan *framework* Bootstrap untuk memastikan tampilan yang rapi dan fleksibel.
4. Tampilan yang *User-Friendly*
5. Menggunakan kombinasi HTML, CSS, dan JavaScript untuk menciptakan tampilan yang menarik dan mudah dipahami.
6. Navigasi dibuat sederhana agar pengguna dapat dengan cepat menemukan informasi wisata, pemesanan *homestay*, dan sejarah budaya.
7. Komponen Utama Antarmuka
8. Halaman utama yang menyajikan informasi utama secara menarik.
9. Formulir pemesanan yang intuitif, mencakup pilihan paket wisata, tanggal kunjungan, jumlah pengunjung, serta metode pembayaran.
10. Dashboard *Admin* & Pengelola *Homestay*, yang menyediakan kontrol penuh terhadap manajemen pengunjung dan pemesanan.

### **Antarmuka PL (*Software Interface*)**

Agar sistem Banceuy *Smart Heritage* dapat berjalan dengan baik, diperlukan integrasi dengan berbagai perangkat lunak lain. Spesifikasi antarmuka perangkat lunak meliputi:

1. Sistem Operasi
   1. Untuk Pengguna: Sistem ini berbasis *web*, sehingga dapat diakses melalui berbagai sistem operasi, seperti Windows, macOS, Android, dan iOS.
   2. Untuk Pengembang: Sistem dikembangkan menggunakan Windows 11 sebagai lingkungan utama pengembangan.
2. *Database Management*
   1. Sistem menggunakan MySQL untuk menyimpan dan mengelola data pengunjung, pemesanan, serta informasi wisata.
   2. Data yang disimpan mencakup informasi akun pengguna, detail pemesanan, histori kunjungan, dan transaksi pembayaran.
3. Bahasa Pemrograman
   1. *Backend* menggunakan PHP dengan *framework* Laravel.
   2. *Frontend* dibangun menggunakan kombinasi HTML, CSS, dan JavaScript.
4. *Browser* yang Didukung
   1. Sistem ini dapat diakses melalui berbagai *browser*, tetapi Google Chrome dan Microsoft Edge lebih direkomendasikan untuk pengalaman yang lebih optimal.
5. Keamanan dan Autentikasi
   1. Sistem menggunakan middleware Laravel untuk mengatur akses pengguna berdasarkan peran (*Admin*, Pengelola *Homestay*, dan Pengunjung).
   2. Data pengguna dan transaksi akan diamankan dengan protokol HTTPS untuk mencegah kebocoran data.

### **Antarmuka PK (*Hardware Interface*)**

Sistem Banceuy *Smart Heritage* membutuhkan perangkat keras tertentu agar dapat digunakan secara optimal. Berikut adalah perangkat yang digunakan:

1. Perangkat *Input*
   1. *Keyboard*: Digunakan untuk memasukkan informasi login, data pemesanan, serta *input* informasi lainnya oleh pengguna dan *admin*.
   2. Mouse: Digunakan untuk navigasi dalam antarmuka pengguna, terutama saat memilih paket wisata, mengisi formulir pemesanan, dan mengelola data di dashboard *admin*.
2. Perangkat *Output*
   1. Monitor: Berfungsi untuk menampilkan informasi kepada pengguna, seperti detail pemesanan, status pembayaran, dan informasi wisata.
   2. Printer (opsional): Dapat digunakan oleh *admin* untuk mencetak laporan data pengunjung dan pemesanan jika diperlukan.
3. Perangkat Jaringan
   1. Sistem ini memerlukan koneksi internet agar pengguna dapat mengakses layanan secara online.
   2. *Server* berbasis cloud digunakan untuk menyimpan data dan mengelola sistem secara *real-time*.

### **Antarmuka Komunikasi (*Communication Interface*)**

Komunikasi dalam sistem ini melibatkan interaksi antara pengunjung, *admin*, dan pengelola *homestay*. Beberapa elemen komunikasi yang diterapkan dalam sistem ini meliputi:

1. Protokol Komunikasi
   1. Sistem menggunakan protokol HTTPS untuk memastikan komunikasi antara pengguna dan *server* aman dari ancaman peretasan.
   2. API yang digunakan untuk komunikasi antara frontend dan backend berbasis RESTful API.
2. Notifikasi dan Komunikasi Pengguna
   1. *Admin* dan Pengelola *Homestay* akan menerima notifikasi otomatis saat ada pemesanan baru.
   2. Pengguna akan menerima konfirmasi pemesanan dan detail pembayaran melalui email atau langsung di dalam sistem.
3. Integrasi dengan Pihak Ketiga
   1. Sistem dapat diintegrasikan dengan layanan pihak ketiga seperti WhatsApp API atau Email Notification untuk mengirimkan informasi penting kepada pengguna.
   2. Kemungkinan integrasi dengan sistem payment gateway untuk transaksi pembayaran online di masa depan.

## **Kebutuhan Fungsional**

### **Pendahuluan**

Kebutuhan fungsional menjelaskan fitur dan fungsi utama yang harus dimiliki sistem Banceuy *Smart Heritage* agar dapat berjalan sesuai tujuan. Sistem ini dikembangkan untuk memudahkan pengelolaan kunjungan wisata, pemesanan paket, serta promosi budaya di Kampung Adat Banceuy.

Sistem harus mampu:

* Mengelola pengguna dengan berbagai peran, yaitu *Admin*, Pengelola *Homestay*, dan Pengunjung.
* Memfasilitasi pemesanan paket wisata dan penginapan secara online.
* Menyediakan informasi wisata dan budaya secara lengkap.
* Mengelola data kunjungan dengan baik.

### ***Input***

Sistem menerima *input* dari berbagai pengguna, antara lain:

1. Pengguna Umum
   1. Melakukan pendaftaran akun.
   2. Melakukan login, logout.
   3. Melalukan pemesanan paket wisata.
   4. Melihat informasi seputar Banceuy.
   5. Melakukan pembayaran paket secara online.
   6. Mengisi buku kunjungan.
   7. Dapat memesan *Homestay.*
2. Pengelola *Homestay*
   1. Memasukkan informasi ketersediaan kamar dan harga.
   2. Mengelola daftar pemesanan *Homestay*.
3. *Admin*
   1. Mengelola data pengunjung.
   2. Mengelola data pesanan paket.
   3. Mengelola data pesanan *Homestay.*
   4. Memasukkan informasi tentang budaya dan wisata di Kampung Adat Banceuy.
4. Midtrans
   * + 1. Melakukan verifikasi Pembayaran.

### **Proses**

Sistem memproses berbagai aktivitas utama, di antaranya:

1. Manajemen Pengguna
   1. *Admin* dapat membuat, mengubah, dan menghapus akun pengguna.
   2. Pengguna dapat melakukan login dan logout.
   3. Pengelola *Homestay* dapat mengelola pesanan *Homestay* yang masuk.
   4. Pengunjung melakukan login, logout, pesan paket, pesan homestay dan juga mengisi buku kunjungan untuk revieuw.
2. Pemesanan dan Pengelolaan Wisata
   1. Pengunjung dapat memilih paket wisata dan melakukan pemesanan.
   2. Sistem mencatat pesanan, termasuk tanggal kunjungan, jumlah pengunjung, dan detail pembayaran.
   3. Pengelola *Homestay* menerima pemesanan *homestay*.
3. Sistem Notifikasi
   1. Pengguna menerima konfirmasi otomatis setelah pemesanan berhasil.
   2. *Admin* dan pengelola *homestay* mendapatkan notifikasi pemesanan baru.
4. Manajemen Informasi Budaya dan Wisata
   1. *Admin* dapat mengunggah informasi sejarah, tradisi, dan budaya lokal.
   2. Pengguna dapat mengakses informasi mengenai atraksi wisata, kuliner, permainan tradisional, dan event lokal.

### ***Output***

Sistem menghasilkan berbagai *output*, seperti:

1. Konfirmasi Pemesanan

Email atau notifikasi dalam sistem yang dikirimkan ke pengguna setelah melakukan pemesanan wisata atau *homestay*.

1. Laporan Data Pengunjung

*Admin* dapat melihat laporan jumlah pengunjung, paket wisata yang dipilih, serta jumlah transaksi yang telah terjadi.

1. Dashboard *Admin*

Menampilkan data pengunjung, pemesanan paket, pemesanan homestay.

1. Tampilan Informasi Wisata dan Budaya

Pengguna dapat melihat halaman yang berisi informasi lengkap tentang sejarah, tradisi, kuliner, dan atraksi di Kampung Adat Banceuy.

## **Kebutuhan Non Fungsional**

|  |  |
| --- | --- |
| **Parameter** | **Requirements** |
| *Availability* | Sistem harus dapat diakses secara 24/7, terutama saat musim liburan untuk memfasilitasi pemesanan wisata. |
| *Portability* | Sistem berbasis *web*, sehingga dapat diakses melalui berbagai perangkat seperti komputer, tablet, dan smartphone. |
| *Ergonomy* | Antarmuka dirancang agar mudah digunakan, dengan navigasi sederhana dan tampilan yang responsif. |
| *Security* | - Sistem menggunakan autentikasi pengguna (login dan peran berdasarkan hak akses).  - Data pengguna dan transaksi diamankan menggunakan protokol HTTPS.  - *Admin* memiliki kontrol penuh atas pengelolaan akun pengguna dan pemesanan. |
| *Bahasa Komunikasi* | Sistem menggunakan Bahasa Indonesia sebagai bahasa utama dalam antarmuka pengguna, dengan kemungkinan tambahan Bahasa Inggris untuk wisatawan mancanegara. |

## ***Design Constraints***

### ***Software Process Requirements***

1. Sistem harus menggunakan metode Agile dalam pengembangannya agar dapat beradaptasi dengan perubahan kebutuhan pengguna.
2. Pengembangan dilakukan secara bertahap, dengan proses Requirement, Design, Development, Testing, Deployment, dan Review.

### ***Software Languages***

1. *Backend*: PHP dengan framework Laravel.
2. *Frontend*: HTML, CSS, dan JavaScript.
3. *Database*: MySQL sebagai sistem manajemen basis data.

## **3.5 *Online User Documentation and Help System Requirements***

1. Sistem harus menyediakan bantuan online bagi pengguna yang mengalami kendala saat menggunakan layanan.
2. Panduan pengguna akan tersedia dalam bentuk FAQ, tutorial, atau chatbot otomatis.

## **3.6 *Purchased Components***

Sistem ini tidak menggunakan komponen berbayar dalam pengembangannya, tetapi dalam versi lanjutan, dapat mempertimbangkan layanan pihak ketiga seperti:

1. *Payment Gateway* untuk transaksi online.
2. Google Maps API untuk menunjukkan lokasi wisata.

## **3.7 *Licensing Requirement***

1. Lisensi sistem mengikuti aturan *open-source*, dengan Laravel sebagai *framework* utama.
2. Jika menggunakan layanan berbayar, maka lisensi harus disesuaikan dengan penyedia layanan terkait.

## **3.8 *Legal, Copyright, and Other Notices***

1. Hak cipta sistem dimiliki oleh tim pengembang dan pengelola Kampung Adat Banceuy.
2. Pengguna dilarang melakukan duplikasi, modifikasi, atau distribusi sistem tanpa izin resmi.
3. Data pengguna akan dilindungi sesuai dengan regulasi Perlindungan Data Pribadi di Indonesia.

