Use Cases

for

Koperasi Simpan Pinjam Institut Teknologi Del

Version 1.0 approved

Jesica Sianturi 11318008

Diana Octaviana 11318012

Nicolas Manurung 11318025

Chandra L Sitorus 11318041

Institut Teknologi Del

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

## Use Case ID and Name

Give each use case a unique integer sequence number identifier. State a concise name for the use case that indicates the value the use case would provide to some user. Begin with an action verb, followed by an object.

## Author and Date Created

Enter the name of the person who initially wrote this use case and the date it was written.

## Primary and Secondary Actors

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the primary actor that will be initiating this use case and any other secondary actors who will participate in completing execution of the use case.

## Trigger

Identify the business event, system event, or user action that initiates the use case. This trigger alerts the system that it should begin testing the preconditions for the use case so it can judge whether to proceed with execution.

## Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

## Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. The system must be able to test each precondition. Number each precondition. Example: PRE-1: User’s identity has been authenticated.

## Post conditions

Describe the state of the system at the successful conclusion of the use case execution. Label each post condition in the form POST-X, where X is a sequence number. Example: POST-1: Price of item in the database has been updated with the new value.

## Normal Flow

Provide a description of the user actions and corresponding system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. Show a numbered list of actions performed by the actor, alternating with responses provided by the system. The normal flow is numbered “X.0”, where “X” is the Use Case ID.

## Alternative Flows

Document other successful usage scenarios that can take place within this use case. State the alternative flow, and describe any differences in the sequence of steps that take place. Number each alternative flow in the form “X.Y”, where “X” is the Use Case ID and Y is a sequence number for the alternative flow. For example, “5.3” would indicate the third alternative flow for use case number 5. Indicate where each alternative flow would branch off from the normal flow, and if pertinent, where it would rejoin the normal flow.

## Exceptions

Describe any anticipated error conditions that could occur during execution of the use case and how the system is to respond to those conditions. Number each alternative flow in the form “X.Y.EZ”, where “X” is the Use Case ID, Y indicates the normal (0) or alternative (>0) flow during which this exception could take place, “E” indicates an exception, and “Z” is a sequence number for the exceptions. For example “5.0.E2” would indicate the second exception for the normal flow for use case number 5. Indicate where in the normal (or an alternative) flow each exception could occur.

## Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. Use the same priority scheme as that used for the functional requirements.

## Frequency of Use

Estimate the number of times this use case will be performed per some appropriate unit of time. This gives an early indicator of throughput, concurrent usage loads, and transaction capacity.

## Business Rules

List any business rules that influence this use case. Don’t include the business rule text here, just its identifier so the reader can find it in another repository when needed.

## Other Information

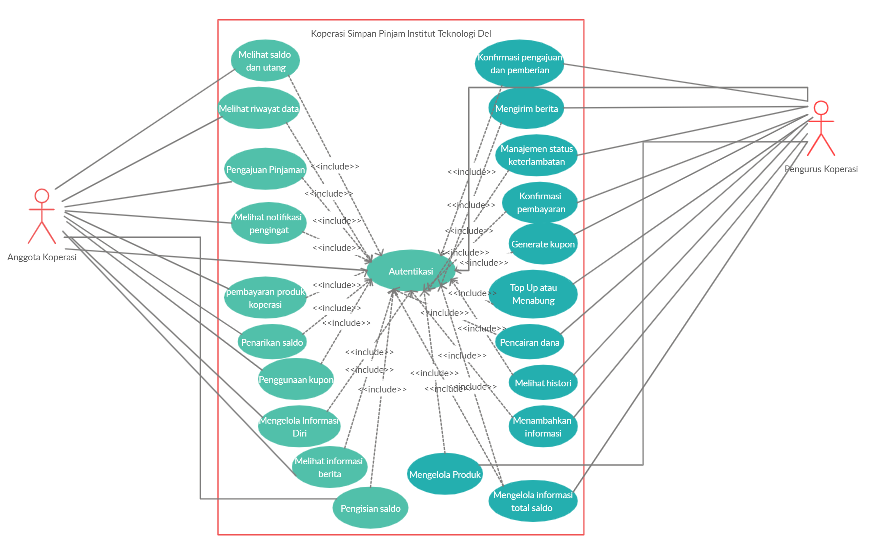
Identify any additional requirements, such as quality attributes, for the use case that may need to be addressed during design or implementation. Also list any associated functional requirements that aren’t a direct part of the use case flows but which a developer needs to know about. Describe what should happen if the use case execution fails for some unanticipated or systemic reason (e.g., loss of network connectivity, timeout). If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception.

