

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Laris Jaya @Rent Car adalah salah satu usaha yang bergerak dibidang jasa yaitu jasa penyewaan kendaraan khususnya penyewaan mobil. Transaksi penyewaan mobil dan pengolahan data pada Laris Jaya @Rent Car masih bersifat manual, dengan menuliskan transaksi penyewaan mobil di kwitansi berangkap sehingga menyebabkan penumpukan dokumen. Untuk pengolahan data penyewaan mobil dapat ditemui kendala yang akan terjadi seperti kekeliruan pada saat penjadwalan mobil yang akan disediakan. Kegiatan penyewaan mobil oleh pelanggan masih dilakukan secara manual dengan menuliskan data pelanggan di form-form data penyewaan.

Selain itu bukti transaksi penyewaan yang dibuat berangkap menggunakan kwitansi menjadi acuan untuk pembuatan laporan penyewaan yang akan dituliskan kedalam buku besar khusus laporan penyewaan. Jika menggunakan sistem ini maka akan banyak sekali terjadi kendala berupa kesalahan penjadwalan penyewaan mobil pelanggan, kesalahan dalam membuat laporan, terjadinya kehilangan data dan terjadi penumpukan dokumen-dokumen.

Dari permasalahan diatas maka penulis tertarik ingin membuatkan suatu sistem penyewaan mobil berupa dekstop untuk Laris Jaya @Rent Car Bekasi agar mempermudah dan mengoptimalkan sistem kerja di perusahaan tersebut guna mendapatkan informasi secara tepat dan akurat.

1.2. Analisa Kebutuhan Sistem

A. Kebutuhan Pengguna

Dalam aplikasi Penyewaan Mobil terdapat dua pengguna yang dapat berinteraksi dalam lingkungan sistem, yaitu: Admin, dan Pemilik. Kedua pengguna tersebut memiliki karakteristik interaksi dengan sistem yang berbeda – beda dan memiliki kebutuhan interaksi yang berbeda – beda seperti berikut :

A1 Skenario Kebutuhan Admin

- a) Mengelola data supir
- b) Mengelola data mobil
- c) Mengelola data penyewa
- d) Mengelola data penyewaan
- e) Mengelola data pengembalian
- f) Membuat laporan

A2 Skenario Kebutuhan Pemilik

- a) Mengelola data admin
- b) Melihat laporan

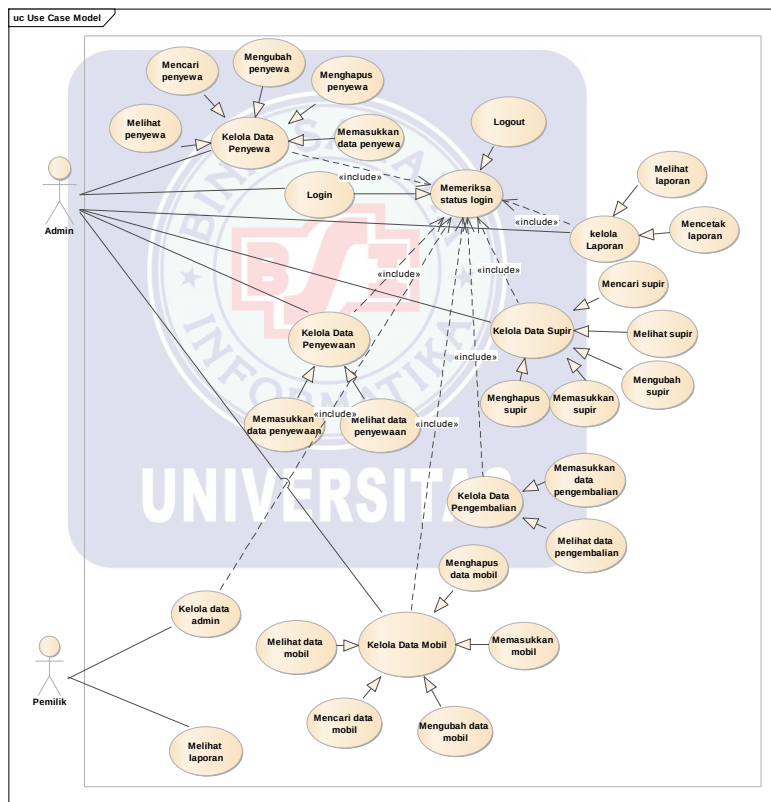
B. Kebutuhan Sistem

- 1) Admin harus melakukan login terlebih dahulu untuk dapat mengakses aplikasi ini dengan memasukkan *username* dan *password* agar privasi ini masih tetap terjaga keamanannya.
- 2) Setelah *login* admin dapat mengelola semua yang ada di aplikasi ini.
- 3) Admin harus melakukan *logout* setelah menggunakan aplikasi ini.
- 4) Sistem ini melakukan proses penyewaan dan pengembalian dengan memberikan form sewa mobil dan perhitungan denda dari mobil yang telah dikembalikan dalam kondisi rusak maupun baik dan keterterlambatan dalam pengembalian.

1.3. Rancangan Diagram *Use Case*

Menurut M.Shalahuddin. (2015:155), “*Use Case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Menurut Windu Grata, Grace dalam (Hendini, 2016) mengemukakan bahwa “*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut.



Gambar I.1 *Use Case Diagram*

Tabel I.1
Deskripsi Use Case Login

<i>Use Case Name</i>	<i>Login</i>
<i>Requirements</i>	Admin mengisi <i>username</i> dan <i>password</i>
<i>Goal</i>	Melakukan <i>login</i>
<i>Pre-Conditions</i>	Admin telah masuk ke halaman <i>login</i>
<i>Post-Conditions</i>	Admin dapat mengakses salah satu menu yang sudah ada di sistem
<i>Failed And Condition</i>	Admin tidak dapat masuk ke menu utama
<i>Actors</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin mengetikan <i>username</i> dan <i>password</i> 2. Pilih <i>button login</i> 3. Sistem menampilkan menu utama
<i>Alternate Flow/Invariant A</i>	A1. <i>Username</i> dan <i>Password</i> yang diisi salah A2. Sistem menampilkan "<i>username/password Salah</i>"

Tabel I.2
Deskripsi Use Case Melihat Data Penyewa

<i>Use Case Name</i>	Melihat penyewa
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin melihat data Penyewa
<i>Pre-Conditions</i>	Sistem menampilkan form penyewa
<i>Post-Conditions</i>	Sistem menampilkan Data Penyewa
<i>Failed And Condition</i>	Sistem tidak menampilkan data penyewa
<i>Actors</i>	<i>Admin</i>

