

BAB III

ANALISA SISTEM BERJALAN

3.1. Tinjauan perusahaan

Kantor Desa Gempolsari adalah instansi pemerintahan yang bergerak di bidang pelayanan masyarakat. Kantor Kepala Desa merupakan satu instansi yang melakukan pendataan penduduk terutama pendataan Kartu Tanda Penduduk (KTP), Kartu Keluarga (KK), Surat Kelahiran, Surat Kematian, dan Surat Keterangan Pindah, dan juga instansi yang merekap masyarakat miskin dari setiap dusun di desa.

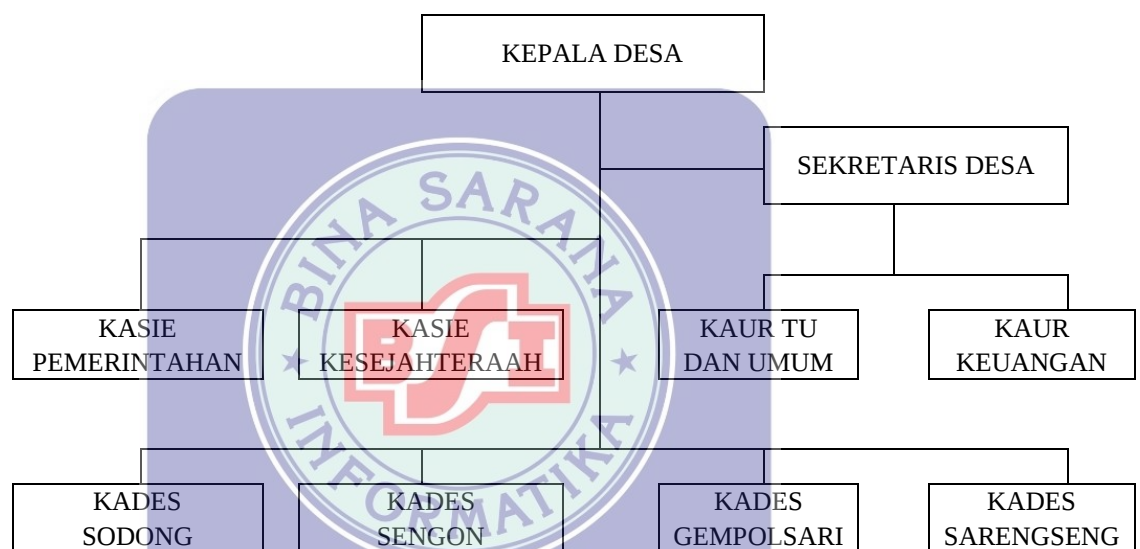
Dalam operasioanalnya, kantor ini mencatat bukti-bukti transaksi masih menggunakan sistem yang sederhana seperti catatan buku kas. Sehingga hal ini tentunya memiliki beberapa resiko, seperti hilangnya bukti transaksi atau dapat menyebabkan kekeliruan data, kesalahan dalam memasukan data kas masuk dan keluar oleh bagian keuangan serta kurangnya efektifnya dalam membuat laporan keuangan. Bagian keuangan harus merekap satu persatu kas masuk dan keluar yang kemudian dimasukkan kedalam buku besar untuk dijadikan laporan keuangan. Untuk itu dibutuhkan sebuah aplikasi kas masuk dan keluar yang diharapkan dapat memecahkan masalah-masalah yang sedang terjadi.

3.1.1. Sejarah Perusahaan

Berdasarkan sejarah, Desa Gempolsari merupakan pecahan Desa Ciberes. Desa Gempolsari adalah desa yang terletak di Jalur Pantura, Kecamatan Patokbeusi, Kabupaten Subang, Jawa Barat. Desa ini terdiri atas 4 dusun, yaitu Gempolsari, Sarengseng, Sodong, dan Sengon, 8 Rukun Warga (RW) , dan 20 Rukun Tetangga (RT).Sebelah utara Desa Gempolsari berbatasan dengan Desa Tambak Jati, sebelah

barat dengan Desa Ciberes, sebelah timur dengan Desa Sukamandi Jaya, sedangkan sebelah selatan berbatasan dengan kecamatan Pabuaran dan desa Rancamulya. Luas hamparan wilayah Desa Gempolsari kurang lebih seluas 507,408 Ha, terdiri dari : areal tanah sawah teknis : 380,000 Ha, areal pemukiman : 106,705 Ha dan tanah lainnya : 20,703 Ha.

3.1.2. Struktur Organisasi Dan Fungsi



Sumber : Kantor Desa Gempolsari

Gambar III.1. Struktur Organisasi Kantor Desa Gempolsari

Adapun fungsi-fungsi setiap jenjang dari struktur organisasi Kantor Desa Gempolsari Kecamatan Patokbeusi Kabupaten Subang tersebut, yaitu:

1. Kepala Desa

- a. Kepala desa/lurah mempunyai tugas menyelenggaraan urusan pemerintahan, pembangunan, dan kemasyarakatan.
- b. Menyelenggarakan pemerintahan desa berdasarkan kebijakan yang ditetapkan bersama BPD

- c. Mengajukan rancangan peraturan Desa
- d. Menetapkan peraturan-peraturan yang telah mendapatkan persetujuan bersama BPD
- e. Menyusun dan mengajukan rancangan peraturan desa mengenai APB Desa untuk dibahas dan ditetapkan bersama BPD
- f. Membina kehidupan masyarakat Desa
- g. Membina ekonomi desa
- h. Mengordinasikan pembangunan desa secara partisipatif
- i. Mewakili desanya di dalam dan luar pengadilan dan dapat menunjuk kuasa hukum untuk mewakilinya sesuai dengan peraturan perundang-undangan; dan
- j. Melaksanakan wewenang lain sesuai dengan peraturan perundangundangan.

2. Sekretaris Desa

- a. Tugas Pokok : Membantu Kepala Desa dalam mempersiapkan dan melaksanakan pengelolaan administrasi Desa, mempersiapkan bahan penyusunan laporan penyelenggaraan Pemerintahan Desa.
- b. Fungsi :
 - 1) Penyelenggara kegiatan administrasi dan mempersiapkan bahan untuk kelancaran tugas Kepala Desa.
 - 2) Melaksanakan tugas kepala desa dalam hal kepala desa berhalangan
 - 3) Melaksanakan tugas kepala desa apabila kepala desa diberhentikan sementara
 - 4) Penyiapan bantuan penyusunan Peraturan Desa
 - 5) Penyiapan bahan Laporan Penyelenggaraan Pemerintahan Desa
 - 6) Pengkoordinasian Penyelenggaraan tugas-tugas urusan; dan
 - 7) Pelaksanaan tugas lain yang diberikan oleh Kepala Desa.