## Assumptions

List any assumptions that were made regarding this use case or how it might execute.

Use Case List

|  |  |
| --- | --- |
| Primary Actor | Use Cases |
| Mahasiswa | 1. Melihat saldo dan utang 2. Melihat riwayat data. 3. Autentikasi. 4. Pengajuan pinjaman. 5. Melihat notifikasi pengingat. 6. Pembayaran produk koperasi. 7. Penarikan saldo 8. Penggunaan kupon (*Redeem*) 9. Pengisian saldo 10. Mengelola informasi diri. 11. Melihat informasi (berita). |
| Staff | 1. Melihat saldo dan utang 2. Melihat riwayat data. 3. Autentikasi. 4. Pengajuan pinjaman. 5. Melihat notifikasi pengingat. 6. Pembayaran produk koperasi. 7. Penarikan saldo 8. Penggunaan kupon (*Redeem*) 9. Pengisian saldo 10. Mengelola informasi diri. 11. Melihat informasi (berita). |
| Dosen | 1. Melihat saldo dan utang 2. Melihat riwayat data. 3. Autentikasi. 4. Pengajuan pinjaman. 5. Melihat notifikasi pengingat. 6. Pembayaran produk koperasi. 7. Penarikan saldo 8. Penggunaan kupon (*Redeem*) 9. Pengisian saldo 10. Mengedit informasi diri. 11. Melihat informasi (berita). |
| Pengurus Koperasi | 1. Mengirim berita 2. Mengonfirmasi pengajuan dan pemberian pinjaman (*on progress and success status change)*. 3. Manajemen status keterlambatan iuran. 4. Mengonfirmasi pembayaran iuran peminjam. 5. Generate kupon / *voucher.* 6. Penambahan saldo pengguna *(top up / menabung).* 7. Pencairan dana (saldo). 8. Melihat histori transaksi 9. Menambahkan informasi (berita / laporan). 10. Autentikasi 11. Melihat informasi total saldo 12. Mengelola produk. |

Use Case Diagram

Use Case Scenario

# 2.1 Use Case Scenario *Melihat Saldo dan Utang*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-01 Melihat Saldo dan Utang | |
| Primary Actor: | Mahasiswa, Dosen, Staff | |
| Description: | Mahasiswa, Dosen dan Staff dapat melihat saldo dan utang pada akun masing-masing. | |
| Preconditions: | 1. Mahasiswa, Dosen dan Staff sudah terdaftar pada aplikasi secara benar. 2. Mahasiswa, Dosen dan Staff login ke aplikasi yang benar. 3. Mahasiswa, Dosen dan Staff memilih halaman beranda. | |
| Postconditions: | 1. Mahasiswa, Dosen dan Staff melihat saldo dan utang. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Mahasiswa, Dosen dan Staff memilih halaman beranda |  |
|  | 1. System menampilkan saldo dan utang. |
| Alternative Flows: | User Action | System Response |
| None | None |
| Error Flow of Events: | User Action | System Response |
| Kembali ke *Previous Page* | Menampilkan *Error Page* |

