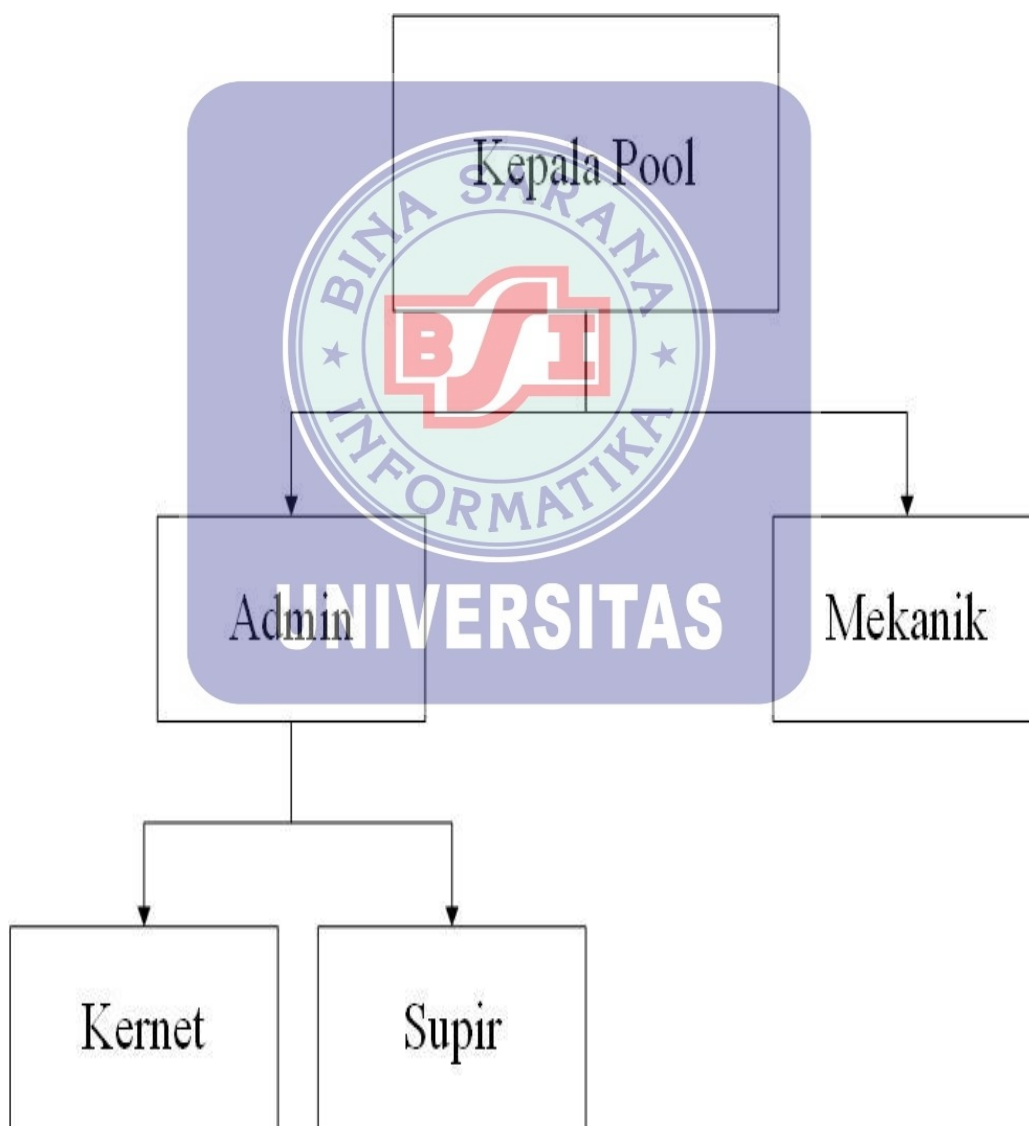


Karawang lebih tepat nya beralamat di Jl Raya Klari, Anggadita Kecamatan Klari Kabupaten Karawang Jawa Barat 41371 dengan nomor Surat Izin Usaha (SIU) : 1636 / 24.1.1 / 31.72 / 1.824.271 / 2015.

3.1.2. Struktur Organisasi

Setiap organisasi maupun perusahaan memiliki tingkatan kerja atau jabatan yang membatasi setiap ruang lingkup pekerjaan, begitu pula pada PO Agra Mas ini. Berikut merupakan struktur organisasi PO Agra Mas Karawang:



Sumber : PO Agra Mas (2019)

Gambar III.1.
Struktur Organisasi PO Agra Mas Karawang

Tugas dan wewenang dari tiap-tiap bagian adalah:

1. Kepala pool

- a. Mengontrol dan memonitor seluruh personil yang ada di bawahnya agar menjalankan tugas dan tanggung jawab sesuai dengan job dan prosedur yang ada.
- b. Mengontrol dan memonitor kinerja Pool yang menjadi tanggung jawabnya supaya dapat berjalan secara efisien.
- c. Menjaga seluruh asset Pool yang menjadi tanggung jawabnya.
- d. Memonitor pengeluaran biaya rutin Pool dan pencatatannya.
- e. Memastikan tercapainya target yang telah ditetapkan sesuai dengan waktu yang telah ditentukan.
- f. Melakukan pelaporan rutin setiap bulannya atas segala sesuatu kegiatan juga proses transaksi yang ada di Pool kepada kantor pusat.

2. Admin

- a. Melaksanakan tugas yang diberikan oleh kepala Pool.
- b. Mencatat data transaksi yang dilakukan pada saat penyewaan bus.
- c. Mengolah data keuangan transaksi yang dilakukan pada saat penyewaan.
- d. Mengontrol kelayakan kendaraan yang beroperasi.

3. Mekanik

- a. Melaksanakan tugas yang diberikan oleh bagian admin.
- b. Melakukan perawatan kendaraan sesuai job order yang diberikan bagian admin.
- c. Melaporkan hasil kegiatan kepada bagian admin.

4. Sopir dan Kernet

- a. Sopir bertugas mengemudi bus sesuai dengan standar operasional yang berlaku
- b. Kernet bertugas membantu sopir dalam perjalanan yang di tempuh.
- c. Kernet bertanggung jawab atas barang penumpang maupun fasilitas juga kelayakan bus pada saat dalam perjalanan.

3.2. Tinjauan Kasus

Pada tinjauan kasus ini penulis menjelaskan tentang analisa sistem berjalan penyewaan bus yang terjadi pada PO Agra Mas Karawang.

3.2.1. Proses Bisnis

Proses bisnis merupakan proses penyewaan yang terjadi, dan tahapan apa saja yang terjadi apabila penyewa akan melakukan proses penyewaan bus. Proses bisnis sistem berjalan sebagai berikut:

1. Prosedur Pemesanan Bus

Penyewa mendatangi kantor PO Agra Mas Karawang kemudian administrasi bus wisata memberikan brosur yang di dalamnya terdapat tujuan wisata, lama penyewaan, *seat*, jenis pelayanan sesuai dengan harga yang diinginkan dan tertera pada brosur tersebut. Setelah penyewa berkonsultasi dan telah mendapatkan kesepakatan yang terbaik, bagian administrasi mencatat data penyewaan pada form penyewaan bus yang terdiri dari dua rangkap lembar penyewaan (satu lembar untuk arsip kantor dan lembar untuk laporan ke kantor pusat). Selain dicatat dalam form penyewaan bus bagian administrasi mencatat ulang pada buku penyewaan, proses pencatatan penyewaan diwajibkan mencantumkan nama lengkap dan nomor telepon penyewa yang dapat dihubungi.

2. Prosedur Pembayaran Penyewaan Bus

Setelah proses pemesanan dilakukan bagian administrasi akan menanyakan apakah biaya akan dibayar penuh atau separuh dari harga yang telah disepakati. Jika penyewa melakukan pembayaran penuh maka bagian administrasi akan memberikan kwitansi yang dikeluarkan langsung oleh perusahaan (berlogo perusahaan), apabila penyewa membayar separuh harga dari yang telah disepakati maka bagian administrasi akan memberikan kwitansi sementara (kwitansi biasa yang digunakan pada umumnya), pembayaran dilakukan secara tunai (*cash*) dan dilunasi sebelum tanggal pemberangkatan. Selain pada kwitansi, bagian administrasi mencatat kembali data pembayaran pada buku penyewaan.

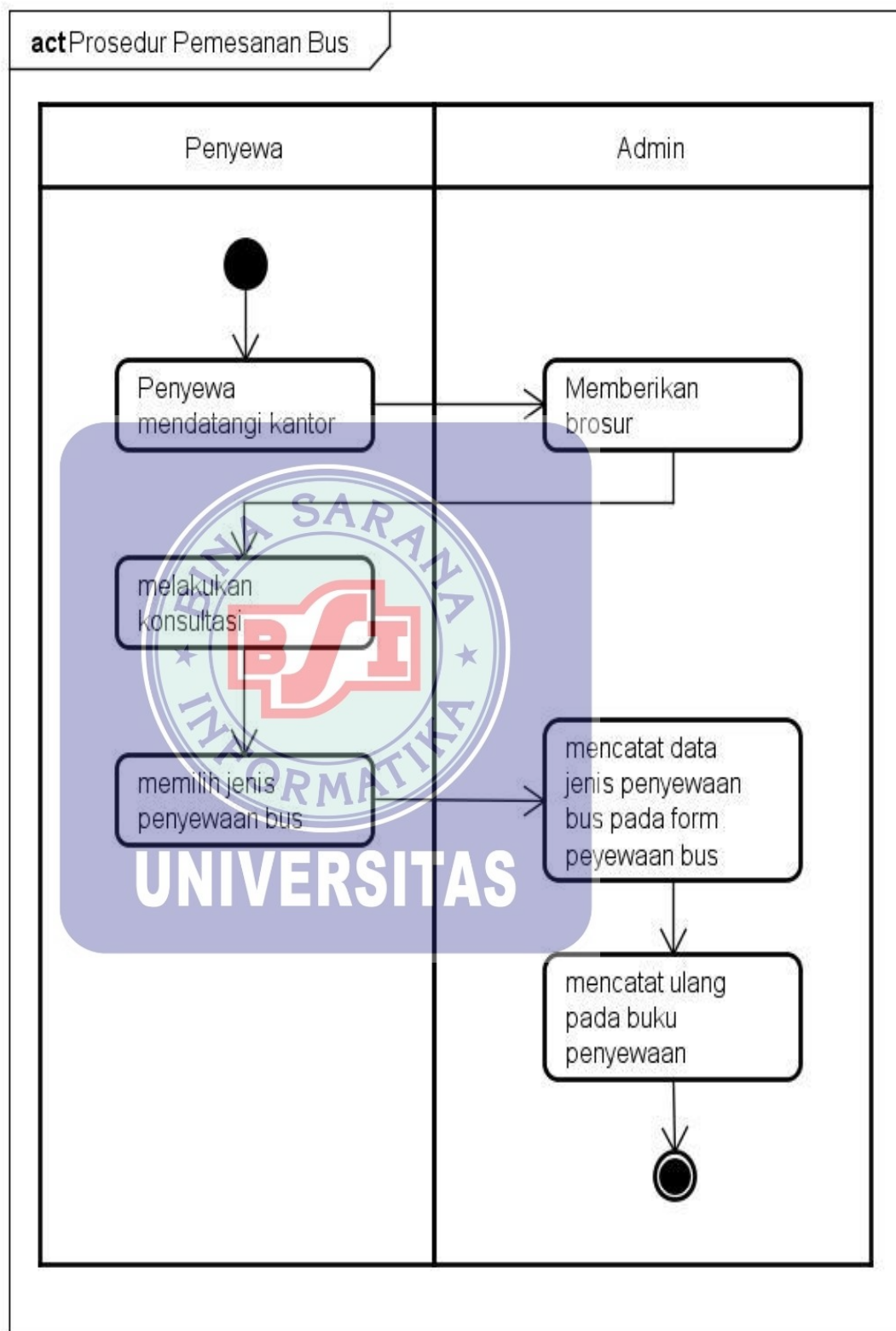
3. Prosedur Pembuatan Laporan Penyewaan Bus

Pembuatan laporan dilakukan setelah proses penyewaan selesai, laporan diambil dari satu form penyewaan bus, kemudian lembar tersebut diberikan kepada kepala pool dan kepala pool akan memeriksa kembali apakah telah sesuai dengan buku penyewaan atau tidak. Setelah itu kepala pool akan menandatangani sebagai tanda menyetujui kemudian kepala pool membawa form penyewaan bus ke kantor pusat PO Agra Mas untuk didata kembali oleh kantor pusat dan diarsipkan di kantor pusat.

3.2.2. Activity Diagram

Activity diagram atau diagram aktivitas merupakan suatu diagram yang menunjukkan alur atau proses transaksi yang terjadi pada suatu perusahaan. Berikut merupakan *activity diagram* sistem berjalan yang terjadi pada PO Agra Mas Karawang:

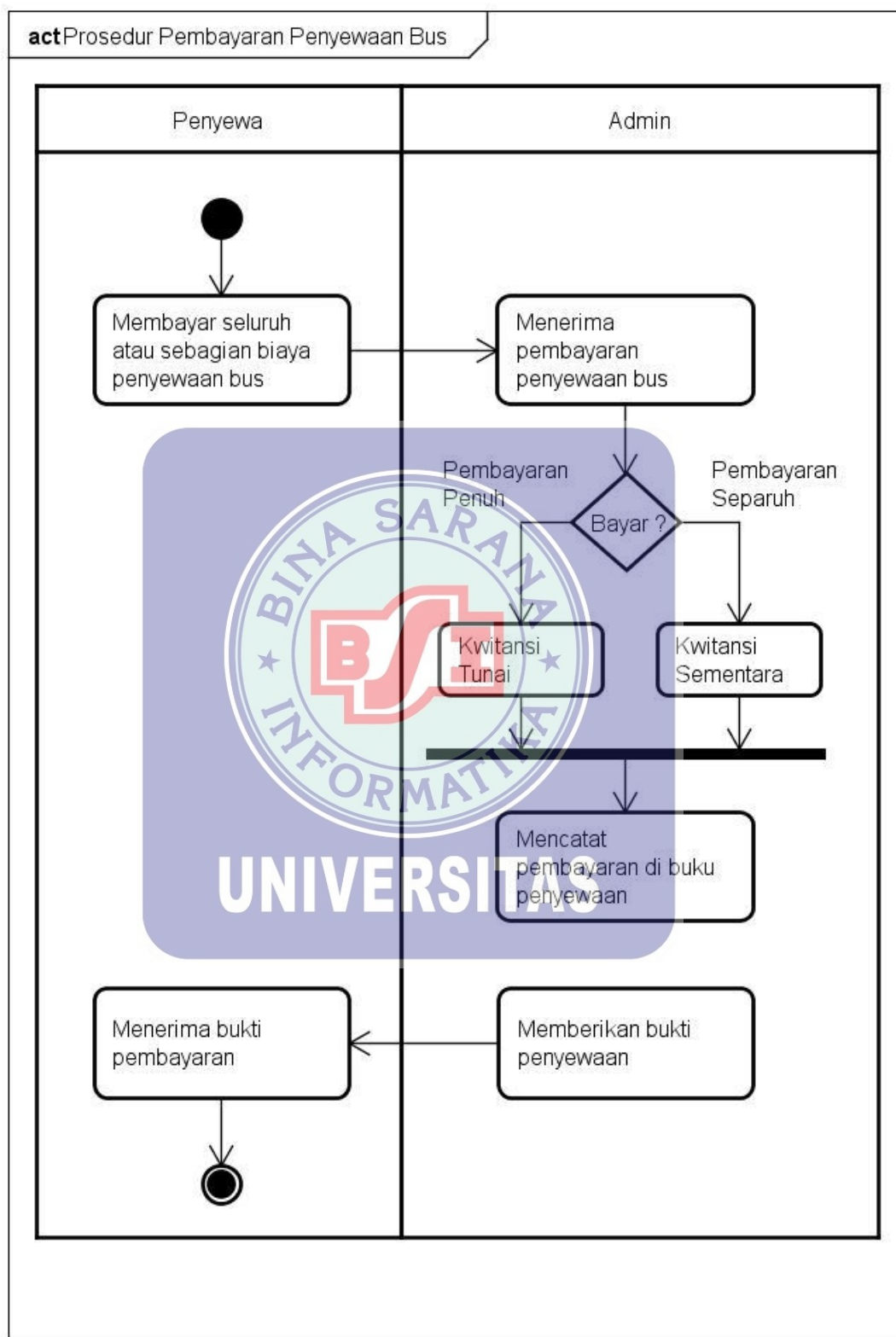
1. Prosedur Pemesanan Bus



Gambar III.2.