<i>Main Flow/Basic Path</i>	1. <i>Admin</i> mengetikkan nama atau <i>id_penyewa</i> 2. Sistem menampilkan data penyewa
<i>Alternate Flow/Invariant A</i>	A1. Sistem menampilkan data penyewa yang dipilih

Tabel I.3
Deskripsi Use Case Memasukan Data Penyewa

<i>Use Case Name</i>	Memasukan Penyewa
<i>Requirements</i>	Admin telah melakukan <i>Login</i>
<i>Goal</i>	Admin melihat data penyewa
<i>Pre-Conditions</i>	Admin menambah data penyewa
<i>Post-Conditions</i>	1. Sistem menyimpan data penyewa ke <i>database</i> 2. Data penyewa yang di tabel akan bertambah
<i>Failed And Condition</i>	Data penyewa tidak tersimpan dan tidak bertambah
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Pilih <i>button</i> tambah 2. Isi semua data yang ada di <i>form</i> 3. Pilih <i>button</i> simpan untuk menyimpan dan pilih <i>button</i> batal untuk membatalkan
<i>Alternate Flow/Invariant A</i>	A1. Admin memilih <i>button</i> simpan A2. Sistem akan menampilkan pesan "Data Berhasil Disimpan" A3. Jika data belum lengkap admin sudah memilih <i>button</i> simpan maka sistem akan menampilkan pesan "Data Belum Lengkap"

Tabel I.4
Deskripsi Use Case Menghapus Data Penyewa

<i>Use Case Name</i>	Menghapus Penyewa
<i>Requirements</i>	Admin sudah melakukan <i>login</i>
<i>Goal</i>	Admin menghapus data penyewa
<i>Pre-Conditions</i>	Sistem menampilkan form penyewa
<i>Post-Conditions</i>	1. Sistem menghapus data penyewa di <i>database</i> berdasarkan data yang dipilih admin 2. Data penyewa yang ada di tabel berkurang
<i>Failed And Condition</i>	Admin tidak dapat menghapus data penyewa
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Admin mengetikkan nama atau <i>id_penyewa</i> 2. Sistem akan menampilkan data yang dicari 3. Kemudian pilih <i>button</i> hapus
<i>Alternate Flow/Invariant A</i>	- A1. Pilih <i>button</i> hapus - A2. Sistem akan menampilkan pesan "Data berhasil dihapus !"

Tabel I.5
Deskripsi Use Case Mencari Data Penyewa

<i>Use Case Name</i>	Mencari Penyewa
<i>Requirements</i>	Admin telah melakukan <i>login</i>
<i>Goal</i>	Admin Mencari data penyewa
<i>Pre-Conditions</i>	Sistem menampilkan form penyewa
<i>Post-Conditions</i>	Sistem menampilkan data yang dicari

<i>Failed And Condition</i>	Data yang dicari tidak dapat ditemukan
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Admin mengetikkan nama atau id_penyewa 2. Sistem menampilkan data yang sudah dicari
<i>Alternate Flow/Invariant A</i>	Sistem akan menampilkan pesan "Data tidak ditemukan"

Tabel I.6
Deskripsi Use Case Mengubah Data Penyewa

<i>Use Case Name</i>	Mengubah Penyewa
<i>Requirements</i>	Admin sudah melakukan login
<i>Goal</i>	Admin mengubah data penyewa
<i>Pre-Conditions</i>	Sistem menampilkan form penyewa
<i>Post-Conditions</i>	1. Sistem mengubah data penyewa yang ada di <i>database</i> 2. Data yang diubah berhasil
<i>Failed And Condition</i>	Admin tidak dapat mengubah data penyewa
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Admin mengetikkan nama atau id_penyewa 2. Memilih data yang akan diubah 3. Mengubah data yang dipilih 4. Pilih <i>button</i>
<i>Alternate Flow/Invariant A</i>	A1. Ketika kita akan mengubah data, sistem akan menampilkan pesan "Data berhasil diubah!"

Tabel I.7
Deskripsi Use Case Melihat Data Supir

<i>Use Case Name</i>	Melihat Supir
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin melihat data supir
<i>Pre-Conditions</i>	Sistem menampilkan form supir
<i>Post-Conditions</i>	Sistem menampilkan data supir
<i>Failed And Condition</i>	Sistem tidak menampilkan data supir
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Admin user mengetikkan nama atau id supir 2. Sistem menampilkan data supir
<i>Alternate Flow/Invariant A</i>	Sistem menampilkan data supir yang dipilih

Tabel I.8
Deskripsi Use Case Memasukan Data Supir

<i>Use Case Name</i>	Memasukan Supir
<i>Requirements</i>	Admin sudah melakukan login
<i>Goal</i>	Admin menambah data supir
<i>Pre-Conditions</i>	Sistem menampilkan form supir
<i>Post-Conditions</i>	1. Sistem menyimpan data supir ke <i>database</i> 2. Data supir yang ada di <i>database</i> bertambah
<i>Failed And Condition</i>	Data supir tidak tersimpan dan tidak bertambah
<i>Actors</i>	<i>Admin</i>

<i>Main Flow/Basic Path</i>	1. Pilih <i>button</i> tambah 2. Isi semua data yang ada di form 3. Pilih <i>button</i> simpan untuk menyimpan pilih <i>button</i> batal untuk membatalkan
<i>Alternate Flow/Invariant A</i>	A1. Admin memilih <i>button</i> simpan A2. Sistem akan menampilkan pesan "Data berhasil disimpan" A3. Jika data belum lengkap, admin sudah memilih <i>button</i> simpan maka akan menampilkan pesan "Data belum lengkap!"

Tabel I.9
Deskripsi Use Case Menghapus Data Supir

<i>Use Case Name</i>	Menghapus supir
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin menghapus data supir
<i>Pre-Conditions</i>	Sistem menampilkan form supir
<i>Post-Conditions</i>	1. Sistem menghapus data petugas di <i>database</i> berdasarkan data pilihan admin 2. Data petugas yang ada ditabel berkurang
<i>Failed And Condition</i>	Admin tidak dapat menghapus data supir
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Admin mengetikkan nama atau id supir 2. Sistem akan menampilkan data yang kita cari 3. Pilih <i>button</i> hapus

<i>Alternate Flow/Invariant A</i>	A1. Pilih <i>button</i> hapus A2. Sistem akan menampilkan pesan "Data berhasil dihapus !"
-----------------------------------	--

Tabel I.10
Deskripsi Use Case Mencari Data supir

<i>Use Case Name</i>	Mencari supir
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin mencari data supir
<i>Pre-Conditions</i>	Sistem menampilkan form supir
<i>Post-Conditions</i>	Sistem menampilkan data yang dicari
<i>Failed And Condition</i>	Data yang dicari tidak ditemukan
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Admin mengetikkan nama atau id supir 2. Sistem menampilkan data yang sudah dicari
<i>Alternate Flow/Invariant A</i>	Jika admin salah mengetikkan nama atau id supir maka sistem akan menampilkan pesan "Data tidak ditemukan !"