## 2.2 Use Case Scenario *Melihat Riwayat Data*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-02 Melihat Riwayat Data | |
| Primary Actor: | Mahasiswa, Dosen, Staff | |
| Description: | Mahasiswa, Dosen dan Staff dapat riwayat data berupa tambahan Sisa Hasil Usaha yang didapatkan dari Rapat Anggota Tahunan dan data masukan dari pengurus setelah melakukan pembayaran. Jumlah *limit* pada Dosen serta Staff berbeda dengan jumlah limit pada Mahasiswa. | |
| Preconditions: | 1. Mahasiswa, Dosen dan Staff sudah terdaftar pada aplikasi secara benar. 2. Mahasiswa, Dosen dan Staff login ke aplikasi yang benar. 3. Mahasiswa, Dosen dan Staff memilih halaman beranda. | |
| Postconditions: | 1. Mahasiswa, Dosen dan Staff melihat riwayat data. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Mahasiswa, Dosen dan Staff menekan menu notifikasi |  |
|  | 1. System menampilkan riwayat data. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## 2.3 Use Case Scenario *Autentikasi*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-03 Autentikasi | |
| Primary Actor: | Mahasiswa, Dosen, Staff dan Pengurus Koperasi | |
| Description: | Mahasiswa, Dosen, Staff dan Pengurus koperasi dapat melakukan autentikasi pada saat login ke aplikasi. | |
| Preconditions: | 1. Mahasiswa, Dosen, Staff dan Pengurus Koperasi sudah terdaftar pada aplikasi secara benar. | |
| Postconditions: | 1. Mahasiswa, Dosen, Staff dan Pengurus Koperasi dapat login ke aplikasi. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Mahasiswa, Dosen, Staff dan Pengurus Koperasi mengetikkan username dan password dengan benar. |  |
|  | 1. System melakukan autentikasi. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Landing Page.* | Menampilkan *Landing Page.* |