Activity Diagram Prosedur Pemesanan Bus Sistem Berjalan

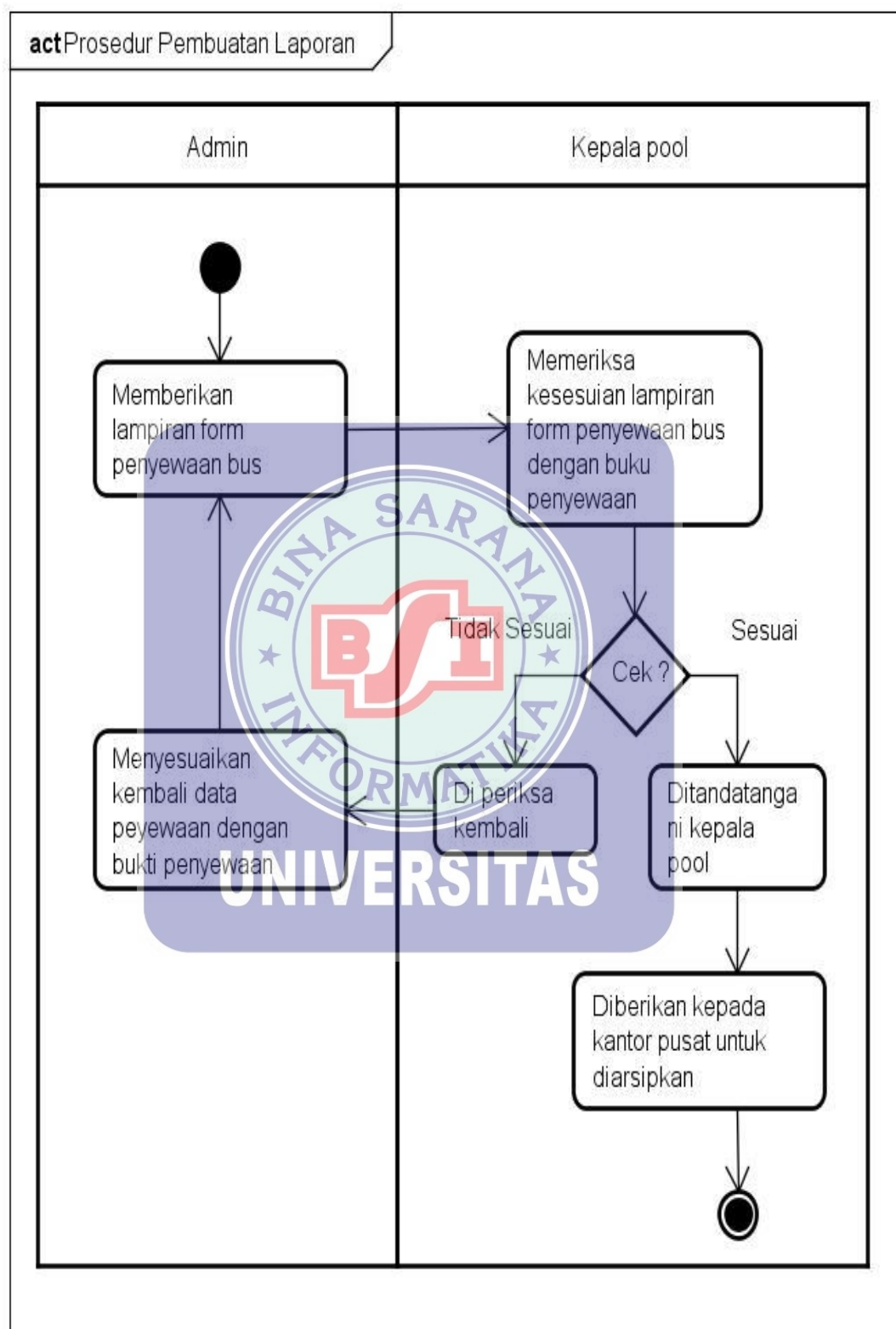
2. Prosedur Pembayaran Penyewaan Bus



Gambar III.3.

Activity Diagram Pembayaran penyewaan Bus Sistem Berjalan

3. Prosedur Pembuatan Laporan Penyewaan Bus



Gambar III.4.

Activity Diagram Pembayaran penyewaan Bus Sistem Berjalan

3.2.3. Dokumen Masukan

Dokumen masukan adalah data yang didapat dari penyewa yang akan dicatat kembali oleh bagian administrasi, berikut merupakan dokumen masukan PO Agra Mas Karawang:

1. Nama Dokumen : Form Penyewaan Bus
 - Fungsi : Sebagai pendataan penyewaan bus
 - Sumber : Penyewa
 - Tujuan : Bagian Administrasi
 - Media : Kertas
 - Jumlah : 2 (Dua) Rangkap
 - Frekuensi : Setiap terjadi transaksi penyewaan bus
2. Nama Dokumen : Buku Penyewaan
 - Fungsi : Sebagai arsip data penyewaan bus dan data pembayaran sewa
 - Sumber : Bagian Administrasi
 - Tujuan : Bagian Administrasi
 - Media : Kertas
 - Jumlah : 1 Lembar
 - Frekuensi : Setiap terjadi transaksi penyewaan dan pembayaran bus

3.2.4. Dokumen Keluaran

Dokumen keluaran merupakan dokumen yang dihasilkan atau dikeluarkan oleh bagian administrasi, berikut adalah dokumen keluaran PO Agra Mas Karawang:

1. Nama Dokumen : Kwitansi Sementara
 - Fungsi : Sebagai bukti pembayaran sementara apabila hanya

sebagian yang telah dilunasi

- | | |
|-----------|--|
| Sumber | : Bagian Administrasi |
| Tujuan | : Penyewa |
| Jumlah | : 1 Lembar |
| Frekuensi | : Setiap terjadi pembayaran pertama (Sebagian pelunasan) |
2. Nama Dokumen : Kwitansi Tunai
- | | |
|-----------|--|
| Fungsi | : Sebagai Bukti Pembayaran Tunai |
| Sumber | : Bagian Administrasi |
| Tujuan | : Penyewa |
| Jumlah | : 1 Lembar |
| Frekuensi | : Setiap Terjadi Pembayaran Tunai Maupun Pelunasan |
3. Nama Dokumen : Buku Penyewaan
- | | |
|-----------|---|
| Fungsi | : Sebagai arsip data penyewaan bus dan data pembayaran sewa |
| Sumber | : Bagian Administrasi |
| Tujuan | : Kepala Pool |
| Jumlah | : 1 Lembar |
| Frekuensi | : Setiap Terjadi Penyetoran dan Pembayaran Penyewaan bus |

3.2.5. Permasalahan Pokok

Hasil penelitian selama melakukan riset dapat disimpulkan bahwa proses penyewaan kas pada PO Agra Mas masih kurang baik, juga proses pengolahan data yang masih menggunakan kertas memungkinkan terjadinya penumpukan kertas.

Permasalahan yang terjadi pada proses penyewaan bus PO Agra Mas Karawang ini adalah:

1. Proses penawaran yang masih menggunakan brosur membuat menutupnya peluang pemasaran penyewaan bus yang terjadi.
2. Proses pencatatan penyewaan bus yang masih menggunakan media kertas dan ditulis tangan memungkinkan terjadinya kekeliruan dalam mencatat data dan juga dapat memungkinkan rusaknya berkas.
3. Kurang efektifnya pembuatan laporan yang masih dibuat secara manual membuat terbuangnya waktu dan kurang

3.2.6. Pemecahan Masalah

Pemecahan masalah atau solusi yang penulis sarankan untuk permasalahan yang ada pada Po Agra Mas adalah sebagai berikut:

1. Memanfaatkan kemajuan teknologi dengan menggunakan UML sehingga mempermudah proses pemasaran penyewaan bus.
2. Penggunaan sistem informasi akuntansi atau program dengan aplikasi *java* yang sudah dibuat oleh penulis, agar proses pendataan ataupun pencatatan pada saat penyewaan bus PO Agra Mas Kawarang bisa beroperasi dengan lebih cepat dan baik lagi.
3. Dengan menggunakan aplikasi yang penulis buat tentunya mempercepat proses pengolahan sehingga mempersingkat waktu dan biaya yang terjadi selama proses penyewaan berlangsung.

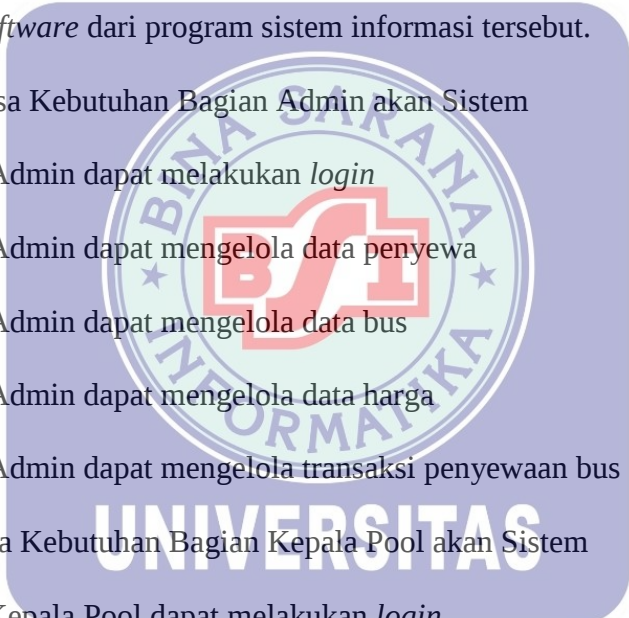
3.3. Analisis Kebutuhan Software

Analisa kebutuhan *software* merupakan tahapan pertama yang harus dilakukan dalam pembuatan sistem yang akan kita butuhkan. Dimana penulis mengembangkan sistem yang telah terkomputerisasi maupun belum terkomputerisasi

sehingga dapat menghasilkan suatu sistem yang lebih baik lagi dan dapat mempercepat proses kerja yang terjadi sebelumnya.

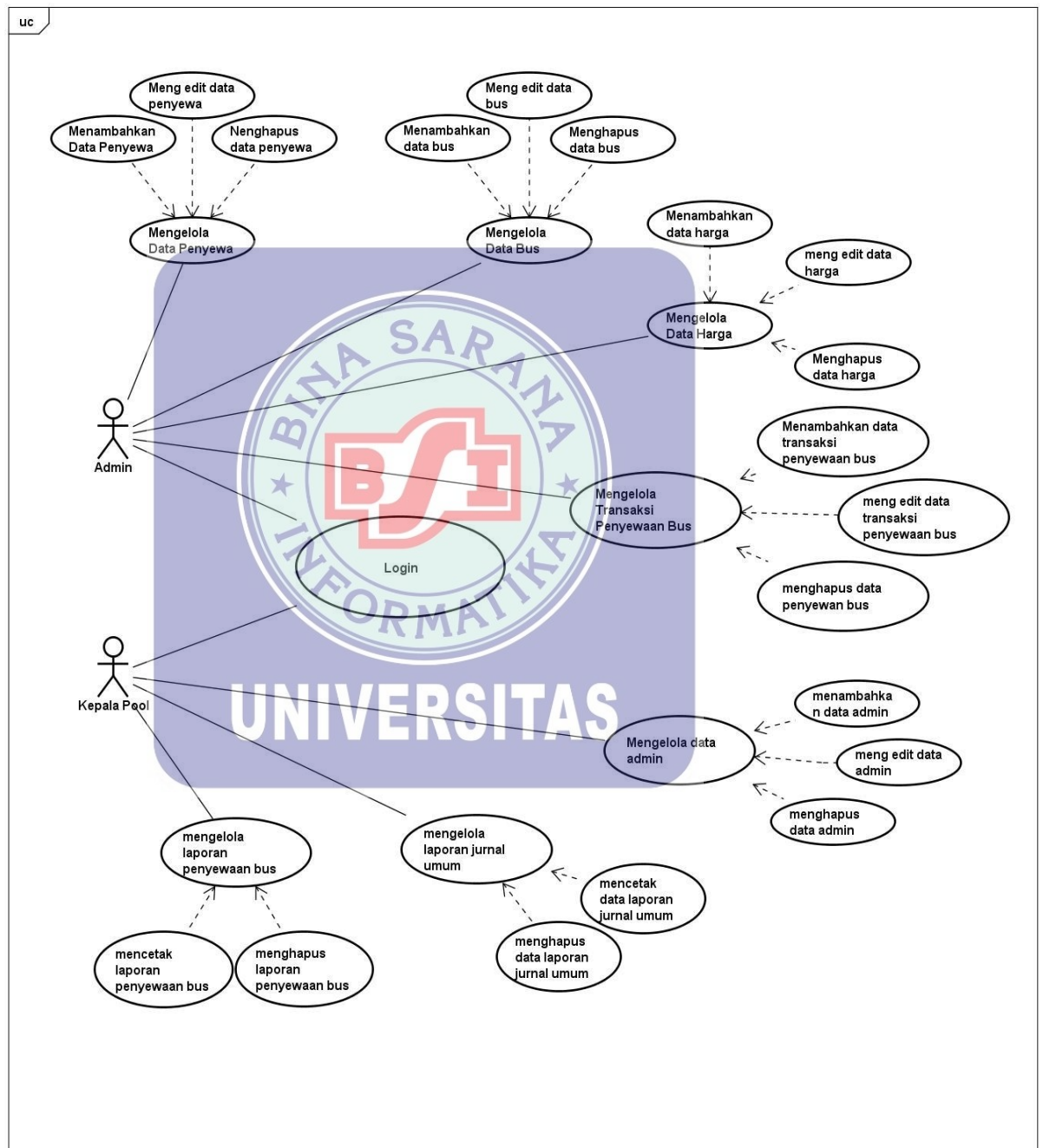
3.3.1. Analisis Kebutuhan

Pada PO Agra Mas Karang proses pencatatan maupun pendataan masih belum terkomputerisasi, masih dicatat didalam lembar yang memiliki lampiran disetiap lembarannya. Agar proses pendataan penyewaan bisa berjalan dengan baik dan lebih efektif lagi, dibutuhkan suatu sistem yang dapat mendukung proses pendataan penyewaan bus dengan beberapa *user*. Berikut merupakan analisa kebutuhan *software* dari program sistem informasi tersebut.

