GOTOBA

VISIT DANAU TOBA INFORMATION WEBSITES

Blibli ***FUTURE*** Program Batch 4.0

Fase 2 – Medan



Oleh:

**Sogumontar Hendra Simangunsong**

**Syntia**

# Daftar Isi

[1. Pendahuluan 4](#_Toc32348548)

[1.1. Tujuan Pembuatan Software 4](#_Toc32348549)

[1.2. Definisi 4](#_Toc32348550)

[1.3. Referensi 5](#_Toc32348551)

[1.4. Sistematika 5](#_Toc32348552)

[2. Deskripsi Umum 6](#_Toc32348553)

[2.1. Perspektif 6](#_Toc32348555)

[2.2. Kegunaan 6](#_Toc32348556)

[2.3. Karakteristik Pengguna 7](#_Toc32348557)

[2.4. Batasan-Batasan 7](#_Toc32348558)

[2.5. Asumsi dan Ketergantungan 8](#_Toc32348559)

[3. Spesifikasi Kebutuhan 9](#_Toc32348560)

[3.1. Kebutuhan Fungsional 9](#_Toc32348562)

[3.1.1. Register 10](#_Toc32348563)

[3.1.2. Log in 11](#_Toc32348564)g

[3.1.3. Mengelola Restoran 12](#_Toc32348565)

[3.1.4. Mengelola Restoran 13](#_Toc32348566)

[3.1.5. Mengelola Restoran 14](#_Toc32348567)

[3.1.6. Mengelola Tempat Wisata 16](#_Toc32348568)

[3.1.7. Mengelola Tempat Wisata 17](#_Toc32348569)

[3.1.8. Mengelola Tempat Wisata 18](#_Toc32348570)

[3.1.9. Mengelola Tempat Wisata 19](#_Toc32348571)

[3.1.10. Mendaftarkan Restoran 20](#_Toc32348572)

[3.1.11. Membeli Tiket Wisata 21](#_Toc32348573)

[3.1.12. Mengelola Jadwal Liburan 23](#_Toc32348574)

[3.1.13. Block Account 24](#_Toc32348575)

[3.2. Kebutuhan Antarmuka Eksternal 26](#_Toc32348576)

[3.2.1. Kebutuhan Antarmuka Pengguna 26](#_Toc32348577)

[3.2.2. Kebutuhan Antarmuka Perangkat Keras 26](#_Toc32348578)

[3.2.3. Kebutuhan Antarmuka Perangkat Lunak 26](#_Toc32348579)

[3.2.4. Kebutuhan Antarmuka Komunikasi 26](#_Toc32348580)

[3.3. Kebutuhan Non-Fungsional 26](#_Toc32348581)

[3.3.1. Struktur Logikal Data 26](#_Toc32348582)

REQUIREMENT DOCUMENT

# Pendahuluan

## Tujuan Pembuatan Software

Tujuan dari pembuatan applikasi ini adalah untuk memperkenalkan tempat wisata yang ada di daerah danau Toba, dan untuk mempromosikan restoran yang ada di sekitar danau Toba, dan juga untuk melakukan promosi daerah danau Toba kepada masyarakat luar, baik dalam negeri maupun luar negeri.

Saat ini pemerintah Indonesia telah menetapkan danau Toba sebagai salah satu destinasi prioritas di Indonesia, dan pemerintah daerah Tobasa sedang melakukan perbaikan dan pembangunan di daerah danau Toba, dan juga banyak dari masyarakat sekitar telah menjadikan danau Toba sebagai mata pencaharian mereka dengan membuka restoran di sekitaran danau Toba .

Dengan adanya aplikasi ini, diharapkan dapat membantu masyarakat di sekitar danau toba untuk mengembangkan usaha mereka, dan juga untuk mempermudah para wisatawan dalam mencari informasi terkait tempat-tempat wisata, dan restoran yang dapat mereka singgahi ketika berkunjung ke daerah danau Toba.

## Definisi

Beberapa definisi kata yang akan sering dipakai :

**Admin** Orang yang dapat mengedit, menambah, atau menghapus data pada sistem; mengakses halaman master.

**Block** Proses memblokir suatu akun; mencabut hak akses dari suatu akun.

**Cart** Keranjang; menyimpan daftar order User.

**Catalog** Daftar koleksi sebuah / beberapa pusat dokumentasi yang disusun menurut sistem tertentu.

**Categories** Kategori; mengelompokkan produk berdasarkan tipe/kategori nya.

**Data** Informasi yang berupa fakta; deskripsi dari suatu kejadian yang dihadapi.

**Database** Sistem yang menyimpan semua informasi yang dimonitor oleh sistem ini.

**ERD** *Entity Relationship Diagram*; Diagram yang menunjukkan

**Home** Halaman utama yang ditampilkan sistem.

**Log in** Mengakses sistem dengan memasukkan identitas dari akun pengguna dan kata sandi untuk mendapatkan hak akses akun.

