

BAB III

PEMBAHASAN

3.1. Tinjauan Perusahaan

Tinjauan perusahaan yang akan dijelaskan adalah mengenai sejarah dan perkembangan, struktur organisasi dan hasil penelitian pada CV Mora Mandiri Cikampek dengan harapan agar perusahaan yang efektif dan efisien.

3.1.1. Sejarah Perusahaan

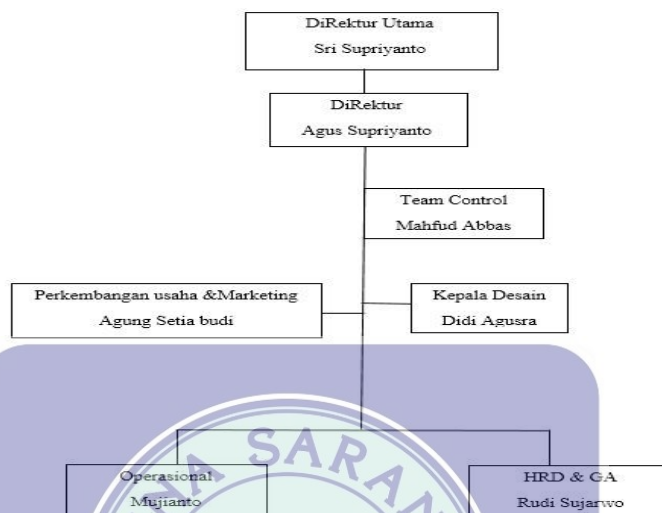
CV Mora Mandiri Cikampek merupakan perusahaan swasta yang bergerak dibidang jasa percetakan. Dalam kegiatan operasionalnya CV Mora Mandiri Cikampek ini memproduksi cetak undangan, buku yasin, pembuatan map, sablon, banner, x banner, dan pembuatan stiker.

Pada awal berdirinya CV Mora Mandiri yaitu tanggal 03 Februari 2005 berlokasi di Jl. Bima Indah Bayur RT. 01/06 Desa Jomin Barat Kecamatan Kota Baru Kabupaten Karawang. Pada saat itu CV Mora Mandiri masih menggunakan nama Bila Mandiri yang merupakan perusahaan pribadi, dengan modal awal sebesar Rp.150.000.000. Kemudian pada tanggal 07 Oktober 2015 CV Bila Mandiri mengubah nama menjadi CV Mora Mandiri Cikampek.

3.1.2. Struktur Organisasi Perusahaan

CV Mora dalam merealisasikan tujuan mempunyai struktur organisasi yang di dalamnya terdapat kedudukan, wewenang, tugas dan tanggung jawab masing-masing anggota sehingga mereka bertanggung jawab pada tugas yang dilaksanakan. Dengan adanya struktur organisasi maka setiap pribadi mendapatkan ruang lingkup pekerjaan yang jelas, sehingga dengan demikian terjalinlah suatu kerja sama yang

baik dalam pencapaian tujuan perusahaan. Berikut ini adalah bagian-bagian yang terdapat dalam struktur organisasi CV Mora Mandiri Cikampek dengan menggunakan masing-masing fungsi.



Sumber: CV Mora Mandiri

Gambar III.1.

Struktur Organisasi UNIVERSITAS

Berikut fungsi dari masing-masing bagian struktur organisasi sesuai uraian tugasnya pada CV Mora Mandiri Cikampek antara sebagai berikut:

1. Direktur Utama sebagai koordinator, komunikator, pengambil keputusan, pemimpin pengelola dan eksekutor dalam menjalankan dan memimpin perusahaan.
2. Direktur, merencanakan dan memutuskan kebijakan strategis yang menyangkut produksi serta memonitoring dan mengarahkan proses-proses diseluruh divisi produksi.

3. Tech Control, untuk pengukuran dan koreksi kinerja bawahan, agar rencana-rencana yang telah dibuat mencapai tujuan-tujuan perusahaan yang telah direncanakan.
4. Kepala desain, membuat dan mengerjakan segala desain yang telah diberikan dari atasan
5. Marketing, memperkenalkan perusahaan dan produk yang dihasilkannya kepada konsumen sehingga perusahaan dan produknya dapat dikenal dan diterima di konsumen.
6. Operasional, bertanggung jawab dalam merekrut dan menempatkan karyawan dalam penyeleksian tenaga kerja.
7. HRD sebagai trainer terhadap pegawai dalam perusahaan

3.2. Tinjauan Kasus

Tinjauan kasus menguraikan tentang gambaran proses dari sistem pembelian peralatan mulai dari sistem pembelian peralatan mulai dari mengecek stok peralatan, mencatat stok peralatan, memesan peralatan, melakukan pembayaran, serta hingga proses akhir membuat laporan keuangan untuk arsip CV. Mora Mandiri Cikampek.

3.2.1. Prosedur Bisnis Sistem Berjalan

Prosedur sistem berjalan merupakan prosedur yang menjelaskan tentang bagaimana sistem pembelian peralatan yang terjadi di CV. Mora Mandiri Cikampek. Berikut adalah prosedur sistem berjalan yang terdapat pada CV. Mora Mandiri Cikampek:

1. Prosedur Permintaan dan pemesanan peralatan

Pada saat persediaan peralatan menunjukkan batas minimal, bagian admin mengecek stok peralatan digudang dan mencatat stok peralatan pada kertas memo yang biasa digunakan admin untuk mencatatnya. Kemudian admin membuat

Laporan Pembelian Peralatan (LPP) dan diajukan kepada direktur, jika telah disepakati/ACC oleh direktur maka admin melakukan pemesanan peralatan kepada *supplier*.

2. Prosedur pembayaran

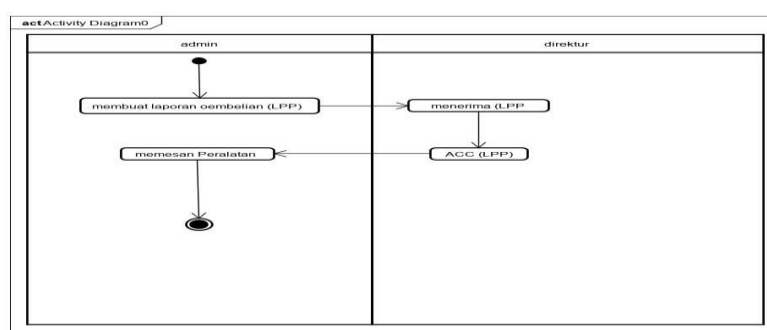
Setelah pembelian dilakukan lalu admin membayar semua total pembelian peralatan kepada *supplier*, kemudian *supplier* menerima pembayaran dan membuatkan nota rangkap 2, nota rangkap 1 berwarna putih disimpan sebagai arsip pada *supplier*. Sedangkan nota rangkap 2 berwarna merah diberikan kepada admin sebagai tanda bukti telah melakukan transaksi pembelian. Agar admin dapat melaporkan pada direktur CV. Mora Mandiri Cikampek.

3. Prosedur Pembuatan Laporan

Setiap akhir periode admin mencatat laporan pembelian peralatan bulanan berdasarkan arsip nota yang ada, selanjutnya laporan akan diberikan kepada direktur untuk proses evaluasi selama satu bulan. Jika telah diperiksa oleh direktur maka laporan disimpan oleh admin sebagai arsip.