3. Kaur Keuangan (Bendahara)

Fungsi utama perbendaharaan meliputi perencanaan kas yang baik, pencegahan terjadinya kebocoran/penyimpangan, pencarian sumber pembiayaan yang paling murah, dan menghindari adanya dana yang menganggur (tidak terserap). Secara lebih rinci tergambar sbb.:

- a. Menerima, menyimpan, menatausahakan, dan membukukan uang/surat berharga dalam pengelolaannya
- b. Melakukan pengujian dan pembayaran berdasarkan perintah
- c. Menolak perintah pembayaran apabila tidak memenuhi persyaratan untuk dibayarkan
- d. Melakukan pemotongan/pemungutan penerimaan negara dari pembayaran yang dilakukannya
- e. Menyetorkan pemotongan/pemungutan kewajiban ke kas negara
- f. Mengelola rekening tempat penyimpanan dan
- g. Menyampaikan Laporan Pertanggungjawaban (LPJ) kepada Kepala Desa.

4. Kaur Umum

Tugas utama : Membantu sekretaris desa dalam melaksanakan administrasi umum, tata usaha dan kearsipan, pengelolaan inventaris kekayaan desa desa serta mempersiapkan agenda rapat dan laporan.

Fungsi :

- 1) Pelaksana kegiatan bidang pembina kehidupan masyarakat desa.
- 2) Pelaksana inventarisasi, pembinaan dan pelestarian kebudayaan yang berlaku di desa.

- 3) Pelaksana kegiatan perencanaan bidang kemasyarakatan dan sosial budaya desa.

5. Kasi Kesejahteraan

a. Tugas :

- 1) Meningkatkan kesejahteraan masyarakat desa meliputi pembinaan di bidang agama, kesehatan, pendidikan, olahraga, dan kesenian.
- 2) Membantu Kepala Desa dalam melaksanakan penyiapan bahan perumusan kebijakan teknis Penyusunan Program Keagamaan serta melaksanakan Program pemberdayaan masyarakat dan sosial kemasyarakatan.

b. Fungsi :

- 1) Penyiapan dan pelaksanaan program perkembangan kehidupan beragama.
- 2) Penyiapan bahan k dan pelaksanaan program, pemberdayaan masyarakat dan sosial kemasyarakatan.
- 3) Pelaksanaan tugas-tugas lain yang diberikan Kepala Desa

6. Kasi Pemerintahan

- ### **a. Tugas Pokok :** Membantu Kepala Desa dalam melaksanakan pengelolaan administrasi kependudukan, administrasi pertanahan, pembinaan, ketentraman dan ketertiban masyarakat Desa, mempersiapkan bahan perumusan kebijakan penataan, Kebijakan dalam Penyusunan produk hukum Desa.

b. Fungsi :

- 1) Pelaksanaan kegiatan administrasi kependudukan
- 2) Persiapan bahan-bahan penyusunan rancangan peraturan Desa dan keputusan Kepala Desa

- 3) Pelaksanaan kegiatan administrasi pertanahan
- 4) Pelaksanaan Kegiatan pencatatan monografi Desa
- 5) Persiapan bantuan dan melaksanakan kegiatan penataan kelembagaan masyarakat untuk kelancaran penyelenggaraan pemerintahan Desa
- 6) Persiapan bantuan dan melaksanakan kegiatan kemasyarakatan yang berhubungan dengan upaya menciptakan ketentraman dan ketertiban masyarakat dan pertahanan sipil; dan
- 7) Pelaksanaan tugas-tugas lain yang diberikan kepada Desa.

7. KADUS (Kepala Dusun)

a. Tugas :

- 1) Membantu pelaksanaan tugas kepala desa dalam wilayah kerjanya.
- 2) Melakukan pembinaan dalam rangka meningkatkan swadaya dan gotong royong masyarakat.
- 3) Melakukan kegiatan penerangan tentang program pemerintah kepada masyarakat.
- 4) Membantu kepala desa dalam pembinaan dan mengkoordinasikan kegiatan RW (RakunWarga) dan RT (Rukun Tetangga) di wilayah kerjanya.
- 5) Melaksanakan tugas lain yang diberikan oleh kepala desa.

b. Fungsi :

- 1) Melakukan koordinasi terhadap jalannya pemerintah desa, pelaksanaan pembangunan dan pembinaan masyarakat di wilayah dusun.
- 2) Melakukan tugas dibidang pembangunan dan pembinaan kemasyarakatan yang menjadi tanggung jawabnya.

- 3) Melakukan usaha dalam rangka meningkatkan partisipasi dan swadaya gotong royong masyarakat dan melakukan pembinaan perekonomian.
- 4) Melakukan kegiatan dalam rangka pembinaan dan pemeliharaan ketentraman dan ketertiban masyarakat.
- 5) Melakukan fungsi-fungsi yang dilimpahkan oleh kepala desa.

3.2. Tinjauan Kasus

Kasus yang diteliti adalah arus kas pada Desa Gempolsari Kabupaten Subang. Pada pembahasan kasus ini dimulai dari proses bisnis sistem berjalan yang dimodelkan kedalam *activity diagram*, dokumen masukan, dokumen keluaran, permasalahan pokok dan pemecahan masalah.

3.2.1. Proses Bisnis Sistem Berjalan

Suatu proses atau tahap-tahapan yang dilakukan sebelum memulai suatu kegiatan untuk menyelesaikan suatu pekerjaan disebut proses sistem. Sesuai dengan yang dibahas dalam Laporan ini, maka proses sistem berjalan yang diambil dari Kantor Desa Gempolsari yaitu sistem pembuatan laporan arus kas masuk dan keluar yang selama ini telah melewati beberapa proses.

Proses bisnis sistem berjalan pembuatan laporan arus kas masuk dan kas keluar secara umum pada Kantor Desa Gempolsari adalah :

1. Proses Pengambilan Dana Desa

Kas masuk desa bersumber dari alokasi dana desa yang diajukan setiap satu tahun sekali. Kepala desa mendampingi bendahara desa pergi ke Bank BJB untuk pengambilan Alokasi Dana Desa (ADD) dengan mengisi slip penarikan yang sudah disediakan Bank BJB. Kepala Desa memberikan dana tersebut kepada Bendahara Desa untuk disimpan dan mengarsipkan slip penarikan yang asli. Kemudian Bendahara Desa mencatat data penerimaan dana ke dalam kas masuk.