**Log out** Keluar dari akun yang sedang aktif pada sistem.

**Master Data** Data utama yang harus dibuat dengan benar supaya transaksi bisa dilakukan

**Merchant** Orang yang dapat menambah produk dan menjual produk dalam web.

**Profile** Berisi informasi User.

**Order** Melakukan pemesanan terhadap suatu produk.

**Rating** Memberi penilaian kuantitatif terhadap suatu produk.

**Review** Memberi penilaian dan/atau ulasan terhadap suatu produk.

**Search** Melakukan pencarian produk berdasarkan judul / kata kunci.

**SRS** *Software Requirement Specification*; Suatu dokumen yang mendeskripsikan semua fungsi dari sistem yang diusulkan dan kendala yang terjadi ketika sistem beroperasi, seperti dokumen ini.

**User** Orang yang mengunjungi / menggunakan web sebagai seorang konsumen.

## Referensi

Adapun referensi yang digunakan dalam membantu proses pembuatan dokumen SRS ini :

* [*http://cheesterzone.blogspot.com/2011/04/dokumen-srs-sistem-informasi.html*](http://cheesterzone.blogspot.com/2011/04/dokumen-srs-sistem-informasi.html)
* [*https://hasrulbakri.wordpress.com/2011/05/23/contoh-dokumen-srs-software-requirement-spesification/*](https://hasrulbakri.wordpress.com/2011/05/23/contoh-dokumen-srs-software-requirement-spesification/)
* http://www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc

## Sistematika

Dokumen SRS ini dibagi menjadi tiga bagian utama, yaitu :

**Pendahuluan**, berisi penjelasan tentang tujuan pembuatan dan lingkup masalah yang dipengaruhi oleh web yang dikembangkan, definisi, referensi dan sistematika.

**Deskripsi Umum**, berisi penjelasan secara umum mengenai web yang akan dikembangkan, meliputi perspektif deskripsi umum sistem,karakteristik penggauna, batasan-batasan dan asumsi penggunaan.

**Spesifikasi Kebutuhan**, berisi uraian kebutuhan web secara lebih rinci.

# Deskripsi Umum



## Perspektif

Produk yang berupa situs web ditujukan kepada para pengguna (konsumen, admin, dan para merchant. Semua pengguna (baik pengguna terdaftar maupun anonim) dapat mengakses situs web ini tanpa melakukan proses login terlebih dahulu.

Pengguna tidak diharuskan mendaftar jika pengguna hanya berkunjung ke web. Namun untuk dapat menggunakan semua fungsi yang disediakan, pengguna diminta untuk mendaftar melalui antarmuka web. Ketika pengguna baru mendaftar, semua data yang diminta akan disimpan dalam database dan sebuah workspace akan diberikan pada pengguna. Nantinya, pengguna dapat melakukan login ke / logout dari sistem kapanpun.

Dari sudut pandang pengguna sebagai konsumen, pengguna diharapkan dapat melihat dan mencari tempat wisata dan restoran yang terdaftar berdasarkan kategori yang tersedia, membeli tiket masuk tempat wisata yang terdaftar, menyimpan jadwal perjalanan yang akan dilakukan beserta tempat wisata yang akan dituju, membeli buku, dan melakukan review terhadap tempat wisata / restoran yang pernah dikunjungi.

Sementara dari sudut pandang pengguna sebagai penjual, pengguna diharapkan dapat mendaftarkan, mengubah maupun menghapus tempat wisata, menambahkan restoran, mengubah maupun menghapus serta mengelola daftar makanan yang ada di dalam restoran tersebut, melihat halaman tempat wisata, restoran masing-masing, serta melihat daftar pengguna yang memesan tiket masuk wisata dari tempat wisata mereka.

Untuk *Admin*, dapat melakukan proses blok terhadap akun yang melakukan pelanggaran, dan juga mendaftarkan tempat wisata sekitaran daerah danau Toba.

## Kegunaan

User dan Merchant dapat melakukan transaksi pembelian tiket masuk tempat wisata, dan pembelian makanan di restoran secara online.