Tabel I.11
Deskripsi Use Case Mengubah Data supir

<i>Use Case Name</i>	Mengubah Supir
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin mengubah data supir
<i>Pre-Conditions</i>	Sistem menampilkan form supir
<i>Post-Conditions</i>	1. Sistem mengubah data supir yang ada di <i>datebase</i> 2. Data yang diubah berhasil

<i>Failed And Condition</i>	Admin tidak dapat mengubah data supir
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Admin mengetikkan nama atau id supir 2. Admin memilih data yang akan diubah 3. Admin mengubah data yang dipilih 4. Admin memilih <i>button</i> ubah
<i>Alternate Flow/Invariant A</i>	Ketika kita akan mengubah data sistem akan menampilkan pesan “Data berhasil dirubah”

Tabel I.12
Deskripsi Use Case Melihat Data Mobil

<i>Use Case Name</i>	Melihat Data Mobil
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin melihat data mobil
<i>Pre-Conditions</i>	Sistem menampilkan form
<i>Post-Conditions</i>	Sistem menampilkan data buku
<i>Failed And Condition</i>	Sistem tidak menampilkan data buku
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Admin mengetikkan kode mobil 2. Sitem akan menampilkan data mobil
<i>Alternate Flow/Invariant A</i>	Sistem menampilkan data mobil

Tabel I.13
Deskripsi use Case Memasukkan Data Mobil

<i>Use Case Name</i>	Memasukan Data mobil
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin memasukan data mobil
<i>Pre-Conditions</i>	Sistem menampilkan data mobil
<i>Post-Conditions</i>	1. Sistem menyimpan data mobil ke <i>database</i> 2. Data mobil yang ada di tabel bertambah
<i>Failed And Condition</i>	Data mobil tidak tersimpan dan tidak bertambah
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Pilih <i>button</i> tambah 2. Isi semua data yang ada di form 3. Pilih <i>button</i> simpan untuk menyimpan atau batal untuk membatalkan
<i>Alternate Flow/Invariant A</i>	A1. Admin memilih <i>button</i> simpan A2. Sistem akan menampilkan pesan "Data berhasil disimpan" A3. Jika data belum lengkap, admin sudah memilih <i>button</i> simpan maka sistem akan menampilkan pesan "Data belum lengkap"

Tabel I.14
Deskripsi Use Case Menghapus Data Mobil

<i>Use Case Name</i>	Menghapus Data Mobil
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin menghapus data mobil

<i>Pre-Conditions</i>	Sistem menampilkan form data mobil
<i>Post-Conditions</i>	1. Sistem menghapus data mobil di <i>database</i> berdasarkan data pilihan user 2. Data mobil yang ada ditabel berkurang
<i>Failed And Condition</i>	<i>Admin</i> tidak dapat menghapus data mobil
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. User mengetikan kode mobil 2. Sistem akan menampilkan data mobil yang dicari 3. Pilih <i>button</i> hapus
<i>Alternate Flow/Invariant A</i>	- A1. Pilih <i>button</i> hapus - A2. Sistem akan menampilkan pesan "Data berhasil dihapus"

Tabel I.15
Deskripsi Use Case Mencari Data Mobil

<i>Use Case Name</i>	Mencari Data Mobil
<i>Requirements</i>	<i>Admin</i> telah melakukan login
<i>Goal</i>	<i>Admin</i> mencari data mobil
<i>Pre-Conditions</i>	Sistem menampilkan form data mobil
<i>Post-Conditions</i>	Menampilkan data yang dicari
<i>Failed And Condition</i>	Data yang dicari tidak ditemukan
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. <i>Admin</i> mengetikan nama atau kode mobil 2. Sistem menampilkan data yang sudah dicari

<i>Alternate Flow/Invariant A</i>	A1. Jika admin salah memasukan nama atau kode mobil sistem akan menampilkan pesan “Data tidak ditemukan”
-----------------------------------	--

Tabel I.16
Deskripsi Use Case Mengubah Data Mobil

<i>Use Case Name</i>	Mengubah Data Mobil
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin mengubah data mobil
<i>Pre-Conditions</i>	Sistem menampilkan <i>form</i> mobil
<i>Post-Conditions</i>	1. Sistem mengubah data mobil yang ada di <i>database</i> 2. Data yang diubah berhasil
<i>Failed And Condition</i>	Admin tidak dapat mengubah data mobil
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Admin mengetikan nama atau kode mobil 2. Admin memilih data yang akan diubah 3. Mengubah data yang dipilih 4. Pilih <i>button</i> ubah
<i>Alternate Flow/Invariant A</i>	A1. Ketika kita akan mengubah data sistem akan menampilkan pesan “Data berhasil diubah”

Tabel I.17
Deskripsi Use Case Melihat Data Peminjaman

<i>Use Case Name</i>	Melihat Data peminjaman
<i>Requirements</i>	Admin telah melakukan login

<i>Goal</i>	Admin melihat data peminjaman
<i>Pre-Conditions</i>	Sistem menampilkan form peminjaman
<i>Post-Conditions</i>	Sistem menampilkan data peminjaman
<i>Failed And Condition</i>	Sistem tidak menampilkan data peminjaman
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	Admin melihat data peminjaman pada tabel peminjaman
<i>Alternate Flow/Invariant A</i>	Sistem menampilkan data peminjaman

Tabel I.18
Deskripsi Use Case Memasukan Data Peminjaman

<i>Use Case Name</i>	Memasukan Data Peminjaman
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin menambah data peminjaman
<i>Pre-Conditions</i>	Sistem menampilkan form peminjaman
<i>Post-Conditions</i>	1. Sistem menyimpan data peminjaman ke <i>database</i> 2. Data peminjaman yang ada bertambah
<i>Failed And Condition</i>	Data peminjaman tidak tersimpan dan tidak bertambah
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Pilih <i>button</i> tambah 2. Isi data di form peminjaman untuk menginput data penyewa 3. Pilih <i>button</i> simpan untuk menyimpan atau batal untuk membatalkan transaksi

<i>Alternate Flow/Invariant A</i>	A1. Admin memilih <i>button</i> simpan, sistem akan menampilkan pesan “Data berhasil disimpan”
-----------------------------------	--