3.2.2. Activity Diagram Proses Bisnis Berjalan

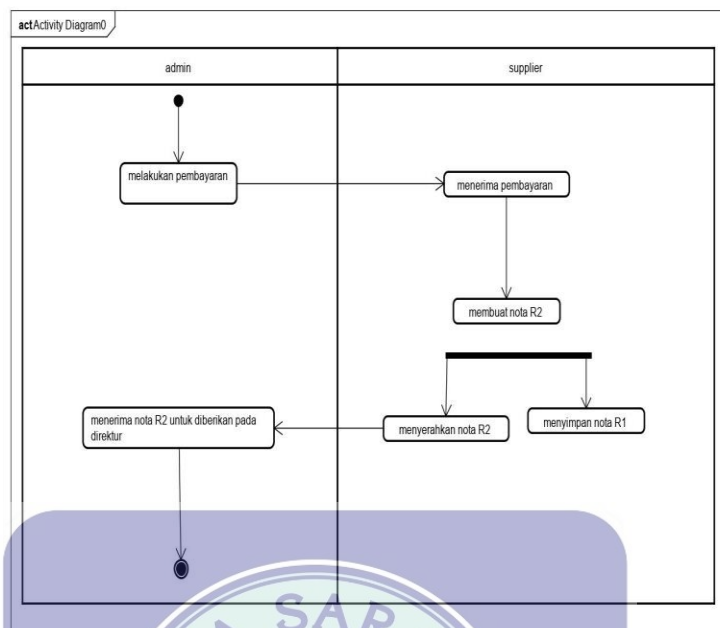
1. Activity Diagram Prosedur Permintaan dan pemesanan peralatan



Sumber: Hasil Penelitian

Gambar III.2.
Activity diagram permintaan dan pemesanan peralatan

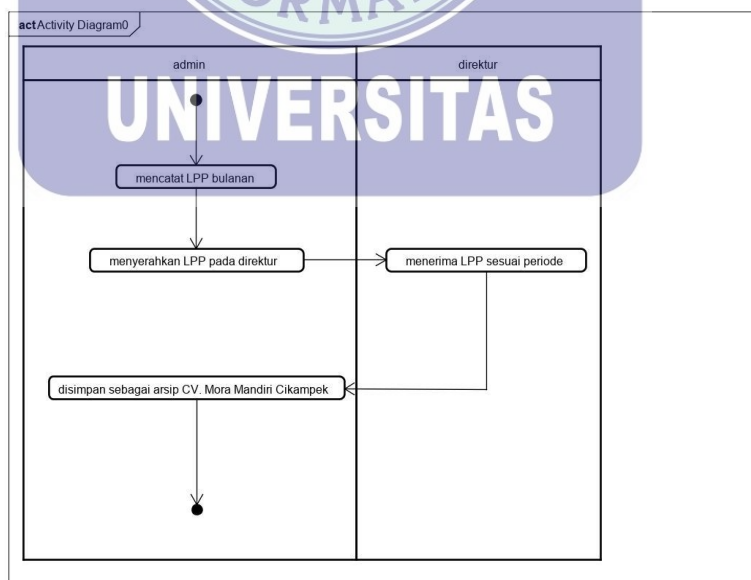
2. Activity Diagram prosedur pembayaran peralatan



Sumber: Hasil Penelitian

Gambar III.3.
Activity Diagram pembayaran peralatan

3. Activity Diagram Prosedur Pembuatan Laporan



Sumber: Hasil Penelitian

Gambar III.4.
Activity Diagram Pembuatan Laporan

3.2.3. Dokumen Masukan

Dokumen masukan adalah segala bentuk dokumen yang berasal dari lingkungan luar sistem yang berupa dokumen-dokumen yang akan diolah dalam suatu proses agar dapat menghasilkan keluaran yang diinginkan. Berikut dokumen masukan tersebut adalah sebagai berikut:

1. Nota Pembelian Peralatan

Nama dokumen	: Nota Pembelian
Fungsi	: Sebagai tanda bukti transaksi pembelian peralatan
Sumber	: Supplier
Tujuan	: Bagian Administrasi
Media	: Kertas
Jumlah	: 1 Rangkap
Frekuensi	: Setiap terjadi pembelian peralatan
Bentuk	: Lihat Lampiran A.1

3.2.4. Dokumen Keluaran

Dokumen keluaran adalah segala bentuk dokumen berupa dokumen-dokumen yang akan mendukung kegiatan serta merupakan hasil pencatatan atau pelaporan. Berikut dokumen keluaran sebagai berikut:

1. Kertas Memo

Nama dokumen	: Laporan Pembelian Peralatan (LPP)
Fungsi	: Sebagai alat mencatat stok peralatan yang telah habis
Sumber	: Bagian administrasi
Tujuan	: Direktur

Media : Kertas
 Jumlah : 1 Lembar
 Frekuensi : Setiap stok peralatan akan habis
 Bentuk : Lihat Lampiran B.1

2. Laporan Pembelian

Nama dokumen : Laporan pembelian
 Fungsi : Untuk mencatat pengeluaran atas pembelian peralatan
 Sumber : Bagian administrasi

Tujuan : Direktur
 Media : Kertas
 Jumlah : 1 Lembar
 Frekuensi : Setiap akhir periode
 Bentuk : Lihat Lampiran B.2

3.2.5. Permasalahan Pokok

Sistem yang berjalan pada CV. Mora Mandiri Cikampek ini pada pembelian peralatan masih menggunakan media sederhana yaitu menggunakan kertas memo sebagai alat mencatat stok pembelian peralatan dan pencatatan laporan. Apabila proses pembelian peralatan dilakukan secara sederhana maka akan mengakibatkan permasalahan seperti berikut:

1. Membutuhkan waktu yang lama dalam pengecekan stok peralatan yang mulai habis
2. Belum memiliki catatan pembelian yang baik yaitu hanya mengumpulkan nota, sehingga terjadi kesalahan dalam perhitungan.
3. Dengan data dan arsip yang masih manual memperlambat proses transaksi.

3.2.6. Pemecahan Masalah

Banyak masalah yang timbul akibat sistem pembelian yang masih dikerjakan secara manual. Hal tersebut tentu tidak lagi sesuai dengan kemajuan teknologi informasi sekarang ini, dimana waktu adalah sesuatu yang berharga. Dengan adanya sistem yang terkomputerisasi tentu akan dapat lebih memudahkan pelaksanaan kerja bagian pembelian dan keuangan, serta dapat menghilangkan pekerjaan yang tidak efektif dan efisien. Berikut adalah pemecahan masalah yaitu:

1. Belum memiliki data stok barang
2. Dalam memproses pembelian lebih cepat dan akurat.
3. Data dapat diolah lebih cepat dan akurat tidak membutuhkan waktu yang lama untuk memperbarui data pada komputer.
4. Membantu kemajuan perusahaan dalam mencapai tujuan dan membantu menganbil keputusan untuk masa yang akan datang.

3.3. Analisa Kebutuhan Software

3.3.1. Analisa Kebutuhan

Analisa kebutuhan software untuk program pembelian peralatan CV. Mora Mandiri Cikampek mengusulkan beberapa prosedur yaitu:

A. Bagian admin

- A.1. admin berhak melakukan login
- A.2. Admin berhak mengelola data peralatan
- A.3. Admin berhak mengelola transaksi
- A.4. Admin berhak mengelola jurnal
- A.5. Admin berhak mengelola laporan