Fungsi-fungsi dari produk adalah :

|  |  |  |
| --- | --- | --- |
| User | Merchant | Admin |
| Login  Register  Profile  Jadwal Liburan  Tempat Wisata  Purchase and Order List  Restoran  Tour Guide Review Search | Login  Register  Storefront Page  Product  Incoming Order | Login  Master Page  Block User |

## Karakteristik Pengguna

Berikut adalah 3 jenis pengguna beserta karakteristiknya yang akan menggunakanproduk.

|  |  |
| --- | --- |
| **Pengguna** | **Karakteristik** |
| User | User dapat melihat dan mencari tempat wisata, restoran dan tour guide yang tersedia, memesan tour guide, membeli tiket tempat wisata, dan memesan makanan di restoran melalui pembayaran online, dan melakukan penjadwalan liburan beserta tempat wisata tujuan. |
| Merchant | Merchant dapat menambah, mengubah dan menghapus restoran, tempat wisata, mengelola makanan di restoran menampilkan halaman profil, dan menerima orderan yang datang. |
| Admin | Admin dapat mengubah data, mengakses halaman master, serta dapat melakukan *block* pada User maupun Merchant. |

## Batasan-Batasan

Beberapa batasan dalam menggunakan web ini :

* Sistem hanya dapat digunakan pada perangkat dengan *browser* yang mendukung JavaScript, seperti Chrome vers. 51, Firefox vers. 54, Edge vers. 14, Safari vers. 10, dan Opera vers. 38.
* User yang dapat melihat / mengubah jadwal liburan, melihat peesanan tiket dan makanan di restoran, melakukan pembayaran, dan memberi ulasan pada tempat wisata adalah User yang sudah terdaftar.
* Merchant yang sudah terdaftar hanya dapat mendaftarkan tempat wisata, dan restoran, dan menyediakan jasa tour guide.
* Admin yang sudah terdaftar hanya dapat melihat master page dan memblokir pengguna.

## Asumsi dan Ketergantungan

Asumsi dan ketergantungan dalam menggunakan web ini :

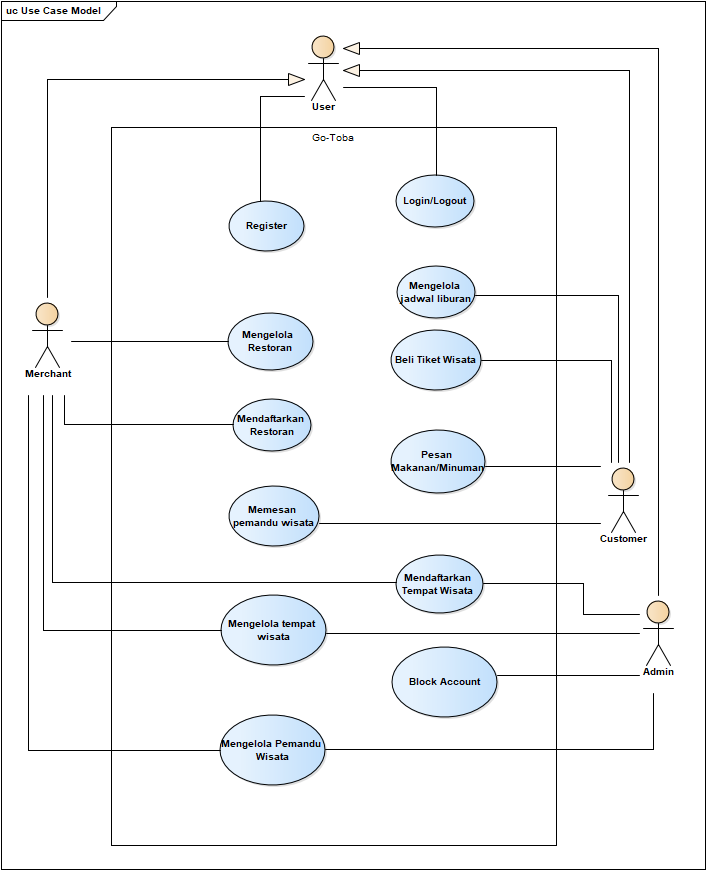
* Pengguna sudah mengerti dalam memakai layanan *e-commerce*
* Semua pembayaran yang dilakukan secara otomatis akan berhasil

# Spesifikasi Kebutuhan



## Kebutuhan Fungsional

Berikut adalah Use Case Diagram .



Gambar Use Case Diagram

Berikut deskripsi dari Use Case Diagram di atas.