2. Proses Pengeluaran Dana Kas

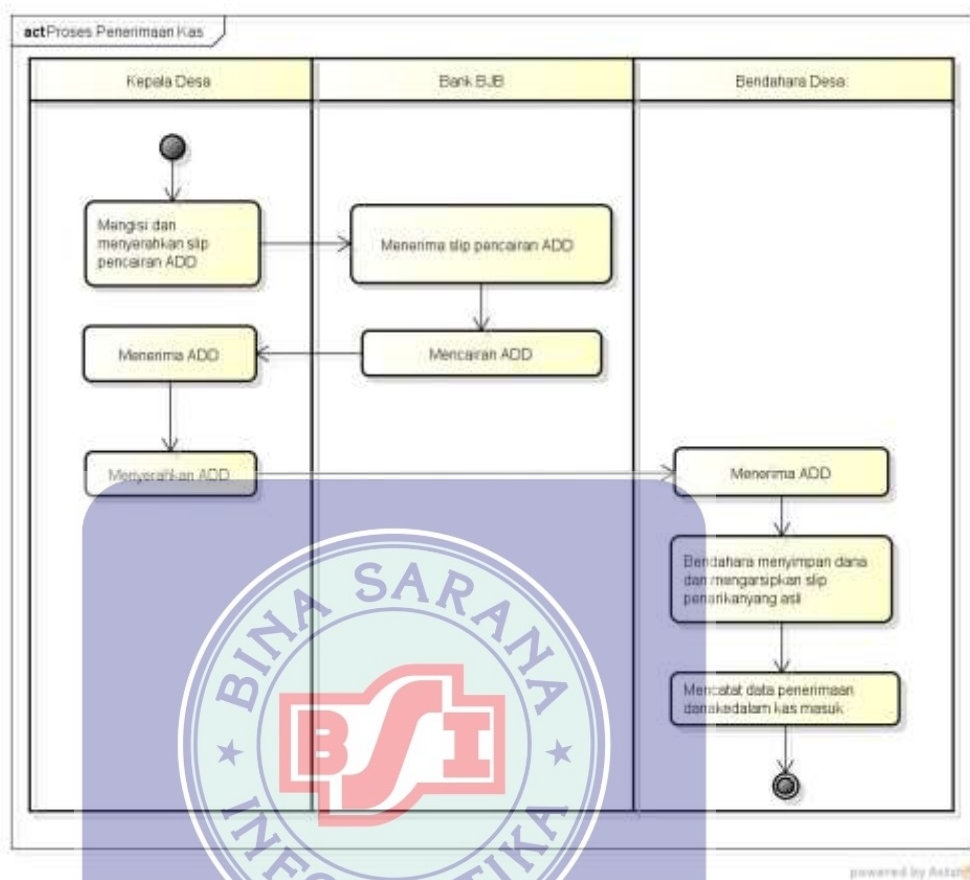
Pada prosedur ini, Bendahara Desa memberikan dana dan rencana anggaran biaya kepada KASI yang sudah disetujui Kepala Desa untuk biaya yang dibutuhkan, kemudian bagian KASI memberikan bukti pengeluaran kas berupa kwitansi kepada Bendahara Desa. Bendahara Desa kemudian mencatatnya ke dalam buku kas keluar dan mengarsipkan kwitansi tersebut.

3. Proses Laporan Keuangan

Bendahara Desa membuat laporan-laporan keuangan yang ada setiap bulannya berupa laporan buku kas umum (BKU), dan laporan pertanggungjawaban penerimaan dan pengeluaran. Laporan buku kas umum diketahui Kepala Desa, sedangkan laporan pertanggungjawaban penerimaan dan pengeluaran tersebut harus diverifikasi Sekretaris Desa terlebih dahulu. Setelah Sekretaris memverifikasi laporan pertanggungjawaban penerima dan pengeluaran tersebut kemudian Bendahara Desa menyerahkan laporan-laporan tersebut.

3.2.2. Activity Diagram

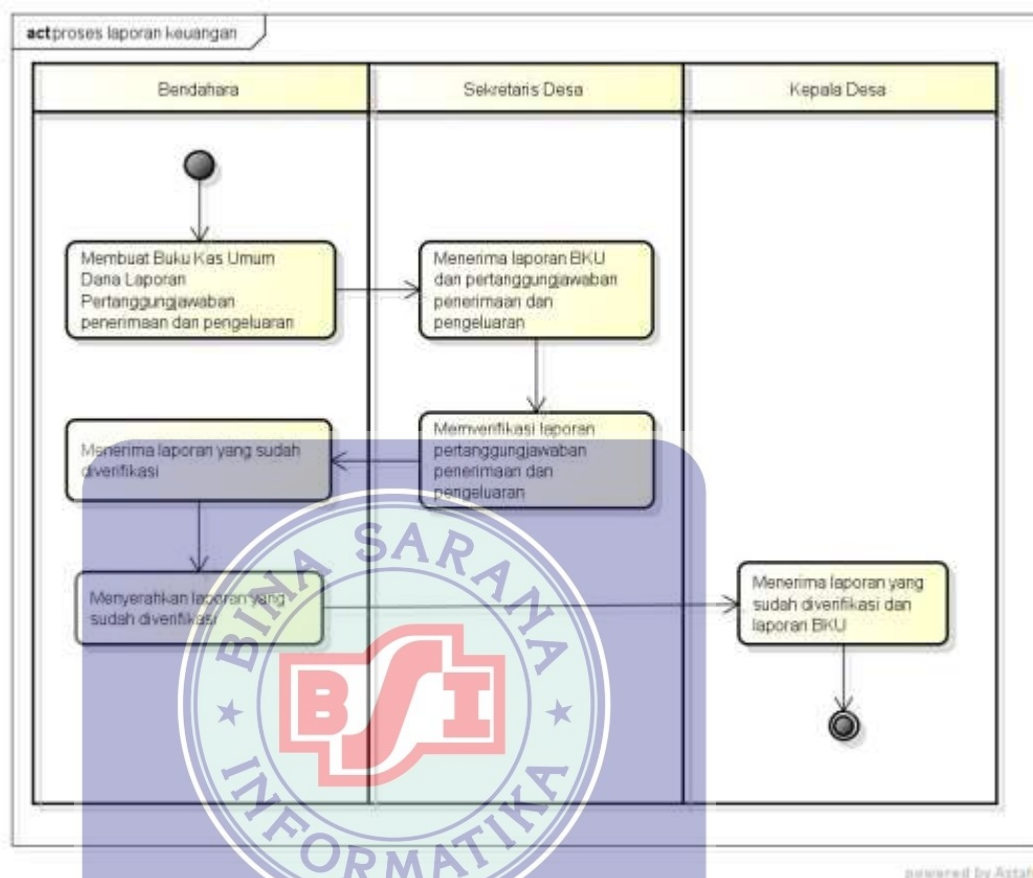
a. Proses Penerimaan Kas



Sumber : Hasil penelitian (2019)