- 
- A. Analisa Kebutuhan Bagian Admin akan Sistem
 - A.1. Admin dapat melakukan *login*
 - A.2. Admin dapat mengelola data penyewa
 - A.3. Admin dapat mengelola data bus
 - A.4. Admin dapat mengelola data harga
 - A.5. Admin dapat mengelola transaksi penyewaan bus
 - B. Analisa Kebutuhan Bagian Kepala Pool akan Sistem
 - B.1. Kepala Pool dapat melakukan *login*
 - B.2. Kepala Pool dapat mengelola data admin
 - B.3. Kepala Pool dapat mengakses laporan
 - B.4. Kepala Pool dapat mengakses jurnal umum

3.3.2. Use Case Diagram

Use Case Diagram adalah diagram yang digunakan untuk menjelaskan siapa yang menggunakan sistem dan apa saja yang bisa dilakukannya. Berikut merupakan *use case diagram* sistem usulan:



Gambar III.5.

***Use Case Diagram* penyewaan Bus Sistem Usulan**

Tabel III.1.

Deksripsi Use Case Diagram Login Bagian Admin

<i>Use case Name</i>	<i>Login Bagian Admin</i>
<i>Recruitment</i>	<i>Use case Login</i>
<i>Goal</i>	Bagian admin melakukan <i>login</i>
<i>Pre- Conditions</i>	Bagian admin telah melakukan <i>login</i>
<i>Post-conditions</i>	Bagian admin berhasil <i>login</i>
<i>Failed and Conditions</i>	Gagal melakukan <i>login</i>
<i>Primary Actor</i>	Bagian admin
<i>Main flow/ Basic path</i>	1. Bagian admin mengelola menu utama 2. Bagian admin mengelola data penyewa 3. Bagian admin mengelola data bus 4. Bagian admin mengelola harga bus 5. Bagian admin mengelola penyewaan bus
<i>Alternate invariant/ Invariant 1</i>	-
<i>Invariant 2</i>	-

Tabel III.2.

Deksripsi Use Case Diagram Data Penyewa Bagian Admin

<i>Use case Name</i>	<i>Data Penyewa Bagian Admin</i>
<i>Recruitment</i>	<i>Use case Data Penyewa</i>
<i>Goal</i>	Bagian admin mengelola data penyewa
<i>Pre- Conditions</i>	Bagian admin telah mengelola data penyewa
<i>Post-conditions</i>	Bagian admin berhasil mengelola data penyewa

<i>Failed and Conditions</i>	Gagal mengelola data penyewa
<i>Primary Actor</i>	Bagian admin
<i>Main flow/ Basic path</i>	1. Bagian admin dapat menambahkan data penyewa 2. Bagian admin mengedit data penyewa
<i>Alternate invariant/ Invariant 1</i>	-
<i>Invariant 2</i>	-

Tabel III.3.

Deksripsi Use Case Diagram Data Bus Bagian Admin

<i>Use case Name</i>	Data Bus Bagian Admin
<i>Recruitment</i>	<i>Use case Data Bus</i>
<i>Goal</i>	Bagian admin mengelola data bus
<i>Pre- Conditions</i>	Bagian admin telah mengelola data bus
<i>Post-conditions</i>	Bagian admin mengelola data bus
<i>Failed and Conditions</i>	Gagal mengelola data bus
<i>Primary Actor</i>	Bagian admin
<i>Main flow/ Basic path</i>	1. Bagian admin dapat menambah data bus 2. Bagian admin mendedit data bus
<i>Alternate invariant/ Invariant 1</i>	-
<i>Invariant 2</i>	-

Tabel III.4.

Deksripsi Use Case Diagram Data Harga Bus Bagian Admin

<i>Use case Name</i>	Data Harga Bus Bagian Admin
<i>Recruitment</i>	<i>Use case Data Harga Bus</i>

<i>Goal</i>	Bagian admin mengelola data harga bus
<i>Pre- Conditions</i>	Bagian admin telah mengelola data harga bus
<i>Post-conditions</i>	Bagian admin mengelola data harga bus
<i>Failed and Conditions</i>	Gagal mengelola data harga bus
<i>Primary Actor</i>	Bagian admin
<i>Main flow/ Basic path</i>	1. Bagian admin dapat menambah data harga bus 2. Bagian admin mendedit data harga bus
<i>Alternate invariant/ Invariant 1</i>	-
<i>Invariant 2</i>	-

Tabel III.5.

Deksripsi Use Case Diagram Data Penyewaan Bus Bagian Admin

<i>Use case Name</i>	Data Penyewaan Bus Bagian Admin
<i>Recruitment</i>	Use case Data Penyewaan Bus
<i>Goal</i>	Bagian admin mengelola data penyewaan bus
<i>Pre- Conditions</i>	Bagian admin telah mengelola data penyewaan bus
<i>Post-conditions</i>	Bagian admin mengelola data penyewaan bus
<i>Failed and Conditions</i>	Gagal mengelola data penyewaan bus
<i>Primary Actor</i>	Bagian admin
<i>Main flow/ Basic path</i>	1. Bagian admin dapat menambah data penyewaan bus 2. Bagian admin mendedit data penyewaan bus
<i>Alternate invariant/ Invariant 1</i>	-
<i>Invariant 2</i>	-

Tabel III.6.

Deksripsi Use Case Diagram Login Bagian Kepala Pool

<i>Use case Name</i>	<i>Login Bagian Kepala Pool</i>
<i>Recruitment</i>	<i>Use case Login</i>
<i>Goal</i>	Bagian kepala pool melakukan <i>login</i>
<i>Pre- Conditions</i>	Bagian kepala pool telah melakukan <i>login</i>
<i>Post-conditions</i>	Bagian kepala pool berhasil <i>login</i>
<i>Failed and Conditions</i>	Gagal melakukan <i>login</i>
<i>Primary Actor</i>	Bagian kepala pool
<i>Main flow/ Basic path</i>	1. Bagian kepala pool mengelola menu utama 2. Bagian kepala pool mengelola data admin 3. Bagian kepala pool mengakses data laporan penyewaan 4. Bagian kepala pool mengakses data jurnal umum
<i>Alternate invariant/ Invariant 1</i>	-
<i>Invariant 2</i>	-

Tabel III.7.

Deksripsi Use Case Diagram Data Admin Bagian Kepala Pool

<i>Use case Name</i>	<i>Data Admin Bagian Kepala Pool</i>
<i>Recruitment</i>	<i>Use case Data Admin</i>
<i>Goal</i>	Bagian kepala pool mengelola data admin
<i>Pre- Conditions</i>	Bagian kepala pool telah mengelola data admin
<i>Post-conditions</i>	Bagian kepala pool berhasil mengelola data admin

<i>Failed and Conditions</i>	Gagal mengelola data admin
<i>Primary Actor</i>	Bagian kepala pool
<i>Main flow/ Basic path</i>	1. Bagian kepala pool menambahkan data admin 2. Bagian kepala pool dapat mengedit data admin
<i>Alternate invariant/ Invariant 1</i>	-
<i>Invariant 2</i>	-

Tabel III.8.

Deksripsi Use Case Diagram Laporan Penyewaan Bagian Kepala Pool

<i>Use case Name</i>	Laporan Penyewaan Bagian Kepala Pool
<i>Recruitment</i>	<i>Use case</i> Laporan Penyewaan
<i>Goal</i>	Bagian kepala pool mengakses laporan penyewaan
<i>Pre- Conditions</i>	Bagian kepala pool telah mengakses laporan penyewaan
<i>Post-conditions</i>	Bagian kepala pool berhasil mengakses laporan penyewaan
<i>Failed and Conditions</i>	Gagal mengakses laporan penyewaan
<i>Primary Actor</i>	Bagian kepala pool
<i>Main flow/ Basic path</i>	Bagian kepala pool mencetak laporan penyewaan
<i>Alternate invariant/ Invariant 1</i>	-
<i>Invariant 2</i>	-

Tabel III.9.

Deksripsi Use Case Diagram Jurnal Umum Bagian Kepala Pool

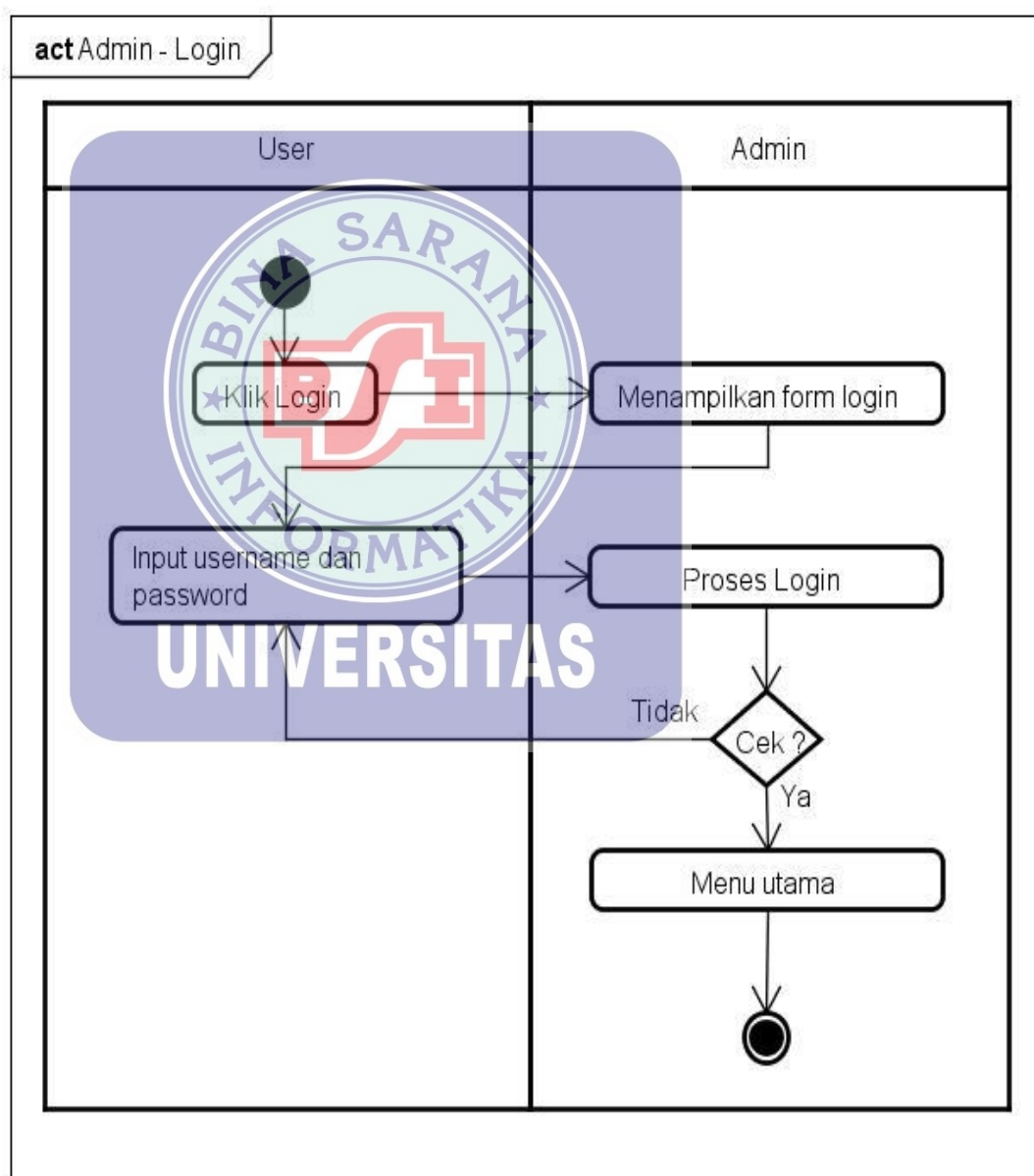
<i>Use case Name</i>	Jurnal Umum Bagian Kepala Pool
<i>Recruitment</i>	<i>Use case</i> Jurnal Umum
<i>Goal</i>	Bagian kepala pool mengakses jurnal umum
<i>Pre- Conditions</i>	Bagian kepala pool telah mengakses jurnal umum
<i>Post-conditions</i>	Bagian kepala pool berhasil mengakses jurnal umum
<i>Failed and Conditions</i>	Gagal mengakses jurnal umum
<i>Primary Actor</i>	Bagian kepala pool
<i>Main flow/ Basic path</i>	Bagian kepala pool mencetak jurnal umum
<i>Alternate invariant/ Invariant 1</i>	-
<i>Invariant 2</i>	-

3.3.3. Activity Diagram

Activity Diagram sistem usulan ini menjelaskan gambaran atau urutan langkah-langkah yang dilakukan sistem dalam mengolah data yang dijalankan oleh pengguna. Berikut merupakan *activity diagram* admin dan kepala pool:

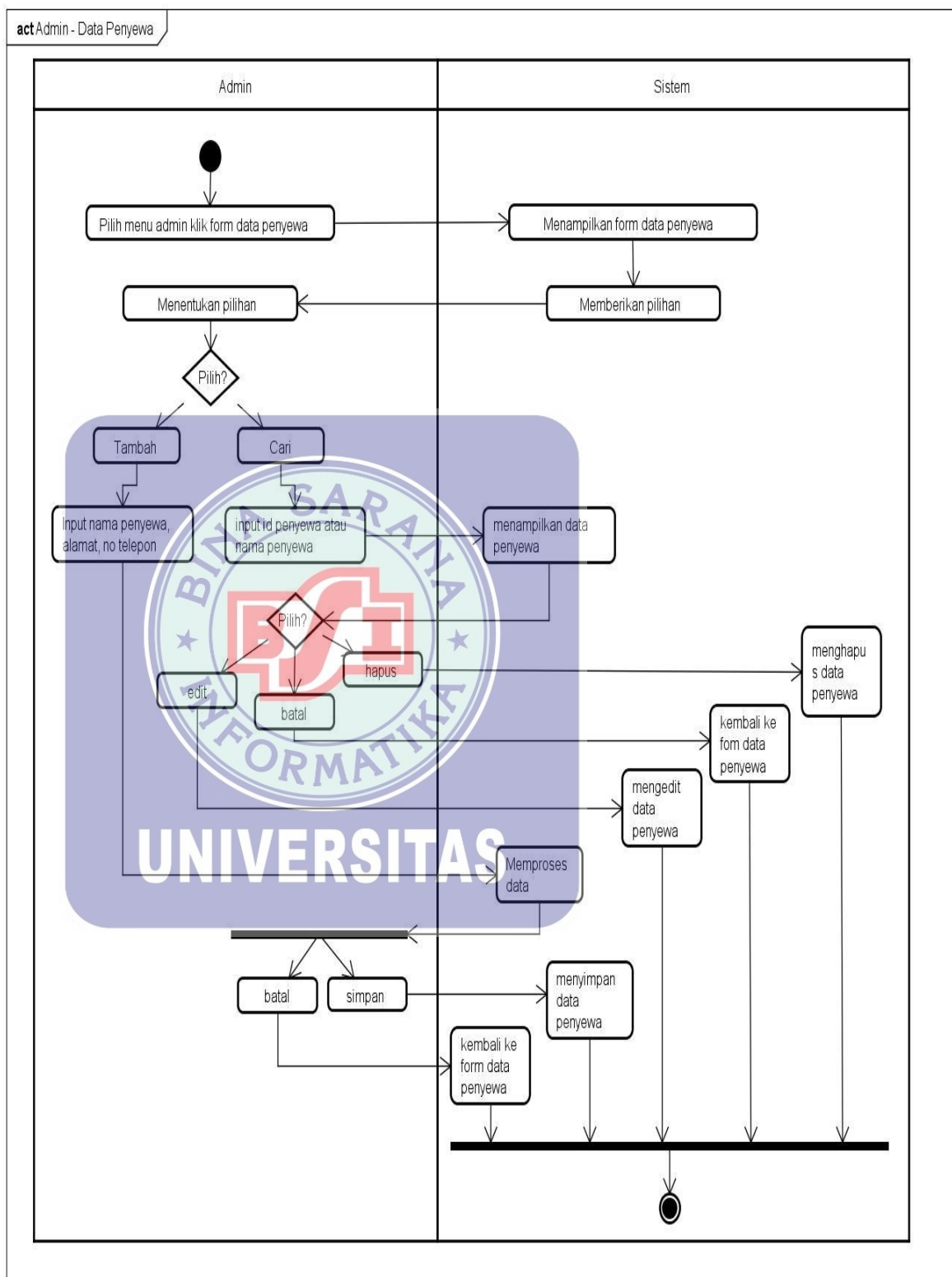
A. Activity Diagram Bagian Admin akan Sistem

A.1. Admin dapat melakukan login



Gambar III.6.
Activity Diagram Login Admin

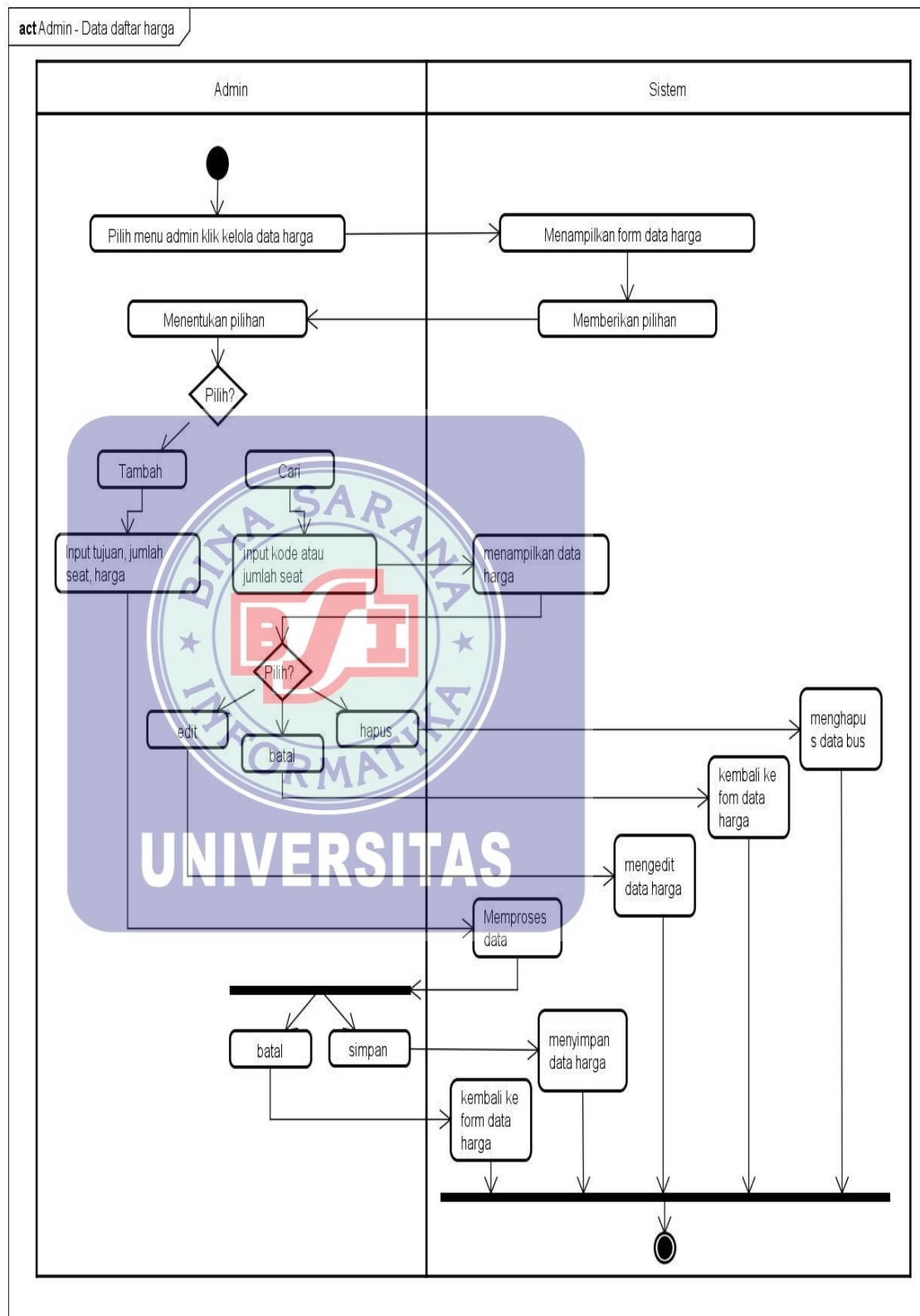
A.2. Admin dapat mengelola data penyewa



Gambar III.7.

Activity Diagram Admin Data Penyewa Sistem Usulan

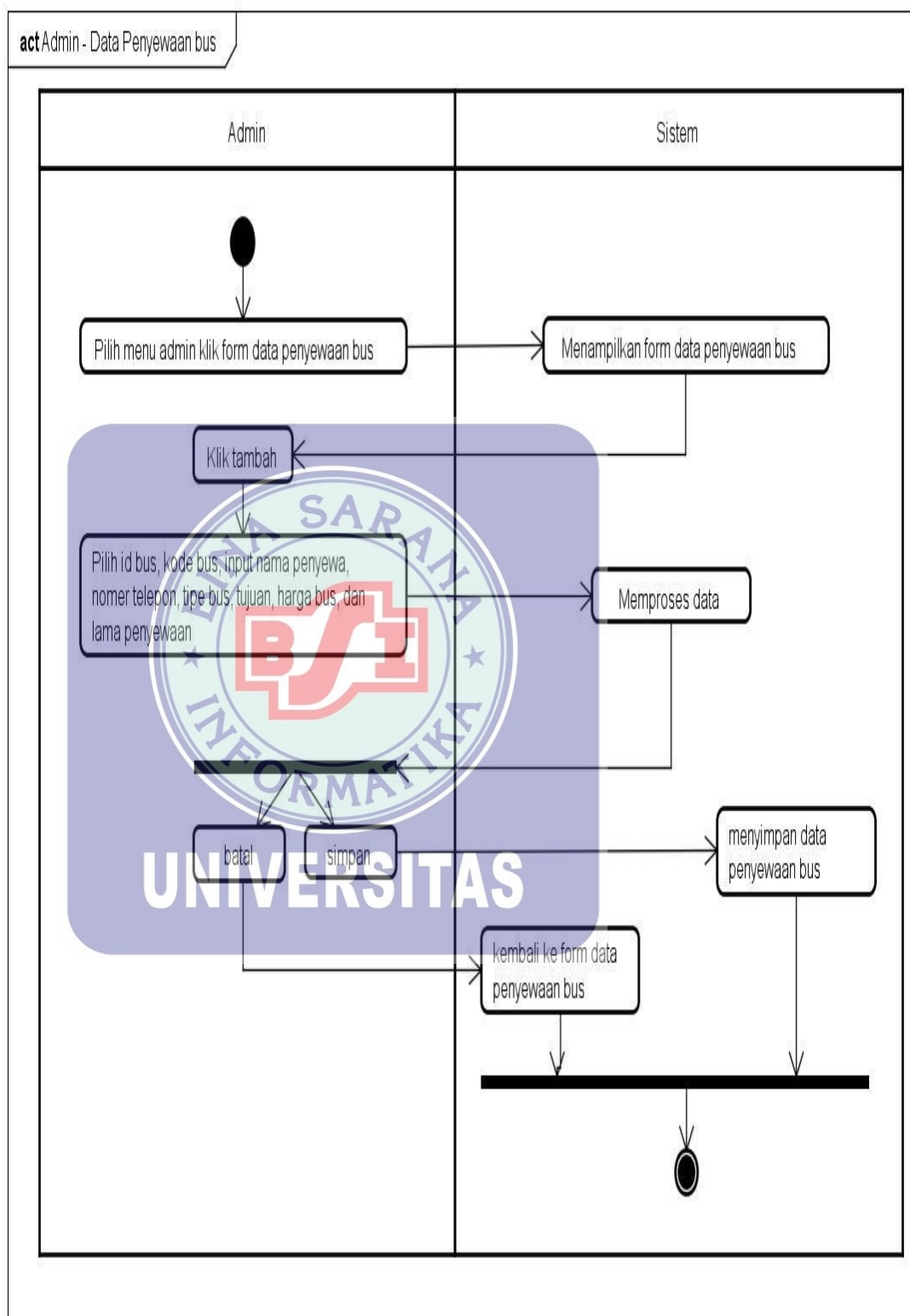
A.4. Admin dapat mengelola data harga



Gambar III.9.

Activity Diagram Admin Data Harga Sistem Usulan

A.5. Admin dapat mengelola data penyewaan bus

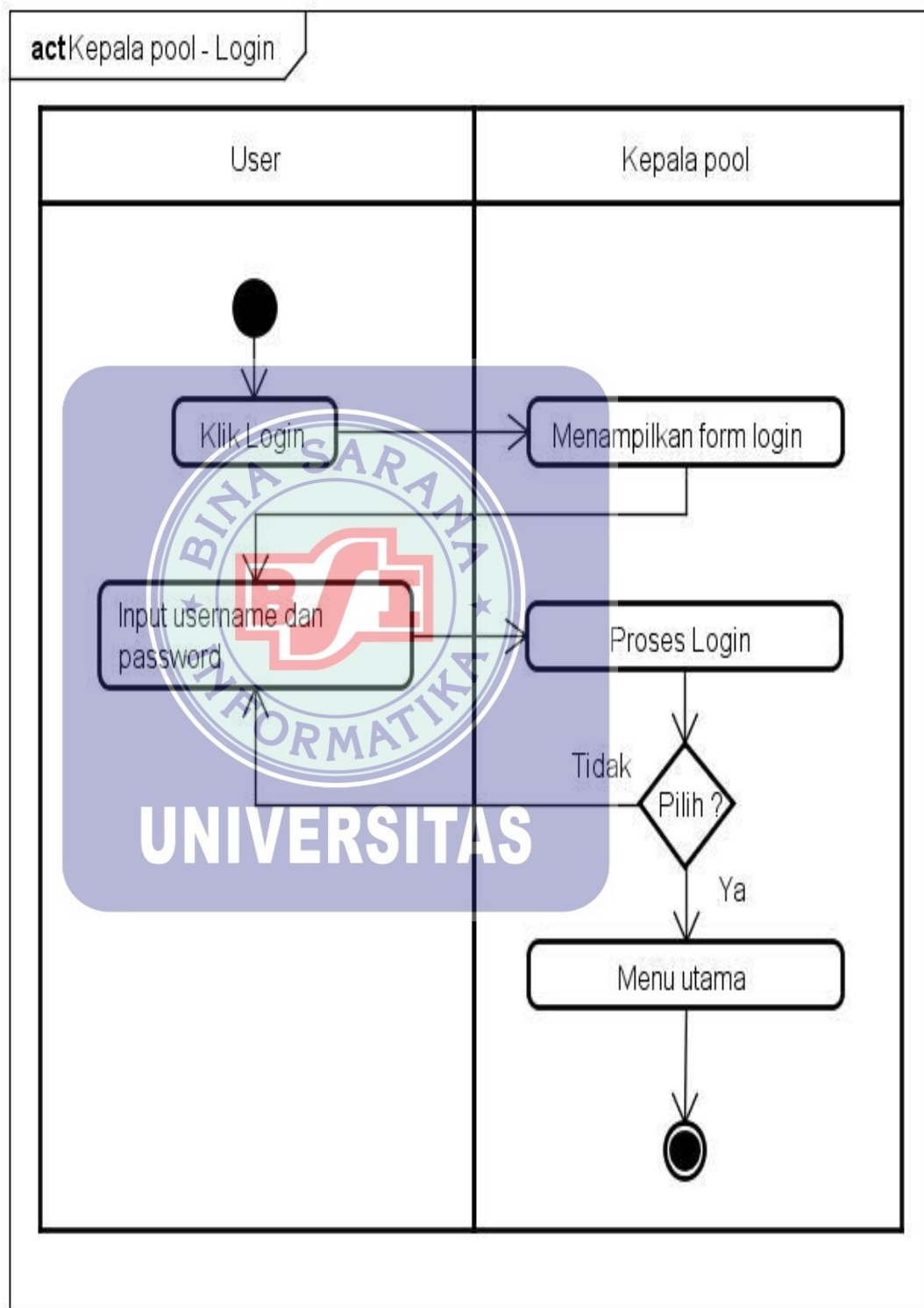


Gambar III.10.