### Register

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_001 | |
| Use Case Name | Register | |
| Brief Description | *Use case* ini menjelaskan bagaimana *customer / merchant* melakukan pendaftaran akun agar dapat menggunakan website ini. | |
| Primary Actor | *Customer/Merchant* | |
| Secondary Actor | - | |
| Precondition | Mengakses link website. | |
| Post Condition | *Customer/Merchant* dapat mengakses website(*login*). | |
| Included Use Case | *-* | |
| Basic Flow Event | Actor | System |
|  | 1. *Customer/Merchant* mengakses website |  |
|  | 2.Menampilkan halaman website. |
| 3. Memilih tombol register. |  |
|  | 4. Menampilkan *form* register. |
| 5. Mengisi *form register,* dan menekan tombol daftar. |  |
|  | 6. Menyimpan data akun ke *database*. |
| 8. Mengarahkan *Customer/ Merchant* *ke* halaman *login*. |  |
| Alternative Flow of Events | - | |

### Log in

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_002 | |
| Use Case Name | *Login* | |
| Brief Description | *Use case* ini menjelaskan bagaimana *user* masuk ke dalam website dengan memasukkan username dan password. | |
| Primary Actor | *Customer, Merchant, Admin* | |
| Secondary Actor | - | |
| Precondition | Register. | |
| Post Condition | Mengakses fitur-fitur yang disediakan. | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. *User* mengakses website. |  |
|  | 2.Menampilkan halaman website. |
| 3. Memilih tombol login. |  |
|  | 4. Menampilkan *form* login. |
| 5. Mengisi *form login ,*  dan menekan tombol login. |  |
|  | 6. Melakukan pengecekan ke dalam DB. |
| 8. Mengarahkan *User*  kehalaman *homepage*. |  |
| Alternative Flow of Events | Kembali ke halaman login dengan notifikasi “Username atau password salah”. | |

### Mengelola Restoran

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_003 | |
| Use Case Name | Mengelola Restoran | |
| Use Case Scenario Name | Menambahkan Makanan/ Minuman | |
| Brief Description | *Use case* ini menjelaskan bagaimana *merchant* menambahkan makanan/minuman ke dalam sistem. | |
| Primary Actor | *Merchant* | |
| Secondary Actor | - | |
| Precondition | *Merchant* harus sudah mendaftarkan restorannya. | |
| Post Condition | Data makanan/minuman merchant bertambah | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. *Merchant* melakukan *login.* |  |
|  | 2.Menampilkan halaman website. |
| 3. Memilih fitur menu restoran. |  |
|  | 4. Menampilkanhalaman restoran. |
| 5. Memilih tombol tambah. |  |
|  | 6. Menampilkan form tambah makanan/minuman. |
| 8. Mengisi form tambah makanan/minuman, dan menekan tombol simpan. |  |
|  |  | 9. Menyimpan data makanan/minuman yang telah di input merchant. Dan mengarahkan merchant ke halaman restoran. |
| Alternative Flow of Events | - | |

### Mengelola Restoran

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_003 | |
| Use Case Name | Mengelola Restoran | |
| Use Case Scenario Name | Update Makanan/ Minuman | |
| Brief Description | *Use case* ini menjelaskan bagaimana *merchant* mengubah data makanan/minuman di dalam sistem. | |
| Primary Actor | *Merchant* | |
| Secondary Actor | - | |
| Precondition | *Merchant* harus sudah mendaftarkan restorannya. | |
| Post Condition | Data makanan/minuman merchant berubah | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. *Merchant* melakukan *login.* |  |
|  | 2.Menampilkan halaman website. |
| 3. Memilih fitur menu restoran. |  |
|  | 4. Menampilkanhalaman restoran. |
| 5. Memilih tombol update. |  |
|  | 6. Menampilkan form update makanan/minuman. |
| 8. Mengubah data makanan/minuman, dan menekan tombol simpan. |  |
|  |  | 9. Menyimpan data makanan/minuman yang telah di ubah merchant. Dan mengarahkan merchant ke halaman restoran. |
| Alternative Flow of Events | - | |