## **3.9 *Applicable Standards***

Sistem harus mematuhi standar berikut:

1. ISO 25010 – Standar kualitas perangkat lunak, mencakup aspek fungsionalitas, performa, dan keamanan.
2. SSL/TLS *Encryption* – Sistem harus menggunakan protokol enkripsi untuk mengamankan komunikasi antara pengguna dan *server.*

# **PEMODELAN PERANGKAT LUNAK**

## **Use Case Diagram**

Gambar 4.1 Use case Banceuy Smart Heritage

*Use Case* di atas melibatkan tiga aktor utama, yaitu *Admin*, Pengelola *Homestay*, dan Pengunjung, yang masing-masing berperan sebagai pengguna dalam Sistem Informasi yang akan dikembangkan. Setiap aktor memiliki hak akses yang berbeda sesuai dengan peran mereka, seperti dijelaskan berikut ini.

### **4.1.1** **Deskripsi Aktor**

Tabel 4.1 Deskripsi Aktor

|  |  |  |
| --- | --- | --- |
| No | Nama Aktor | Deskripsi |
| 1. | *Admin* | Admin memiliki akses penuh dalam sistem. Admin dapat mengelola data pengunjung, memverifikasi pembayaran, mengelola pesanan paket wisata termasuk mengatur tanggal kunjungan, serta mengelola pengguna (tambah, edit, hapus user). Admin juga berperan dalam mengelola data kunjungan dan memantau aktivitas dalam sistem. |
| 2. | Pengelola *Homestay* | Pengelola homestay bertanggung jawab mengelola pesanan homestay. Mereka dapat mengatur ketersediaan homestay, menerima pesanan homestay dari pengunjung, dan melakukan konfirmasi terhadap pesanan homestay. |
| 3. | Pengunjung | Pengunjung adalah pengguna umum yang dapat mengakses informasi wisata, memesan paket wisata, memesan homestay, mengisi buku kunjungan, serta melakukan pendaftaran akun (register). Pengunjung dapat menentukan tanggal kunjungan dan melihat informasi budaya, sejarah, makanan tradisional, serta kesenian yang tersedia di Banceuy Smart Heritage. |
| 4. | Midtrans | Midtrans adalah aktor yang memverifikasi pembayaran dalam pesanan yang dilakukan oleh Pengunjung. |

Setelah hak akses masing-masing aktor ditentukan, setiap *Use Case* juga memiliki deskripsi tersendiri. Penjelasan dari setiap *Use Case* dapat dilihat pada tabel berikut.

### **4.1.2** **Deskripsi Use Case Diagram**

Tabel 4.2 Deskripsi Use Case Diagram

|  |  |  |
| --- | --- | --- |
| No | *Use Case* | Deskripsi |
| 1. | *Login* | Fitur yang memungkinkan semua aktor (Admin, Pengelola Homestay, dan Pengunjung) masuk ke dalam sistem untuk mengakses fitur sesuai perannya. |
| 2. | *Logout* | Fitur untuk semua aktor keluar dari akun mereka setelah menggunakan sistem. |
| 3. | *Register* | Fitur yang memungkinkan pengunjung mendaftarkan akun baru agar bisa mengakses layanan sistem. |
| 4. | Kelola Pesanan Paket Wisata | Admin dapat mengelola pesanan paket wisata, termasuk menerima, memproses, dan mengonfirmasi pesanan dari pengunjung. |
| 6. | Kelola Data Pengunjung | Admin dapat mengelola data pengunjung, termasuk verifikasi, perbaikan, dan pencatatan informasi pengunjung. |
| 7. | Verifikasi Pembayaran | Midtrans melakukan verifikasi pembayaran dari pengunjung untuk pemesanan paket wisata atau homestay. |
| 8. | Kelola User | Admin dapat menambah, mengedit, dan menghapus user yang terdaftar dalam sistem. |
| 9. | Tambah User | Admin dapat menambahkan user baru ke dalam sistem. |
| 10. | Edit User | Admin dapat memperbarui informasi user yang sudah ada. |
| 11. | Hapus User | Admin dapat menghapus user dari sistem. |
| 12. | Akses Informasi Wisata | Pengunjung dapat mengakses informasi tentang wisata, budaya, sejarah, makanan tradisional, dan homestay yang tersedia. |
| 13. | Pesan Paket Wisata | Pengunjung dapat memilih dan memesan paket wisata yang tersedia melalui sistem. |
| 14. | Isi Buku Kunjungan | Pengunjung dapat mengisi buku kunjungan sebagai catatan pengalaman setelah berkunjung. |
| 15. | Pesan Homestay | Pengunjung dapat memesan homestay yang tersedia untuk mendukung kunjungan wisata mereka. |
| 16. | Pesanan Homestay | Pengelola Homestay dapat menerima, dan mengonfirmasi pesanan homestay dari pengunjung. |
| 17. | Print Data Pesanan Paket | Pengelola Homestay dapat mengatur jumlah homestay yang tersedia dan statusnya di sistem. |
| 18. | Print Data Pesanan Homestay | Pengelola Homestay dapat memberikan konfirmasi terhadap pemesanan homestay yang diterima. |
| 19. | Edit Data | Admin dapat memperbarui informasi data pengunjung yang sudah ada ajika terdapat kesalahan. |
| 20. | Hapus Data | Admin dapat menghapus informasi data pengunjung yang sudah ada bila harus di hapus. |
| 21. | Tambah Data | Admin dapat menambah informasi data pengunjung bila perlu. |
| 22. | Edit Pesanan | Admin dapat memperbarui pesanan pengunjung yang sudah ada bila ada kesalahan. |
| 23. | Hapus Pesanan | Admin dapat menghapus pesanan pengunjung yang sudah ada bila harus di hapus. |

## **4.1.3 Scenario Use Case Diagram**

UseCase scenario login adalah deskripsi dari serangkaian langkah yang diambil oleh pengguna untuk mengakses sistem dengan menggunakan kredensial yang valid.

1. Login

Tabel 3.1 Skenario Login

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-01 | |
| Nama *Usecase* | | Login | |
| Tujuan | | Memungkinkan semua aktor (Admin, Pengunjung, dan Pengelola Homestay) untuk masuk ke dalam sistem dengan akun mereka masing-masing. | |
| Deskripsi | | Use case ini menggambarkan proses aktor memasukkan kredensial (username/email dan password) untuk mengakses sistem dan mendapatkan hak akses sesuai perannya. | |
| Nama aktor | | Admin, Pengunjung, Pengelola Homestay | |
| Triger Event | | Aktor membuka halaman login dan memasukkan data login | |
| Stakeholder | | Admin: Akses fitur manajemen sistem  Pengunjung: Akses layanan pemesanan wisata  Pengelola Homestay: Akses manajemen homestay | |
| Related *usecase* | | Menampilkan Form Login – Sistem menampilkan halaman login saat user mengakses sistem.  Memverifikasi Kredensial – Sistem mengecek validitas username/email dan password.  Menampilkan Dashboard Sesuai Hak Akses – Setelah login berhasil, sistem menampilkan halaman dashboard sesuai peran aktor. | |
| Skenario utama | | |  |
| Pre kondisi :  Aktor telah memiliki akun dalam sistem. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Aktor mengakses halaman login. | |  | |
|  | | 2. Sistem menampilkan form login. | |
| 3. Aktor mengisi username/email dan password. | |  | |
|  | | 4. Sistem melakukan validasi kredensial. | |
| 5. Jika valid, sistem mengarahkan ke dashboard sesuai role. | | 6. Sistem menampilkan halaman dashboard berdasarkan peran user. | |
| *Post*-kondisi: Aktor berhasil masuk dan diarahkan ke dashboard sesuai dengan hak aksesnya. | |  | |
| **Exception** | |  | |
| 3.1 Kredensial tidak lengkap:  Jika username/email atau password kosong, sistem menampilkan pesan error:  "Harap lengkapi semua kolom sebelum melanjutkan."  3.2 Kredensial tidak valid:  Jika username/email atau password salah, sistem menampilkan pesan error:  "Username atau password salah. Silakan coba lagi." | | | |

1. Logout

Tabel 3.2 Skenario Logout

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-02 | |
| Nama *Usecase* | | Logout | |
| Tujuan | | Memungkinkan semua aktor (Admin, Pengunjung, dan Pengelola Homestay) untuk keluar dari sistem dengan aman dan mengakhiri sesi pengguna. | |
| Deskripsi | | Use case ini menggambarkan proses aktor keluar dari sistem untuk mengakhiri sesi aktifnya. Setelah logout, pengguna akan diarahkan kembali ke halaman login atau halaman utama sistem. | |
| Nama aktor | | Admin, Pengunjung, Pengelola Homestay | |
| Triger Event | | Aktor mengklik tombol Logout di dalam sistem.  Sistem mengakhiri sesi pengguna.  Sistem mengarahkan pengguna kembali ke halaman login atau halaman utama. | |
| Stakeholder | | Admin: Dapat keluar dari sistem setelah menyelesaikan tugasnya.  Pengunjung: Dapat logout setelah melakukan pemesanan atau melihat informasi wisata.  Pengelola Homestay: Dapat logout setelah mengelola pesanan homestay. | |
| Related *usecase* | | Login (UC-01) – Pengguna harus login kembali untuk mengakses sistem setelah logout.  Kelola Sesi Pengguna – Sistem memastikan bahwa setelah logout, sesi pengguna benar-benar ditutup agar tidak bisa diakses tanpa login ulang. | |
| Skenario utama | | |  |
| Pre kondisi :  Aktor telah login ke dalam sistem. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Aktor mengakses menu logout. | |  | |
|  | | 2. Sistem menampilkan konfirmasi logout (opsional). | |
| 3. Aktor mengonfirmasi logout. | |  | |
|  | | 4. Sistem mengakhiri sesi pengguna. | |
| 5. Sistem mengarahkan aktor ke halaman login atau halaman utama. | | 6. Menampilkan notifikasi bahwa logout berhasil. | |
| *Post*-kondisi: Aktor berhasil keluar dari sistem, dan sesi pengguna telah diakhiri. | |  | |
| **Exception** | |  | |
| 3.1 Gangguan Koneksi Saat Logout:  Jika terjadi gangguan koneksi internet saat logout, sistem menampilkan pesan error: "Gagal logout. Silakan coba lagi."  3.2 Otomatis Logout Karena Timeout:  Jika pengguna tidak aktif dalam waktu tertentu (misalnya 30 menit), sistem otomatis mengakhiri sesi dan mengarahkan pengguna ke halaman login.  Pesan yang ditampilkan:  "Sesi Anda telah berakhir karena tidak ada aktivitas. Silakan login kembali." | | | |