Activity Diagram Admin Data Penyewaan Bus Sistem Usulan

B. Analisa Kebutuhan Bagian Kepala Pool akan Sistem

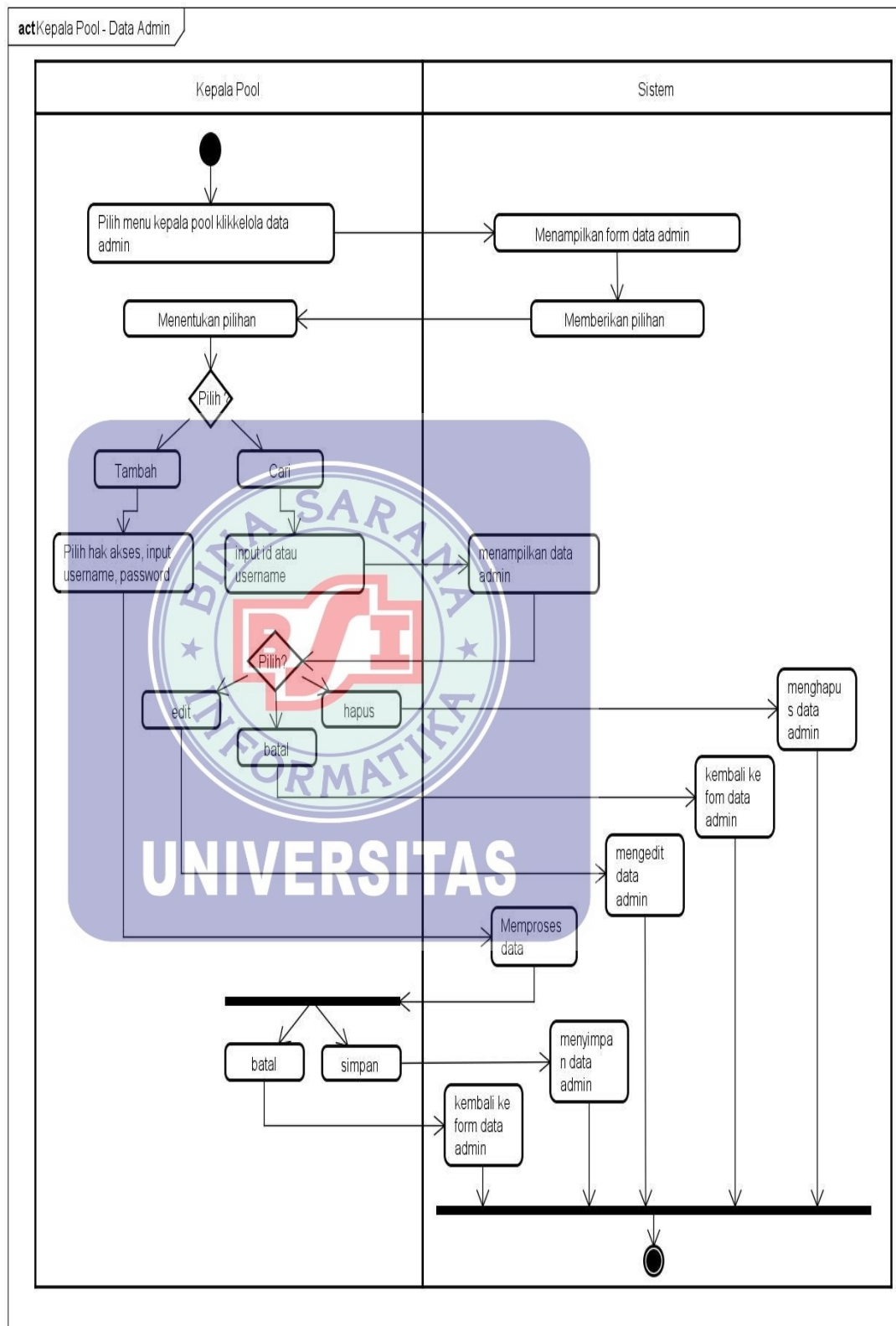
B.1. Kepala Pool dapat melakukan login



Gambar III.11.

Activity Diagram Login Kepala Pool/Pemilik

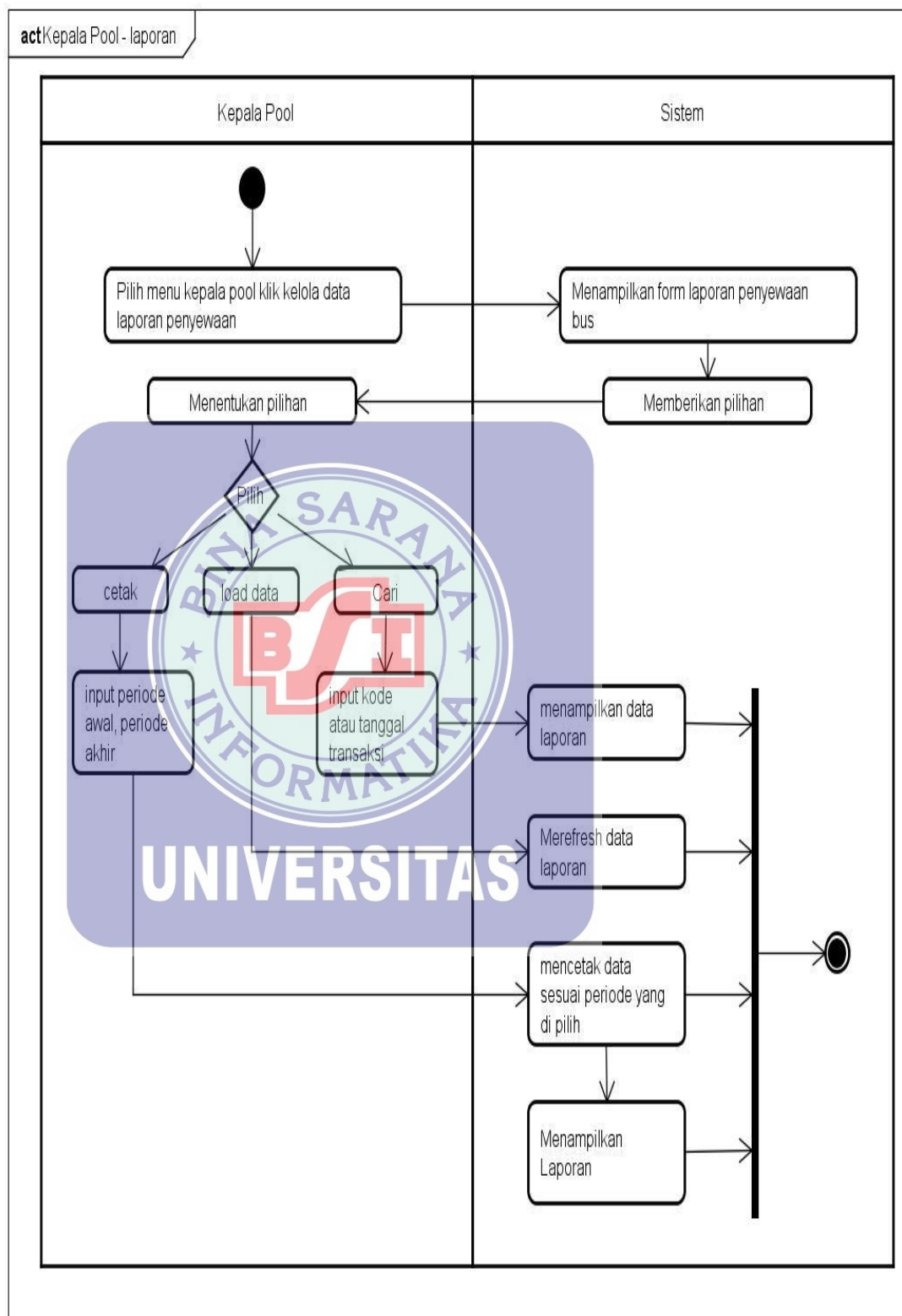
B.2. Kepala Pool dapat mengelola data admin



Gambar III.12.

Activity Diagram Kepala Pool Data Admin Sistem Usulan

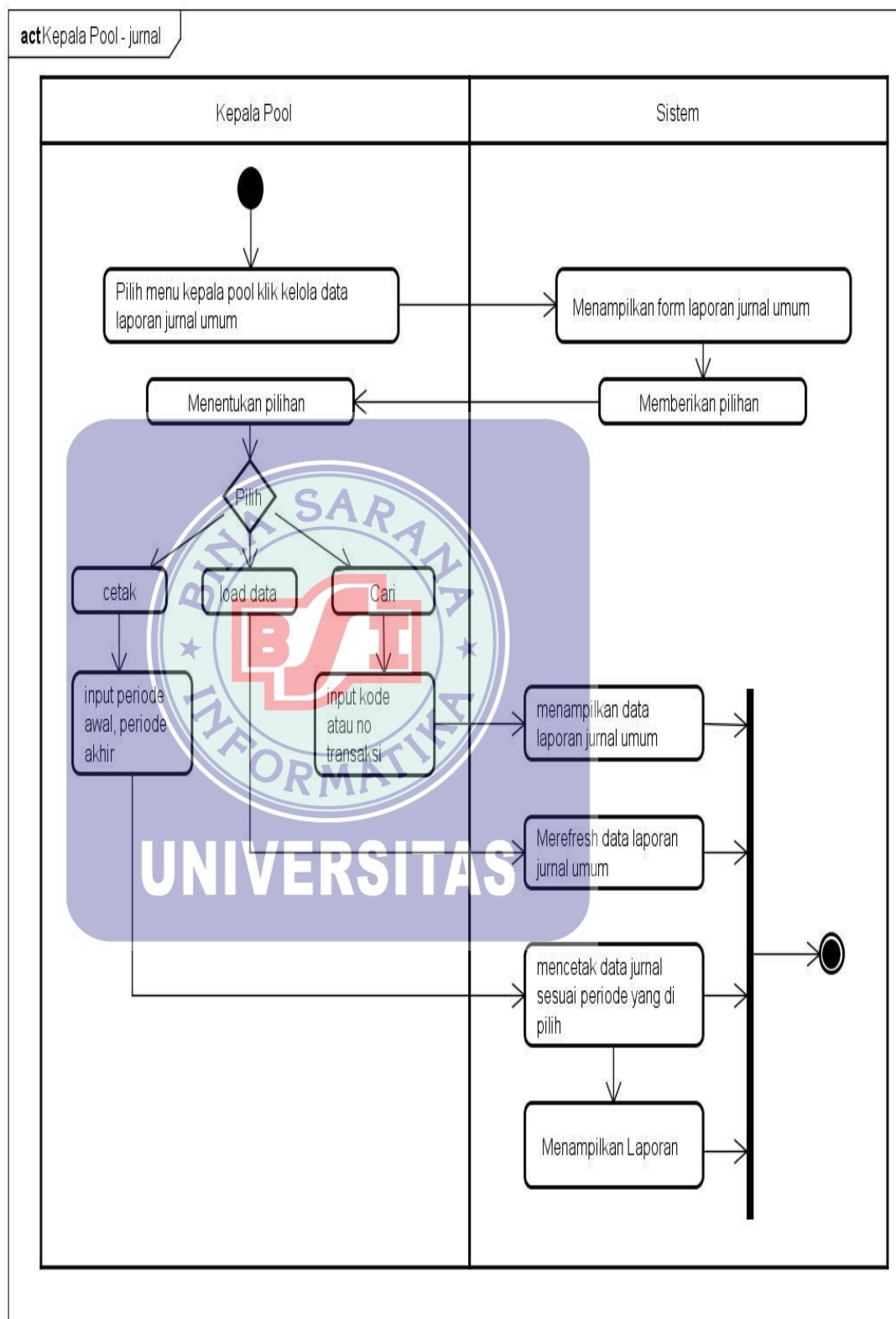
B.3. Kepala Pool dapat mengakses laporan



Gambar III.13.

Activity Diagram Kepala Pool Laporan Penyewaan Bus Sistem Usulan

B.4. Kepala Pool dapat mengakses jurnal umum



Gambar III.14.

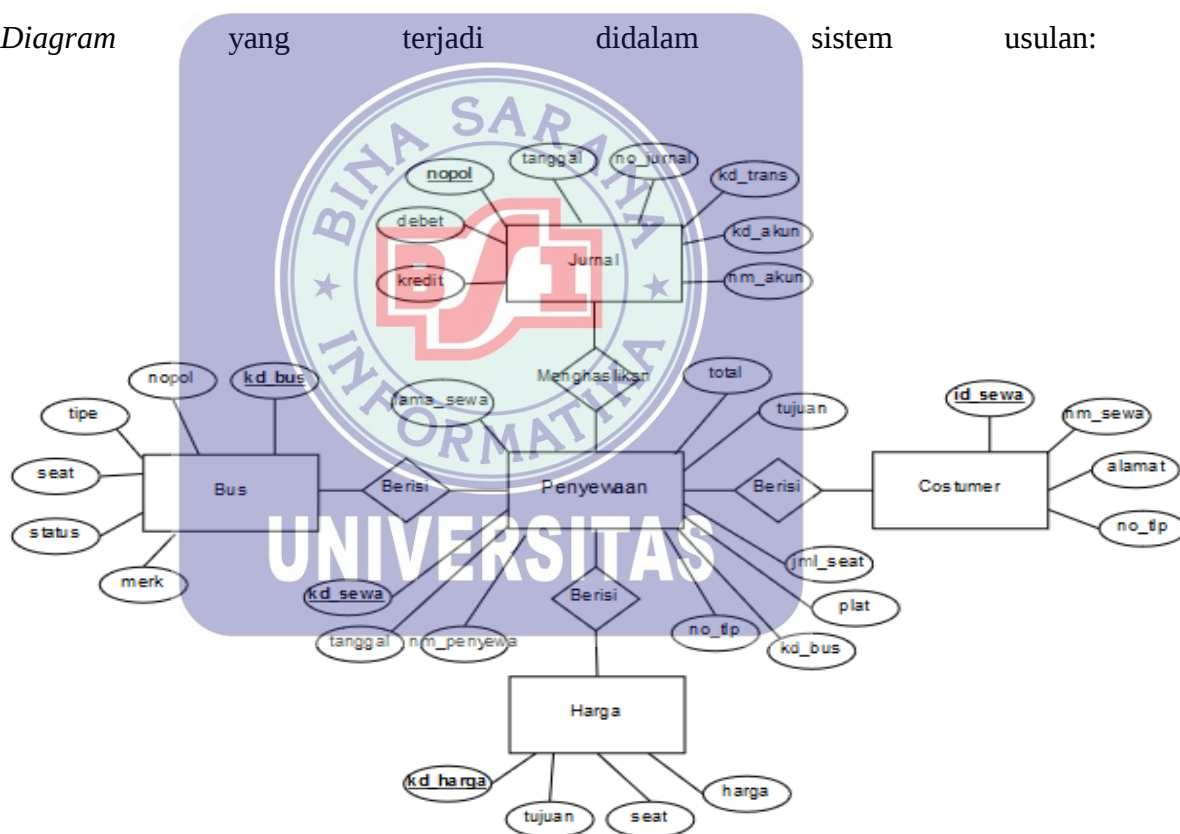
Activity Diagram Kepala Pool Laporan Jurnal Umum Sistem Usulan

3.4. Desain

Desain merupakan tahap penggambaran bentuk sistem yang akan dirancang, juga penjelasan mengenai rancangan yang penulis buat dan alur kegunaan atau fungsi dari setiap tampilan rancangan yang penulis buat.