### Mengelola Restoran

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_003 | |
| Use Case Name | Mengelola Restoran | |
| Use Case Scenario Name | Menghapus Makanan/ Minuman | |
| Brief Description | *Use case* ini menjelaskan bagaimana *merchant* menghapus makanan/minuman dari dalam sistem. | |
| Primary Actor | *Merchant* | |
| Secondary Actor | - | |
| Precondition | *Merchant*  harus sudah mendaftarkan restorannya. | |
| Post Condition | Data makanan/minuman merchant di hapus. | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. *Merchant* melakukan *login.* |  |
|  | 2.Menampilkan halaman website. |
| 3. Memilih fitur menu restoran. |  |
|  | 4. Menampilkanhalaman restoran. |
| 5. Memilih tombol hapus. |  |
|  | 6. Menampilkan notifikasi persetujuan untuk melanjutkan aksi. |
| 8. Menekan tombol ya. |  |
|  |  | 9. Menghapus data makanan/minuman. |
| Alternative Flow of Events | - | |

### Mendaftarkan Restoran

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_004 | |
| Use Case Name | Mendaftarkan Restoran | |
| Use Case Scenario Name | - | |
| Brief Description | *Use case* ini menjelaskan bagaimana *merchant* mendaftarkan restoran ke dalam sistem. | |
| Primary Actor | *Merchant* | |
| Secondary Actor | - | |
| Precondition | *Merchant* harus sudah login. | |
| Post Condition | Data restoran bertambah. | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. *Merchant* melakukan *login.* |  |
|  | 2.Menampilkan halaman website. |
| 3. Memilih fitur menu restoran. |  |
|  | 4. Menampilkanhalaman restoran. |
| 5. Memilih tombol tambah. |  |
|  | 6. Menampilkan form restoran. |
| 8. Mengisi data restoran, dan menekan tombol simpan. |  |
|  |  | 9. Menyimpan data retoran. Dan mengarahkan merchant ke halaman utama. |
| Alternative Flow of Events | - | |

### Mendaftarkan Tempat Wisata

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_005 | |
| Use Case Name | Mendaftarkan Tempat Wisata | |
| Use Case Scenario Name | - | |
| Brief Description | *Use case* ini menjelaskan bagaimana *merchant* mendaftarkan tempat wisata ke dalam sistem. | |
| Primary Actor | *Merchant* | |
| Secondary Actor | - | |
| Precondition | *Merchant* harus sudah login. | |
| Post Condition | Data tempat wisata bertambah. | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. *Merchant* melakukan *login.* |  |
|  | 2.Menampilkan halaman website. |
| 3. Memilih fitur menu tempat wisata. |  |
|  | 4. Menampilkanhalaman tempat wisata. |
| 5. Memilih tombol tambah. |  |
|  | 6. Menampilkan form tempat wisata. |
| 8. Mengisi data tempat wisata, dan menekan tombol simpan. |  |
|  |  | 9. Menyimpan data tempat wisata. Dan mengarahkan merchant ke halaman utama. |
| Alternative Flow of Events | - | |

### Mengelola Tempat Wisata

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_006 | |
| Use Case Name | Mengelola Tempat Wisata | |
| Use Case Scenario Name | Menghapus Tempat Wisata | |
| Brief Description | *Use case* ini menjelaskan bagaimana *merchant* menghapus tempat wisata dari dalam sistem. | |
| Primary Actor | *Merchant*, *Admin* | |
| Secondary Actor | - | |
| Precondition | *Admin, Merchant*  harus sudah mendaftarkan tempat wisatanya. | |
| Post Condition | Data tempat wisata merchant di hapus. | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. Admin, *Merchant* melakukan *login.* |  |
|  | 2.Menampilkan halaman website. |
| 3. Memilih fitur menu tempat wisata. |  |
|  | 4. Menampilkanhalaman tempat wisata. |
| 5. Memilih tombol hapus. |  |
|  | 6. Menampilkan notifikasi persetujuan untuk melanjutkan aksi. |
| 8. Menekan tombol ya. |  |
|  |  | 9. Menghapus data tempat wisata. |
| Alternative Flow of Events | - | |

### Mengelola Tempat Wisata

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_006 | |
| Use Case Name | Mengelola Tempat Wisata | |
| Use Case Scenario Name | Update data tempat wisata | |
| Brief Description | *Use case* ini menjelaskan bagaimana *merchant* melakukan update data tempat wisata dari dalam sistem. | |
| Primary Actor | Admin, *Merchant* | |
| Secondary Actor | - | |
| Precondition | Admin, *Merchant*  harus sudah mendaftarkan tempat wisatanya. | |
| Post Condition | Data tempat wisata merchant di update. | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. Admin, *Merchant* melakukan *login.* |  |
|  | 2.Menampilkan halaman website. |
| 3. Memilih fitur menu tempat wisata. |  |
|  | 4. Menampilkanhalaman tempat wisata. |
| 5. Memilih tombol update. |  |
|  | 6. Menampilkan form tempat wisata. |
| 8. Mengisi form dan menekan tombol simpan. |  |
|  |  | 9. Menyimpan perubahan data dan mengarahkan merchant ke halaman tempat wisata. |
| Alternative Flow of Events | - | |