Gambar III.2. Activity Diagram Proses Penerimaan Kas pada Kantor Desa Gempolsari

c. Proses Laporan Keuangan



Sumber : Hasil penelitian (2019)

**Gambar III.4. Activity Diagram Proses Laporan Keuangan pada Kantor
Desa Gempolsari**

3.2.3. Spesifikasi Bentuk 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. Adapun dokumen-dokumen masukan tersebut adalah sebagai berikut :

1. Kwitansi

Nama Dokumen : Kwitansi

Fungsinya : Sebagai bukti dari pengeluaran kas

Sumber : KASI

Jumlah : Satu lembar

Media : Kertas

Tujuan : Bendahara

Frekuensi : Setiap terjadi pengeluaran kas

Bentuk Dokumen : Lampiran A.1

2. Slip penarikan

Nama Dokumen : Slip penarikan

Fungsinya : Sebagai bukti dari penarikan kas

Sumber : Bank BJB

Jumlah : Satu lembar

Media : Kertas

Tujuan : Bendahara Desa

Frekuensi : Setiap terjadi pengeluaran kas

Bentuk Dokumen : Lampiran A.2



3.2.4. Spesifikasi Bentuk Dokumen Keluaran

Dokumen keluaran adalah segala bentuk dokumen perusahaan berupa dokumen-dokumen yang akan mendukung kegiatan manajemen serta merupakan dokumen hasil pencatatan atau laporan. Adapun dokumen-dokumen keluaran tersebut adalah sebagai berikut :

1. Rencana Anggaran Biaya

Nama Dokumen : Rencana Anggaran Biaya

Fungsi : Sebagai syarat persetujuan dari rencana anggaran biaya

Sumber : Bendahara Desa

Jumlah : Satu lembar

Media : Kertas

Tujuan : Kepala Desa

Frekuensi : Setiap terjadi perencanaan anggaran biaya

Bentuk Dokumen : Lampiran B.1

2. Buku Kas Umum

Nama Dokumen : Buku Kas Umum

Fungsinya : Sebagai laporan dari kas masuk dan kas keluar
dalam bentuk buku kas umum

Sumber : Bendahara Desa

Jumlah : Satu lembar

Media : Kertas

Tujuan : Kepala Desa

Frekuensi : Setiap akhir bulan

Bentuk Dokumen : Lampiran B.2

3. Laporan Pertanggung jawaban

Nama : Laporan Pertanggungjawaban

Fungsinya : Sebagai laporan pertanggungjawaban atas penerima
kas dan pengeluaran kas

Sumber : Bendahara Desa

Jumlah : Satu lembar

Media : Kertas

Tujuan : Kepala Desa

Frekuensi : Setiap akhir bulan

Bentuk Dokumen : Lampiran B.3

3.2.5. Permasalahan Pokok

Dari hasil pengamatan serta uraian yang telah penulis paparkan di atas, dapat diambil kesimpulan bahwa dalam proses pelaporan kas masuk dan kas keluar yang terjadi di Kantor Desa Gempolsari masih dilakukan secara sistem yang sederhana. Dengan dilakukan proses pengolahan data dalam penerima kas yang masih secara sederhana dimana dalam melakukan pembuatan laporan kas masuk dan kas keluar masih menggunakan kertas sehingga kinerjanya masih belum efektif.

1. Pencatatan dan perekapan data pengeluaran dana dalam buku membutuhkan waktu lama sehingga laporan keuangan sering terlambat dilaporkan kepada atasan.
2. Sering terjadi kekeliruan dalam pengisian data sehingga data kurang akurat dan efektif.
3. Proses pencatatan data lama.
4. Membutuhkan ruang penyimpanan data yang luas untuk pengarsipan dokumen dana kas keluar dan kas masuk.

3.2.6. Pemecahan Masalah

Untuk memecahkan masalah yang ada di Kantor Desa Gempolsari, khususnya pada bagian yang mengelola pelaporan arus kas, penulis mengusulkan untuk menggunakan sistem yang telah terkomputerisasi sehingga permasalahan yang sering terjadi dapat teratasi dengan baik, dan juga dengan menggunakan sistem komputerisasi maka dapat mempermudah dalam penggunaannya serta mengefisiensikan proses kerja agar lebih baik dan cepat. Selain itu, apabila sebuah perusahaan/instansi telah menggunakan sistem yang terkomputerisasi dapat meminimalisasi adanya kecurangan juga kesalahan yang disebabkan oleh kesalahan manusia (*human error*) .

Penulis mengusulkan sebuah sistem yang terkomputerisasi untuk menyelesaikan atau mengatasi permasalahan yang ada pada kantor desa Gempolsari. Aplikasi berbasis desktop dengan menggunakan bahasa pemrograman *Java Netbeans*. media penyimpanan menggunakan XAMPP yang merupakan *database* server.

Dengan aplikasi tersebut maka :

1. Pencatatan dan proses perekapan dapat dilakukan dengan cepat, karena laporan dibuat otomatis dan aplikasi yang merekam setiap transaksi.
2. Pencatatan untuk pengeluaran dan pemasukan dana yang bersifat tetap hanya perlu diisi sekali, karena untuk transaksi berikutnya cukup mengambil data dari *database*.
3. Mempercepat proses pencarian dengan memanfaatkan fitur pencarian pada aplikasi.
4. Tidak membutuhkan ruang penyimpanan yang luas, karena data tersimpan dalam bentuk digital dengan memanfaatkan *database*.

3.3. Analisa kebutuhan software

Analisa kebutuhan merupakan langkah awal untuk menentukan gambaran perangkat yang akan dihasilkan ketika pengembang melaksanakan sebuah proyek pembuatan perangkat lunak. Perangkat lunak yang baik dan sesuai dengan kebutuhan pengguna sangat tergantung pada keberhasilan dalam melakukan analisis kebutuhan. Untuk proyek-proyek perangkat lunak yang besar, analisa kebutuhan dilakukan setelah aktivitas sistem informasi *engineering* dan *software project* dilakukan.