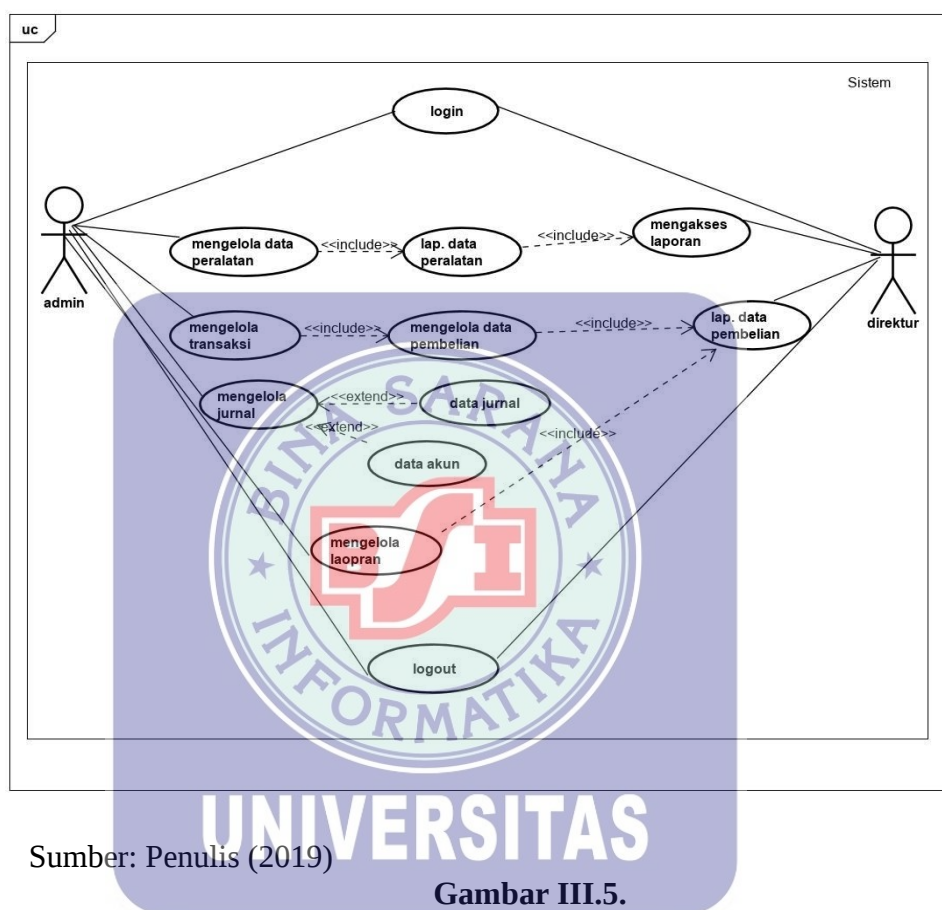
B. Bagian Direktur

B.1. Direktur berhak melakukan login

B.2. Direktur berhak mengakses laporan

3.3.2 Use Case Diagram

Berikut adalah *use case diagram* yang diusulkan penulis:



Sumber: Penulis (2019)

Gambar III.5.
Use Case Diagram Usulan

1. Use Case Diagram Login

Tabel III.1

Deskripsi Use Case Diagram

<i>Use Case Name</i>	Login
<i>Requirment</i>	Admin dapat melakukan login
<i>Goal</i>	Admin dapat login dan masuk ke halaman menu utama
<i>Pre-Conditions</i>	Admin dapat melakukan login
<i>Post-Conditions</i>	Admin berhasil login
<i>Failed end Conditions</i>	Admin Gagal login

<i>Actors</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin memasukkan username dan password 2. Admin dapat masuk ke halaman menu utama
<i>Alternate Flow/Invariant 1</i>	1. Admin memasukkan username dan password 2. Admin dapat melakukan login 3. Admin dapat melakukan logout
<i>Invariant 2</i>	1. Admin memasukkan username dan password yang salah 2. Admin tidak dapat login

2. Use Case Diagram Admin Mengelola Data peralatan

Tabel III.2.
Deskripsi Use Case Diagram Admin Mengelola Data Konsumen

<i>UseCase Name</i>	Data peralatan
<i>Requirment</i>	Admin dapat mengelola data peralatan
<i>Goal</i>	Untuk mengelola data peralatan
<i>Pre-Conditions</i>	Admin dapat mengelola data peralatan
<i>Post-Conditions</i>	Data peralatan dapat dikelola
<i>Failed end Conditions</i>	Gagal mengelola data peralatan
<i>Actors</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin menyimpan data peralatan 2. Admin memproses peralatan 3. Admin menghapus data peralatan 4. Admin mengedit data peralatan 5. Admin memproses data peralatan 6. Admin mencari data peralatan
<i>Alternate Flow/Invariant 1</i>	A.1. Admin dapat mengelola data peralatan
<i>Invariant 2</i>	B.1. Admin dapat membuka menu utama kemudian mengelola data peralatan

3. Use Case Diagram Admin Mengelola Transaksi

Tabel III.3.
Deskripsi Use Case Diagram Admin Mengelola Transaksi

<i>UseCase Name</i>	Mengelola Transaksi
<i>Requirment</i>	Admin dapat mengelola transaksi
<i>Goal</i>	Untuk mengelola transaksi
<i>Pre-Conditions</i>	Admin dapat mengelola transaksi
<i>Post-Conditions</i>	Transaksi dapat dikelola
<i>Failed end Conditions</i>	Gagal mengelola transaksi
<i>Actors</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin menambah data transaksi 2. Admin menyimpan data transaksi 3. Admin menghapus data transaksi 4. Admin memproses data transaksi 5. Admin mencari data peralatan
<i>Alternate Flow/Invariant 1</i>	A.1. Admin dapat mengelola data transaksi
<i>Invariant 2</i>	B.1. Admin dapat membuka menu utama kemudian mengelola data barang pada menu data master

4. Use Case Diagram Admin mengelola data karyawan

Tabel III.4.
Deskripsi Use Case Diagram Admin Mengelola Data jurnal

<i>UseCase Name</i>	Data Barang
<i>Requirment</i>	Admin dapat mengelola data jurnal
<i>Goal</i>	Untuk mengelola data jurnal
<i>Pre-Conditions</i>	Admin dapat mengelola data jurnal
<i>Post-Conditions</i>	Data jurnal dapat dikelola
<i>Failed end Conditions</i>	Gagal mengelola data jurnal
<i>Actors</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin menambah data jurnal 2. Admin menyimpan data jurnal 3. Admin menghapus data jurnal 4. Admin mengedit data jurnal 5. Admin mencari data jurnal 6. Admin mengelola data jurnal
<i>Alternate Flow/Invariant 1</i>	A.1. Admin dapat mengelola jurnal
<i>Invariant 2</i>	B.1. Admin dapat membuka menu utama kemudian

	mengelola jurnal
--	------------------

5. Use Case Diagram Admin mengelola laporan

Tabel III.5.
Deskripsi Use Case Diagram Admin Mengelola Laporan

<i>UseCase Name</i>	Laporan
<i>Requirment</i>	Admin dapat mengelola laporan
<i>Goal</i>	Admin dapat mengelola laporan pembelian
<i>Pre-Conditions</i>	Admin dapat mengelola laporan
<i>Post-Conditions</i>	Admin dapat mengelola laporan
<i>Failed end Conditions</i>	Admin gagal mengelola laporan
<i>Actors</i>	Admin
<i>Main Flow/Basic Path</i>	1. Admin melakukan login terlebih dahulu 2. Admin mengelola data peralatan 3. Admin mengelola data jurnal 4. Admin mengelola menu transaksi 5. Admin mengelola laporan pembelian
<i>Alternate Flow/Invariant 1</i>	A.1. Admin dapat mengelola transaksi pembelian peralatan
<i>Invariant 2</i>	B.1. Admin dapat membuka menu utama kemudian mengelola laporan

6. Use Case Diagram Direktur Login

Tabel III.6.
Deskripsi Use Case Diagram Pemilik Login

<i>Use Case Name</i>	Login
<i>Requirment</i>	Pemilik dapat melakukan login
<i>Goal</i>	Pemilik dapat login dan masuk ke halaman menu utama
<i>Pre-Conditions</i>	Pemilik dapat melakukan login
<i>Post-Conditions</i>	Pemilik berhasil login

<i>Failed end Conditions</i>	Pemilik Gagal login
<i>Actors</i>	Pemilik
<i>Main Flow/Basic Path</i>	1. Pemilik memasukkan username dan password 2. Pemilik dapat masuk ke halaman menu utama
<i>Alternate Flow/Invariant 1</i>	1. Pemilik memasukkan username dan password 2. Pemilik dapat melakukan login 3. Pemilik dapat melakukan logout
<i>Invariant 2</i>	1. Pemilik memasukkan username dan password yang salah 2. Pemilik tidak dapat login

7. Use Case Diagram Admin mengelola laporan

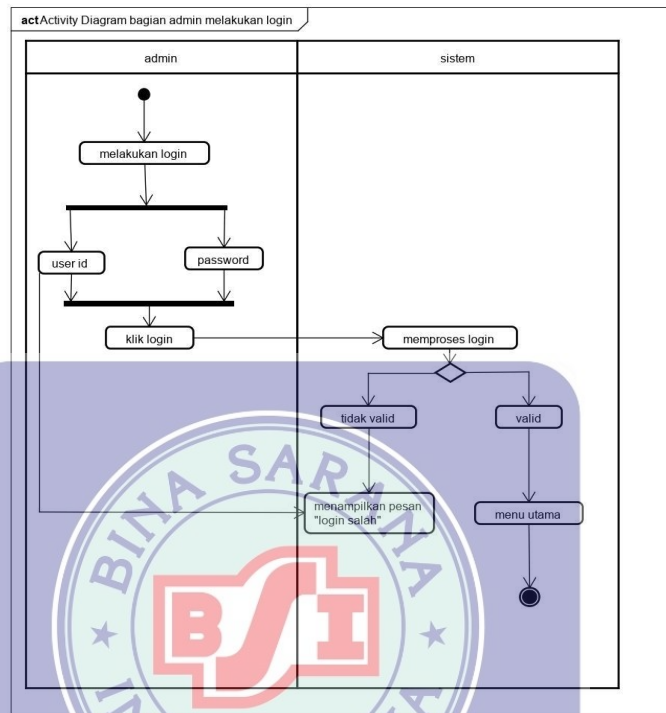
Tabel III.7.
Deskripsi Use Case Diagram Admin Mengelola Laporan