Tabel I.19
Deskripsi Use Case Melihat Data Pengembalian

<i>Use Case Name</i>	Melihat Data Pengembalian
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin melihat data pengembalian
<i>Pre-Conditions</i>	Sistem menampilkan <i>form</i> pengembalian
<i>Post-Conditions</i>	Sistem menampilkan data pengembalian
<i>Failed And Condition</i>	Sistem tidak bisa menampilkan data pengembalian
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Admin mengetikkan no_sewa 2. Sistem akan menampilkan data pengembalian
<i>Alternate Flow/Invariant A</i>	Sistem menampilkan pengembalian yang dipilih

Tabel I.20
Deskripsi Use Case Memasukan Data Pengembalian

<i>Use Case Name</i>	Memasukan data pengembalian
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin menambahkan data pengembalian
<i>Pre-Conditions</i>	Sistem menampilkan <i>form</i> pengembalian

<i>Post-Conditions</i>	1. Sistem menyimpan data pengembalian ke <i>database</i> 2. Data pengembalian yang ada ditabel bertambah
<i>Failed And Condition</i>	Data pengembalian tidak tersimpan dan tidak bertambah
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Pilih <i>button</i> tambah 2. Isi data di <i>form</i> input peminjaman 3. Pilih <i>button</i> kondisi mobil yang sesuai dengan mobil yang dikembalikan 4. Jika penyewa mengembalikan lebih dari waktu yang di tentukan maka akan dikenakan denda 100 ribu untuk luar kota dan 50 ribu untuk dalam kota 5. jika penyewa mengembalikan mobil dalam keadaan rusak atau hilang maka penyewa akan di kenakan denda sesuai ketentuan rental
<i>Alternate Flow/Invariant A</i>	A1. Admin memilih <i>button</i> simpan sistem akan menampilkan pesan “Data berhasil disimpan”

Tabel I.21
Deskripsi Use Case Melihat Laporan

<i>Use Case Name</i>	Melihat Laporan
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin melihat Laporan
<i>Pre-Conditions</i>	Sistem menampilkan <i>form</i> laporan
<i>Post-Conditions</i>	Sistem menampilkan laporan yang dipilih

<i>Failed And Condition</i>	Sistem tidak menampilkan laporan
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	1. Admin memilih laporan yang akan di lihat 2. Pilih <i>button</i> cetak
<i>Alternate Flow/Invariant A</i>	Sistem menampilkan data laporan

Tabel I.22
Deskripsi Use Case Mencetak Laporan

<i>Use Case Name</i>	Mencetak Laporan
<i>Requirements</i>	Admin telah melakukan login
<i>Goal</i>	Admin mencetak laporan
<i>Pre-Conditions</i>	Sistem menampilkan form laporan
<i>Post-Conditions</i>	Sistem menampilkan laporan
<i>Failed And Condition</i>	Sistem tidak dapat mencetak laporan
<i>Actors</i>	1. Admin 2. Pemilik
<i>Main Flow/Basic Path</i>	1. Pilih laporan yang akan dicetak 2. Pilih <i>button</i> cetak 3. Pilih Print
<i>Alternate Flow/Invariant A</i>	Sistem akan menampilkan hasil laporan

Tabel I.23
Deskripsi Use Case Logout

<i>Use Case Name</i>	<i>Logout</i>
<i>Requirements</i>	<i>Admin</i> telah selesai mengelola <i>form</i>
<i>Goal</i>	Admin melakukan <i>Logout</i>

<i>Pre-Conditions</i>	Sistem menampilkan menu utama
<i>Post-Conditions</i>	1. Admin keluar dari menu utama 2. Sistem menampilkan halaman utama
<i>Failed And Condition</i>	Admin tidak dapat keluar dari menu utama
<i>Actors</i>	<i>Admin</i>
<i>Main Flow/Basic Path</i>	Admin memilih menu logout yang ada di menu utama
<i>Alternate Flow/Invariant A</i>	Ketika admin memilih menu logout sistem akan kembali kehalaman utama (login)

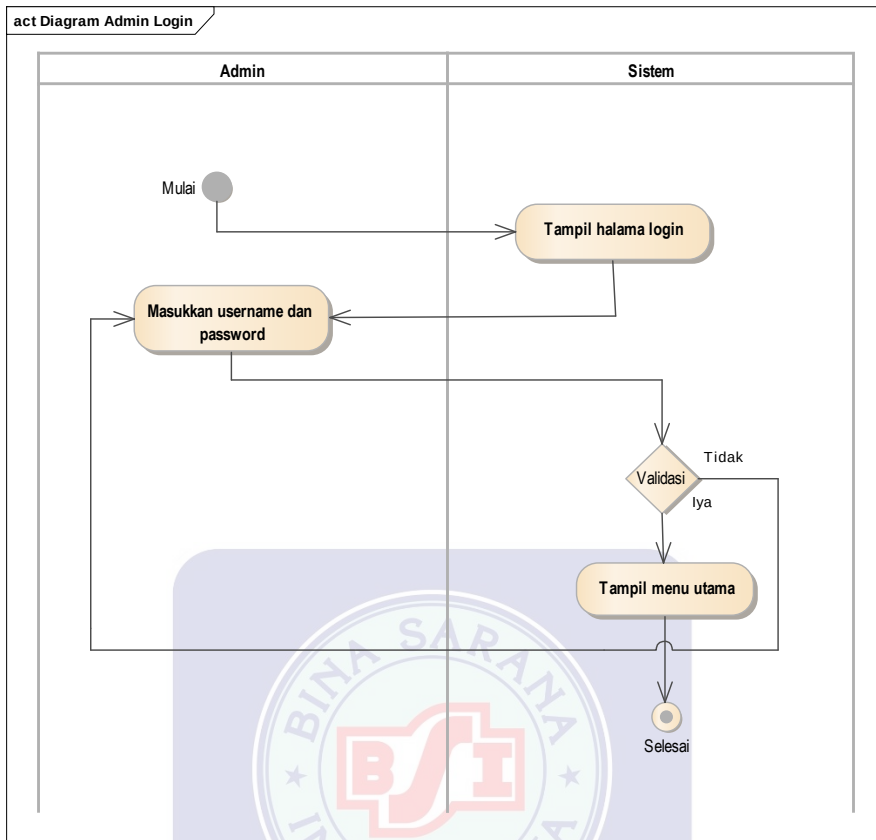
1.4. Rancangan Diagram Aktivitas

Menurut Sukamto;M.Shalahuddin. (2015:61) “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Menurut Windu Grata, Grace dalam (Hendini, 2016) mengemukakan bahwa “*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