3.3.1. Analisa kebutuhan

A. Bagian Keuangan

A.1. Mengakses *Login*

A.2. Mengelola Data Master

A.2.1. Mengelola Master Akun

A.2.2. Mengelola Saldo Awal

A.3. Mengelola Transaksi

A.3.1. Mengelola kas masuk

A.3.2. Mengelola Kas keluar

A.3.3. Mengelola Jurnal Umum

A.4. Mangelola Cetak Laporan

A.4.1. Mengelola cetak laporan kas masuk

A.4.2. Mengelola cetak laporan kas keluar

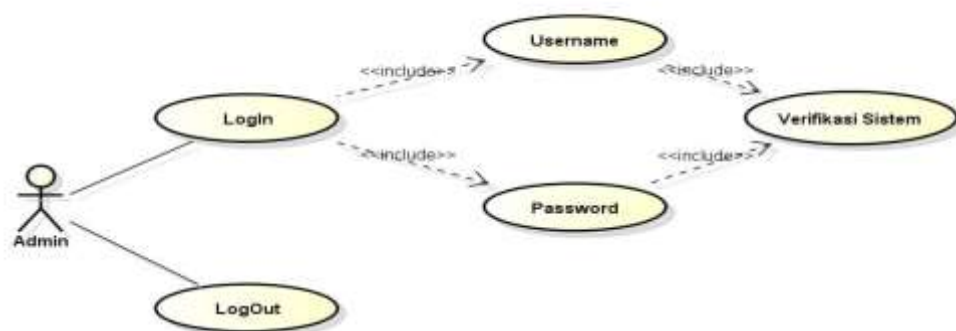
A.4.3. Mengelola cetak laporan jurnal

3.3.2. Use Case Diagram

Tahapan analisis yang telah diuraikan akan dimodelkan menjadi *usecase diagram* dengan maksud untuk menggambarkan sistem menjadi model logika yang dapat dipahami lebih lanjut atau memberikan deskripsi lanjut mengenai rancang bangun sistem informasi akuntansi arus kas pada desa gempolsari kecamatan patobeusi kabupaten subang. Hasil pemodelan sistem menjadi *usecase diagram* dapat dilihat pada gambar berikut ini.

1. Use Case diagram Login

Berikut adalah proses login ke menu utama:



Sumber : Hasil penelitian (2019)

Gambar III.5. UseCase Diagram Login

Pada *Use case* Diagram login ini menjelaskan tentang langkah-langkah yang dilakukan admin untuk bisa masuk ke sistem.

Tabel III.1.

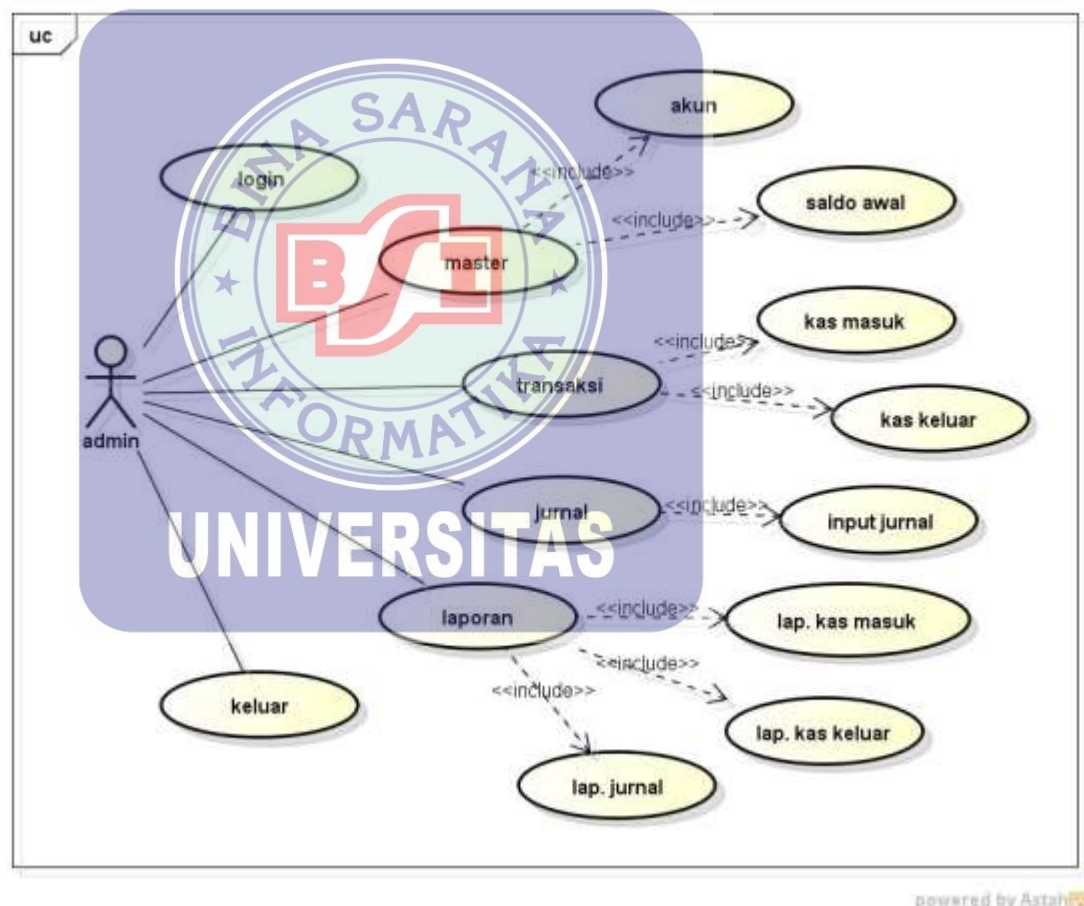
Deskripsi Use Case Login

Use Case Narrative Login

Tujuan	user melakukan login dan mengisi <i>username</i> dan <i>password</i> agar bisa masuk ke menu utama.
Deskripsi	Sistem ini memungkinkan user untuk mengakses sistem yang ada pada menu utama.
Skenario Utama	
Aktor	<i>User</i>
Kondisi Awal	User Membuka Aplikasi pelaporan arus kas masuk dan keluar
Aksi Aktor	Reaksi Sistem

1. Aktor Memilih tombol login	Sistem akan menampilkan <i>textbox</i> untuk mengisi <i>username</i> dan <i>password</i>
2. Aktor Memilih tombol batal	Sistem akan menutup aplikasi
Kondisi Akhir	Jika perintah sesuai maka sistem akan masuk kedalam aplikasi pelaporan arus kas maka aktor dapat melakukan aktivitas didalam sistem.

2. UseCase Diagram Usulan Sistem Informasi Arus Kas Desa Gempolsari



Sumber : Hasil penelitian (2019)

Gambar III.6. UseCase Diagram Arus Kas

Pada Gambar III.6 dapat di jelaskan bahwa rancangan sistem informasi dapat mengolah data master, mengolah data transaksi, data jurnal, dan juga data laporan.

Tabel III.2.