UseCase Name	Laporan
Requirment	Direktur dapat mengelola laporan
Goal	Direktur dapat mengelola laporan pembelian
Pre-Conditions	Direktur dapat mengelola laporan
Post-Conditions	Direktur dapat mengelola laporan
Failed end Conditions	Direktur gagal mengelola laporan
Actors	Direktur
<i>Main Flow/Basic Path</i>	1. Direktur melakukan login terlebih dahulu 2. Direktur mengelola data peralatan 3. Direktur mengelola data jurnal 4. Direktur mengelola menu transaksi 5. Direktur mengelola laporan pembelian
<i>Alternate Flow/Invariant 1</i>	A.1. Direktur dapat mengelola transaksi pembelian peralatan
<i>Invariant 2</i>	B.1. Direktur dapat membuka menu utama kemudian mengelola laporan

3.3.3. Activity Diagram

1. Kebutuhan Admin

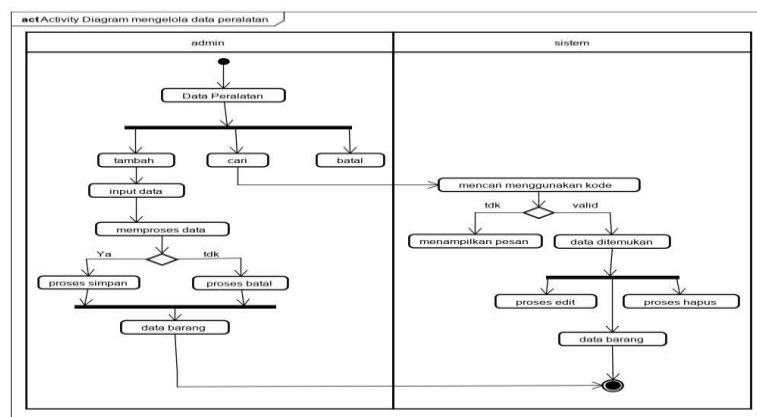
A.1. Activity Diagram Admin Login



Sumber:Hasil Penelitian

Gambar.III.6.
Activity Diagram Admin Login

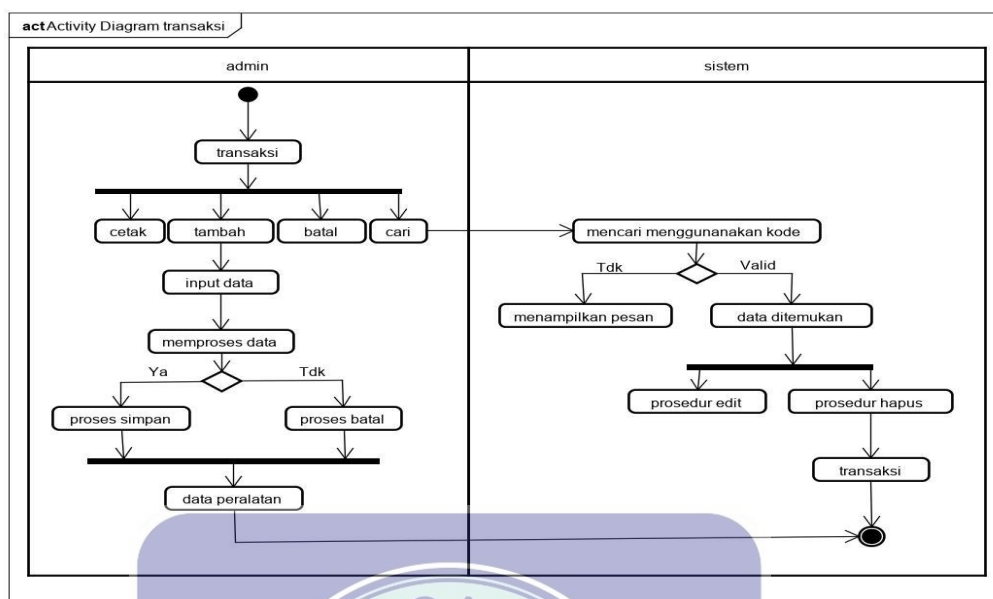
A.2. Activity Diagram Admin Mengelola Data peralatan



Sumber:Hasil Penelitian

Gambar.III.7.
Activity Diagram Admin Mengelola Data peralatan

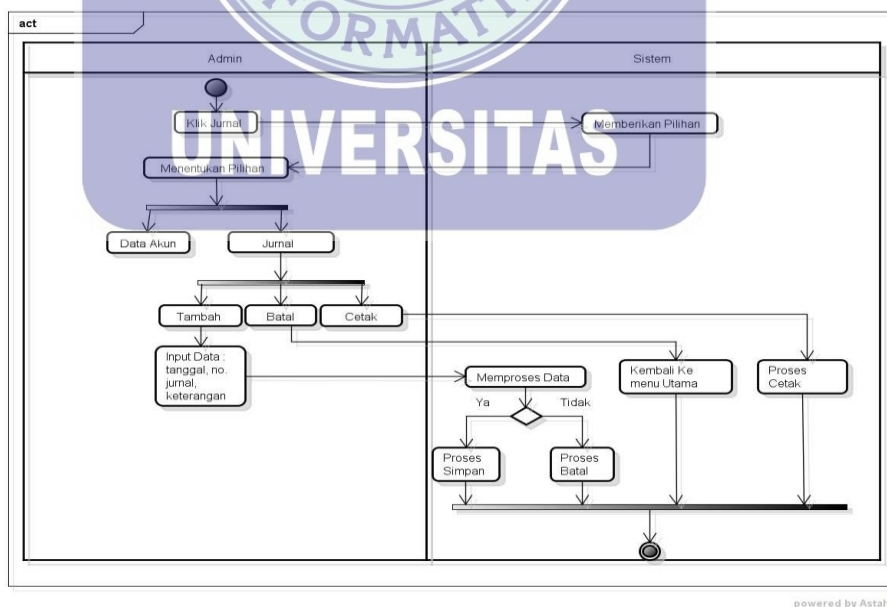
A.3. Activity Diagram Admin mengelola Transaksi



Sumber: Hasil Penelitian

Gambar.III.8.
Activity Diagram Admin mengelola Transaksi

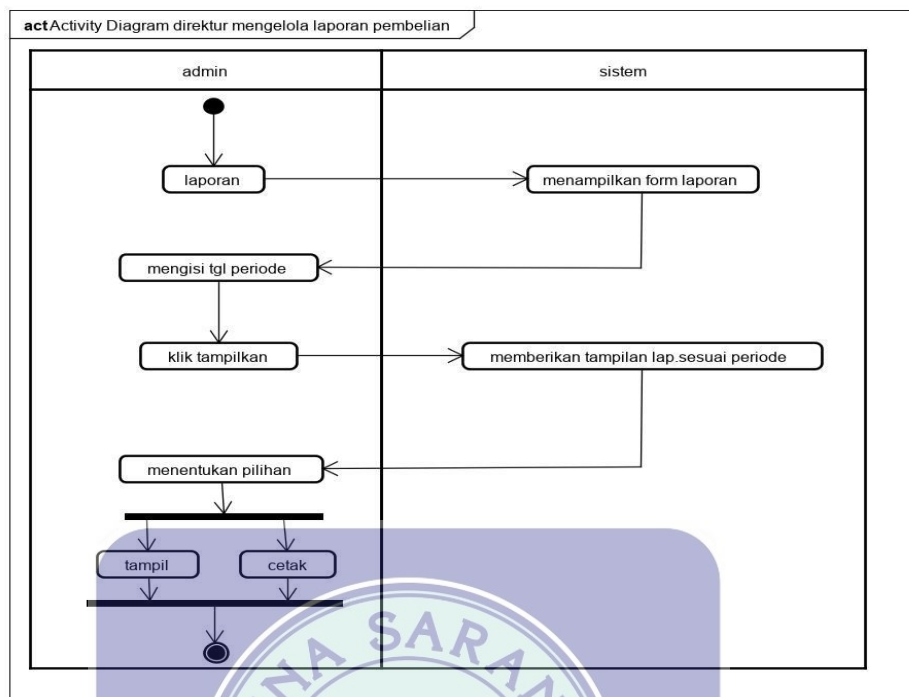
A.4. Activity Diagram Admin Mengelola jurnal