|  |  |
| --- | --- |
|  |  |

1. Register

Tabel 3.3 Skenario Register

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-03 | |
| Nama *Usecase* | | Register | |
| Tujuan | | Memungkinkan Pengunjung untuk mendaftarkan akun baru di sistem agar dapat mengakses fitur-fitur yang memerlukan autentikasi, seperti pemesanan paket wisata atau homestay. | |
| Deskripsi | | Use case ini menggambarkan proses registrasi pengguna baru ke dalam sistem dengan mengisi formulir pendaftaran, memverifikasi informasi, dan menyimpan data pengguna ke dalam basis data. | |
| Nama aktor | | Pengunjung | |
| Triger Event | | Pengunjung mengakses halaman Register.  Pengunjung mengisi formulir pendaftaran dengan informasi yang diperlukan.  Sistem memverifikasi data yang dimasukkan.  Jika valid, sistem menyimpan data pengguna dan menampilkan konfirmasi bahwa akun telah berhasil dibuat.  Pengunjung dapat login ke dalam sistem menggunakan akun yang baru dibuat. | |
| Stakeholder | | Pengunjung: Mendaftar akun agar dapat menggunakan fitur sistem seperti pemesanan paket wisata dan homestay. | |
| Related *usecase* | | Menampilkan Form Registrasi – Sistem menampilkan halaman pendaftaran.  Memverifikasi Data Registrasi – Sistem memeriksa validitas data.  Menampilkan Notifikasi Sukses – Sistem memberikan notifikasi bahwa akun berhasil dibuat. | |
| Skenario utama | | |  |
| Pre kondisi :  Pengunjung belum memiliki akun di sistem. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Pengunjung mengakses halaman Register. | |  | |
|  | | 2. Sistem menampilkan formulir pendaftaran. | |
| 3. Pengunjung mengisi formulir pendaftaran (nama, email, password, dll.). | |  | |
|  | | 4. Sistem memvalidasi data yang dimasukkan. | |
| 5. Jika valid, sistem menyimpan data akun baru. | | 6. Sistem menampilkan pesan sukses dan mengarahkan ke halaman login. | |
| *Post*-kondisi: Akun pengunjung berhasil dibuat dan dapat digunakan untuk login. | |  | |
| **Exception** | |  | |
| 3.1 Data Tidak Lengkap:  Jika ada kolom kosong, sistem menampilkan pesan error: "Harap lengkapi semua data sebelum melanjutkan."  3.2 Email Sudah Terdaftar:  Jika email sudah digunakan, sistem menampilkan pesan error: "Email sudah terdaftar. Silakan gunakan email lain atau login." | | | |

|  |  |
| --- | --- |
|  |  |

1. Pesan Paket Wisata

Tabel 3.4 Skenario Pesan Paket Wisata

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-04 | |
| Nama *Usecase* | | Pesan Paket Wisata | |
| Tujuan | | Memungkinkan Pengunjung untuk memilih dan memesan paket wisata. | |
| Deskripsi | | Use case ini menggambarkan proses pemesanan paket wisata oleh pengunjung. | |
| Nama aktor | | Pengunjung | |
| Triger Event | | Pengunjung memilih paket wisata dan mengisi data pemesanan. | |
| Stakeholder | | Pengunjung: Dapat memilih paket wisata dan melakukan pemesanan. | |
| Related *usecase* | | Pilih Paket – Pengunjung memilih paket wisata yang tersedia.  Pembayaran – Pengunjung melakukan pembayaran untuk paket wisata yang dipilih.  Konfirmasi Pesanan – Sistem mengonfirmasi bahwa pesanan telah diterima. | |
| Skenario utama | | |  |
| Pre kondisi :  Pengunjung telah login ke sistem. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Pengunjung mengakses daftar paket wisata. | |  | |
|  | | 2. Pengunjung memilih paket wisata yang diinginkan. | |
| 3. Sistem menampilkan detail paket wisata. | |  | |
|  | | 4. Pengunjung mengisi data pemesanan dan memilih metode pembayaran. | |
| 5. Pengunjung melakukan pembayaran. | | 6. Sistem memverifikasi pembayaran dan mengonfirmasi pesanan. | |
| *Post*-kondisi: Pesanan paket wisata berhasil dibuat, dan pengunjung menerima notifikasi konfirmasi. | |  | |
| **Exception** | |  | |
| 4.1 Data pemesanan tidak lengkap:  Jika ada data yang kosong, sistem menampilkan pesan error: "Harap lengkapi semua data sebelum melanjutkan."  4.2 Pembayaran gagal:  Jika transaksi gagal, sistem menampilkan pesan error: "Pembayaran gagal. Silakan coba lagi atau gunakan metode lain." | | | |

1. Kelola Data Pengunjung

Tabel 3.5 Skenario Kelola Data Pengunjung

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-05 | |
| Nama *Usecase* | | Kelola Data Pengunjung | |
| Tujuan | | Memungkinkan admin untuk mengelola data kunjungan wisatawan ke kawasan Banceuy Smart Heritage. | |
| Deskripsi | | Admin dapat melihat, menambah, mengedit, dan menghapus data kunjungan berdasarkan catatan transaksi atau data yang diinput manual. | |
| Nama aktor | | Admin | |
| Triger Event | | Admin membuka halaman kelola data kunjungan. | |
| Stakeholder | | Admin: Memantau dan mengelola histori kunjungan wisatawan.  Pihak Pengelola: Mendapatkan data statistik jumlah kunjungan. | |
| Related *usecase* | | Lihat Detail Kunjungan, Tambah/Edit/Hapus Kunjungan | |
| Skenario utama | | |  |
| Pre kondisi :  Admin telah login ke dalam sistem. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Admin membuka menu kelola data kunjungan. | |  | |
|  | | 2. . Sistem menampilkan daftar data kunjungan. | |
| 3. Admin memilih tindakan (tambah, edit, hapus, lihat). | |  | |
|  | | 4. Sistem menampilkan form atau detail sesuai tindakan. | |
| 5. Admin menyelesaikan aksi yang dipilih. | | 6. Sistem memproses dan menyimpan perubahan. | |
| *Post*-kondisi: Data kunjungan diperbarui sesuai aksi yang dilakukan. | |  | |
| **Exception** | |  | |
| 3.1 Data tidak ditemukan → Pesan: "Data kunjungan tidak tersedia."  5.1 Field kosong saat tambah/edit → Pesan: "Lengkapi semua data sebelum menyimpan."  6.1 Gagal simpan → Pesan: "Data tidak berhasil disimpan." | | | |

1. Kelola Pesanan Paket Wisata

Tabel 3.6 Skenario Kelola Pesanan Paket Wisata

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-06 | |
| Nama *Usecase* | | Kelola Pesanan Paket Wisata | |
| Tujuan | | Memungkinkan admin untuk memantau dan memproses pesanan paket wisata dari pengunjung. | |
| DeskripsiAdmin | | Admin dapat melihat daftar pesanan paket wisata, mengecek status pembayaran, serta memperbarui status konfirmasi atau penolakan pesanan. | |
| Nama aktor | | Admin | |
| Triger Event | | Admin membuka menu kelola pesanan paket wisata. | |
| Stakeholder | | Admin: Mengelola dan memverifikasi data pesanan wisata.  Pengunjung: Mendapatkan informasi status pesanan yang mereka lakukan. | |
| Related *usecase* | | Verifikasi Pembayaran, Kelola Tanggal Kunjungan | |
| Skenario utama | | |  |
| Pre kondisi :  Admin sudah login ke sistem dan terdapat data pesanan dari pengunjung. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Admin membuka menu kelola pesanan paket wisata. | |  | |
|  | | 2. Sistem menampilkan daftar pesanan. | |
| 3. Admin memilih salah satu pesanan. | |  | |
|  | | 4. Sistem menampilkan detail pesanan. | |
| 5. Admin memverifikasi status pembayaran. | | 6. Sistem menampilkan status pembayaran dan pilihan tindakan. | |
| 7. Admin memilih konfirmasi atau tolak pesanan. | | 8. Sistem menyimpan status baru dan mengirim notifikasi ke pengunjung. | |
| *Post*-kondisi: Status pesanan telah diperbarui. | |  | |
| **Exception** | |  | |
| .1 Tidak ada pesanan → Pesan: "Belum ada pesanan masuk."  5.1 Data pembayaran tidak ditemukan → Pesan: "Data pembayaran tidak tersedia."   * 1. Gagal simpan → Pesan: "Status pesanan gagal diperbarui." | | | |

1. Pesanan Homestay

Tabel 3.7 Skenario Pesanan Homestay

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-07 | |
| Nama *Usecase* | | Pesanan Homestay | |
| Tujuan | | Memungkinkan Pengelola Homestay untuk mengelola pesanan yang dilakukan oleh Pengunjung. | |
| Deskripsi | | Use case ini menggambarkan proses penerimaan dan konfirmasi homestay. | |
| Nama aktor | | Pengelola Homestay | |
| Triger Event | | Pengelola Homestay membuka menu pesanan homestay. | |
| Stakeholder | | Pengelola Homestay: Dapat melihat, menerima, dan mengonfirmasi pesanan homestay.  Pengunjung: Dapat mengetahui status pesanan homestay mereka. | |
| Related *usecase* | | Pengelola Homestay menerima pesanan dari Pengunjung dan Sistem memperbarui status pesanan setelah dikonfirmasi oleh Pengelola. | |
| Skenario utama | | |  |
| Pre kondisi :  Ada pesanan homestay dari Pengunjung. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Pengelola Homestay membuka menu pesanan homestay. | |  | |
|  | | 2. Sistem menampilkan daftar pesanan yang masuk. | |
| 3. Pengelola Homestay memilih pesanan yang ingin diproses. | |  | |
|  | | 4. Pengelola Homestay menerima dan mengonfirmasi pesanan. | |
| 5. Sistem memperbarui status pesanan menjadi "Dikonfirmasi". | | 6. Pengunjung menerima notifikasi status pesanan. | |
| *Post*-kondisi: Pesanan homestay berhasil dikonfirmasi oleh Pengelola Homestay. | |  | |
| **Exception** | |  | |
| 4.1 Tidak ada pesanan masuk:  Sistem menampilkan pesan "Belum ada pesanan baru."  4.2 Pesanan dibatalkan oleh Pengunjung:  Sistem memperbarui status pesanan menjadi "Dibatalkan oleh pengguna." | | | |

1. Verifikasi Pembayaran

Tabel 3.8 Skenario Verifikasi Pembayaran

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-08 | |
| Nama *Usecase* | | Verifikasi Pembayaran | |
| Tujuan | | Memungkinkan Midtrans untuk mengelola transaksi pembayaran dari Pengunjung. | |
| Deskripsi | | Use case ini mencakup verifikasi pembayaran, pencatatan transaksi, dan pembaruan status pesanan. | |
| Nama aktor | | Midtrans | |
| Triger Event | | Midtrans dalam menu pembayaran di sistem. | |
| Stakeholder | | Midtrans: Memverifikasi pembayaran yang masuk.  Pengunjung: Mengetahui status pembayaran dan konfirmasi pemesanan. | |
| Related *usecase* | | Sistem mengecek keabsahan pembayaran yang dilakukan Pengunjung. Setelah pembayaran berhasil diverifikasi, pesanan dikonfirmasi. | |
| Skenario utama | | |  |
| Pre kondisi :  Ada transaksi pembayaran dari Pengunjung. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Midtrans membuka menu pembayaran. | |  | |
|  | | 2. Sistem menampilkan daftar transaksi pembayaran. | |
| 3. Midtrans memilih transaksi yang akan diverifikasi. | |  | |
|  | | 4. Memproses bukti pembayaran. | |
| 5. Jika valid, Midtrans mengonfirmasi pembayaran. | | 6. Sistem memperbarui status pesanan menjadi "Dibayar". | |
| *Post*-kondisi: Pembayaran terverifikasi, dan pesanan Pengunjung dikonfirmasi. | |  | |
| **Exception** | |  | |
| 4.1 Bukti pembayaran tidak valid:  Sistem menampilkan pesan "Bukti pembayaran tidak valid. Silakan unggah ulang."  4.2 Transaksi tidak ditemukan:  Sistem menampilkan pesan "Pembayaran tidak ditemukan. Silakan cek kembali data transaksi." | | | |