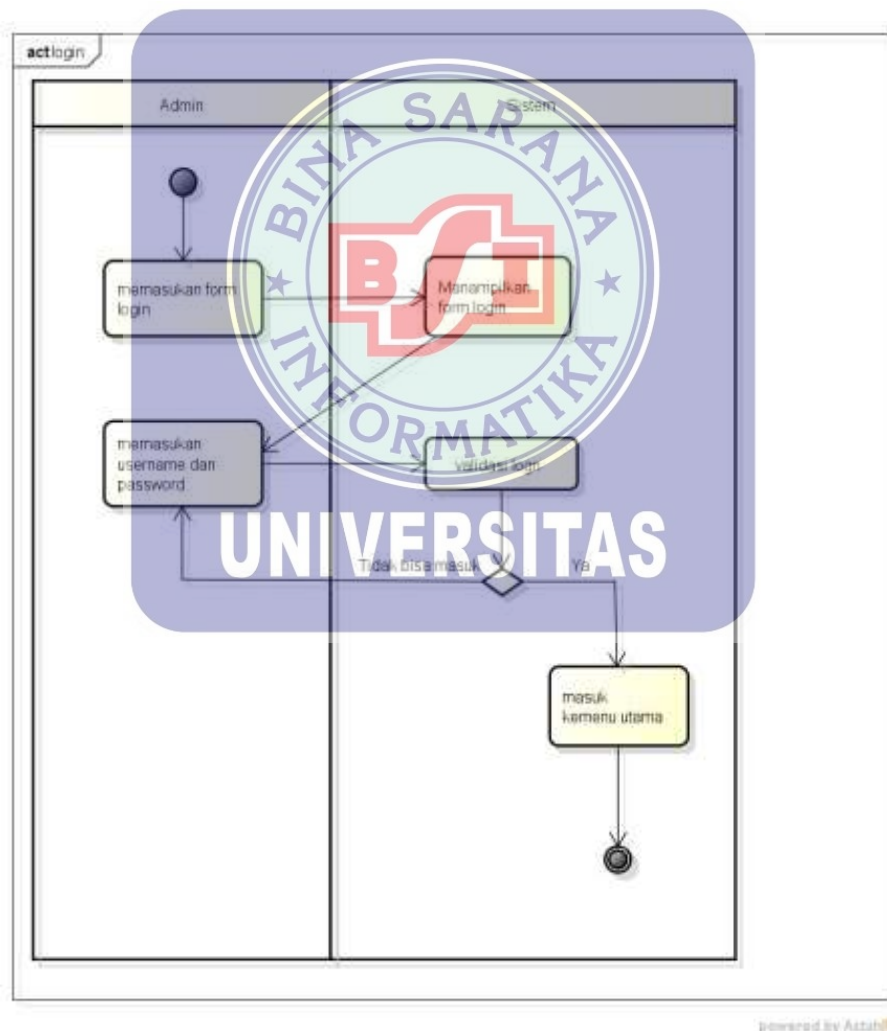
Deskripsi Use Case Diagram Sistem Informasi Arus kas

Use Case Narative Sistem Informasi Arus kas	
Tujuan	user dapat melakukan pengolahan data yang ada di menu utama ya itu data master, transaksi, jurnal dan laporan.
Deskripsi	Sistem ini memungkinkan user untuk mengelola sistem pencatatan arus kas, mulai dari input data sampai ke laporan.
Skenario Utama	
Aktor	User
Kondisi awal	User Membuka tampilan menu utama pada aplikasi
Aksi Aktor	Reaksi Sistem
1. Aktor memilih tombol master	Sistem akan menampilkan data akun, dan saldo awal.
2. Aktor memilih tombol transaksi	Sistem akan menampilkan data kas masuk, kas keluar.
3. Aktor memilih tombol jurnal	Sistem akan menampilkan input jurnal umum
4. Aktor memilih tombol cetak laporan	Sistem akan menampilkan cetak laporan jurnal, laporan arus kas masuk, dan laporan arus kas keluar
5. Aktor memilih tombol keluar	Sistem akan keluar dari menu utama.
Kondisi Akhir	Jika perintah sesuai maka sistem akan masuk kedalam aplikasi pelaporan arus kas maka aktor dapat melakukan aktivitas didalam sistem.

3.3.3. Activity Diagram

Setiap *usecase* pada *usecase diagram* ini akan diuraikan lebih lanjut sesuai dengan urutan kegiatannya menggunakan *activity diagram*. Hasil pemodelan menggunakan *activity diagram* untuk Arus Kas Desa Gempolsari Kecamatan Patokbeusi Kabupaten Subang dapat dilihat berikut ini:

1. Activity Diagram Login

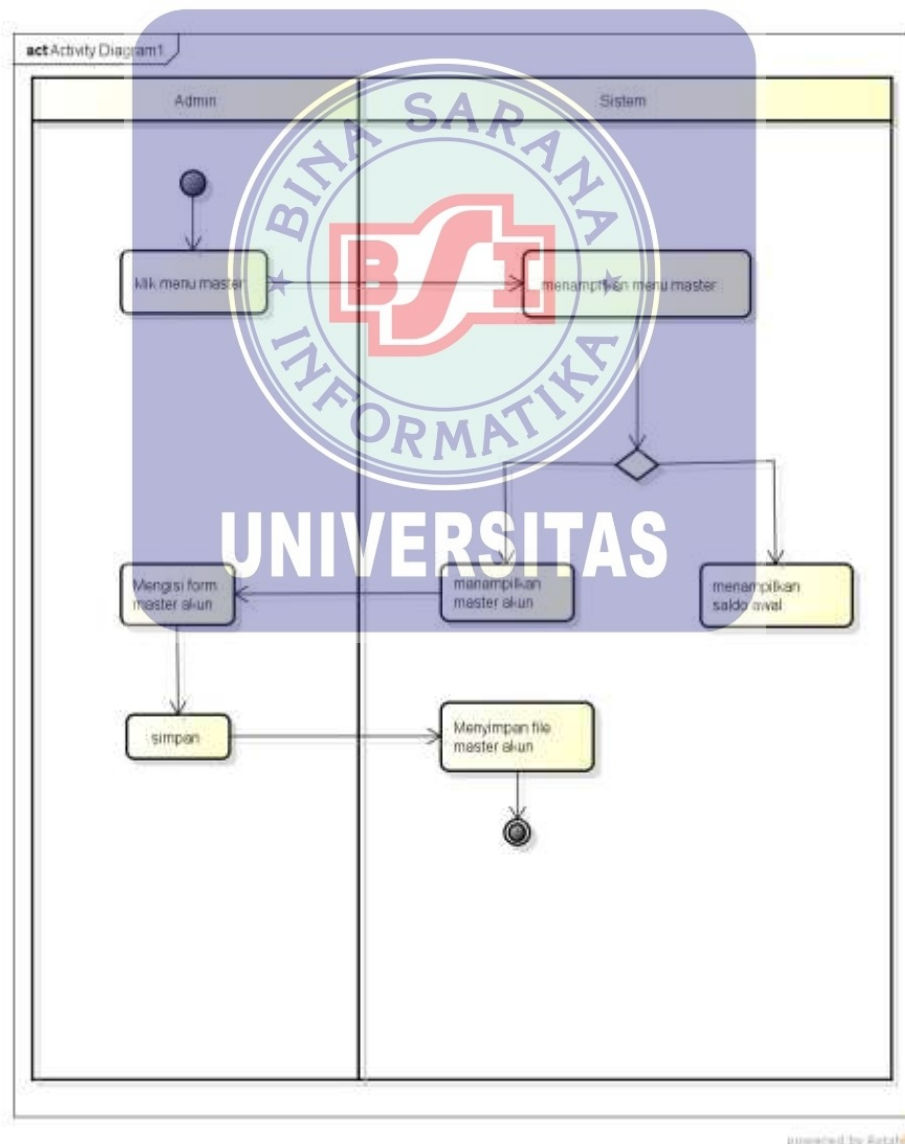


Sumber : Hasil penelitian (2019)