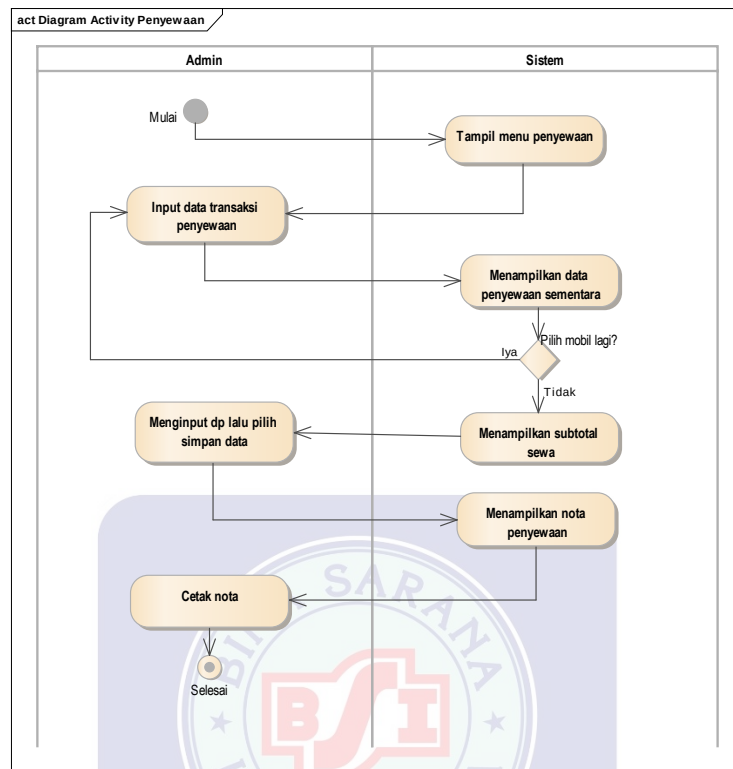
1.4.1. Activity Diagram

1. *Activity Diagram Login Admin.*



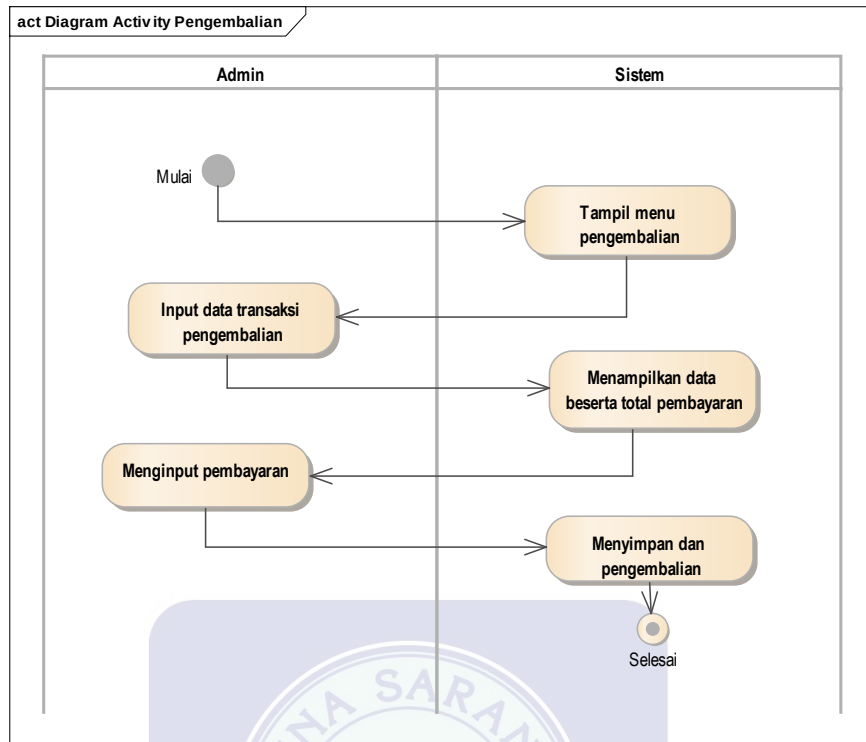
Gambar I.2 Activity Diagram Login Admin

2. Activity Diagram Penyewaan



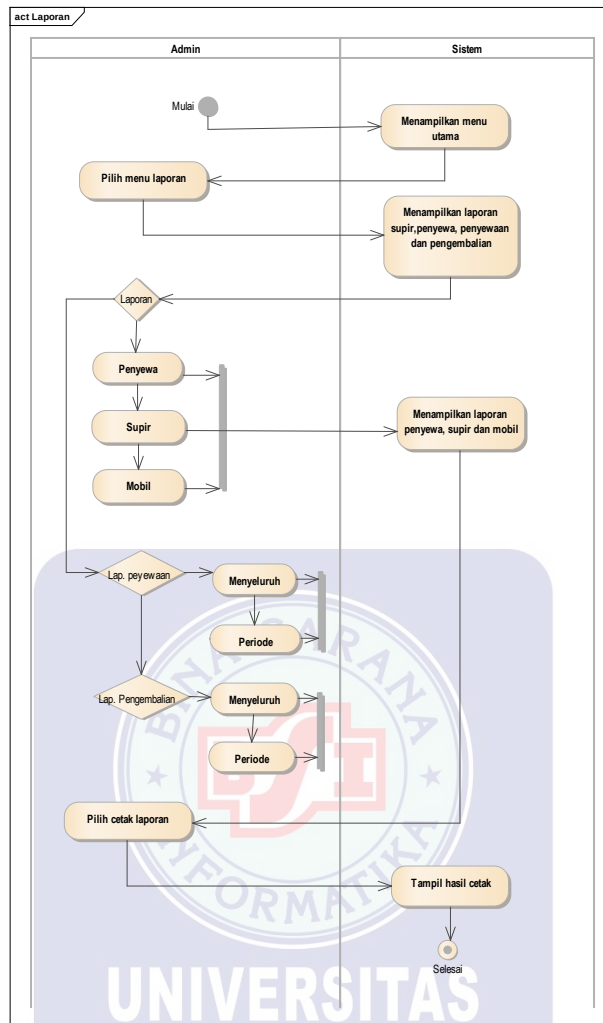
Gambar 1.3. Activity Diagram Penyewaan

3. Activity Diagram Pengembalian.



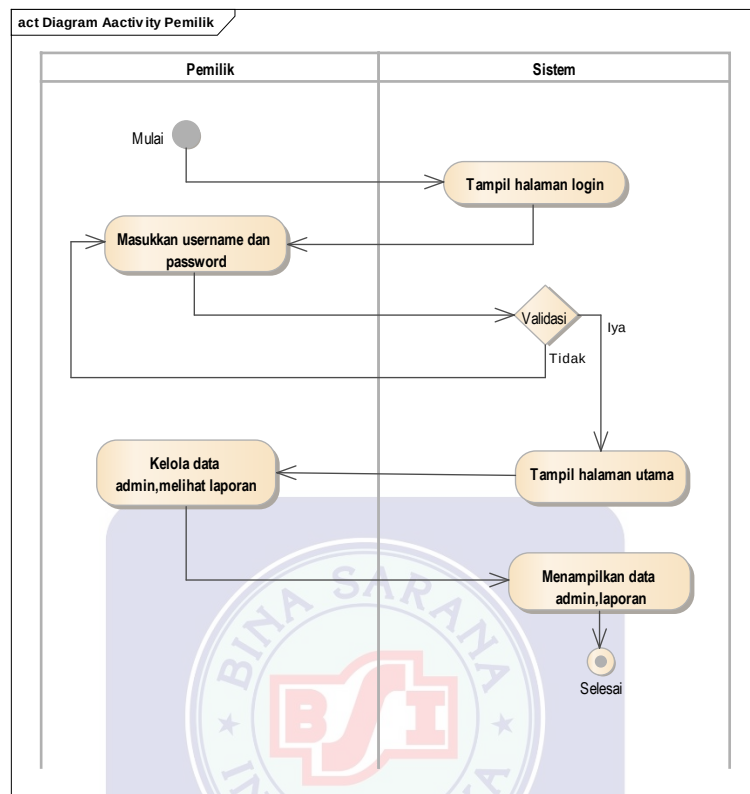
Gambar 1.4. Activity Diagram Pengembalian

4. Activity Diagram Laporan



Gambar 1.5. Activity Diagram Laporan

5. Activity Diagram Pemilik

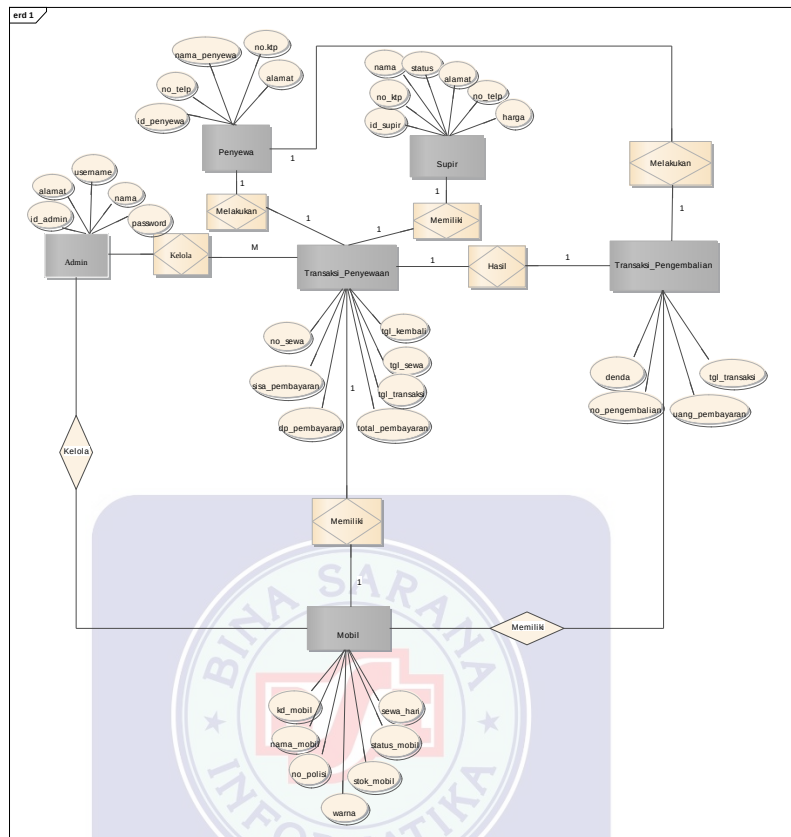


Gambar 1.6. Activity Diagram Pemilik

1.5. Entity Relationship Diagram (ERD)

Menurut Yanto (2016:32) dalam (Lestari et al., n.d.), "*Entity Relationship Diagram* (ERD) adalah suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional. ERD