3.4.1. Entity Relation Diagram (ERD)

Entity Relation Diagram merupakan diagram yang menggambarkan hubungan data dalam *database* yang saling berelasi antara satu objek dengan objek dasar lainnya yang ada di dalam *database*. Berikut merupakan *Entity Relation Diagram* yang terjadi didalam sistem usulan:

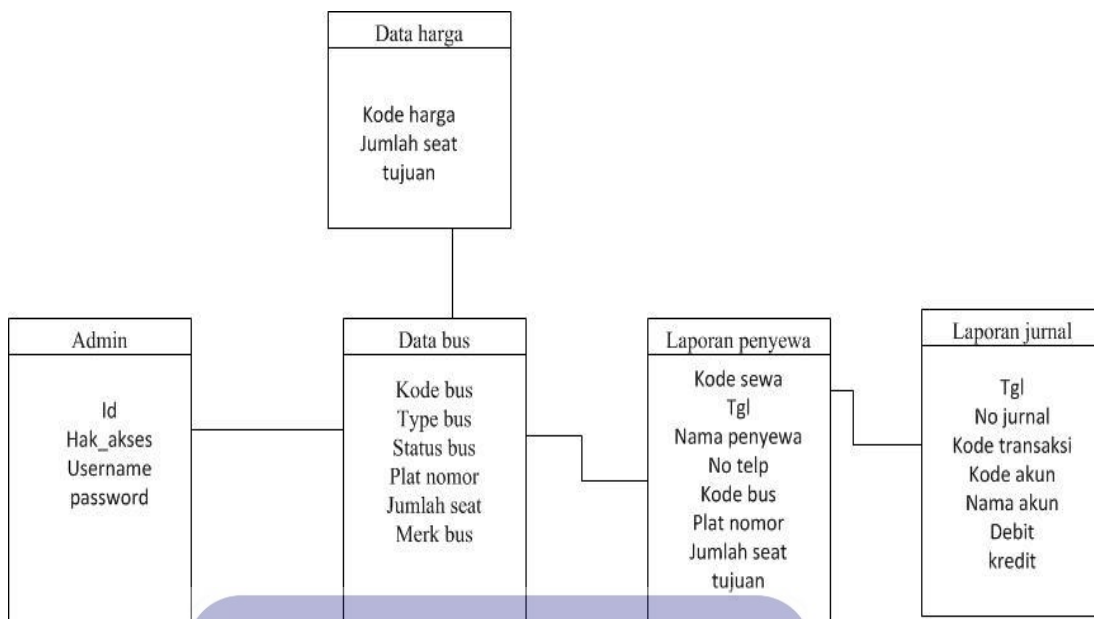


Gambar III.15.

Entity Relation Diagram Penyewaan Bus

3.4.2. Logical Record Structure (LRS)

Logical Record Structure merupakan gambaran urutan suatu sistem yang digambarkan dalam bentuk table. Berikut merupakan *Logical Record Structure* sistem usulan:



Gambar III.16.
Logical Record Structure Penyewaan Bus

3.4.3. Spesifikasi File

Spesifikasi *file* menjelaskan format yang diperlukan dalam pembuatan sistem dan sesuai dengan kebutuhan juga kegunaannya. Berikut merupakan spesifikasi file dari setiap sistem yang dibuat:

1. Spesifikasi *File* Tabel Admin

Tabel III.10.

Spesifikasi *File* Tabel Admin

No	Element Data	Nama <i>File</i>	<i>Type</i>	<i>Size</i>	Keterangan
1.	Id Admin	id_admin	<i>Int</i>	8	<i>Primary key</i>
2.	<i>Username</i>	<i>username</i>	<i>Varchar</i>	35	
3.	<i>Password</i>	<i>password</i>	<i>Varchar</i>	35	
4.	Hak Akses	hak_akses	<i>Varchar</i>	10	

Nama *File* : *File Admin*
 Fungsi : Digunakan untuk mengelola data pengguna
 Akronim : Data Admin
 Tipe *File* : *File Master*
 Akses *File* : *Random*
 Panjang *Record* : 88
 Kunci *File* : id_admin

2. Spesifikasi *File* Tabel Data Bus

Tabel III.11.
Spesifikasi *File* Tabel Data Bus

No	Element Data	Nama <i>File</i>	Type	Size	Keterangan
1.	Kode Bus	kd_bus	★ Int	8	Primary key
2.	Plat Nomor	nopol	Varchar	12	
3.	Tipe Bus	tipe	Varchar	20	
4.	Jumlah Seat	seat	Varchar	8	
5.	Status Bus	status	Varchar	15	
6.	Merk Bus	merk	Varchar	20	

Nama *File* : *File Data Bus*
 Fungsi : Digunakan untuk mengelola data bus
 Akronim : Data Bus
 Tipe *File* : *File Master*
 Akses *File* : *Random*
 Panjang *Record* : 83
 Kunci *File* : kd_bus

3. Spesifikasi *File* Tabel Data Penyewa**Tabel III.12.****Spesifikasi *File* Tabel Data Penyewa**

No	Element Data	Nama <i>File</i>	<i>Type</i>	<i>Size</i>	Keterangan
1.	Id Penyewa	id_sewa	<i>Int</i>	8	<i>Primary key</i>
2.	Nama Penyewa	nm_sewa	<i>Varchar</i>	35	
3.	Alamat	alamat	<i>Text</i>		
4.	No Tlp	no_tlp	<i>Varchar</i>	15	

Nama *File* : *File* Data Penyewa

Fungsi : Digunakan untuk mengelola data penyewa

Akronim : Data Penyewa

Tipe *File* : *File* Master

Akses *File* : *Random*

Panjang *Record* : 58

Kunci *File* : id_sewa

4. Spesifikasi *File* Tabel Data Harga**Tabel III.13.****Spesifikasi *File* Tabel Data Harga Bus**

No	Element Data	Nama <i>File</i>	<i>Type</i>	<i>Size</i>	Keterangan
1.	Kode Harga	kd_harga	<i>Int</i>	8	<i>Primary key</i>
2.	Tujuan Wisata	tujuan	<i>Text</i>		
3.	Jumlah <i>Seat</i>	<i>seat</i>	<i>Varchar</i>		
4.	Harga	int	<i>Int</i>	10	

Nama *File* : *File* Data Harga
 Fungsi : Digunakan untuk mengelola data harga bus
 Akronim : Data Harga Bus
 Tipe *File* : *File* Master
 Akses *File* : *Random*
 Panjang *Record* : 18
 Kunci *File* : kd_harga

5. Spesifikasi *File* Tabel Data Jurnal

Tabel III.14.
Spesifikasi *File* Tabel Data Jurnal

No	Element Data	Nama <i>File</i>	Type	Size	Keterangan
1.	Nomor	nomor	<i>Int</i>	5	<i>Primary key</i>
2.	Tanggal	tanggal	<i>date</i>		
3.	Nomor Jurnal	no_jurnal	<i>Varchar</i>	8	
4.	Kode Transaksi	kd_trans	<i>Varchar</i>	8	
5.	Kode Akun	kd_akun	<i>Varchar</i>	8	
6.	Nama Akun	nm_akun	<i>Varchar</i>	30	
7.	Debet	debet	<i>Int</i>	10	
8.	Kredit	kredit	<i>Int</i>	10	

Nama *File* : *File* Data Jurnal
 Fungsi : Digunakan untuk mengelola data jurnal umum
 Akronim : Data Harga Jurnal
 Tipe *File* : *File* Laporan
 Akses *File* : *Random*

Panjang *Record* : 89

Kunci *File* : nomor

6. Spesifikasi *File* Tabel Data Penyewaan Bus

Tabel III.15.

Spesifikasi *File* Tabel Data Penyewaan Bus

No	Element Data	Nama <i>File</i>	<i>Type</i>	<i>Size</i>	Keterangan
1.	Kode Sewa	kd_sewa	<i>Varchar</i>	8	<i>Primary key</i>
2.	Tanggal	tanggal	<i>date</i>		
3.	Nama Penyewa	nm_penyewa	<i>Varchar</i>	35	
4.	No Telepon	no_tlp	<i>Varchar</i>	15	
5.	Kode Bus	kd_bus	<i>Varchar</i>	8	
6.	Plat Nomor	plat	<i>Varchar</i>	10	
7.	Jumlah <i>Seat</i>	jml_seat	<i>Varchar</i>	5	
8.	Tujuan	Tujuan	<i>Text</i>		
9.	Lama Sewa	lama_sewa	<i>Int</i>	5	
10.	Total Biaya	Total	<i>Int</i>	10	

Nama *File* : *File* Data Penyewaan Bus

Fungsi : Digunakan untuk mengelola data penyewaan bus

Akronim : Data Penyewaan

Tipe *File* : *File* Transaksi

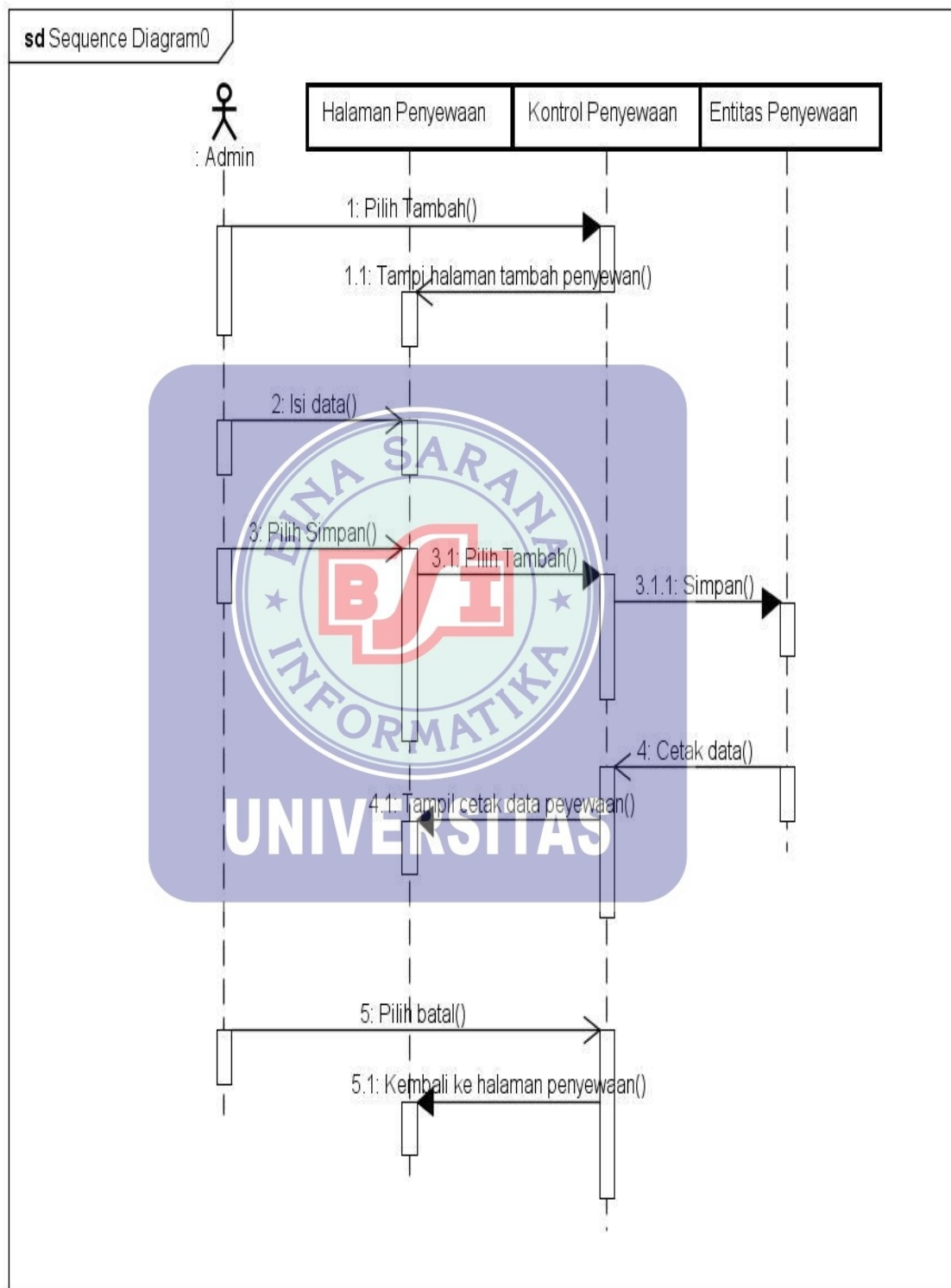
Akses *File* : *Random*

Panjang *Record* : 96

Kunci *File* : kd_sewa

3.4.4. Sequence Diagram

Berikut merupakan *sequence diagram* dalam sistem usulan PO Agra Mas:

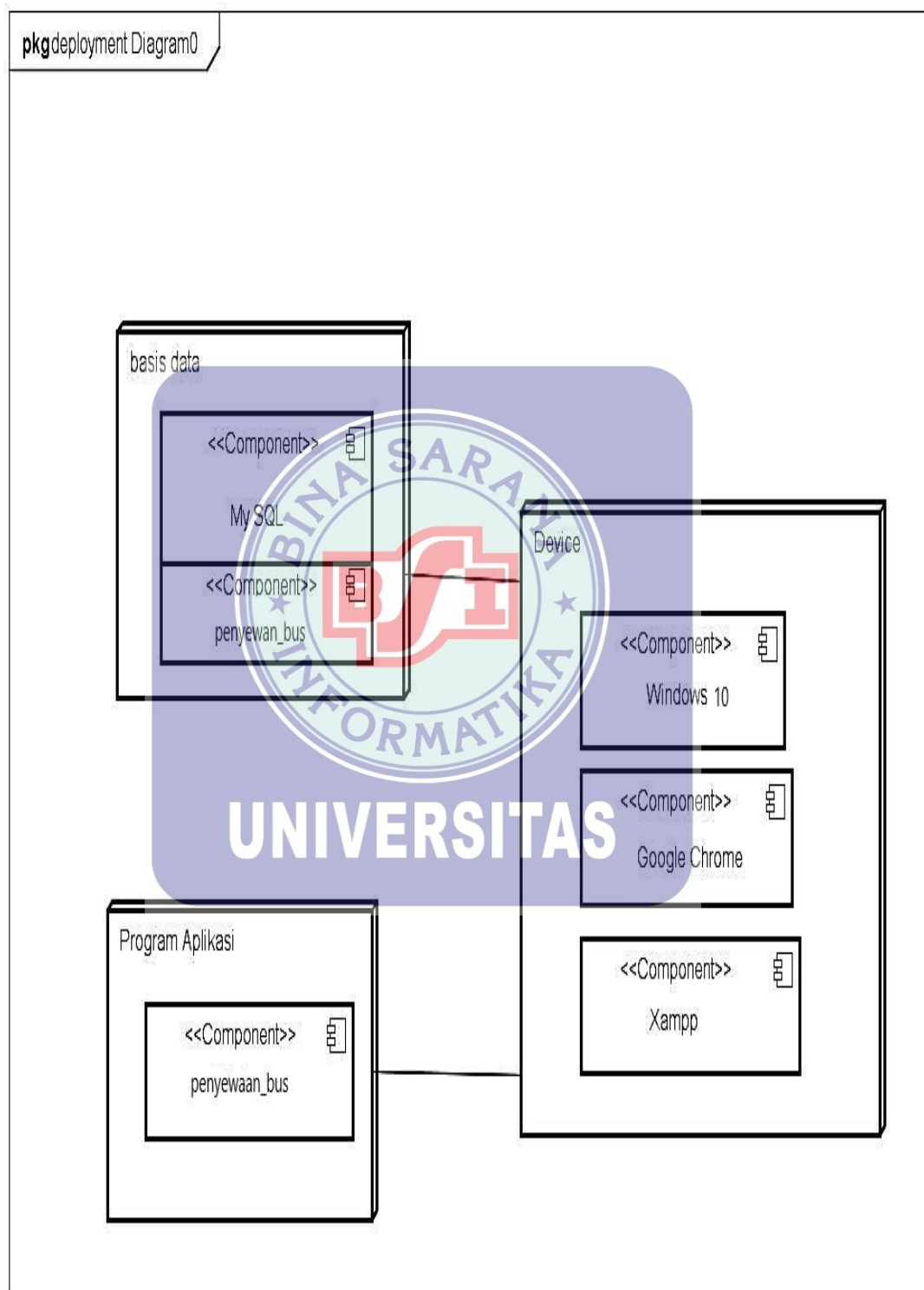


Gambar III.17.