## Use Case Scenario *Pengajuan Pinjaman*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-04 Pengajuan Pinjaman | |
| Primary Actor: | Mahasiswa, Dosen, Staff | |
| Description: | Mahasiswa, Dosen dan Staff dapat melakukan pengajuan permohonan pinjaman kepada pengurus koperasi. | |
| Preconditions: | 1. Mahasiswa, Dosen dan Staff sudah terdaftar pada aplikasi secara benar. 2. Mahasiswa, Dosen dan Staff login ke aplikasi yang benar. 3. Mahasiswa, Dosen dan Staff memilih halaman beranda. | |
| Postconditions: | 1. Mahasiswa, Dosen dan Staff melakukan pengajuan pinjaman. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Mahasiswa, Dosen dan Staff memilih menu pengajuan pinjaman |  |
|  | 1. System menampilkan form pengajuan pinjaman. |
| 1. Mahasiswa, Staff dan Dosen mengisi form pengajuan pinjaman dan mengirimkan kepada pengurus koperasi. |  |
|  | 1. System mengirimkan data yang di-*input* kepada pengurus koperasi dan mengirimkan konfirmasi kepada anggota. |
|  | 1. Mahasiswa, Staff dan Dosen mendapatkan notifikasi konfirmasi. |  |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Melihat Notifikasi Pengingat*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-05 Melihat Notifikasi Pengingat | |
| Primary Actor: | Mahasiswa, Dosen, Staff | |
| Description: | Mahasiswa, Dosen dan Staff dapat melihat notifikasi pengingat dari pengurus koperasi jika belum membayar pinjaman atau iuran bulanan anggota. | |
| Preconditions: | 1. Mahasiswa, Dosen dan Staff sudah terdaftar pada aplikasi secara benar. 2. Mahasiswa, Dosen dan Staff login ke aplikasi yang benar. 3. Mahasiswa, Dosen dan Staff memilih halaman beranda. | |
| Postconditions: | 1. Mahasiswa, Dosen dan Staff melihat riwayat data. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Mahasiswa, Dosen dan Staff memilih halaman beranda. |  |
|  | 1. System menampilkan pengingat di halaman beranda. |
| Alternative Flows: | User Action | System Response |
| None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Pembayaran Produk Koperasi*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-06 Pembayaran Produk Koperasi | |
| Primary Actor: | Mahasiswa, Dosen, Staff | |
| Description: | Mahasiswa, Dosen dan Staff dapat melakukan pembayaran produk koperasi melalui aplikasi. | |
| Preconditions: | 1. Mahasiswa, Dosen dan Staff sudah terdaftar pada aplikasi secara benar. 2. Mahasiswa, Dosen dan Staff login ke aplikasi yang benar. 3. Mahasiswa, Dosen dan Staff memilih menu produk koperasi. | |
| Postconditions: | 1. Mahasiswa, Dosen dan Staff melihat riwayat data. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Mahasiswa, Dosen dan Staff memilih menu produk koperasi. |  |
|  | 1. System menampilkan halaman produk koperasi. |
| 1. Mahasiswa, Dosen dan Staff memilih produk koperasi. |  |
|  | 1. System mengirimkan menampilkan halaman transfer koperasi. |
| 1. Mentransfer biaya produk koperasi |  |
|  | 1. Menyimpan data. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Penarikan Saldo*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-07 Penarikan Saldo | |
| Primary Actor: | Mahasiswa, Dosen, Staff | |
| Description: | Mahasiswa, Dosen dan Staff melakukan penarikan saldo. | |
| Preconditions: | 1. Mahasiswa, Dosen dan Staff sudah terdaftar pada aplikasi secara benar. 2. Mahasiswa, Dosen dan Staff login ke aplikasi yang benar. 3. Mahasiswa, Dosen dan Staff menuju halaman beranda. | |
| Postconditions: | 1. Mahasiswa, Dosen dan Staff melakukan penarikan saldo | |
| Primary Flow of Events: | User Action | System Response |
| 1. Mahasiswa, Dosen dan Staff memilih tombol tarik saldo. |  |
|  | 1. System menampilkan halaman tarik saldo. |
| 1. Mahasiswa, Dosen dan Staff memasukkan jumlah saldo yang ingin ditarik dan menekan tombol konfirmasi. |  |
|  | 1. System dengan mengirim notifikasi konfirmasi. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Penggunaan Kupon / Redeem*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-08 Penggunaan Kupon / Redeem | |
| Primary Actor: | Mahasiswa, Dosen, Staff | |
| Description: | Mahasiswa, Dosen dan Staff dapat menggunakan kupon yang didapat dari pengurus koperasi. | |
| Preconditions: | 1. Mahasiswa, Dosen dan Staff sudah terdaftar pada aplikasi secara benar. 2. Mahasiswa, Dosen dan Staff login ke aplikasi yang benar. 3. Mahasiswa, Dosen dan Staff memilih menu produk koperasi | |
| Postconditions: | 1. Mahasiswa, Dosen dan Staff menggunakan kupon. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Mahasiswa, Dosen dan Staff memilih menu bayar. |  |
|  | 1. System menampilkan halaman menu bayar. |
| 1. Mahasiswa, Dosen dan Staff mengisi kolom kupon untuk digunakan saat melakukan pembayaran terhadap produk koperasi. |  |
|  | 1. System mengonfirmasi penggunaan kupon. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Pengisian Saldo*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-09 Pengisian Saldo | |
| Primary Actor: | Mahasiswa, Dosen, Staff | |
| Description: | Mahasiswa, Dosen dan Staff dapat melakukan pengisian saldo. | |
| Preconditions: | 1. Mahasiswa, Dosen dan Staff sudah terdaftar pada aplikasi secara benar. 2. Mahasiswa, Dosen dan Staff login ke aplikasi yang benar. 3. Mahasiswa, Dosen dan Staff memilih menu saldo. | |
| Postconditions: | 1. Mahasiswa, Dosen dan Staff melakukan pengisian saldo. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Mahasiswa, Dosen dan Staff memilih saldo. |  |
|  | 1. System menampilkan halaman menu saldo. |
| 1. Mahasiswa, Dosen dan Staff mengisi jumlah saldo. |  |
|  | 1. System secara otomatis melakukan pengisian saldo, dan mengirim notifikasi saldo. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## 