1. Akses Informasi Wisata

Tabel 3.9 Skenario Akses Informasi Wisata

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-09 | |
| Nama *Usecase* | | Akses Informasi Wisata | |
| Tujuan | | Memungkinkan Pengunjung untuk melihat informasi wisata yang tersedia di sistem. | |
| Deskripsi | | Use case ini mencakup akses informasi mengenai sejarah, kuliner, kaulinan barudak, kesenian, wisata alam, adat tradisi dan lainya. | |
| Nama aktor | | Pengunjung | |
| Triger Event | | Pengunjung membuka halaman informasi wisata. | |
| Stakeholder | | Pengunjung: Dapat melihat berbagai informasi wisata yang tersedia. | |
| Related *usecase* | | Lihat Paket Wisata – Pengunjung melihat daftar paket wisata.  Lihat Kuliner – Pengunjung mengakses informasi kuliner khas daerah dan seterusnya. | |
| Skenario utama | | |  |
| Pre kondisi :  Pengunjung mengakses sistem. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Pengunjung membuka halaman informasi wisata. | |  | |
|  | | 2. Sistem menampilkan kategori informasi wisata. | |
| 3. Pengunjung memilih kategori yang ingin dilihat (sejarah, kuliner, kesenian, paket wisata dll.). | |  | |
|  | | 4. Sistem menampilkan daftar informasi yang sesuai. | |
| 5. Pengunjung dapat membaca detail informasi. | |  | |
| *Post*-kondisi: Pengunjung mendapatkan informasi yang dibutuhkan mengenai wisata. | |  | |
| **Exception** | |  | |
| 4.1 Data tidak ditemukan: Sistem menampilkan pesan "Informasi belum tersedia." | | | |

1. Kelola User

Tabel 3.10 Skenario Kelola User

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-10 | |
| Nama *Usecase* | | Kelola User | |
| Tujuan | | Memungkinkan Admin untuk mengelola akun pengguna dalam sistem. | |
| Deskripsi | | Use case ini mencakup pembuatan, pengeditan, dan penghapusan akun pengguna. | |
| Nama aktor | | Admin | |
| Triger Event | | Admin membuka menu manajemen pengguna. | |
| Stakeholder | | Admin: Memiliki hak akses untuk mengelola akun pengguna.  Pengguna lain: Akun mereka dapat dikelola oleh Admin jika diperlukan. | |
| Related *usecase* | | Tambah User – Admin dapat membuat akun baru.  Edit User – Admin dapat mengedit informasi pengguna.  Hapus User – Admin dapat menghapus akun pengguna. | |
| Skenario utama | | |  |
| Pre kondisi :  Admin memiliki akses ke sistem manajemen pengguna. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Admin membuka menu kelola pengguna. | |  | |
|  | | 2. Sistem menampilkan daftar pengguna. | |
| 3. Admin memilih opsi (Tambah/Edit/Hapus). | |  | |
|  | | 4. Admin melakukan perubahan yang diperlukan. | |
| 5. Sistem memperbarui data pengguna. | |  | |
| *Post*-kondisi: Perubahan data pengguna berhasil dilakukan. | |  | |
| **Exception** | |  | |
| 4.1 Data user tidak valid:  Sistem menampilkan pesan "Harap isi data dengan benar."  4.2 User tidak ditemukan:  Sistem menampilkan pesan "User tidak ditemukan atau telah dihapus." | | | |

1. Isi Buku Kunjungan

Tabel 3.11 Isi Buku Kunjungan

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-11 | |
| Nama *Usecase* | | Isi Buku Kunjungan | |
| Tujuan | | Pengunjung mengisi nama, alamat, no hp, umur, tanggal kunjungan, instansi dan ulasan. | |
| Deskripsi | | Use case ini mencakup pengisian data kunjungan. | |
| Nama aktor | | Pengunjung | |
| Triger Event | | Pengunjung membuka fitur isi buku kunjungan. | |
| Stakeholder | | Pengunjung: Bisa mengisi data dan memberikan ulasan.  Admin: Bisa melihat, mengelola data dan ulasan yang masuk. | |
| Related *usecase* | |  | |
| Skenario utama | | |  |
| Pre kondisi :  Pengunjung telah selesai berwisata. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Pengunjung membuka halaman buku kunjungan. | |  | |
|  | | 2. Sistem menampilkan formulir ulasan. | |
| 3. Pengunjung mengisi ulasan dan mengirimkannya. | |  | |
|  | | 4. Sistem menyimpan ulasan dan menampilkannya di buku kunjungan. | |
| *Post*-kondisi: Ulasan pengunjung tersimpan. | |  | |
| **Exception** | |  | |
| 3.1 Ulasan kosong:  Sistem menampilkan pesan "Harap isi ulasan sebelum mengirimkan." | | | |

1. Pesan Homestay

Tabel 3.13 Skenario Pesan Homestay

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-13 | |
| Nama *Usecase* | | Pesan Homestay | |
| Tujuan | | Memungkinkan pengunjung untuk memesan homestay sebagai bagian dari perjalanan wisata mereka atau untuk kepentingan penelitian. | |
| Deskripsi | | Use case ini mencakup pemilihan homestay yang tersedia dan proses pemesanan. | |
| Nama aktor | | Pengunjung | |
| Triger Event | | Pengunjung ingin menginap selama kunjungan wisata. | |
| Stakeholder | | Pengunjung: Bisa memesan homestay untuk menginap.  Pengelola Homestay: Bisa menerima dan mengelola pesanan homestay. | |
| Related *usecase* | |  | |
| Skenario utama | | |  |
| Pre kondisi : Ada homestay yang tersedia dalam sistem. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Pengunjung membuka halaman pemesanan homestay. | | 2. Sistem menampilkan daftar homestay yang tersedia. | |
| 3. Pengunjung memilih homestay yang diinginkan. | | 4. Sistem menampilkan detail homestay dan opsi pemesanan. | |
| 5. Pengunjung mengonfirmasi pemesanan homestay. | | 6. Sistem menyimpan pemesanan dan mengirim notifikasi ke pengelola homestay. | |
| *Post*-kondisi: Pesanan homestay telah dibuat dan diteruskan ke pengelola homestay. | |  | |
| **Exception** | |  | |
| 2.1 Tidak ada homestay yang tersedia: Sistem menampilkan pesan "Homestay tidak tersedia untuk tanggal yang dipilih." | | | |

1. Hapus User

Tabel 3.14 Skenario Hapus User

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-14 | |
| Nama *Usecase* | | Hapus User | |
| Tujuan | | Memungkinkan admin untuk menghapus akun pengguna dari sistem. | |
| Deskripsi | | Use case ini mencakup proses penghapusan akun pengguna dari sistem oleh admin. | |
| Nama aktor | | Admin | |
| Triger Event | | Admin ingin menghapus akun pengguna. | |
| Stakeholder | | Admin: Bisa menghapus akun pengguna yang tidak aktif atau melanggar ketentuan.  Pengguna: Akun mereka dapat dihapus jika diperlukan. | |
| Related *usecase* | |  | |
| Skenario utama | | |  |
| Pre kondisi : Akun pengguna yang akan dihapus tersedia dalam sistem. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Admin membuka halaman pengelolaan pengguna. | | 2. Sistem menampilkan daftar pengguna yang terdaftar. | |
| 3. Admin memilih pengguna yang akan dihapus. | | 4. Sistem menampilkan konfirmasi penghapusan. | |
| 5. Admin mengonfirmasi penghapusan pengguna. | | Sistem menghapus akun pengguna dari database. | |
| *Post*-kondisi: Akun pengguna berhasil dihapus dari sistem. | |  | |
| **Exception** | |  | |
| 2.1 Akun tidak ditemukan: Sistem menampilkan pesan "Akun pengguna tidak ditemukan."  2.2 Gagal menghapus akun: Sistem menampilkan pesan "Penghapusan akun gagal, silakan coba lagi." | | | |

1. Edit User

Tabel 3.15 Skenario Edit User

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-15 | |
| Nama *Usecase* | | Edit User | |
| Tujuan | | Memungkinkan admin untuk mengedit informasi pengguna yang terdaftar dalam sistem. | |
| Deskripsi | | Use case ini mencakup proses perubahan data pengguna seperti nama, email, dan peran dalam sistem. | |
| Nama aktor | | Admin | |
| Triger Event | | Admin ingin mengubah informasi pengguna. | |
| Stakeholder | | Admin: Bisa mengedit informasi pengguna jika diperlukan.  Pengguna: Data mereka dapat diperbarui oleh admin jika ada kesalahan atau perubahan informasi. | |
| Related *usecase* | |  | |
| Skenario utama | | |  |
| Pre kondisi : Akun pengguna yang akan diedit tersedia dalam sistem. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Admin membuka halaman pengelolaan pengguna. | | 2. Sistem menampilkan daftar pengguna yang terdaftar. | |
| 3. Admin memilih pengguna yang ingin diedit. | | 4. Sistem menampilkan formulir dengan data pengguna saat ini. | |
| 5. Admin mengubah informasi yang diperlukan. | | 6. Sistem memperbarui informasi pengguna dalam database. | |
| *Post*-kondisi: Data pengguna berhasil diperbarui dalam sistem. | |  | |
| **Exception** | |  | |
| 2.1 Akun tidak ditemukan: Sistem menampilkan pesan "Akun pengguna tidak ditemukan."  2.2 Gagal menyimpan perubahan: Sistem menampilkan pesan "Perubahan data gagal disimpan, silakan coba lagi." | | | |

1. Tambah User

Tabel 3.16 Skenario Tambah User

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-16 | |
| Nama *Usecase* | | Tambah User | |
| Tujuan | | Memungkinkan admin untuk menambahkan pengguna baru ke dalam sistem. | |
| Deskripsi | | Use case ini mencakup proses pendaftaran akun baru oleh admin. | |
| Nama aktor | | Admin | |
| Triger Event | | Admin ingin menambahkan akun pengguna baru. | |
| Stakeholder | | Admin: Bisa menambahkan pengguna baru untuk mengakses sistem.  Pengguna Baru: Bisa mendapatkan akun dan mengakses sistem setelah didaftarkan. | |
| Related *usecase* | |  | |
| Skenario utama | | |  |
| Pre kondisi : Admin memiliki akses untuk menambahkan pengguna baru. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Admin membuka halaman pengelolaan pengguna. | | 2. Sistem menampilkan daftar pengguna yang terdaftar. | |
| 3. Admin memilih opsi "Tambah User." | | 4. Sistem menampilkan formulir pendaftaran pengguna baru. | |
| 5. Admin mengisi data pengguna dan menyimpan. | | 6. Sistem menyimpan data pengguna baru dan menampilkan konfirmasi. | |
| *Post*-kondisi: Akun pengguna baru berhasil ditambahkan ke sistem. | |  | |
| **Exception** | |  | |
| 2.1 Data tidak lengkap: Sistem menampilkan pesan "Silakan lengkapi semua kolom yang diperlukan."  2.2 Gagal menyimpan data: Sistem menampilkan pesan "Pendaftaran pengguna gagal, silakan coba lagi." | | | |