Sequence Diagram Penyewaan Bus

3.4.5. Deployment Diagram

Berikut merupakan *deployment diagram* dalam sistem usulan PO Agra Mas:



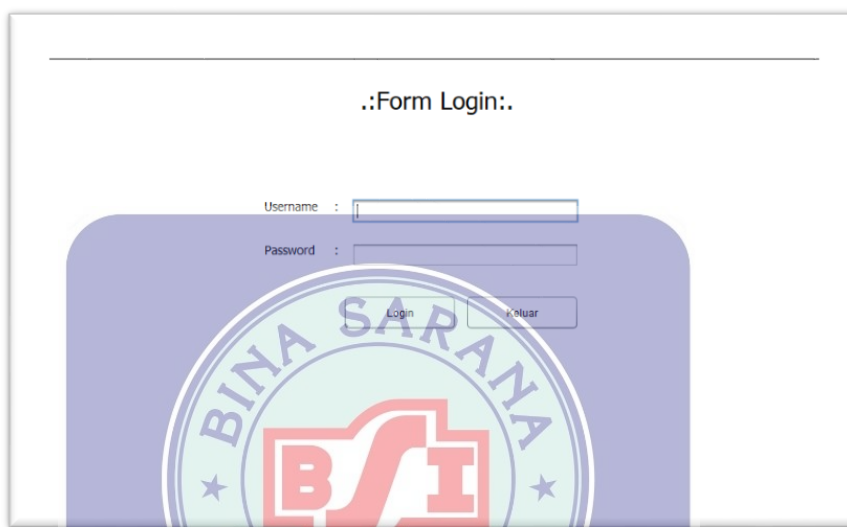
Gambar III.18.

Deployment Diagram Penyewaan Bus

3.4.6. User Interface

User Interface menggambarkan tampilan dari setiap halaman di dalam sebuah sistem. Berikut merupakan *User Interface* yang terjadi di dalam sistem usulan sebagai berikut:

a. Form Login

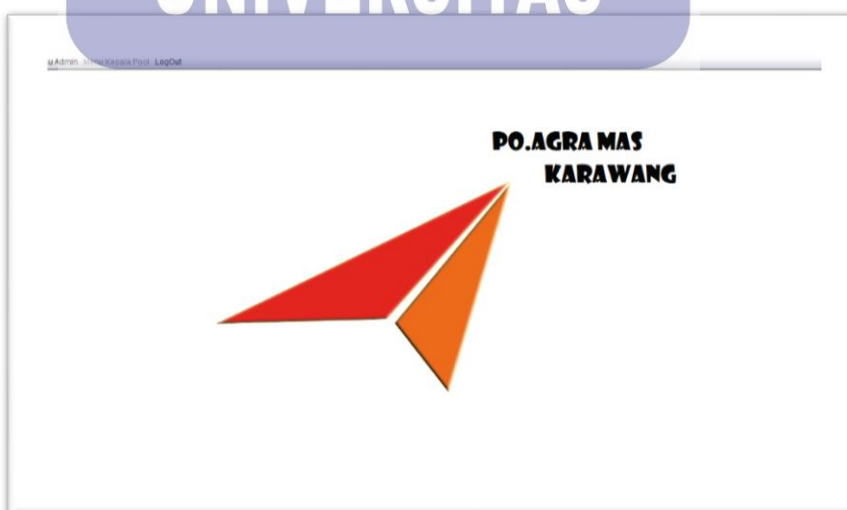


The screenshot shows a web browser window with the title "Form Login". The page contains a login form with the following elements:

- A title "Form Login:" at the top.
- A label "Username :" followed by a text input field.
- A label "Password :" followed by a password input field.
- Two buttons: "Login" and "Keluar" (Logout).
- A large, semi-transparent watermark in the background featuring a circular logo with the text "BINA SARANA INFORMATIKA" and "UNIVERSITAS" below it. The logo also contains the letters "BSI" in a stylized red and white design.

Gambar III.19.
Form login

b. Menu Utama Bagian Admin



Gambar III.20.
Menu utama bagian admin

c. Form Data Penyewa Bagian Admin

Form Data Penyewa

Id Penyewa : Alamat :

Nama Penyewa : No Tlp :

Masukan Id / Nama :

id_sewa	nm_sewa	alamat	no_tlp
30001	budiman	ckampek barat	081546279846
30002	pryono	karawang barat	081354739283
30003	memel	ckampek	0921745439

Gambar III.21.
Form data penyewa bagian admin

d. Form Data Bus Bagian Admin

Form Data Bus

Kode Bus : Plat Nomor :

Tipe Bus : Jumlah Seat :

Status Bus : Merk Bus :

Masukan Kode / Plat No :

id_bus	nopol	tipe	seat	status	merk
50001	B 5478 CG	Big Bus	48	tersedia	SCANIA
50002	B 5389 DGF	Eksekutif	19	tersedia	mercedes-benz
50003	B 8765 GTY	Medium Bus	29	tersedia	mercedes-benz

Gambar III.22.
Form data bus bagian admin

e. Form Data Harga Harga Bagian Admin

kd_harga	tujuan	seat	harga
70001	cikarang	19	1350000
70002	cikarang	29	1700000
70003	cikarang	48	2200000
70004	dalam kota jakarta/jatuhur/purw...	19	1350000
70005	dalam kota jakarta/jatuhur/purw...	29	1700000
70006	dalam kota jakarta/jatuhur/purw...	48	2400000

Gambar III.23.
Form Data harga bagian admin

f. Form Penyewaan Bus Bagian Admin

kd_sewa	tanggal	nm_penyewa	no_tlp	kd_bus	plat	jml_seat	tujuan	lama_sewa	total

Gambar III.24.
Form penyewaan bus bagian admin

g. Menu Utama Bagian Kepala Pool



Gambar III.25.
Menu utama bagian kepala Pool

h. Form Data Admin Bagian Kepala Pool

id_admin	username	hak_akses
20001	ica	admin
20002	m.sodik	pemilik

Gambar III.26.
Form Data admin bagian kepala Pool

i. Form Laporan Penyewaan Bagian Kepala Pool

3.5. Implementasi

Implementasi menjelaskan penerapan yang dilakukan di dalam sistem informasi yang telah dirancang juga pengujian terhadap kegunaan sistem apakah berjalan dengan baik atau tidak.

3.5.1. Code Generation

Code Generation berisi instruksi atau kode program yang telah dibuat didalam suatu sistem yang berguna untuk menerjemahkan kegunaan kode program agar dapat beroperasi sesuai kebutuhan dan fungsinya.

```
package program;
import Database.KoneksiDatabase;
import Database.ResultSetTable;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public class form_bus extends javax.swing.JFrame {

    ResultSet rs;
    KoneksiDatabase con;

    /**
     * Creates new form form_admin
     */
    public form_bus() {
        con = new KoneksiDatabase(new Database.Parameter().HOST_DB, new
        Database.Parameter().USERNAME_DB, new Database.Parameter().PASSWORD_DB);
        initComponents();
        setTitle("Form Data Bus");
        loadtabel();
        nonaktif();
        tambah.setEnabled(true);
        batal.setEnabled(false);
        simpan_E.setEnabled(false);
        simpan_B.setEnabled(false);
        edit.setEnabled(false);
        hapus.setEnabled(false);
    }

    private String id() {
        String no = null;
        try {
            con.koneksiDatabase();
            String sql = "Select right(kd_bus,4)+1 from bus ";
            ResultSet rs = con.eksekusiQuery(sql);
            if (rs.next()) {
                rs.last();
            }
        } catch (SQLException ex) {
            Logger.getLogger(form_bus.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

```

        no = rs.getString(1);
        while (no.length() < 4) {
            no = "000" + no;
            no = "5" + no;
            kd_bus.setText(no);
        }
    } else {
        no = "50001";
        kd_bus.setText(no);
    }
} catch (Exception e) {
}
return no;
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jekel = new javax.swing.ButtonGroup();
    jPanel2 = new javax.swing.JPanel();
    simpan_E = new javax.swing.JButton();
    simpan_B = new javax.swing.JButton();
    jLabel32 = new javax.swing.JLabel();
    jLabel33 = new javax.swing.JLabel();
    tipe = new javax.swing.JComboBox<>();
    batal = new javax.swing.JButton();
    tambah = new javax.swing.JButton();
    jLabel30 = new javax.swing.JLabel();
    jLabel31 = new javax.swing.JLabel();
    tcari = new javax.swing.JTextField();
    cari = new javax.swing.JButton();
    edit = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    tbl_bus = new javax.swing.JTable();
    hapus = new javax.swing.JButton();
    jLabel34 = new javax.swing.JLabel();
    jLabel35 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    jLabel20 = new javax.swing.JLabel();
    jLabel21 = new javax.swing.JLabel();
    jLabel26 = new javax.swing.JLabel();
    jLabel27 = new javax.swing.JLabel();
    kd_bus = new javax.swing.JTextField();
    plat = new javax.swing.JTextField();
    jLabel36 = new javax.swing.JLabel();
    jLabel37 = new javax.swing.JLabel();
    status = new javax.swing.JComboBox<>();
    jLabel38 = new javax.swing.JLabel();
    jLabel39 = new javax.swing.JLabel();
    merk = new javax.swing.JTextField();
    jumlah = new javax.swing.JTextField();
    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

```

```

jPanel2.setBackground(new java.awt.Color(255, 255, 255));
jPanel2.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jPanel2.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

simpan_E.setText("Simpan");
simpan_E.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        simpan_EActionPerformed(evt);
    }
});
jPanel2.add(simpan_E, new org.netbeans.lib.awtextra.AbsoluteConstraints(490, 250, 190, 50));

simpan_B.setText("Simpan");
simpan_B.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        simpan_BActionPerformed(evt);
    }
});
jPanel2.add(simpan_B, new org.netbeans.lib.awtextra.AbsoluteConstraints(490, 250, 190, 50));

jLabel32.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel32.setForeground(new java.awt.Color(51, 51, 51));
jLabel32.setText("Tipe Bus");
jPanel2.add(jLabel32, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 140, -1, -1));

jLabel33.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel33.setForeground(new java.awt.Color(51, 51, 51));
jLabel33.setText(":");
jPanel2.add(jLabel33, new org.netbeans.lib.awtextra.AbsoluteConstraints(400, 140, 10, -1));

tipe.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "PILIH=",
"Eksekutif", "Medium Bus", "Big Bus", "HDC" }));
tipe.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tipeActionPerformed(evt);
    }
});
jPanel2.add(tipe, new org.netbeans.lib.awtextra.AbsoluteConstraints(420, 140, 120, -1));

batal.setText("Batal");
batal.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        batalActionPerformed(evt);
    }
});
jPanel2.add(batal, new org.netbeans.lib.awtextra.AbsoluteConstraints(710, 250, 190, 50));

tambah.setText("Tambah");
tambah.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tambahActionPerformed(evt);
    }
});
jPanel2.add(tambah, new org.netbeans.lib.awtextra.AbsoluteConstraints(270, 250, 190, 50));

jLabel30.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel30.setForeground(new java.awt.Color(51, 51, 51));
jLabel30.setText("Masukan Kode / Plat No");

```

```

jPanel2.add(jLabel30, new org.netbeans.lib.awtextra.AbsoluteConstraints(250, 360, -1, -1));

jLabel31.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel31.setForeground(new java.awt.Color(51, 51, 51));
jLabel31.setText(":");
jPanel2.add(jLabel31, new org.netbeans.lib.awtextra.AbsoluteConstraints(400, 360, 10, -1));
jPanel2.add(tcari, new org.netbeans.lib.awtextra.AbsoluteConstraints(410, 360, 200, -1));

cari.setText("Cari");
cari.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cariActionPerformed(evt);
    }
});
jPanel2.add(cari, new org.netbeans.lib.awtextra.AbsoluteConstraints(620, 360, 90, -1));

edit.setText("Edit");
edit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        editActionPerformed(evt);
    }
});
jPanel2.add(edit, new org.netbeans.lib.awtextra.AbsoluteConstraints(730, 360, 90, -1));

jScrollPane1.setForeground(new java.awt.Color(255, 255, 255));

tbl_bus.setBackground(new java.awt.Color(51, 51, 51));
tbl_bus.setForeground(new java.awt.Color(255, 255, 255));
tbl_bus.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null, null},
        {null, null, null, null, null, null},
        {null, null, null, null, null, null},
        {null, null, null, null, null, null},
        {null, null, null, null, null, null},
        {null, null, null, null, null, null},
        {null, null, null, null, null, null},
        {null, null, null, null, null, null},
        {null, null, null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4", "Title 5", "Title 6"
    }
));
jScrollPane1.setViewportView(tbl_bus);

jPanel2.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(100, 400, 1040,
170));

hapus.setText("Hapus");
hapus.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        hapusActionPerformed(evt);
    }
});
jPanel2.add(hapus, new org.netbeans.lib.awtextra.AbsoluteConstraints(840, 360, 90, -1));

jLabel34.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel34.setForeground(new java.awt.Color(51, 51, 51));
jLabel34.setText("Jumlah Seat");

```