juga merupakan gambaran yang merelasikan antara objek yang satu dengan objek yang lain dari objek di dunia nyata yang sering dikenal dengan hubungan antar entitas".

Menurut Sukanto;M.Shalahuddin. (2015:50) "*Entity Relationship Diagram* (ERD) adalah bentuk paling awal dalam melakukan perancangan basis data relasional".

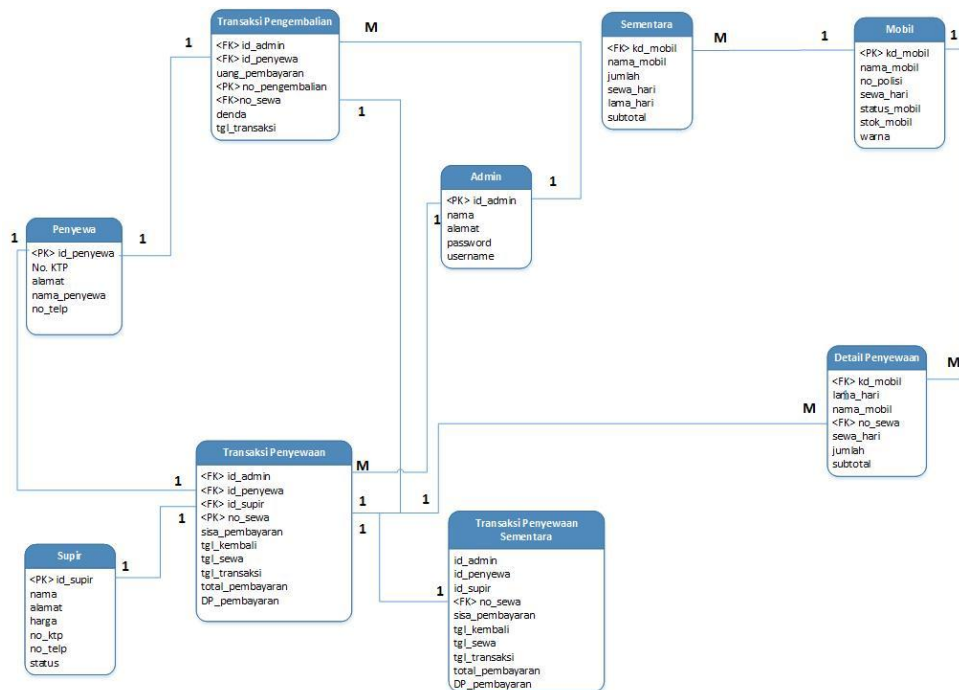


Gambar 1.7 Entity Relationship Diagram

1.6. Logical Record Structure (LRS)

Menurut Fridayanthie dan Mahdiati (2016:132) dalam (Lestari et al., n.d.), “*Logical Record Structure (LRS)* adalah representasi dari struktur *record – record* pada tabel – tabel yang terbentuk dari hasil relasi antar himpunan entitas”. Menentukan kardinalitas, jumlah tabel, dan *ForeignKey (FK)*.