Sumber:Hasil Penelitian

Gambar.III.9.
Acitivity Diagram Admin Mengelola jurnal

A.5. Activity Diagram Admin mengelola Laporan

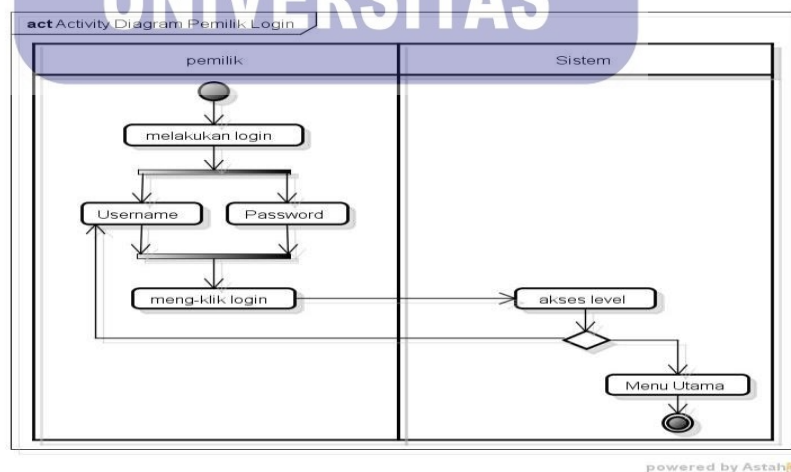


Sumber: Hasil Penelitian

Gambar.III.10.
Activity Diagram Admin Mengelola Laporan

B. Analisis Kebutuhan Pemilik

B.1. Activity diagram Direktur mengakses login

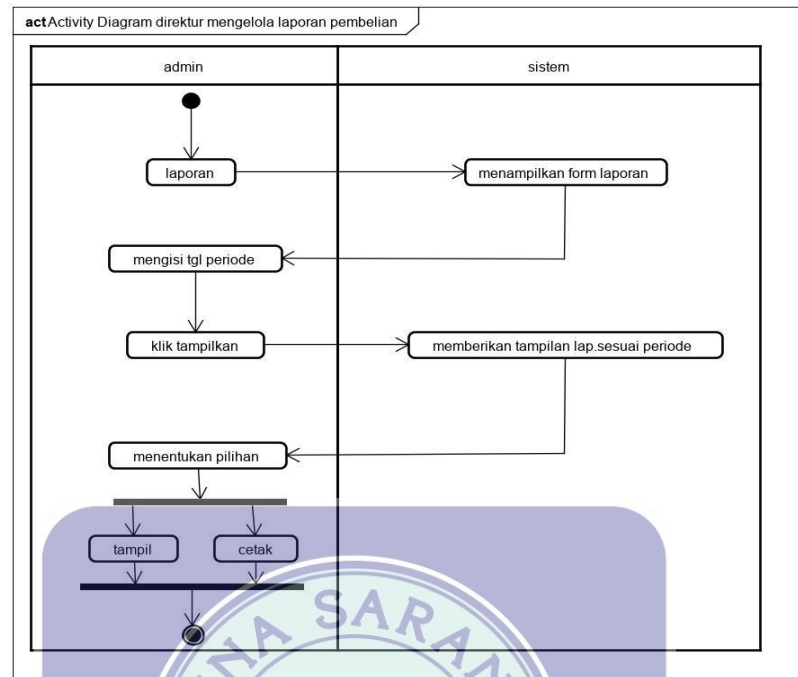


Sumber:Hasil penelitian

Gambar.III.11.

Activity diagram Direktur mengakses login

B.2. Activity Diagram Direktur Mengakses Laporan

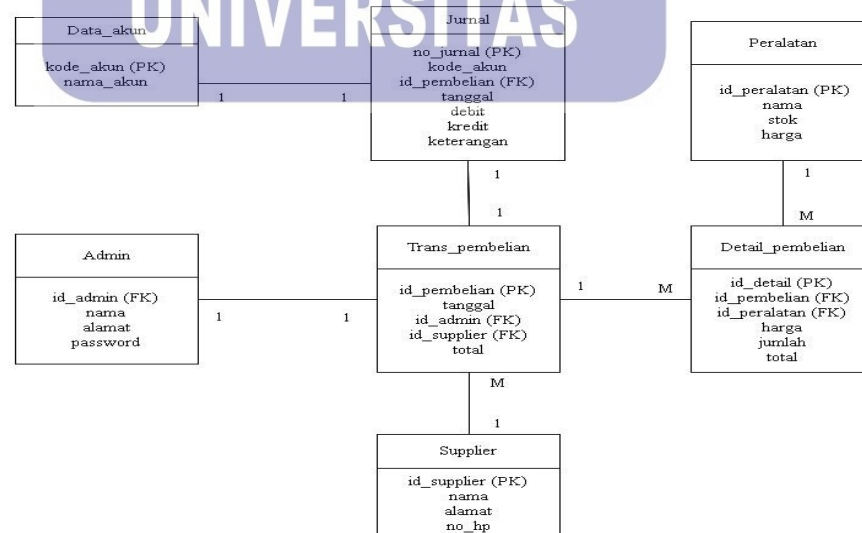


Sumber: Hasil Penelitian

Gambar III.12.
Activity Diagram Direktur Mengakses Laporan

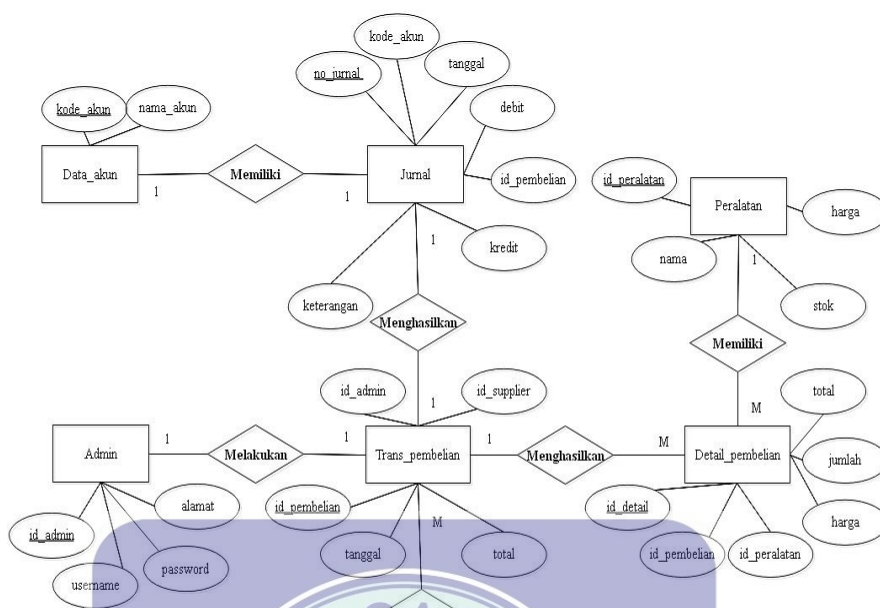
3.4.Desain

3.4.1. Logical Record Structure (LRS)



Gambar III.13.
Logical Record Structure (LRS)

3.4.2. Entity Relationship Diagram (ERD)



Gambar III.14.
Entity Relationship Diagram (ERD)

3.4.3. Spesifikasi File

Spesifikasi *file* mempermudah pemahaman terhadap spesifikasi data maka sebelumnya diuraikan kamus data yang merupakan elemen atau *field* yang digunakan. Adapun *file-file* yang akan digunakan dalam sistem komputerisasi dengan nama *database* dba_suciase dapat dijelaskan sebagai berikut:

1. Spesifikasi File Data Pengguna

Nama File	: tabel admin
Akronim	: admin
Fungsi	: Untuk menyimpan data admin
Tipe File	: File Master

Akses File : Random
 Media : Harddisk
 Kunci Field : id_admin
 Panjang Record : 41 Karakter
 Nama Database : pembelian
 Software : MySQL

Tabel IV.10.