1. Tambah Data

Tabel 3.19 Skenario Tambah Data

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-19 | |
| Nama *Usecase* | | Tambah Data | |
| Tujuan | | Menambahkan data baru ke dalam sistem. | |
| Deskripsi | | Use case ini dilakukan oleh admin untuk menambahkan data baru ke dalam sistem. | |
| Nama aktor | | Admin | |
| Triger Event | | Admin memilih opsi “Tambah Data” dari menu sistem. | |
| Stakeholder | | Admin | |
| Related *usecase* | | Kelola User, Kelola Data Pengunjung | |
| Skenario utama | | |  |
| Pre kondisi : Admin telah login ke sistem. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Admin memilih menu Tambah Data | | 2. Sistem menampilkan form isian | |
| 3. Admin mengisi data yang diminta | | 4. Sistem melakukan validasi | |
| 5. Admin menyimpan data | | 6. Sistem menyimpan data dan menampilkan notifikasi berhasil | |
| *Post*-kondisi: Data berhasil ditambahkan ke dalam basis data sistem. | |  | |
| **Exception** | |  | |
| Format data tidak sesuai  Data duplikat | | | |

1. Edit Data

Tabel 3.20 Skenario Edit Data

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-20 | |
| Nama *Usecase* | | Edit Data | |
| Tujuan | | Mengubah data yang telah ada di dalam sistem. | |
| Deskripsi | | Admin dapat memperbarui data pengguna atau pengunjung yang sudah ada. | |
| Nama aktor | | Admin | |
| Triger Event | | Admin memilih opsi “Edit Data” pada data yang dipilih. | |
| Stakeholder | | Admin | |
| Related *usecase* | | Kelola User, Kelola Data Pengunjung | |
| Skenario utama | | |  |
| Pre kondisi : Data telah tersedia dan admin telah login. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Admin memilih data yang akan diedit | | 2. Sistem menampilkan data tersebut dalam form | |
| 3. Admin mengubah data | | 4. Sistem melakukan validasi | |
| 5. Admin menyimpan perubahan | | 6. Sistem menyimpan dan menampilkan notifikasi berhasil | |
| *Post*-kondisi: Data berhasil diperbarui di dalam sistem. | |  | |
| **Exception** | |  | |
| Data tidak ditemukan  Format perubahan tidak valid | | | |

1. Hapus Data

Tabel 3.21 Skenario Hapus Data

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-21 | |
| Nama *Usecase* | | Hapus Data | |
| Tujuan | | Menghapus data dari sistem. | |
| Deskripsi | | Admin dapat menghapus data pengunjung atau user yang tidak lagi diperlukan. | |
| Nama aktor | | Admin | |
| Triger Event | | Admin memilih opsi “Hapus” pada data yang ditampilkan. | |
| Stakeholder | | Admin | |
| Related *usecase* | | Kelola User, Kelola Data Pengunjung | |
| Skenario utama | | |  |
| Pre kondisi : dmin telah login dan data tersedia di sistem. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Admin memilih data yang ingin dihapus | | 2. Sistem menampilkan konfirmasi penghapusan | |
| 3. Admin mengonfirmasi penghapusan | | 4. Sistem menghapus data dan menampilkan pesan sukses | |
| *Post*-kondisi: Data tidak lagi tersedia di dalam sistem. | |  | |
| **Exception** | |  | |
| Data tidak ditemukan  Gagal koneksi ke basis data | | | |

1. Edit Pesanan

Tabel 3.22 Skenario Edit Pesanan

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-22 | |
| Nama *Usecase* | | Edit Pesanan | |
| Tujuan | | Mengubah detail pesanan yang telah dibuat. | |
| Deskripsi | | Pengunjung atau admin dapat memperbarui data pesanan sebelum diverifikasi. | |
| Nama aktor | | Admin | |
| Triger Event | | Admin memilih opsi edit pada daftar pesanan. | |
| Stakeholder | | Admin | |
| Related *usecase* | | Kelola Pesanan Paket Wisata, Pesanan Homestay | |
| Skenario utama | | |  |
| Pre kondisi : Pesanan belum diverifikasi. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Admin memilih edit Pesanan | | 2. Sistem menampilkan form edit pesanan | |
| 3. Admin mengubah data | | 4. Sistem memvalidasi perubahan | |
| 5. Admin menyimpan perubahan | | 6. Sistem memperbarui data pesanan | |
| *Post*-kondisi: Post-kondisi: Data pesanan diperbarui. | |  | |
| **Exception** | |  | |
| Pesanan tidak ada | | | |

1. Hapus Pesanan

Tabel 3.23 Skenario Hapus Pesanan

|  |  |  |  |
| --- | --- | --- | --- |
|  | Identifikasi | | |
| No. *Usecase* | | UC-23 | |
| Nama *Usecase* | | Hapus Pesanan | |
| Tujuan | | Menghapus pesanan yang tidak jadi dilakukan. | |
| Deskripsi | | Admin dapat membatalkan pesanan yang belum dikonfirmasi. | |
| Nama aktor | | Admin | |
| Triger Event | | Admin memilih tombol hapus dari daftar pesanan. | |
| Stakeholder | | Admin | |
| Related *usecase* | | Kelola Pesanan Paket Wisata | |
| Skenario utama | | |  |
| Pre kondisi : Pesanan belum dikonfirmasi. | |  | |
| Aktor | | Reaksi sistem | |
| 1. Admin memilih hapus pesanan | | 2. Sistem menampilkan konfirmasi hapus | |
| 3. admin menekan ”ya” | | 4. Sistem menghapus pesanan. | |
| *Post*-kondisi: Pesanan berhasil dihapus dari sistem. | |  | |
| **Exception** | |  | |
| Pesanan sudah dikonfirmasi  Data pesanan tidak ditemukan | | | |

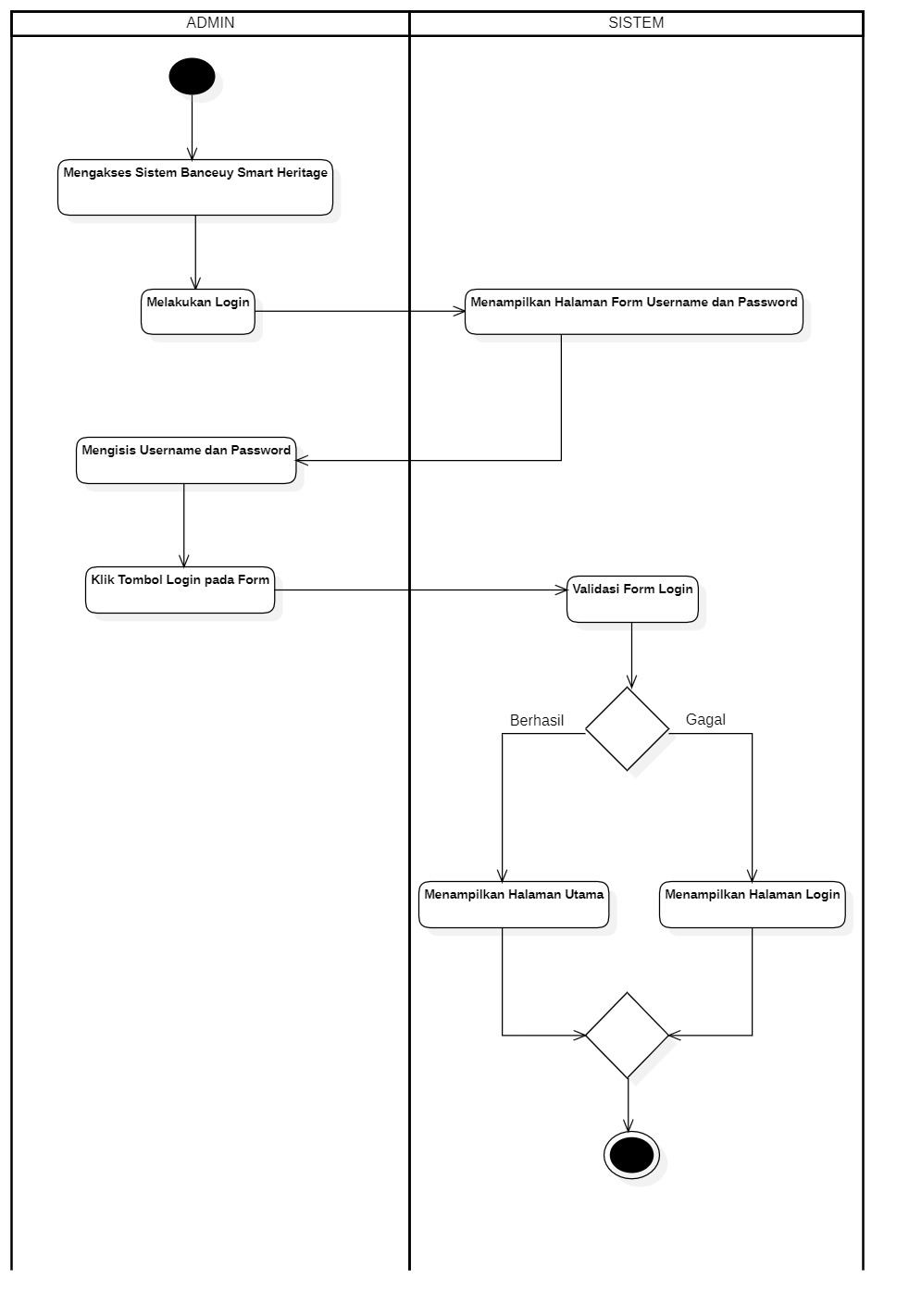
## **4.2 Activity Diagram**

Activity diagram, dalam bahasa Indonesia diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. Activity diagram merupakan pengembangan dari Use Case yang memiliki alur aktivitas.