## Use Case Scenario *Mengelola Informasi Diri*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-10 Mengelola Informasi Diri | |
| Primary Actor: | Mahasiswa, Dosen, Staff | |
| Description: | Mahasiswa, Dosen dan Staff dapat melakukan pengeditan terhadap informasi pada akun masing-masing. | |
| Preconditions: | 1. Mahasiswa, Dosen dan Staff sudah terdaftar pada aplikasi secara benar. 2. Mahasiswa, Dosen dan Staff login ke aplikasi yang benar. 3. Mahasiswa, Dosen dan Staff memilih menu profil. | |
| Postconditions: | 1. Mahasiswa, Dosen dan Staff melakukan pengeditan informasi diri. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Mahasiswa, Dosen dan Staff memilih menu profil. |  |
|  | 1. System menampilkan halaman menu profil. |
| 1. Mahasiswa, Dosen dan Staff mengisi dan melengkapi profil serta menyimpan. |  |
|  | 1. System secara otomatis menyimpan perubahan terhadap informasi profil. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Melihat Informasi*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-11 Melihat Informasi | |
| Primary Actor: | Mahasiswa, Dosen, Staff | |
| Description: | Mahasiswa, Dosen dan Staff dapat melakukan pengisian saldo. | |
| Preconditions: | 1. Mahasiswa, Dosen dan Staff sudah terdaftar pada aplikasi secara benar. 2. Mahasiswa, Dosen dan Staff login ke aplikasi yang benar. 3. Mahasiswa, Dosen dan Staff memilih menu informasi. | |
| Postconditions: | 1. Mahasiswa, Dosen dan Staff melihat informasi. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Mahasiswa, Dosen dan Staff memilih menu informasi. |  |
|  | 1. System menampilkan halaman menu informasi. |
| 1. Mahasiswa, Dosen dan Staff melihat salah satu informasi yang dibutuhkan. |  |
|  | 1. System secara otomatis menampilkan detail informasi yang dipilih. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Mengirim Informasi / Berita*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-12 Mengirim Informasi / Berita | |
| Primary Actor: | Pengurus Koperasi | |
| Description: | Pengurus koperasi dapat mengirim informasi atau berita seperti membagikan laporan atau informasi hasil rapat ke anggota koperasi. | |
| Preconditions: | 1. Pengurus koperasi sudah terdaftar pada aplikasi secara benar. 2. Pengurus koperasi login ke aplikasi yang benar. 3. Pengurus koperasi memilih menu laporan. | |
| Postconditions: | 1. Pengurus koperasi membagikan laporan kepada anggota koperasi. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Pengurus koperasi memilih menu laporan. |  |
|  | 1. System menampilkan halaman menu laporan . |
| 1. Pengurus koperasi memilih menu laporan yang akan dibagikan dan mengirimkan ke semua anggota koperasi. |  |
|  | 1. System mengirimkan laporan yang dipilih oleh pengurus koperasi ke semua anggota koperasi. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Mengonfirmasi Pengajuan dan Pemberian Pinjaman*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-13 Mengonfirmasi Pengajuan dan Pemberian Pinjaman | |
| Primary Actor: | Pengurus Koperasi | |
| Description: | Pengurus koperasi dapat mengonfirmasi pengajuan dan pemberian pinjaman yang dilakukan anggota koperasi *(on progress and success status change).* | |
| Preconditions: | 1. Pengurus koperasi sudah terdaftar pada aplikasi secara benar. 2. Pengurus koperasi login ke aplikasi yang benar. 3. Pengurus koperasi memilih menu pinjaman. | |
| Postconditions: | 1. Pengurus koperasi mengonfirmasi pengajuan dan pemberian pinjaman | |
| Primary Flow of Events: | User Action | System Response |
| 1. Pengurus koperasi memilih menu pinjaman. |  |
|  | 1. System menampilkan halaman menu pinjaman. |
| 1. Pengurus koperasi memilih peminjam dan menekan tombol “tolak” atau “terima”. |  |
|  | 1. System menampilkan *descriptive* *column*. |
|  | 1. Pengurus koperasi mengisi deskripsi penerimaan atau penolakan. |  |
|  |  | 1. Menuju halaman penarikan pinjaman. |
|  | 1. Memilih salah satu peminjam dan mengonfirmasi peminjam. |  |
|  |  | 1. Sistem menyimpan hasil konfirmasi. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Manajemen Status Keterlambatan Iuran*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-14 Manajemen Status Keterlambatan Iuran | |
| Primary Actor: | Pengurus Koperasi | |
| Description: | Pengurus koperasi dapat mengaktifkan maupun menonaktifkan pengingat keterlambatan pembayaran. | |
| Preconditions: | 1. Pengurus koperasi sudah terdaftar pada aplikasi secara benar. 2. Pengurus koperasi login ke aplikasi yang benar. 3. Pengurus koperasi memilih menu penunggak. | |
| Postconditions: | 1. Pengurus koperasi memanajemen status keterlambatan iuran. | |
| Primary Flow of Events: | User Action | User Action |
| 1. Pengurus koperasi memilih menu penunggak. |  |
|  | 1. Pengurus koperasi memilih menu penunggak. |
| 1. Pengurus koperasi memilih salah satu penunggak. |  |
|  | 1. Pengurus koperasi memilih salah satu penunggak. |
| Alternative Flows: | User Action | User Action |
| 1. Bila tunggakan sudah dibayar, pengurus koperasi memilih menu penunggak aktif. |  |
|  | 1. Bila tunggakan sudah dibayar, pengurus koperasi memilih menu penunggak aktif. |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Mengonfirmasi Pembayaran Iuran Peminjam*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-15 Mengonfirmasi Pembayaran Iuran Peminjam | |
| Primary Actor: | Pengurus Koperasi | |
| Description: | Pengurus koperasi dapat mengonfirmasi pembayaran iuran peminjam yang dilakukan oleh anggota koperasi. | |
| Preconditions: | 1. Pengurus koperasi sudah terdaftar pada aplikasi secara benar. 2. Pengurus koperasi login ke aplikasi yang benar. 3. Pengurus koperasi memilih menu iuran. | |
| Postconditions: | 1. Pengurus koperasi mengonfirmasi pembayaran iuran peminjam. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Pengurus koperasi memilih menu iuran. |  |
|  | 1. System menampilkan halaman menu iuran. |
| 1. Pengurus koperasi memilih salah satu list iuran peminjam. |  |
|  | 1. System otomatis menampilkan halaman informasi iuran. |
| 1. Pengurus koperasi mengonfirmasi iuran. |  |
|  | 1. System menyimpan konfirmasi iuran. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario Generate *Kupon /* Voucher