```

jPanel2.add(jLabel34, new org.netbeans.lib.awtextra.AbsoluteConstraints(620, 140, -1, -1));

jLabel35.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel35.setForeground(new java.awt.Color(51, 51, 51));
jLabel35.setText(":");
jPanel2.add(jLabel35, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 140, 10, -1));

jLabel1.setFont(new java.awt.Font("Tahoma", 0, 28)); // NOI18N
jLabel1.setForeground(new java.awt.Color(51, 51, 51));
jLabel1.setText("Form Data Bus");
jPanel2.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(530, 10, -1, -1));

jLabel20.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel20.setForeground(new java.awt.Color(51, 51, 51));
jLabel20.setText("Kode Bus");
jPanel2.add(jLabel20, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 90, -1, -1));

jLabel21.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel21.setForeground(new java.awt.Color(51, 51, 51));
jLabel21.setText(":");
jPanel2.add(jLabel21, new org.netbeans.lib.awtextra.AbsoluteConstraints(400, 90, 10, -1));

jLabel26.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel26.setForeground(new java.awt.Color(51, 51, 51));
jLabel26.setText("Plat Nomor");
jPanel2.add(jLabel26, new org.netbeans.lib.awtextra.AbsoluteConstraints(620, 90, -1, -1));

jLabel27.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel27.setForeground(new java.awt.Color(51, 51, 51));
jLabel27.setText(":");
jPanel2.add(jLabel27, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 90, 10, -1));

kd_bus.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        kd_busActionPerformed(evt);
    }
});
jPanel2.add(kd_bus, new org.netbeans.lib.awtextra.AbsoluteConstraints(420, 90, 80, -1));
jPanel2.add(plat, new org.netbeans.lib.awtextra.AbsoluteConstraints(710, 90, 170, -1));

jLabel36.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel36.setForeground(new java.awt.Color(51, 51, 51));
jLabel36.setText("Status Bus");
jPanel2.add(jLabel36, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 190, -1, -1));

jLabel37.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel37.setForeground(new java.awt.Color(51, 51, 51));
jLabel37.setText(":");
jPanel2.add(jLabel37, new org.netbeans.lib.awtextra.AbsoluteConstraints(400, 190, 10, -1));

status.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "pilih=",
"tersedia", "disewa" }));
status.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        statusActionPerformed(evt);
    }
});
jPanel2.add(status, new org.netbeans.lib.awtextra.AbsoluteConstraints(420, 190, 120, -1));

```

```

jLabel38.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel38.setForeground(new java.awt.Color(51, 51, 51));
jLabel38.setText("Merk Bus");
jPanel2.add(jLabel38, new org.netbeans.lib.awtextra.AbsoluteConstraints(620, 190, -1, -1));

jLabel39.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel39.setForeground(new java.awt.Color(51, 51, 51));
jLabel39.setText(":");
jPanel2.add(jLabel39, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 190, 10, -1));
jPanel2.add(merk, new org.netbeans.lib.awtextra.AbsoluteConstraints(710, 190, 170, -1));
jPanel2.add(jumlah, new org.netbeans.lib.awtextra.AbsoluteConstraints(710, 140, 170, -1));

getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1250,
660));

pack();
} // </editor-fold>

private void tambahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    tambah.setEnabled(false);
    batal.setEnabled(true);
    simpan_E.setVisible(false);
    simpan_B.setEnabled(true);
    simpan_B.setVisible(true);
    edit.setEnabled(false);
    aktif();
    id();
}

private void simpan_BActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    simpanB();
}

private void simpan_EActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        if (!kd_bus.getText().isEmpty() && !tipe.getSelectedItem().toString().isEmpty() &&
!plat.getText().isEmpty() && !jumlah.getText().isEmpty()) {
            String kolom[] = {"kd_bus", "nopol", "tipe", "seat", "status", "merk"};
            String isi[] = {kd_bus.getText(), plat.getText(), tipe.getSelectedItem().toString(),
jumlah.getText(), status.getSelectedItem().toString(), merk.getText()};
            con.queryUpdate("bus", kolom, isi, "kd_bus=" + String.valueOf(kd_bus.getText()) + "");
            JOptionPane.showMessageDialog(this, "Data Berhasil Diedit");
            loadtabel();
            clear();
            nonaktif();
            tambah.setEnabled(true);
            batal.setEnabled(false);
            simpan_E.setVisible(false);
            simpan_B.setEnabled(false);
            simpan_B.setVisible(true);
            edit.setEnabled(false);
            hapus.setEnabled(false);
            tcari.setText("");
        } else {
            JOptionPane.showMessageDialog(this, "Data isian ada yang kosong");
        }
    }
}

```

```

        plat.requestFocus();
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(this, "Error edit data");
}
loadtabel();
clear();
nonaktif();
tambah.setEnabled(true);
batal.setEnabled(false);
simpan_E.setVisible(false);
simpan_B.setEnabled(false);
simpan_B.setVisible(true);
edit.setEnabled(false);
}

private void batalActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    tambah.setEnabled(true);
    batal.setEnabled(false);
    simpan_E.setVisible(false);
    simpan_B.setEnabled(false);
    simpan_B.setVisible(true);
    edit.setEnabled(false);
    hapus.setEnabled(false);
    nonaktif();
    clear();
    loadtabel();
    tcari.setText("");
}

private void cariActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        if (!tcari.getText().isEmpty()) {
            rs = con.querySelectAll("bus", "kd_bus LIKE '%" + tcari.getText() + "%' OR nopol LIKE '%" + tcari.getText() + "%'");
            tbl_bus.setModel(new Database.ResultSetTable(rs));
            edit();
            tambah.setEnabled(false);
            batal.setEnabled(true);
            edit.setEnabled(true);
            hapus.setEnabled(true);
        } else {
            JOptionPane.showMessageDialog(this, "isi data pencarian");
            tcari.requestFocus();
        }
    } catch (Exception e) {
    }
}

private void editActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    aktif();
    simpan_E.setVisible(true);
    simpan_E.setEnabled(true);
    simpan_B.setVisible(false);

```

```

}

private void hapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        String kode=String.valueOf(kd_bus.getText());
        if (JOptionPane.showConfirmDialog(this, "Yakin menghapus data", "Peringatan",
JOptionPane.OK_CANCEL_OPTION) == JOptionPane.OK_OPTION) {
            con.queryDelete("bus","kd_bus="+kode);
            JOptionPane.showMessageDialog(this, "Data Berhasil Dihapus");
            loadtabel();
            clear();
            nonaktif();
            tambah.setEnabled(true);
            batal.setEnabled(false);
            edit.setEnabled(false);
            hapus.setEnabled(false);
            tcari.setText("");
        }else{
            return;
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Pilih data terlebih dahulu");
    }
    loadtabel();
    clear();
    nonaktif();
}

private void tipeActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int seat = 0;
    if (tipe.getSelectedItem() == "Eksekutif"){
        seat = 19;
    }
    else if (tipe.getSelectedItem() == "Medium Bus"){
        seat = 29;
    }
    else if (tipe.getSelectedItem() == "Big Bus"){
        seat = 48;
    }
    else if (tipe.getSelectedItem() == "HDC"){
        seat = 59;
    }
    else {seat = 0;}
    jumlah.setText(String.valueOf(seat));
}

private void statusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void kd_busActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

/**
 * @param args the command line arguments

```

```

*/
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(form_bus.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(form_bus.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(form_bus.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(form_bus.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
    }
    //</editor-fold>
    //</editor-fold>
    //</editor-fold>
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new form_bus().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton batal;
private javax.swing.JButton cari;
private javax.swing.JButton edit;
private javax.swing.JButton hapus;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel27;
private javax.swing.JLabel jLabel30;
private javax.swing.JLabel jLabel31;
private javax.swing.JLabel jLabel32;
private javax.swing.JLabel jLabel33;

```

```

private javax.swing.JLabel jLabel34;
private javax.swing.JLabel jLabel35;
private javax.swing.JLabel jLabel36;
private javax.swing.JLabel jLabel37;
private javax.swing.JLabel jLabel38;
private javax.swing.JLabel jLabel39;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.ButtonGroup jekel;
private javax.swing.JTextField jumlah;
private javax.swing.JTextField kd_bus;
private javax.swing.JTextField merk;
private javax.swing.JTextField plat;
private javax.swing.JButton simpan_B;
private javax.swing.JButton simpan_E;
private javax.swing.JComboBox<String> status;
private javax.swing.JButton tambah;
private javax.swing.JTable tbl_bus;
private javax.swing.JTextField tcari;
private javax.swing.JComboBox<String> tipe;
// End of variables declaration
private void loadtabel() {
    String namaKolom[] = {"kd_bus", "nopol", "tipe", "seat", "status", "merk"};
    rs = con.querySelect(namaKolom, "bus");
    tbl_bus.setModel(new ResultSetTable(rs));
}

private void simpanB() {
    try {
        if (!kd_bus.getText().isEmpty() && !tipe.getSelectedItem().toString().isEmpty() &&
            !plat.getText().isEmpty() && !jumlah.getText().isEmpty()) {
            String kolom[] = {"kd_bus", "nopol", "tipe", "seat", "status", "merk"};
            String isi[] = {kd_bus.getText(), plat.getText(), tipe.getSelectedItem().toString(),
                jumlah.getText(), status.getSelectedItem().toString(), merk.getText()};

            System.out.println(con.queryInsert("bus", kolom, isi));

            JOptionPane.showMessageDialog(this, "Data Berhasil Disimpan");
            loadtabel();
            clear();
            nonaktif();
            tambah.setEnabled(true);
            batal.setEnabled(false);
            simpan_E.setVisible(false);
            simpan_B.setEnabled(false);
            simpan_B.setVisible(true);
            edit.setEnabled(false);
        } else {
            JOptionPane.showMessageDialog(this, "Data isian ada yang kosong");
            plat.requestFocus();
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Error input data");
        System.out.println("salah");
    }
    loadtabel();
    clear();
    nonaktif();
    tambah.setEnabled(true);

```

```

        batal.setEnabled(false);
        simpan_E.setVisible(false);
        simpan_B.setEnabled(false);
        simpan_B.setVisible(true);
        edit.setEnabled(false);
    }

    private void clear() {
        kd_bus.setText("");
        plat.setText("");
        jumlah.setText("");
        tipe.setSelectedItem("=PILIH=");
        status.setSelectedItem("=pilih=");
        merk.setText("");
    }

    private void nonaktif() {
        kd_bus.setEnabled(false);
        plat.setEnabled(false);
        jumlah.setEnabled(false);
        tipe.setEnabled(false);
        merk.setEnabled(false);
        status.setEnabled(false);
    }

    private void aktif() {
        kd_bus.setEnabled(false);
        plat.setEnabled(true);
        jumlah.setEnabled(false);
        tipe.setEnabled(true);
        status.setEnabled(true);
        merk.setEnabled(true);
    }

    private void edit() {
        String st = (String) tcari.getText();
        ResultSet rst = con.querySelectAll("bus", "kd_bus=" + st + "OR nopol=" + st + "");
        try {
            while (rst.next()) {
                this.kd_bus.setText(rst.getString("kd_bus"));
                this.plat.setText(rst.getString("nopol"));
                this.jumlah.setText(rst.getString("seat"));
                this.tipe.setSelectedItem(rst.getString("tipe"));
                this.status.setSelectedItem(rst.getString("status"));
                this.merk.setText(rst.getString("merk"));
            }
        } catch (SQLException ex) {
            Logger.getLogger(form_bus.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

3.5.2. Blackbox Testing

Kotak pengujian yang menguji kelayakan sistem yang dirancang dan dibangun untuk memudahkan pendataan sesuai dengan kebutuhana dan fungsinya.

Berikut beberapa pengujian terhadap form yang ada didalam sistem usulan:

1. Pengujian Terhadap Form *Login*

Tabel III.16.

Hasil Blackbox Testing Form *Login*

N o	Skenario Pengujian	Text case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Masukkan <i>Username</i> , mengosongka n <i>Password</i> . Klik tombol <i>Login</i>	<i>Username</i> , <i>Password</i> [Kosong]	Sistem akan menolak akses dan menampilkan pesan “ <i>Login</i> Gagal”	Sesuai harapan	Valid
2.	Masukkan <i>username</i> kosong, Masukkan <i>password</i> . Klik tombol <i>Login</i>	<i>Username</i> [Kosong], <i>Password</i>	Sistem akan menolak akses dan menampilkan pesan “ <i>Login</i> Gagal”	Sesuai harapan	Valid
3.	Masukkan <i>username</i> kosong, Masukkan <i>password</i> kosong. Klik tombol <i>Login</i>	<i>Username</i> [Kosong], <i>Password</i> [Kosong]	Sistem akan menolak akses dan menampilkan pesan “ <i>Login</i> Gagal”	Sesuai harapan	Valid
4.	Masukkan <i>username</i> , Masukkan <i>Password</i> . Klik tombol	<i>Username</i> , <i>Password</i>	Sistem menerima akses dan menampilkan “Selamat	Sesuai Harapan	Valid

	Login		datang (username) ”		
--	-------	--	------------------------	--	--

2. Pengujian Terhadap Form Penyewaan Bus

Tabel III.17.

Hasil Blackbox Testing Form Penyewaan Bus

N o	Skenario Pengujian	Text case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Masukkan Tujuan wisata, Tujuan wisata, Mengosongkan Harga, Lama sewa. Klik tombol simpan	Tujuan wisata, Harga [kosong], Lama sewa	Sistem akan menolak akses dan menampilkan pesan “Data isian ada yang kosong”	Sesuai harapan	Valid
2.	Masukkan Tujuan wisata, Tujuan wisata, Masukkan Harga, mengosongkan lama sewa. Klik tombol simpan	Tujuan wisata, Harga, Lama sewa [kosong],	Sistem akan menolak akses dan menampilkan pesan “Data isian ada yang kosong”	Sesuai harapan	Valid
3.	Masukkan Tujuan wisata, Tujuan wisata, Mengosongkan harga, mengosongkan lama sewa. Klik tombol simpan	Tujuan wisata, Harga [kosong], Lama sewa [kosong],	Sistem akan menolak akses dan menampilkan pesan “Data isian ada yang kosong”	Sesuai harapan	Valid
4.	Masukkan Tujuan wisata, Tujuan wisata, Masukkan Harga, Lama sewa. Klik	Tujuan wisata, harga, lama sewa	Sistem menerima akses dan menampilkan “Data berhasil	Sesuai Harapan	Valid

	tombol simpan		disimpan, kwitansi sedang dicetak ”		
--	------------------	--	--	--	--

3.5.3. Spesifikasi *Hardware* dan *Software*

Spesifikasi *hardware* dan *software* merupakan penjelasan mengenai kelebihan maupun kekurangan perangkat keras dan perangkat lunak yang digunakan dalam pembuatan sistem informasi akuntansi. Berikut merupakan spesifikasi *hardware* dan *software* yang digunakan:

Kebutuhan	Keterangan
Sistem Operasi	Microsoft Windows 10
Processor	Intel
RAM	2 GB
Keyboard	108 ke
Printer	Canon
Mouse	Standard
Software	Bahasa Pemrograman : Java Neatbeans 8.1 Aplikasi Pendukung : Jasper Report DBMS : MY SQL Anti Virus : Smadav Xampp : Xampp Control Panel Browser : Mozilla Firefox