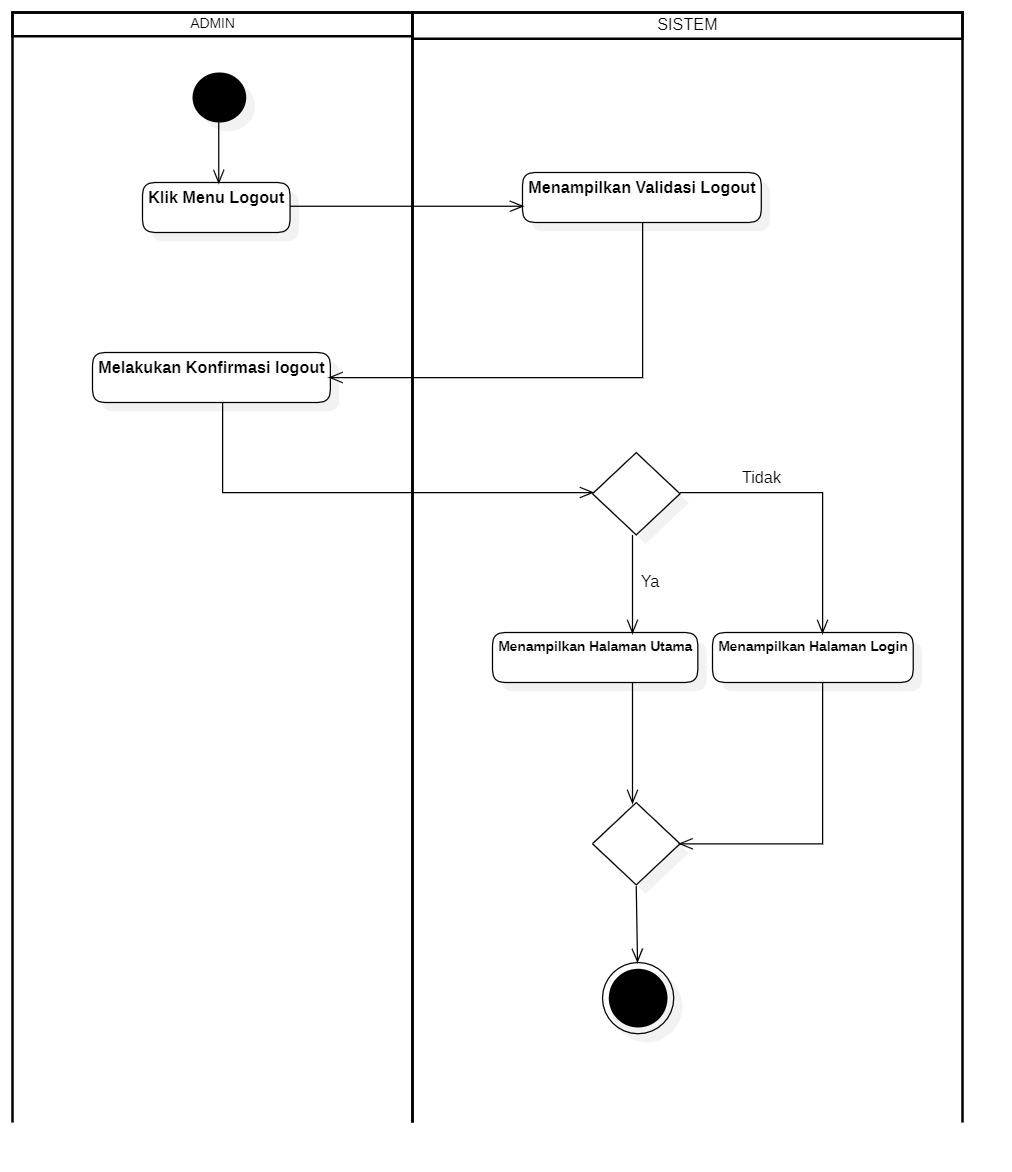
| **No** | ***Use Case*** | ***Activity State*** |
| --- | --- | --- |
|  | Kelola Pesanan Paket Wisata | Use case dimulai saat Admin mengakses menu Kelola Pesanan Paket Wisata. Kelola Pesanan Paket Wisata; Terima Pesanan; Konfirmasi Pesanan. |
|  | **Kelola Data Pengunjung** | Use case dimulai saat Admin mengakses menu Kelola Data Pengunjung. Verifikasi Data Pengunjung; Perbarui Data Pengunjung; Cari Data Pengunjung. |
|  | **Verifikasi Pembayaran** | Use case dimulai saat Midtrans mengakses menu Verifikasi Pembayaran. |
|  | **Kelola User** | Use case dimulai saat Admin mengakses menu Kelola User. Kelola User; Tambah User; Edit User; Hapus User. |
|  | **Login** | Use case dimulai saat Admin atau Pengunjung mengakses sistem dengan akun yang valid. Login. |
|  | **Logout** | Use case dimulai saat Admin atau Pengunjung melakukan logout dari sistem. Logout. |
|  | **Register** | Use case dimulai saat Pengunjung melakukan registrasi akun baru. Register. |
|  | **Kelola Pesanan Homestay** | Use case dimulai saat Pengelola Homestay mengakses menu Kelola Pesanan Homestay. Kelola Pesanan Homestay; Terima Pesanan Homestay; Konfirmasi Pesanan Homestay. |
|  | **Pesan Paket Wisata** | Use case dimulai saat Pengunjung melakukan pemesanan paket wisata. Pesan Paket Wisata. |
|  | **Akses Informasi Wisata** | Use case dimulai saat Pengunjung mengakses informasi seputar wisata dan budaya. Akses Informasi Wisata; Lihat Sejarah; Lihat Makanan Tradisional; Lihat Kesenian; Lihat Kaulinan Barudak; Lihat Budaya; Lihat Paket Wisata; Lihat Homestay. |
|  | **Isi Buku Kunjungan** | Use case dimulai saat Pengunjung mengisi buku kunjungan sebagai catatan pengalaman wisata. Isi Buku Kunjungan. |
|  | **Pesan Homestay** | Use case dimulai saat Pengunjung melakukan pemesanan homestay. Pesan Homestay. |
|  | **Kelola Tanggal Kunjungan** | Use case dimulai saat Admin mengakses menu Kelola Tanggal Kunjungan. Kelola Tanggal Kunjungan; Tentukan Tanggal Kunjungan |
|  | **Hapus User** | Use case dimulai saat Admin memilih untuk menghapus data pengguna. Hapus User. |
|  | **Edit User** | Use case dimulai saat Admin memilih untuk mengedit data pengguna. Edit User. |
|  | **Tambah User** | Use case dimulai saat Admin memilih untuk menambah data pengguna. Tambah User. |
|  | **Konfirmasi Pesanan Homestay** | Use case dimulai saat Pengelola Homestay mengonfirmasi pesanan homestay yang diterima. Konfirmasi Pesanan Homestay. |
|  | **Ketersediaan Homestay** | Use case dimulai saat Pengelola Homestay mengakses menu Mengatur Ketersediaan Homestay. Mengatur Ketersediaan Homestay. |
|  | **Tambah Data** | Use case dimulai saat Admin memilih untuk menambah data pengguna. Tambah Data. |
|  | **Edit Data** | Use case dimulai saat Admin memilih untuk memperbarui data pengguna. Edit Data. |
|  | **Hapus Data** | Use case dimulai saat Admin memilih untuk menghapus data pengguna. Hapus Pengguna. |
|  | **Edit Pesanan** | Use case dimulai saat Admin memilih untuk edit pesanan. Edit Pesanan. |
|  | **Hapus Pesanan** | Use case dimulai saat Admin memilih untuk menghapus pesanan. Hapus Pesanan. |