Menurut (Yuniva & Maulina, 2018) “*Logical Record Structure (LRS)* adalah representasi dari struktur *record – record* pada tabel – tabel yang terbentuk dari hasil relasi antar himpunan entitas”.

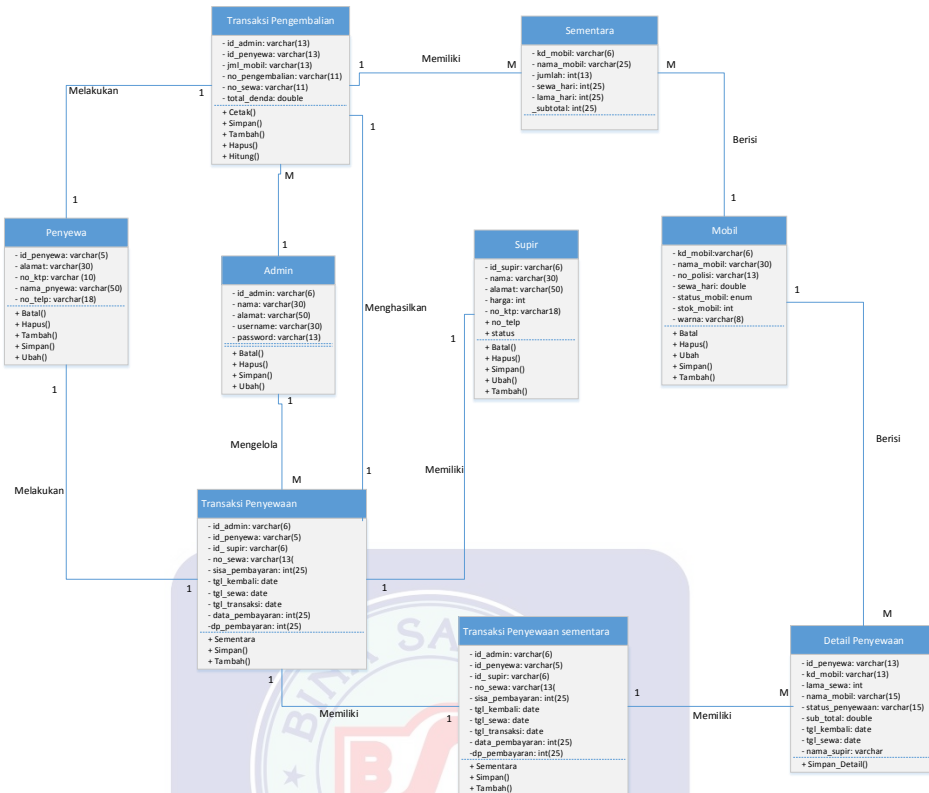


Gambar 1.8 Logical Record Structure

1.7. Class Model / Class Diagrams

Menurut Sukamto;M.Shalahuddin. (2015:141) “*Class Diagram* Menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

Menurut Indrajani. (2015). “*Class Diagram* digunakan untuk menggambarkan perbedaan yang mendasar antara class – class, hubungan antar-class, dan dimana sub-sistem kelas tersebut. Pada kelas diagram terdapat nama kelas *operations*, serta asosiasi (hubungan antar-kelas)”.



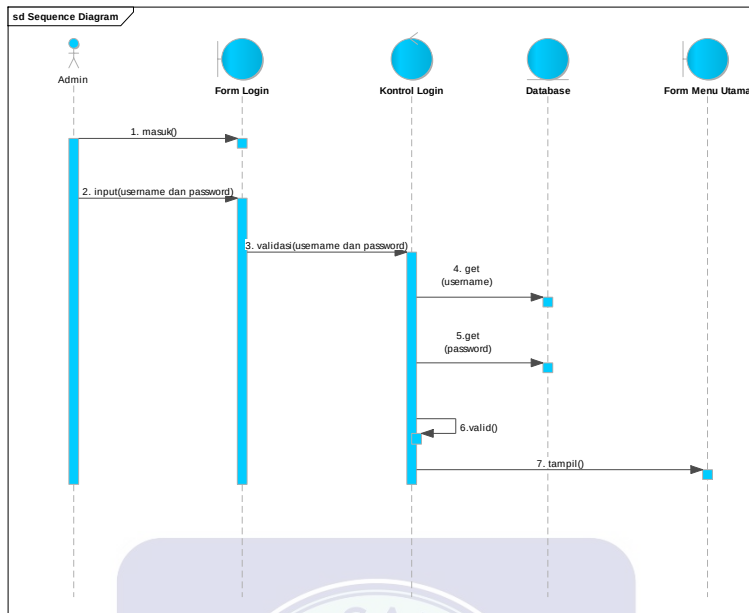
Gambar I.9 Class Diagram

1.8. Sequence Diagram

Menurut Sukamto;M.Shalahuddin. (2015:165) “Diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan di terima antar objek”.

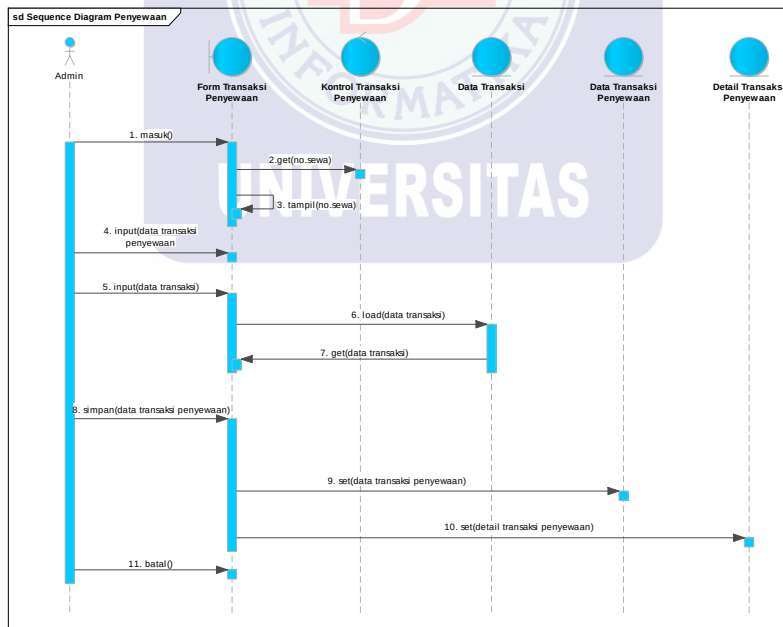
Menurut Hariyanto, B. (2017:29). “*Diagram Sequence* adalah pandangan grafis satu skenario yang menunjukkan interaksi dalam objek basis data, yaitu mana yang lebih dulu dan mana yang berikutnya. *Diagram Sequence* mengukuhkan peran objek dan membantu menyediakan informasi esensi untuk menentukan tanggung jawab (*responsibilities*) dan antar muka objek”.