Gambar III.7. Activity Diagram Login

Pada Gambar III.7 dapat di jelaskan bahwa sebelum masuk ke menu utama pada rancangan sistem arus kas Desa Gempolsari pengguna harus melakukan *login* terlebih dahulu. Setelah *form login* tampil, pengguna di minta untuk memasukan nama pengguna dan kata sandi lalu pilih tombol masuk kemudian sistem akan melakukan validasi *login*, jika nama pengguna dan kata sandi benar maka sistem akan menampilkan *form* menu utama, namun jika nama pengguna salah dan kata sandi maka akan di minta untuk kembali memasukan nama dan kata sandi.

2. Activity Diagram Master Akun

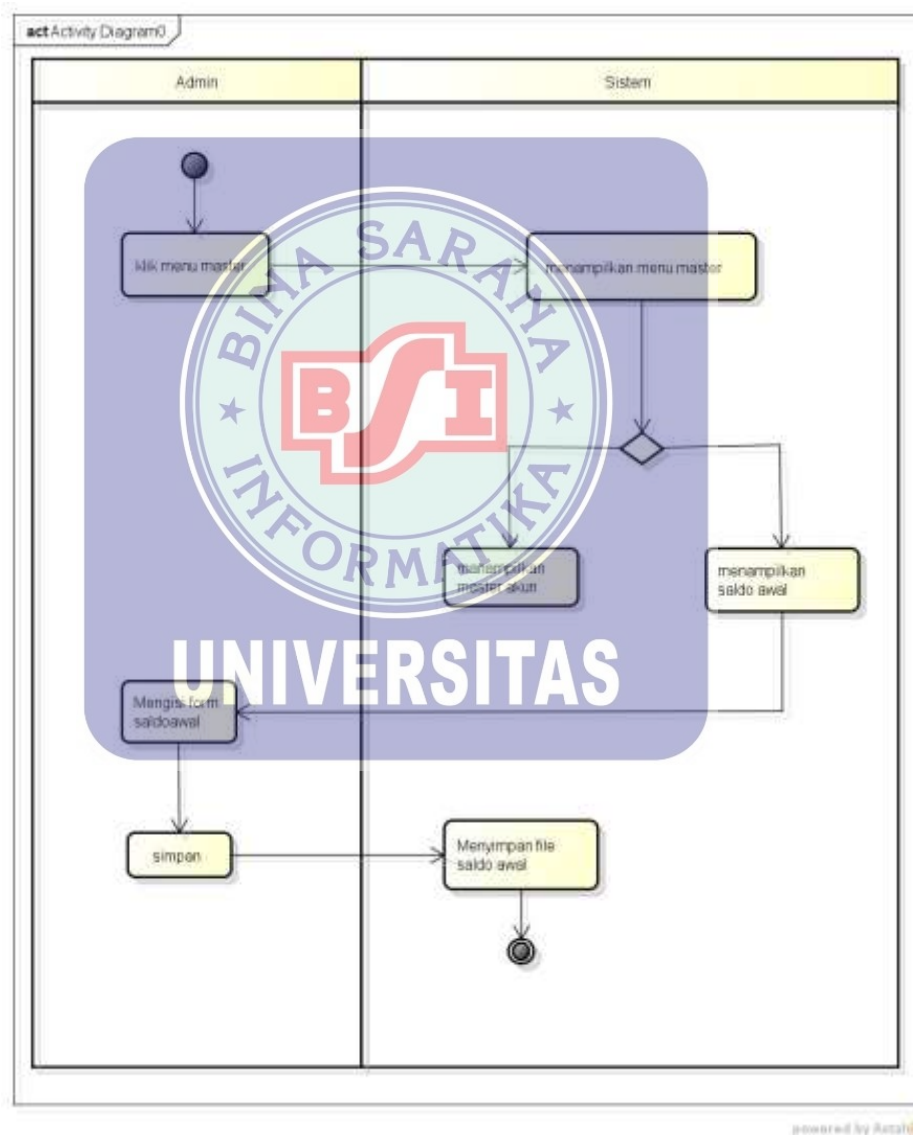


Sumber : Hasil penelitian (2019)

Gambar III.8. Activity Diagram Master Akun

Pada gambar III.8. dapat dijelaskan apabila pengguna ingin input data, maka pengguna harus memilih menu master kemudian akan menampilkan submenu master akun setelah itu mengisi form master akun lalu menekan tombol simpan dan sistem akan memproses data akun.