Activity Diagram

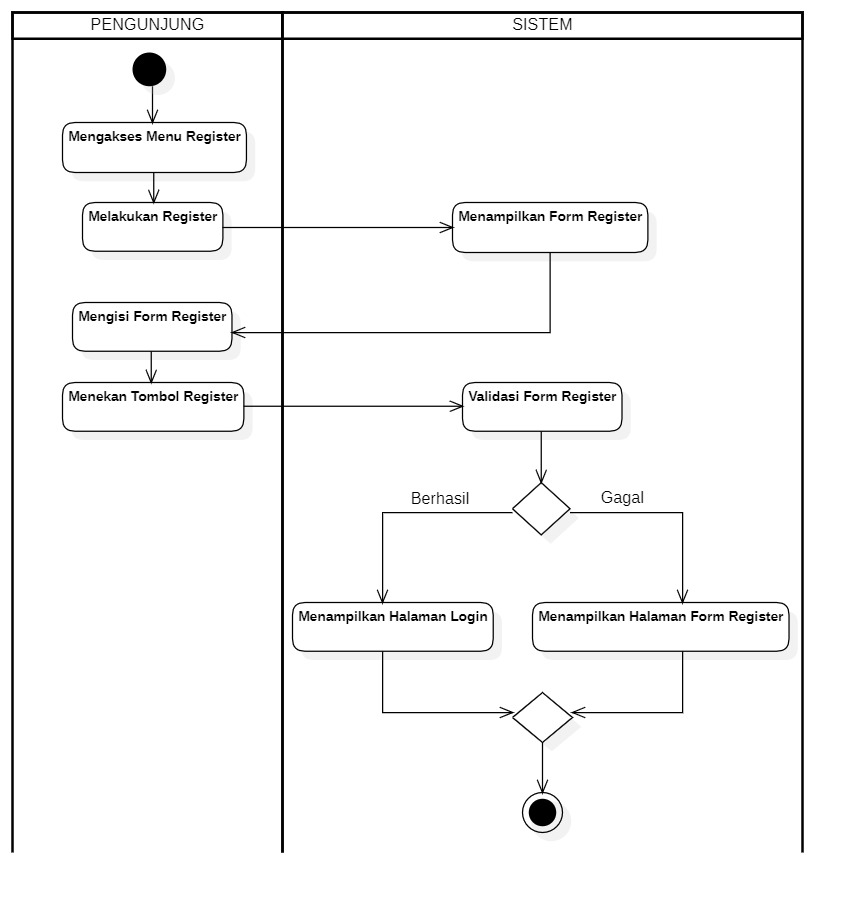
* + 1. Login

****

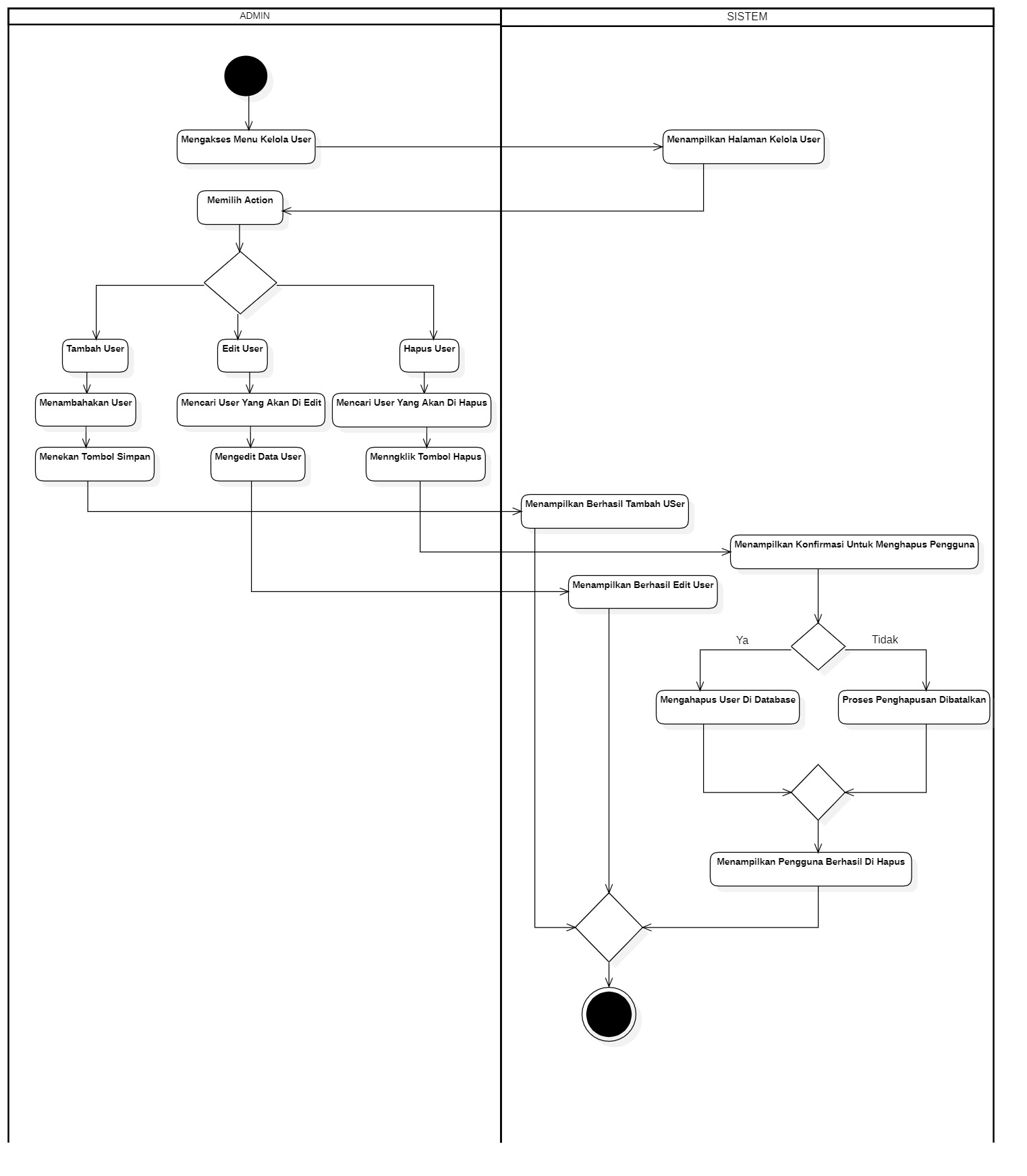
* + 1. Logout

****

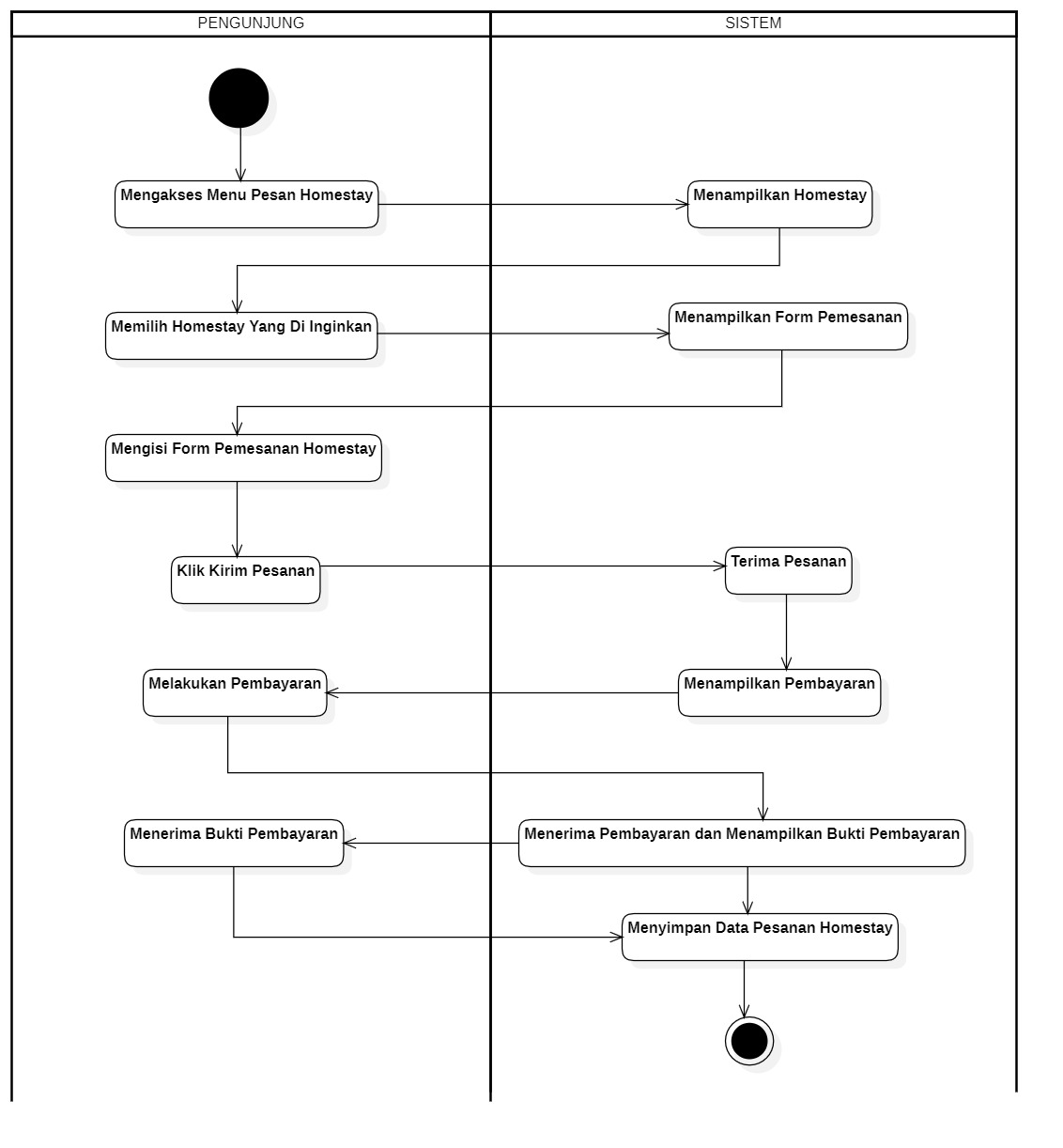
* + 1. Register

****

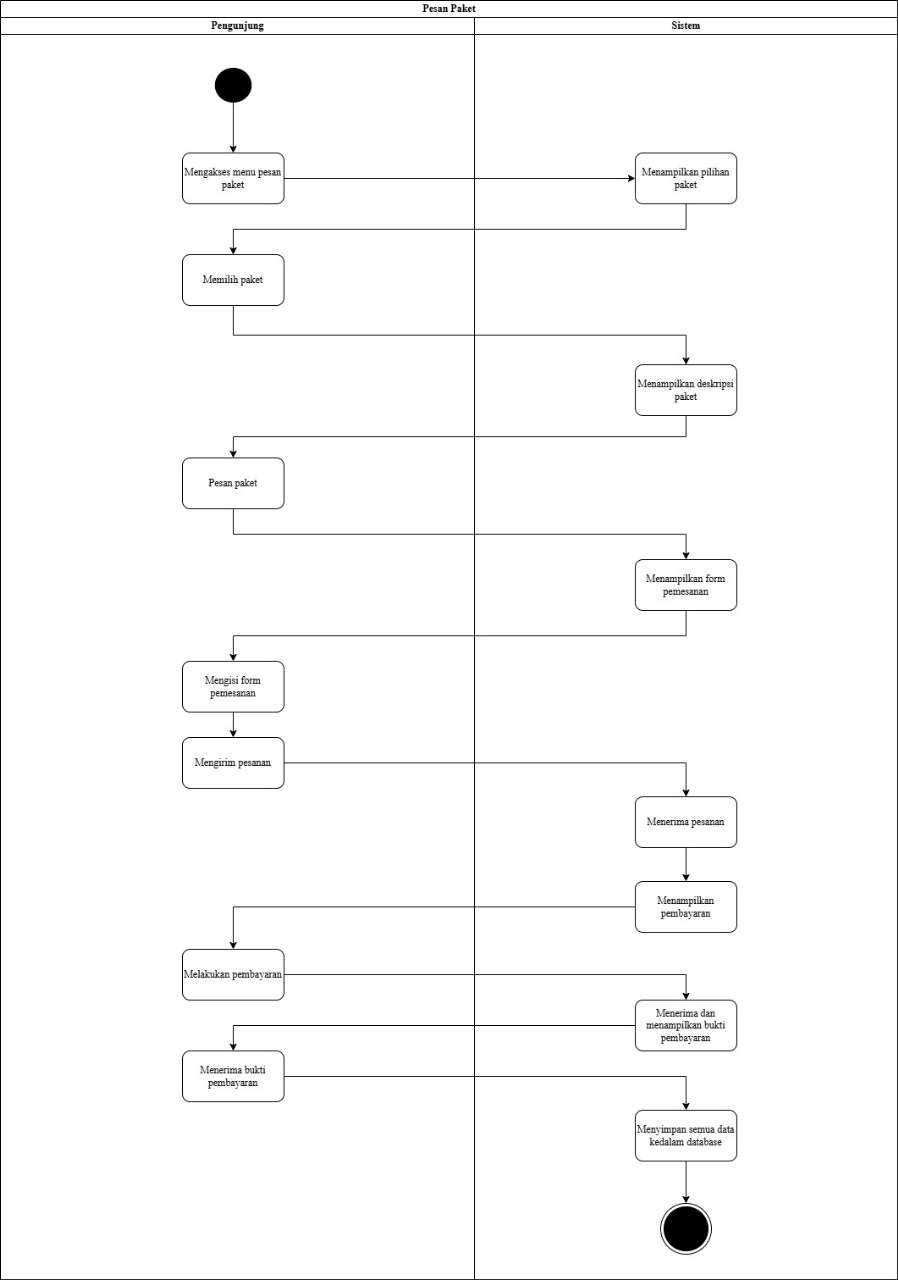
* + 1. Kelola User

****

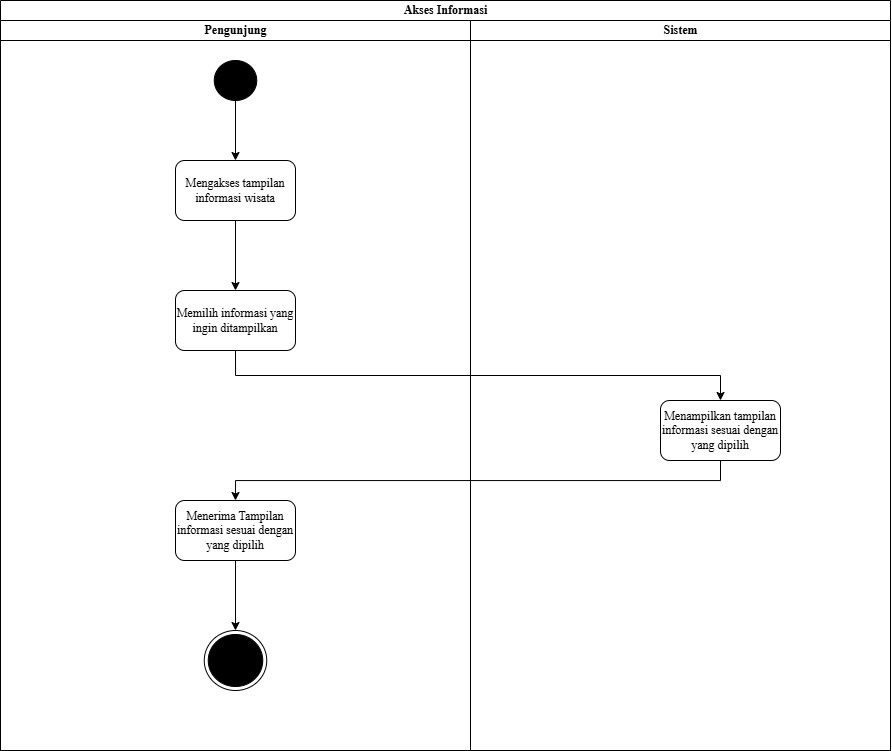
* + 1. Pesan Homestay

****

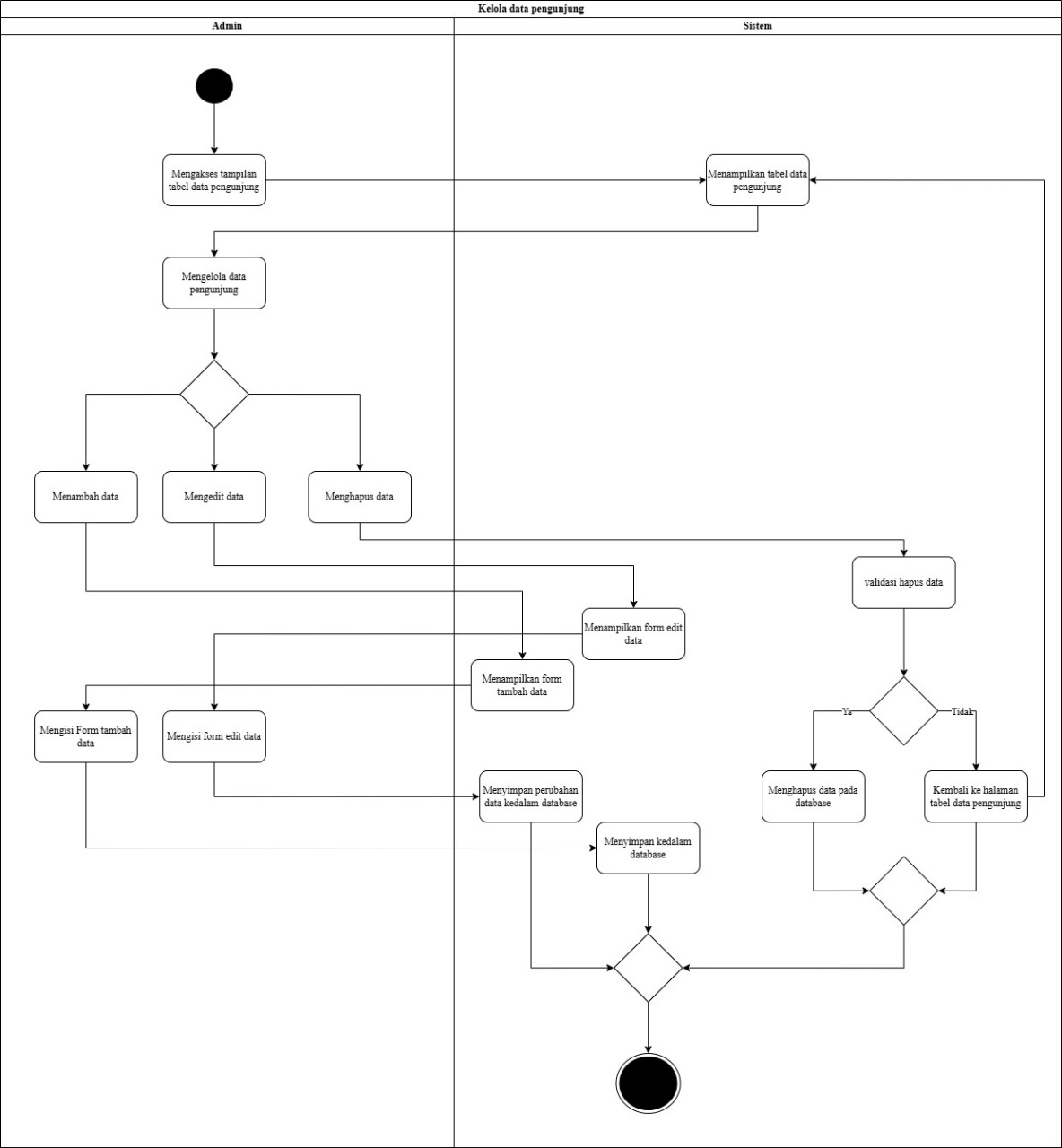