Spesifikasi File Admin

No	Elemen data	Akronim	Tipe Data	Size	Keterangan
1	Id_admin	ID. Admin	Int	15	Primary Key
2	Nama	Nama	Varchar	10	-
3	Alamat	Alamat	Varchar	6	
4	Password	Password	Varchar	10	

2. Spesifikasi File master_akun

Nama File : Tabel master_akun
 Akronim : data akun
 Fungsi : Untuk menyimpan data akun
 Tipe File : File Master
 Akses File : Random
 Media : Harddisk
 Kunci Field : Kode_akun
 Ukuran Record : 25 Karakter
 Nama Database : pembelian
 Software : MySQL

Table IV.11.
Spesifikasi File master akun

No	Elemen data	Akronim	Tipe Data	Size	Keterangan
1	Kode_akun	Kode akun	<i>Varchar</i>	5	<i>Primary Key</i>
2	Nama_akun	Nama akun	<i>Varchar</i>	10	-
3	Jenis	Jenis	<i>Varchar</i>	10	-

3. Spesifikasi *File* data pembelian

Nama *File* : Tabel pembelian

Akronim : trans_pembelian

Fungsi : Untuk menyimpan data pembelian

Tipe *File* : Transaksi

Akses *File* : *Random*

Media : *Harddisk*

Kunci *Field* : id_pembelian

Ukuran *Record* : 41 Karakter

Nama *Database* : pembelian

Software : MySQL

Table IV.12.

Spesifikasi File Data pembelian

No	Elemen data	Akronim	Tipe Data	Size	Keterangan
1	Id_pembelian	ID. Pembelian	<i>Int</i>	15	<i>Primary Key</i>
2	Tanggal	Tanggal	<i>Varchar</i>	15	-
3	Id_admin	Id_admin	<i>Int</i>	11	-

4. Spesifikasi *File* Detail_pembelian

Nama *File* : Tabel detail pembelian

Akronim : pembelian

Fungsi : Untuk menyimpan data pembelian

Tipe File : Transaksi

Akses File : *Random*

Media : *Harddisk*

Kunci Field : id_detail

Ukuran Record : 58 Karakter

Nama Database : pembelian

Software : MySQL

Table IV.13.
Spesifikasi File Detail pembelian

No	Elemen data	Akronim	Tipe Data	Size	Keterangan
1	Id_detail	Id detail	<i>Int</i>	11	<i>Primary Key</i>
2	Id_pembelian	Id pembelian	<i>Varchar</i>	7	<i>Foreign Key</i>
3	Id_barang	Id supplier	<i>Varchar</i>	7	-
4	Harga	Id peralatan	<i>Varchar</i>	11	-
5	Jumlah	Harga	<i>Varchar</i>	11	-
6	Total	Jumlah	<i>Int</i>	11	-

5. Spesifikasi File data supplier

Nama File : Tabel data supplier

Akronim : data supplier

Fungsi : Untuk menyimpan data supplier

Tipe File : *File master data*

Akses File : *Random*

Media : *Harddisk*

Kunci Field : id_supplier

Ukuran Record : 45 Karakter

Nama Database : pembelian

Software : MySQL

Table IV.14.
Spesifikasi File data

No	Elemen data	Akronim	Tipe Data	Size	Keterangan
1	Id_supplier	ID. Supplier	<i>Int</i>	15	<i>Primary Key</i>
2	Nama	Nama	<i>Varchar</i>	10	-
3	Alamat	Alamat	<i>Varchar</i>	10	-
4	No_hp	No. Hp	<i>Varchar</i>	10	-

6. Spesifikasi *File* Data peralatan

Nama *File* : Tabel peralatan

Akronim : peralatan

Fungsi : Untuk menampilkan data peralatan

Tipe *File* : *File* master

Akses *File* : *Random*

Media : *Harddisk*

Kunci *Field* : id_peralatan

Ukuran *Record* : 30 Karakter

Nama *Database* : pembelian

Software : MySQL

Table IV.15.

Spesifikasi File peralatan

No	Elemen data	Akronim	Tipe Data	Size	Keterangan
1	Id_peralatan	Peralatan	<i>Varchar</i>	15	<i>Primary key</i>
2	Nama	Nama	<i>Varchar</i>		
3	Stok	Stok	<i>Varchar</i>		
4	Harga	Harga	<i>Varchar</i>	15	-

7. Spesifikasi *File* jurnal

Nama *File* : Tabel jurnal

Akronim : jurnal

Fungsi : Untuk menyimpan data transaksi

Tipe File : File laporan

Akses File : Random

Media : Harddisk

Kunci Field : no_referensi

Ukuran Record : 106 Karakter

Nama Database : pembelian

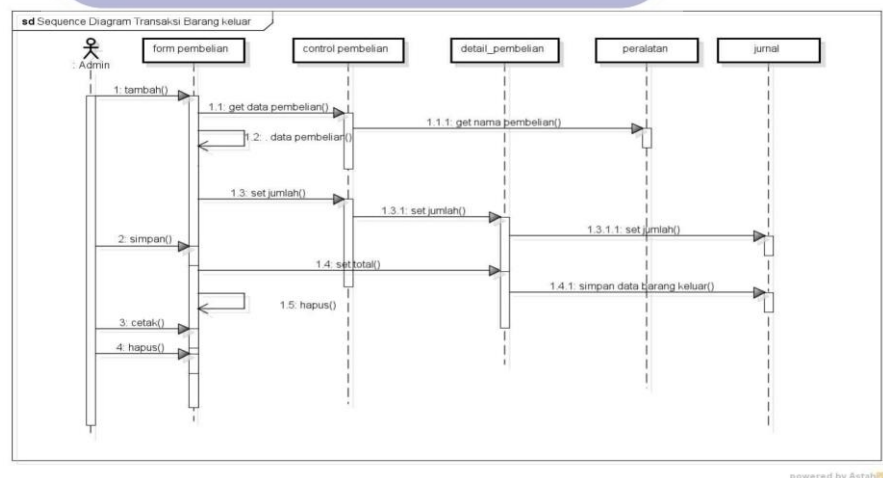
Software : MySQL

Table IV.16.
Spesifikasi File jurnal

No	Elemen data	Akronim	Tipe Data	Size	Keterangan
1	No_jurnal	No referensi	Varchar	14	Primary Key
2	Kode_akun	Kode Akun	Varchar	14	-
3	Tanggal	Tanggal	Date	-	
4	debit	Debit	Int	14	
5	Kredit	Kredit	Int	14	
6	Keterangan	Keterangan	Varchar	15	

3.4.4. Sequence Diagram

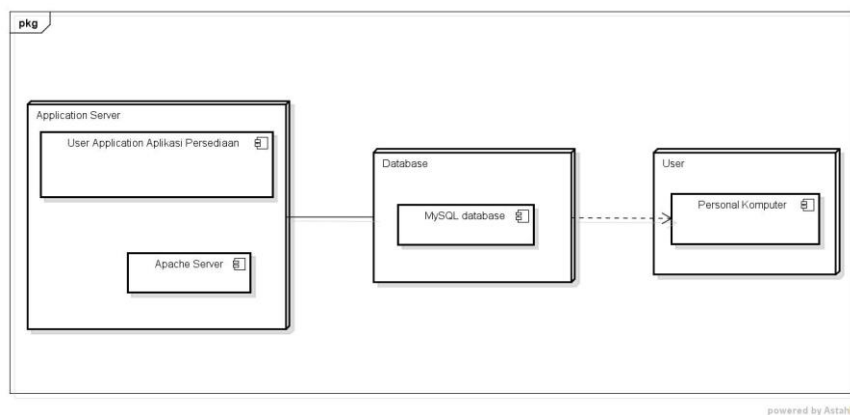
1. Sequence Diagram Transaksi pembelian



Sumber: Hasil Penelitian

Gambar.III.25.
Sequence Diagram transaksi pembelian

3.4.5. Deployment Diagram



Sumber:Hasil Penelitian

Gambar.III.27.
Deployment Diagram

3.4.6. User Interface

1. Form Login

The screenshot shows a web-based login form. At the top, there's a header area with a logo and the text 'BINA SARANA INFORMATIKA UNIVERSITAS'. Below this, the word 'LOGIN' is prominently displayed. The form itself has two input fields: one for 'Username' and one for 'Password'. A blue button with the text 'Masuk' is positioned at the bottom of the form. The entire interface is overlaid with a large, semi-transparent watermark of the university's logo.