1. Rancangan *Sequence Diagram Login*



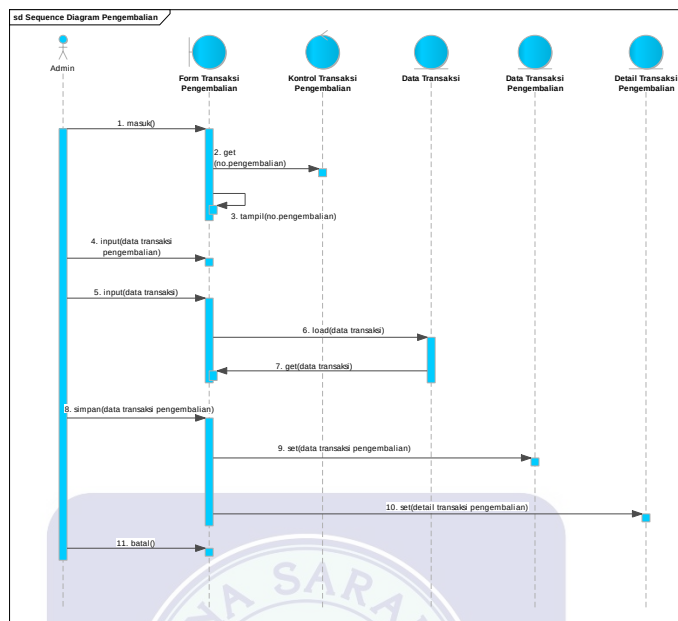
Gambar I.10 Rancangan *Sequence Diagram Login*

2. Rancangan *Sequence Diagram Transaksi Penyewaan*



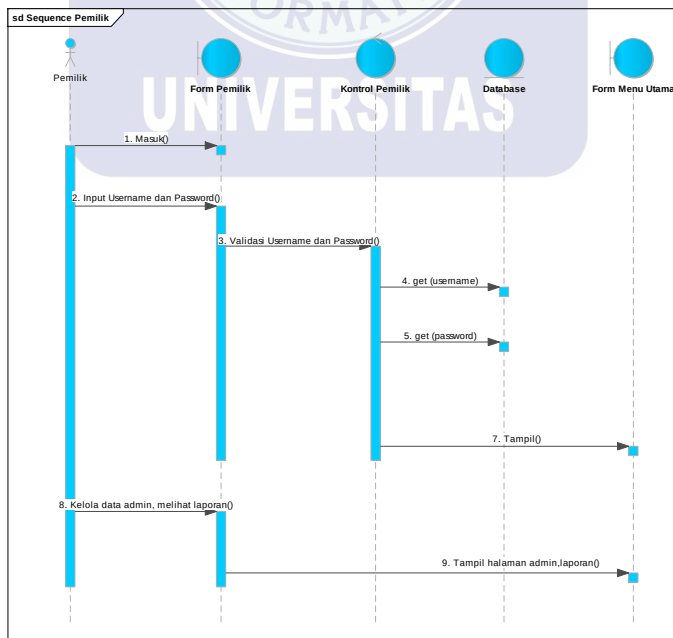
Gambar I.11 Rancangan *Sequence Diagram Transaksi Penyewaan*

3. Rancangan *Sequence Diagram* Transaksi Pengembalian



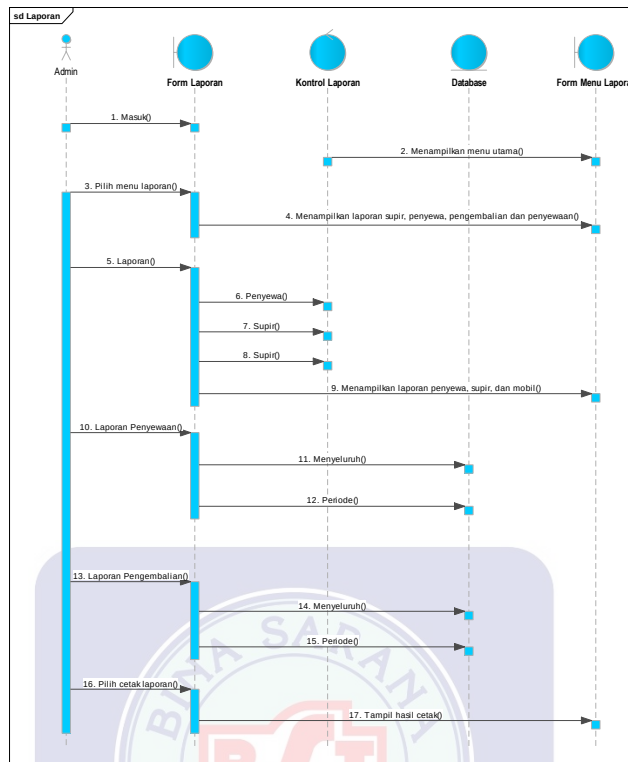
Gambar I.12 Rancangan *Sequence Diagram* Transaksi Pengembalian

4. Rancangan *Sequence Diagram* Pemilik



Gambar 1.13. Rancangan *Sequence Diagram* Pemilik

5. Rancangan Sequence Diagram Laporan



Gambar 1.14. Rancangan Sequence Diagram Laporan

1.9. Spesifikasi *Hardware* dan *Software*

1. Spesifikasi *Hardware*

a. *Server*

1. CPU

- (a) *Processor corei3*
- (b) *RAM sebesar 4 GB*
- (c) *Hard disk 500 GB*

2. *Mouse*

3. *Keyboard*

4. Monitor dengan resolusi layar minimum 1024x768 pixels

b. *Client*

1. CPU

- (a) *Processor minimal pentium 4*
- (b) *RAM sebesar 2 GB*

(c) *Hard Disk 500 GB*

2. *Mouse*

3. *Keyboard*

4. Monitor dengan resolusi layar minimum 1024x768 pixels

2. Spesifikasi Software

a. *Server*

1. Sistem operasi yang umum digunakan seperti : *Microsoft Windows, Mac OS X 10.45*, dll. Minimal Windows 7

2. *JDK-8u144*

3. *Netbeans 8.2*

4. *Xampp 3.3.2*

5. *iReport 4.5.0 plugin*

6. *iReport 4.5.0 plugin*

b. *Client*

1. Sistem operasi yang umum digunakan seperti : *Microsoft Windows, Mac OS X 10.45*, dll. Minimal Windows 7.