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-16 *Generate* Kupon *Voucher* | |
| Primary Actor: | Pengurus Koperasi | |
| Description: | Pengurus koperasi dapat melakukan *generate* terhadap kupon atau *voucher.* | |
| Preconditions: | 1. Pengurus koperasi sudah terdaftar pada aplikasi secara benar. 2. Pengurus koperasi login ke aplikasi yang benar. 3. Pengurus koperasi memilih menu kupon. | |
| Postconditions: | 1. Pengurus koperasi melakukan *generate* kupon. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Pengurus koperasi memilih menu kupon |  |
|  | 1. System menampilkan halaman menu kupon. |
| 1. Pengurus koperasi memasukkan kode *voucher /* kupon dan jumlah diskon. |  |
|  | 1. System menampilkan produk yang akan diberi voucher. |
|  | 1. Pengurus koperasi memilih produk. |  |
|  |  | 1. System menampilkan halaman masa berlaku dan limit tukar voucher. |
|  | 1. Pengurus koperasi memilih masa berlaku dan limit tukar voucher. |  |
|  |  | 1. System menyimpan data voucher |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Penambahan Saldo Pengguna (Top Up / Menabung)*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-17 Penambahan saldo pengguna (*Top Up /Menambung)* | |
| Primary Actor: | Pengurus Koperasi | |
| Description: | Pengurus koperasi dapat menambahkan saldo pengguna pada saat pengguna menabung atau melakukan *top up.* | |
| Preconditions: | 1. Pengurus koperasi sudah terdaftar pada aplikasi secara benar. 2. Pengurus koperasi login ke aplikasi yang benar. 3. Pengurus koperasi memilih menu pengisian saldo. | |
| Postconditions: | 1. Pengurus koperasi menambahkan saldo pengguna. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Pengurus koperasi memilih menu pengisian saldo. |  |
|  | 1. System menampilkan halaman menu pengisian saldo. |
| 1. Pengurus koperasi memilih salah satu pengguna untuk ditambahkan saldonya. |  |
|  | 1. System otomatis menambahkan saldo kepada pengguna yang dipilih. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Pencairan Dana (Saldo)*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-18 Pencairan Dana (Saldo) | |
| Primary Actor: | Pengurus Koperasi | |
| Description: | Pengurus koperasi melakukan pencairan dana (saldo) anggota koperasi. | |
| Preconditions: | 1. Pengurus koperasi sudah terdaftar pada aplikasi secara benar. 2. Pengurus koperasi login ke aplikasi yang benar. 3. Pengurus koperasi memilih menu pencairan. | |
| Postconditions: | 1. Pengurus koperasi melakukan pencairan dana. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Pengurus koperasi memilih menu pencairan. |  |
|  | 1. System menampilkan halaman menu pencairan. |
| 1. Pengurus koperasi memilih salah satu pengguna. |  |
|  | 1. System mengurangi saldo pada pengguna. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Melihat Histori Transaksi