Gambar.III.28.

Tampilan menu *Form Login*

2. Form Data Supllier

DATA SUPPLIER

ID Supplier

Nama Supplier

Alamat

NO HP

ID Supplier	Nama	Alamat	No Hp
SPL001	ptyuu	Cikampek	0899454

Gambar.III.29.

Tampilan menu Data *Supllier*3. *Form* Data Pembelian

Data Barang Pembelian

ID Pembelian	Tanggal	ID Admin	ID Supplier	Barang	Harga	Jumlah	Total
1	2019-07-09	1	SPL001	PRL001	20000	2	40000
2	2019-07-09	1	SPL001	PRL001	20000	2	40000
3	2019-07-09	1	SPL001	PRL001	20000	2	40000
4	2019-07-10	1	SPL001	PRL001	20000	2	40000
5	2019-07-10	1	SPL001	PRL001	20000	2	40000

Jumlah Data

Gambar.III.30.

Tampilan menu Data Pembelian

4. Menu Utama



Gambar.III.31.

Tampilan Menu Utama

5. Form Data peralatan

ID Peralatan	Nama	Stok	Harga
PRL001	Baja	17	20000
PRL002	undangan	12	10000

Gambar.III.32.

Tampilan Menu Data Peralatan

3.5. Implementasi

Pada bab ini menjelaskan tentang implementasi dan pengujian sistem. Tahap terakhir dari rancangan sistem persediaan barang pada CV Mora Mandiri Cikampek adalah menerapkan sistem yang telah dirancang untuk diterapkan pada keadaan sebenarnya. Jadwal implementasi dalam mewujudkan sistem yang baru dirancang tersebut sebagai berikut:

1. Analisa Sistem

Analisa sistem yaitu suatu kegiatan untuk mempelajari sistem yang berjalan serta masalah yang ada, yang bertujuan untuk mendapatkan gambaran tentang permasalahan yang terdapat pada CV Mora Mandiri Cikampek, sehingga mengurangi kesalahan antara sistem yang diusulkan dengan sistem yang berjalan.

2. Pengumpulan Data

Data-data yang ada di CV Mora Mandiri Cikampek dikumpulkan serta dikelompokkan sesuai dengan program yang akan dibuat.

3. Pembuatan Program dan Pengetesan Program

Dari data-data yang telah dikelompokkan dapat dibuat sesuai dengan data-data yang ada. Kemudian dilakukan pengetesan program agar program yang telah dibuat ataupun yang dirancang dapat diketahui kekurangannya.

4. Tes Sistem

kegiatan ini bertujuan untuk melaksanakan uji coba terhadap sistem secara langsung keseluruhan dan sampai dimana sistem ini dapat dipahami dengan baik dan benar.

3.5.1. Code Generation

* To change this template file, choose Tools | Templates

* and open the template in the editor.

```

*/

package pembelian;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

*

* @author Nisa

*/

public class Frm Pembelian extends javax.swing.JFrame {

    DefaultTableModel tb;

    /**
     * Creates new form NewJFrame
     */

    pembelian.koneksi1 konek = new pembelian.koneksi1();
    pembelian.koneksi conn = new pembelian.koneksi();

    public Frm Pembelian() {
        initComponents()
        tabelmodel();
    }

    public void tabelmodel(){

        tb =new DefaultTableModel();

        tb.addColumn("ID Pembelian");

        tb.addColumn("Tanggal");

        tb.addColumn("ID Admin");

        tb.addColumn("ID Supplier");

        tb.addColumn("Barang");

        tb.addColumn("Harga");

        tb.addColumn("Jumlah");
    }

```

```

tb.addColumn("Total");

tblanggaran.setModel(tb);

try {

    ResultSet res = conn.ambilData

    ("SELECT trans_pembelian.id_pembelian,

    trans_pembelian.tanggal, trans_pembelian.

    id_admin, trans_pembelian.id_supplier,

    detail_pembelian.id_barang, detail_pembelian.

    harga, detail_pembelian.jumlah, detail_pembelian.

    total as Total FROM trans_pembelian JOIN detail_pembelian

    ON detail_pembelian.id_pembelian = trans_pembelian.id_pembelian");

    while (res.next()){

        tb.addRow(new Object[]{res.getString(1),

        res.getString(2),res.getString(3),res.

        getString(4),res.getString(5),res.getString(6),

        res.getString(7),res.getString(8)});

    }

    tblanggaran.setModel(tb);

} catch (SQLException ex) {

    JOptionPane.showMessageDialog(this, ex);

}

}

/**

 * 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()

```

```

jButton1 = new javax.swing.JButton();

jPanel1 = new javax.swing.JPanel();

jLabel8 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

tblanggaran = new javax.swing.JTable()

{ public boolean isCellEditable(int rowIndex, int colIndex)

{ return false; } };

lblcount_rows = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

txtcari = new javax.swing.JTextField();

jButton3 = new javax.swing.JButton();

jButton1.setText("jButton1");

setDefaultCloseOperation

(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jPanel1.setBackground(new java.awt.Color(51, 51, 255));

jLabel8.setFont(new java.awt.Font("Lucida Grande", 0, 22)); // NOI18N

jLabel8.setForeground(new java.awt.Color(255, 255, 255));

jLabel8.setText("Data Barang Pembelian")

javax.swing.GroupLayout

jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup

(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(27, 27, 27)

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE,

244, javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

);

```

```

jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup
(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup())
.addContainerGap()
.addComponent(jLabel8)
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

tblanggaran.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}
},
new String [] {
"ID Barang Masuk", "Tanggal Masuk", "ID Barang",
"ID Admin", "Nama Barang", "ID Supplier", "jumlah", "Harga", "Total"
}
));

tblanggaran.addMouseListener(
(new java.awt.event.MouseAdapter() {
public void mouseClicked(java.awt.event.MouseEvent evt) {
tblanggaranMouseClicked(evt);
}
public void mouseReleased(java.awt.event.MouseEvent evt) {
tblanggaranMouseReleased(evt);
}
}));

tblanggaran.addKeyListener(new java.awt.event.KeyAdapter() {

```

```

public void keyPressed(java.awt.event.KeyEvent evt) {
    tblanggaranKeyPressed(evt);
}

public void keyReleased(java.awt.event.KeyEvent evt) {
    tblanggaranKeyReleased(evt);
}

});

jScrollPane1.setViewportView(tblanggaran);

lblcount_rows.setFont(new java.awt.Font
("Lucida Grande", 0, 12)); // NOI18N
lblcount_rows.setText("Jumlah Data ");
jButton2.setText("TAMBAH");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jButton3.setText("Cari");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

javax.swing.GroupLayout
layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);

layout.setHorizontalGroup
(layout.createParallelGroup
(javax.swing.GroupLayout.Alignment.LEADING)

```

```

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(layout.createSequentialGroup())

.addContainerGap()

.addGroup(layout.createParallelGroup
(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing
.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.
DEFAULT_SIZE, 745, Short.MAX_VALUE)

.addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup
(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblcount_rows, javax.swing.
GroupLayout.PREFERRED_SIZE, 228, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(layout.createSequentialGroup())

.addComponent(txtcari, javax.swing.
GroupLayout.PREFERRED_SIZE, 157, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addComponent(jButton3)

.addPreferredGap(javax.swing.LayoutStyle.
ComponentPlacement.RELATED)

.addComponent(jButton2)))

.addGap(0, 0, Short.MAX_VALUE)))

.addContainerGap())

);

layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.
GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup())