### Mengelola Tempat Wisata

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_006 | |
| Use Case Name | Mengelola Tempat Wisata | |
| Use Case Scenario Name | Menambahkan tempat wisata | |
| Brief Description | *Use case* ini menjelaskan bagaimana *merchant* menambahkan tempat wisata yang baru. | |
| Primary Actor | Admin, *Merchant* | |
| Secondary Actor | - | |
| Precondition | Admin, *Merchant*  harus sudah login. | |
| Post Condition | Data tempat wisata merchant bertambah. | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. Admin, *Merchant* melakukan *login.* |  |
|  | 2.Menampilkan halaman website. |
| 3. Memilih fitur menu tambah tempat wisata. |  |
|  | 4. Menampilkanform tambah tempat wisata. |
| 5. Mengisi form tempat wisata dan menekan tombol simpan. |  |
|  | 6. Menyimpan data tempat wisata ke *database*, dan mengarahkan merchant ke halaman tempat wisata. |
| Alternative Flow of Events | - | |

### Mengelola Tempat Wisata

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_006 | |
| Use Case Name | Mendaftarkan Tempat Wisata. | |
| Brief Description | Use case ini menjelaskan bagaimana merchant mendaftarkan tempat wisata mereka . | |
| Primary Actor | *Admin, Merchant* | |
| Secondary Actor | - | |
| Precondition | Admin, *Merchant* melakukan login. | |
| Post Condition | Merchant dapat menambahkan tempat wisata ke dalam system. | |
| Included Use Case | Login | |
| Basic Flow Event | Actor | System |
|  | 1. Admin, *Merchant* melakukan login. |  |
|  | 2.Menampilkan halaman website. |
| 3. Memilih fitur tambah tempat wisata. |  |
|  | 4. Menampilkan form tambah tempat wisata. |
| 8. Mengisi form, dan menekan tombol simpan. |  |
|  |  | 9. Menyimpan data tempat wisata yang telah ditambahkan merchant. Dan mengarahkan merchant ke daftar tempat wisata yang dimilikinya. |
| Alternative Flow of Events | - | |

### Membeli Tiket Wisata

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_007 | |
| Use Case Name | Membeli Tiket Wisata | |
| Brief Description | *Use case* ini menjelaskan bagaimana *customer* membeli tiket wisata. | |
| Primary Actor | *Customer* | |
| Secondary Actor | - | |
| Precondition | *Customer*  harus sudah login. | |
| Post Condition | *Customer* dapat membeli tiket wisata. | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. *Customer* melakukan *login.* |  |
|  | 2.Menampilkan halaman website. |
| 3. Memilih fitur menu tempat wisata. |  |
|  | 4. Menampilkanseluruh tempat wisata yang terdaftar. |
| 5. Memilih tempat wisata yang ingin di kunjungi, dan menekan tombol detail. |  |
|  | 6. Menampilkan detail dari tempat wisata yang di pilih customer. |
| 8. Menekan tombol pesan tiket. |  |
|  |  | 9. melakukan pemesanan tiket, dan mengarahkan customer ke daftar tiket yang di miliki. |
|  | 10. *Customer*  melakukan scanning QR code untuk masuk ke tempat wisata |  |
| Alternative Flow of Events | - | |

### Mengelola Jadwal Liburan

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_006 | |
| Use Case Name | Jadwal Liburan | |
| Brief Description | *Use case* ini menjelaskan bagaimana *Customer* melakukan penjadwalan liburan dengan melakukan drag and drop ke kalender yang disediakan oleh sistem. | |
| Primary Actor | *Customer* | |
| Secondary Actor | - | |
| Precondition | *Customer*  harus sudah *login*. | |
| Post Condition | *Customer* mendapatkan notifikasi sesuai penjadwalan yang telah ditentukan. | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. *Customer* melakukan *login.* |  |
|  | 2.Menampilkan halaman website. |
| 3. Memilih fitur Jadwal Liburan. |  |
|  | 4. Menampilkanhalaman jadwal, yang berisi kalender, dan tempat wisata. |
| 5. Drag tempat wisata ke salah satu tanggal di kalender. |  |
|  | 6. Menyimpan jadwal sesuai tanggal yang telah di tentukan dan tempat wisata yang akan dikunjungi. |
|  | 7. Sistem akan memberikan notifikasi ketika sudah hari H. |
| Alternative Flow of Events | - | |