* + 1. Pesan Paket Wisata



* + 1. Akses Informasi

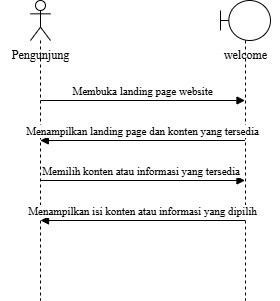


* + 1. Kelola Data Pengunjung

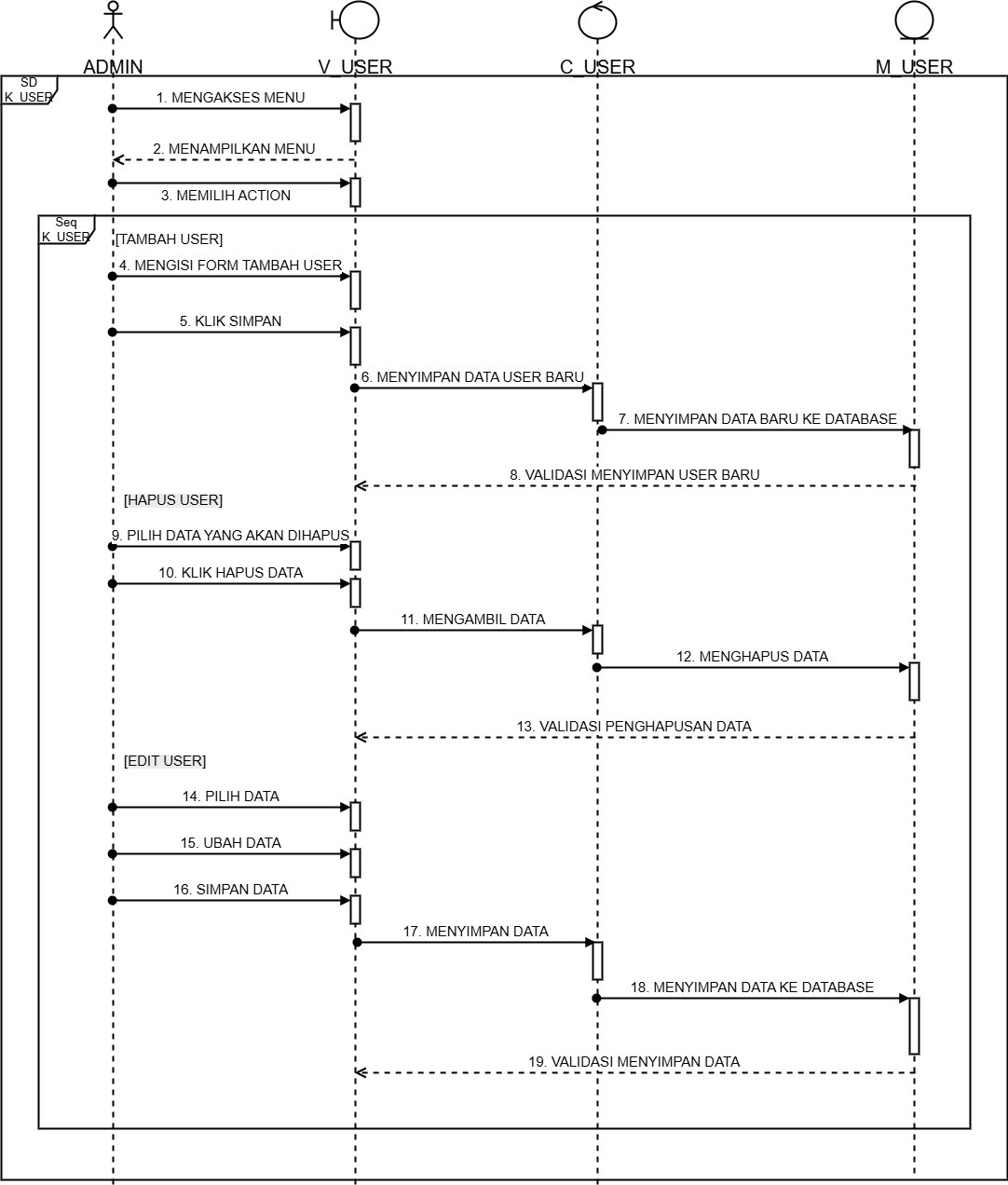
****

Sequence Diagram

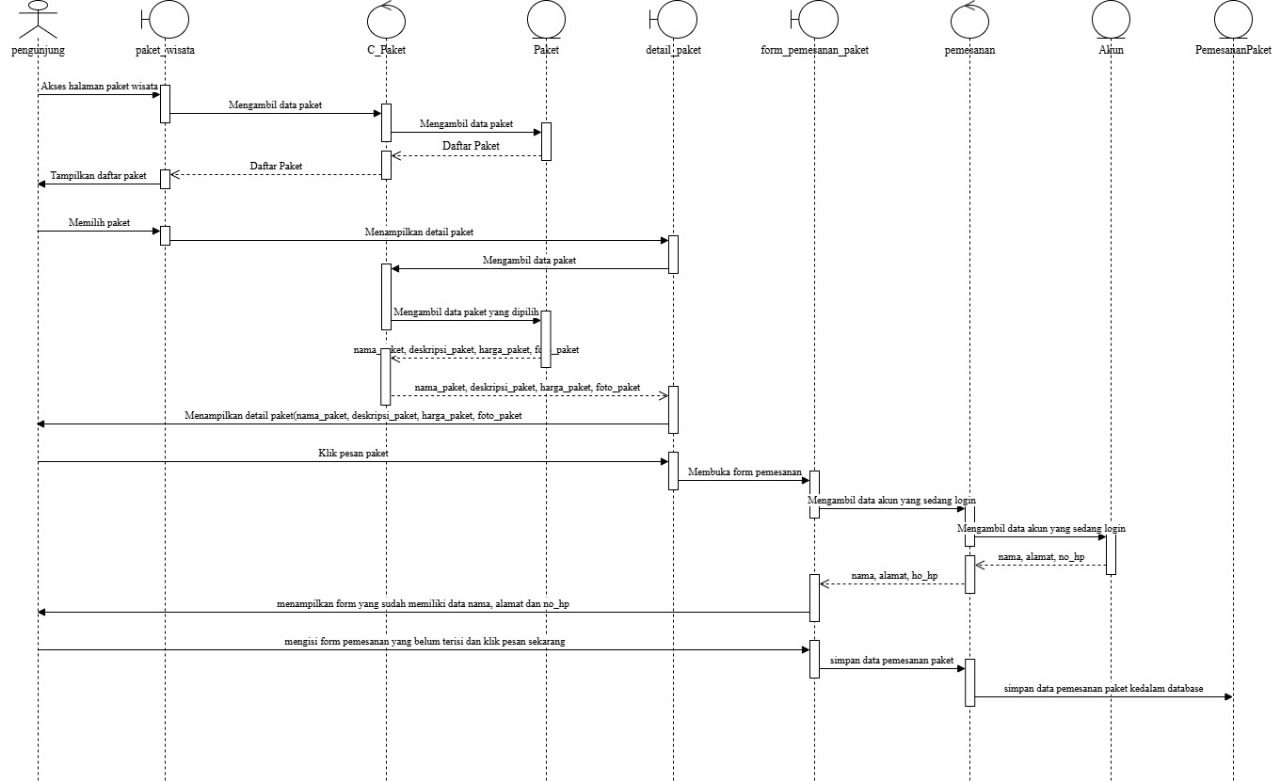
* + - 1. Login
      2. Logout
      3. Register
      4. Akses Informasi



* + - 1. Kelola User



* + - 1. Pemesanan Paket



* + - 1. Isi Buku Kunjungan