```

```

.addComponent(jPanel1, javax.swing.
GridLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.
ComponentPlacement.RELATED)
.addGroup(layout.createParallelGroup
(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jButton2)
.addComponent(txtcari, javax.swing.
GridLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton3))
.addGap(18, 18, 18)
.addComponent(jScrollPane1, javax.swing.
GridLayout.PREFERRED_SIZE, 307, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.
ComponentPlacement.RELATED)
.addComponent(lblcount_rows, javax.swing.
GridLayout.PREFERRED_SIZE, 26, javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

setSize(new java.awt.Dimension(781, 486));

setLocationRelativeTo(null);

} // </editor-fold>

private void tblanggaranMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

}

private void tblanggaranMouseReleased(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

}

```

```

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

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

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    new Frmtranspembelian(this, rootPaneCheckingEnabled).show();
    this.dispose();// TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    DefaultTableModel tabelnyo = new DefaultTableModel();
    tabelnyo.addColumn("ID Pembelian");
    tabelnyo.addColumn("Barang");
    tabelnyo.addColumn("Harga");
    tabelnyo.addColumn("Jumlah");
    tabelnyo.addColumn("Total");
    try{
        Connection conn = konek.openkoneksi();

        java.sql.Statement stm = conn.createStatement();

        java.sql.ResultSet sql = stm.executeQuery

        ("SELECT * FROM detail_pembelian where id_pembelian like

        '%" + txtcari.getText()

        + '%" + "or id_barang like '%" + txtcari.getText() + '%"");

        while (sql.next()) {
            tabelnyo.addRow(new Object[]{

            sql.getString(2),

            sql.getString(3),

            sql.getString(4),

```

```

sql.getString(5),
sql.getString(6)
});
}

tblanggaran.setModel(tabelnyo);
} catch (Exception e){
}
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    <code>//<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
            (Frmpembelian.class.getName()).
            log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger
            (Frmpembelian.class.getName()).
            log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger
            (Frmpembelian.class.getName()).
            log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger
            (Frmpembelian.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 Frmmpembelian().setVisible(true);
});
}

{new Frmtranspembelian(this, rootPaneCheckingEnabled).show();
this.dispose();// TODO add your handling code here:
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

```

```

DefaultTableModel tabelnyo = new DefaultTableModel();

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JLabel jLabel8;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JLabel lblcount_rows;

private javax.swing.JTable tblanggaran;

private javax.swing.JTextField txtcari;

// End of variables declaration
}

pembelian.koneksi1 konek = new pembelian.koneksi1();
pembelian.PetugasSession PetugasSession = new pembelian.PetugasSession();

public login() {
    initComponents();
}

private void initUI(){
    getContentPane().setBackground(new Color(245, 245, 245));

    Dimension windowSize = getSize();

    GraphicsEnvironment ge = GraphicsEnvironment.getLocalGraphicsEnvironment();

    Point centerPoint = ge.getCenterPoint();

    int dx = centerPoint.x - windowSize.width / 2;

    int dy = centerPoint.y - windowSize.height / 2;

    setLocation(dx, dy);
}

private void btnloginActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    String row_txtusername = txtusername.getText();

```

```

String row_txtpassword = txtpassword.getText();

if (txtusername.getText().equals("")) {

JOptionPane.showMessageDialog(this,"username  Harus Di Isi");

}

else if (txtpassword.getText().equals("")) {

JOptionPane.showMessageDialog(this,"password dan  Harus Di Isi");

}

else{

try {

Connection conn = konek.openkoneksi();

java.sql.Statement stm = conn.createStatement()

java.sql.ResultSet rsLogin = stm.executeQuery

("SELECT * FROM admin WHERE nama = " + row_txtusername

+ " AND password = " + row_txtpassword + "");

rsLogin.next();

rsLogin.last();

if (rsLogin.getRow()==1){

PetugasSession.setU_id(rsLogin.getInt("id_admin"))

PetugasSession.setU_username(rsLogin.getString("nama"));

new menu_utama().setVisible(true);

this.dispose();

}

else

JOptionPane.showMessageDialog(null, "Maaf, Username atau password anda salah.");

txtusername.setText("")

txtpassword.setText("");

txtusername.requestFocus();

}

konek.closekoneksi();

} catch (SQLException e) {

```

```

OptionPane.showMessageDialog(null, "Error " + e);

} catch (ClassNotFoundException ex) {

Logger.getLogger(login.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

/**

 * @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;
            }
        }
    }

    //</editor-fold>

    //</editor-fold>

    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {

```

```

public void run() {

new login().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton btnlogin;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPasswordField txtpassword;

private javax.swing.JTextField txtusername;// End of variables declaration

```

3.5.2. Blackbox Testing

1. Pengujian terhadap form login

No	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Klik Tombol Login	User name dan password tidak di isi	Sistem akan menolak dan akan muncul pemberitahuan "user name belum di isi"	Sesuai harapan	Valid
2.	Klik tombol Login	User name diisi dan password kosong	Sistem akan menolak dan akan muncul pemberitahuan "password belum di isi"	Sesuai harapan	Valid
3.	Klik text login	User name tidak di isi dan password di isi	Sistem akan menolak dan akan muncul pemberitahuan "user name belum di isi"	Sesuai harapan	Valid

4	Klik tombol Login	User name dan password di isi	Sistem akan menerima dan akan muncul menu utama	Sesuai harapan	Valid
---	-------------------	-------------------------------	---	----------------	-------

2. Pengujian terhadap *form supplier*

No	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Klik tombol simpan	Kode supplier sama	Sistem akan menolak dan akan muncul pemberitahuan “kode suplier sudah pernah di simpan”	Sesuai harapan	Valid
2.	Klik tombol simpan	Input text ada yang kosong	Sistem akan menolak dan akan muncul pemberitahuan “terdapat inputan kosong”	Sesuai harapan	Valid
3.	Klik tombol Simpan	Text di isi semua dan kode supliier berbeda	Sistem akan menerima dan menyimpan data	Sesuai harapan	Valid

3.5.3. Spesifikasi Hardware dan Software

1. Perangkat Keras (*hardware*)

Perangkat keras atau *hardware* adalah serangkaian unsur-unsur yang terdiri dari beberapa perangkat keras yang membentuk suatu sistem komputer yang digunakan. Perangkat keras memiliki beberapa komponen diantaranya komponen masukan (*input device*), komponen pemroses (*Processing device*), komponen keluaran (*output device*), dan komponen simpanan luar (*storage*).

Adapun perangkat keras yang penulis sarankan agar proses pengolahan data pada CV Mora Mandiri Cikampek berjalan dengan baik dan maksimal adalah sebagai berikut:

- a. *Processor* : CPU N1350
- b. *Memory Size* : 4.00 GB (RAM)
- c. *Hardisk* : 350 GB
- d. *Mouse* : *Standard Mouse*
- e. *Keyboard* : 108 Keys
- f. *Monitor* : SVGA 15 Inch
- g. *Printer* : *Inkjet*

2. Perangkat Lunak (*software*)

Perangkat lunak atau *software* adalah serangkaian unsur-unsur yang terdiri dari beberapa perangkat lunak yang membentuk suatu sistem komputer yang digunakan untuk mengoperasikan kerja pengguna. Fungsi perangkat lunak adalah untuk mengidentifikasi dan menyiapkan aplikasi program sehingga tata kerja seluruh peralatan komputer dapat terkontrol, serta membuat aktifitas kerja menjadi lebih efisien. Perangkat lunak yang digunakan adalah sebagai berikut:

- a. Sistem Operasi : *Windows 8*
- b. Program Aplikasi : *Java Application*
- c. Bahasa Pemrograman : *Netbeans IDE 8.1*
- d. *Database* : *MySQL Database*
- e. Web Browser : *Google Chrome*
- f. Server : *Xampp*