### Block Account

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_007 | |
| Use Case Name | Block Account | |
| Brief Description | *Use case* ini menjelaskan bagaimana *Admin*  melakukan blocking terhadap akun yang melakukan pelanggaran. | |
| Primary Actor | *Admin.* | |
| Secondary Actor | - | |
| Precondition | *Admin*  harus sudah *Login*. | |
| Post Condition | *Admin* dapat block atau unblock akun *user / merchant*. | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. *Admin* melakukan *login.* |  |
|  | 2.Menampilkan halaman *admin*. |
| 3. Memilih fitur account. |  |
|  | 4. Menampilkandaftar akun. |
| 5. Memilih akun yang akan di block, dan menekan tombol block. |  |
|  | 6. mengganti status akun di database, agar akun tersebut tidak bias melakukan login ke sistem. |
| Alternative Flow of Events | - | |

### Memesan Pemandu Wisata

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_008 | |
| Use Case Name | Memesan Pemandu Wisata | |
| Brief Description | *Use case* ini menjelaskan bagaimana *Customer*  melakukan pemesanan terhadap pemandu wisata. | |
| Primary Actor | *Customer.* | |
| Secondary Actor | - | |
| Precondition | *Customer*  harus sudah *Login*. | |
| Post Condition | *Customer* dapat menggunakan jasa pemandu wisata. | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. *Customer* melakukan *login.* |  |
|  | 2.Menampilkan halaman utama. |
| 3. Memilih fitur pemandu wisata. |  |
|  | 4. Menampilkandaftar pemandu wisata yang terdaftar di dalam sistem. |
| 5. Memilih pemandu wisata yang akan di pesan, dan menekan tombol pesan. |  |
|  | 6. menampilkan form untuk pemesanan. |
|  | 7. *Customer* mengisi form pemesanan dan menekan tombol pesan. |  |
|  |  | 8. Menyimpan data pemesanan pemandu wisata. |
| Alternative Flow of Events | - | |

### Memesan Makanan/Minuman

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_009 | |
| Use Case Name | Memesan Makanan/Minuman | |
| Brief Description | *Use case* ini menjelaskan bagaimana *Customer*  melakukan pemesanan terhadap makanan/minuman yang terdaftar di dalam sistem. | |
| Primary Actor | *Customer.* | |
| Secondary Actor | - | |
| Precondition | *Customer*  harus sudah *Login*. | |
| Post Condition | *Customer* dapat menikmati makanan/minuman yang dipesan. | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. *Customer* melakukan *login.* |  |
|  | 2.Menampilkan halaman utama. |
| 3. Memilih fitur restoran. |  |
|  | 4. Menampilkandaftar makanan/minuman yang terdaftar di dalam restoran masing-masing. |
| 5. Memilih makanan yang akan di pesan, dan menekan tombol pesan. |  |
|  | 6. menampilkan form untuk pemesanan. |
|  | 7. *Customer* mengisi form pemesanan dan menekan tombol pesan. |  |
|  |  | 8. Menyimpan data pemesanan makanan/minuman. |
| Alternative Flow of Events | - | |

### Mendaftarkan Pemandu Wisata

|  |  |  |
| --- | --- | --- |
| Use Case Id | UC\_010 | |
| Use Case Name | Mendaftarkan Pemandu Wisata | |
| Brief Description | *Use case* ini menjelaskan bagaimana *Merchant/Admin*  mendaftarkan pemandu wisata. | |
| Primary Actor | *Customer, Admin.* | |
| Secondary Actor | - | |
| Precondition | *Merchant/Admin* harus sudah *Login*. | |
| Post Condition | *Merchant/Admin* berhasil mendaftarkan pemandu wisata. | |
| Included Use Case | *Login* | |
| Basic Flow Event | Actor | System |
|  | 1. *Merchant/Admin* melakukan *login.* |  |
|  | 2.Menampilkan halaman utama. |
| 3. Merchant/Admin Memilih fitur pemandu wisata. |  |
|  | 4. Menampilkandaftar pemandu wisata yang terdaftar di dalam sistem. |
| 5. Memilih tombol tambah pemandu wisata. |  |
|  | 6. menampilkan form untuk pennambahan pemandu wisata. |
|  | 7. *Merchant/Admin* mengisi form pendaftaran dan menekan tombol simpan. |  |
|  |  | 8. Menyimpan data pemandu wisata. |
| Alternative Flow of Events | - | |

## Kebutuhan Antarmuka Eksternal

### Kebutuhan Antarmuka Pengguna

Antarmuka pengguna harus bisa berinteraksi dengan antarmuka yang ditampilkan sistem pada layar *smartphone* Android (utama), laptop, komputer, tablet, dan sebagainya.