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-19 Melihat Histori Transaksi | |
| Primary Actor: | Pengurus Koperasi | |
| Description: | Pengurus koperasi dapat melihat histori transaksi berupa riwayat penggunaan saldo, penarikan saldo, dan pembayaran iuran pengguna. | |
| Preconditions: | 1. Pengurus koperasi sudah terdaftar pada aplikasi secara benar. 2. Pengurus koperasi login ke aplikasi yang benar. 3. Pengurus koperasi memilih notifikasi. | |
| Postconditions: | 1. Pengurus koperasi melihat histori transaksi. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Pengurus koperasi memilih notifikasi. |  |
|  | 1. System menampilkan halaman notifikasi. |
| 1. Pengurus koperasi melihat histori transaksi. |  |
|  | 1. System menampilkan histori transaksi. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Menambahkan Informasi*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-20 Menambahkan Informasi | |
| Primary Actor: | Pengurus Koperasi | |
| Description: | Pengurus koperasi dapat menambahkan informasi baik berupa berita maupun laporan. | |
| Preconditions: | 1. Pengurus koperasi sudah terdaftar pada aplikasi secara benar. 2. Pengurus koperasi login ke aplikasi yang benar. 3. Pengurus koperasi memilih menu informasi. | |
| Postconditions: | 1. Pengurus koperasi menambahkan informasi. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Pengurus koperasi memilih menu informasi. |  |
|  | 1. System menampilkan halaman menu informasi. |
| 1. Pengurus koperasi menambahkan informasi dengan mengisi data. |  |
|  | 1. System menambahkan informasi baru. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Melihat Informasi Total Saldo*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-21 Melihat Informasi Total Saldo | |
| Primary Actor: | Pengurus Koperasi | |
| Description: | Pengurus koperasi dapat melihat informasi total saldo. | |
| Preconditions: | 1. Pengurus koperasi sudah terdaftar pada aplikasi secara benar. 2. Pengurus koperasi login ke aplikasi yang benar. 3. Pengurus koperasi memilih halaman beranda. | |
| Postconditions: | 1. Pengurus koperasi melihat informasi total saldo. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Pengurus koperasi memilih halaman beranda. |  |
|  | 1. System menampilkan informasi saldo. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |

## Use Case Scenario *Mengelola Produk*

|  |  |  |
| --- | --- | --- |
| UC ID and Name: | UC-22 Mengelola Produk | |
| Primary Actor: | Pengurus Koperasi | |
| Description: | Pengurus koperasi dapat mengelola produk. | |
| Preconditions: | 1. Pengurus koperasi sudah terdaftar pada aplikasi secara benar. 2. Pengurus koperasi login ke aplikasi yang benar. 3. Pengurus koperasi memilih menu produk. | |
| Postconditions: | 1. Pengurus koperasi menambahkan data produk. | |
| Primary Flow of Events: | User Action | System Response |
| 1. Pengurus koperasi memilih menu produk. |  |
|  | 1. System menampilkan halaman menu produk. |
| 1. Pengurus koperasi menambahkan produk dan mengisi data produk. |  |
|  | 1. System menyimpan data produk yang ditambahkan. |
| Alternative Flows: | User Action | System Response |
|  | None | None |
| Error Flow of Events: | User Action | System Response |
|  | Kembali ke *Previous Page* | Menampilkan *Error Page* |