### Kebutuhan Antarmuka Perangkat Keras

Perangkat keras yang digunakan harus mampu untuk terhubung dengan koneksi internet, serta memiliki daya yang cukup.

### Kebutuhan Antarmuka Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak ialah:

* Sistem Operasi

Laptop, komputer : Windows 10

* Browser : Chrome
* Database : PostgreSQL 11

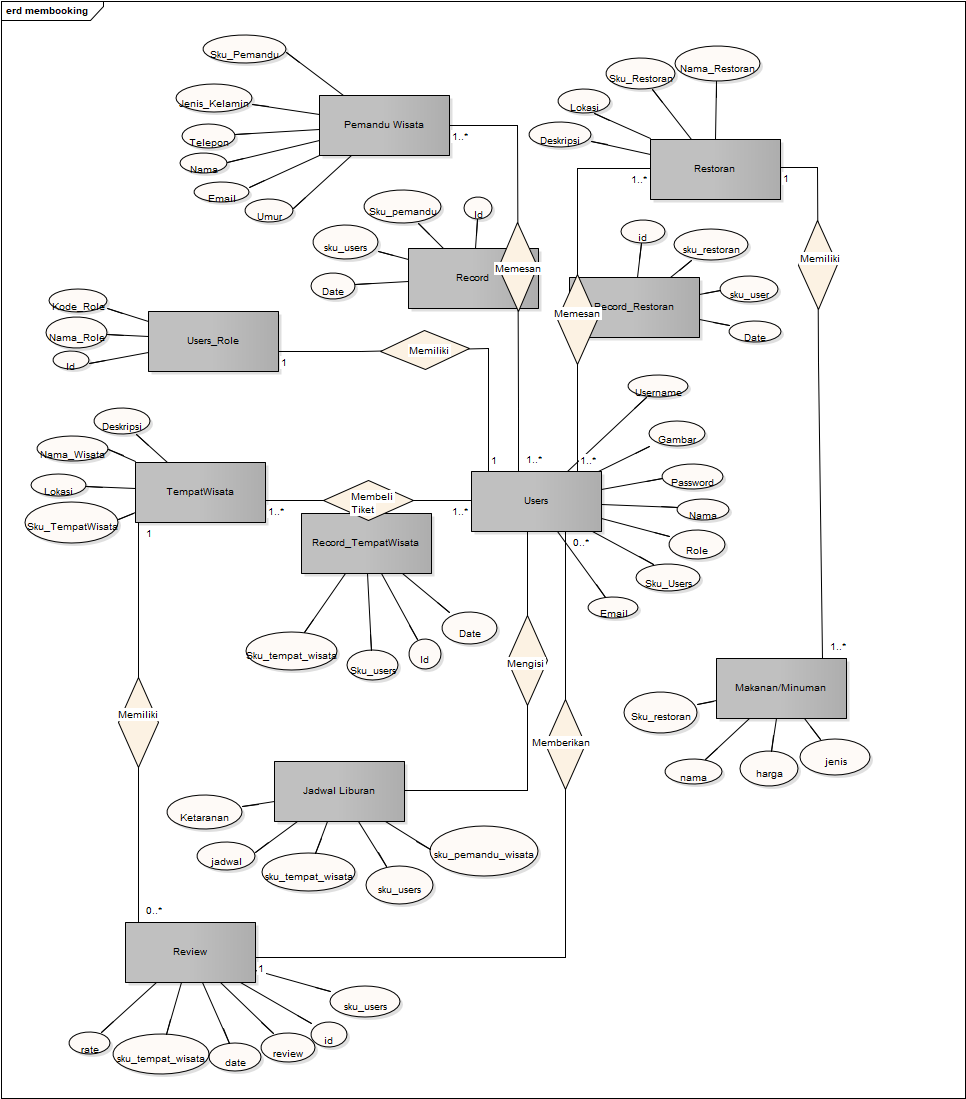
### Kebutuhan Antarmuka Komunikasi

Tidak diterapkan pada sistem ini.

## Kebutuhan Non-Fungsional

### Struktur Logikal Data

**ERD**

****