3. Activity Diagram Saldo Awal

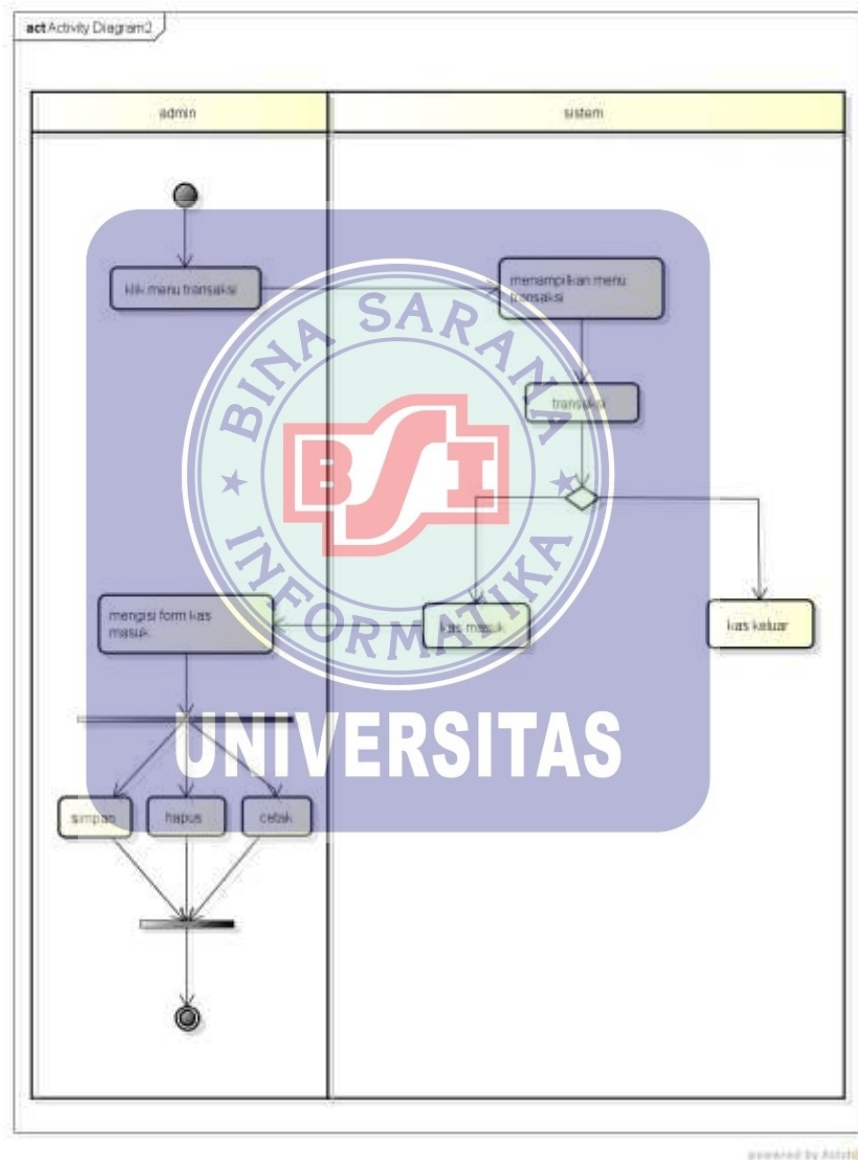


Sumber : Hasil penelitian (2019)

Gambar III.9. Activity Diagram Master Akun

Pada gambar III.9. dapat dijelaskan apabila pengguna ingin input data, maka pengguna harus memilih menu master kemudian akan menampilkan submenu saldo awal setelah itu mengisi form saldo awal lalu menekan tombol simpan dan sistem akan memproses saldo awal.

4. Activity Diagram Kas Masuk

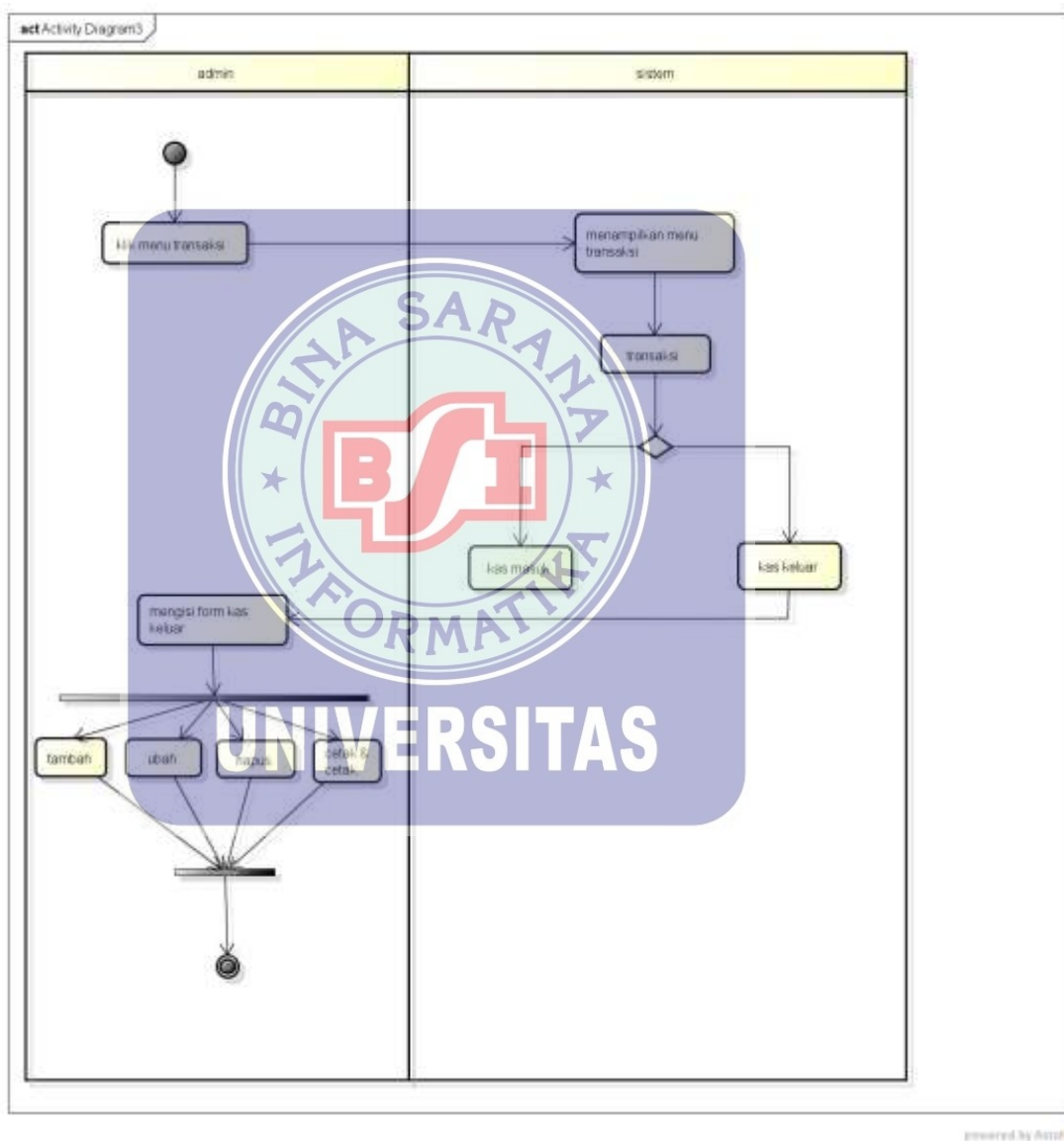


Sumber : Hasil penelitian (2019)

Gambar III.10. Activity Diagram Kas Masuk

Pada Gambar III.10 sistem informasi kas masuk akan memilih menu transaksi dan submenu kas masuk, lalu sistem akan menampilkan *form* kas masuk. Pengguna dapat memilih tombol simpan, hapus, dan cetak maka sistem akan memproses data kas masuk.

5. Activity Diagram Kas Keluar



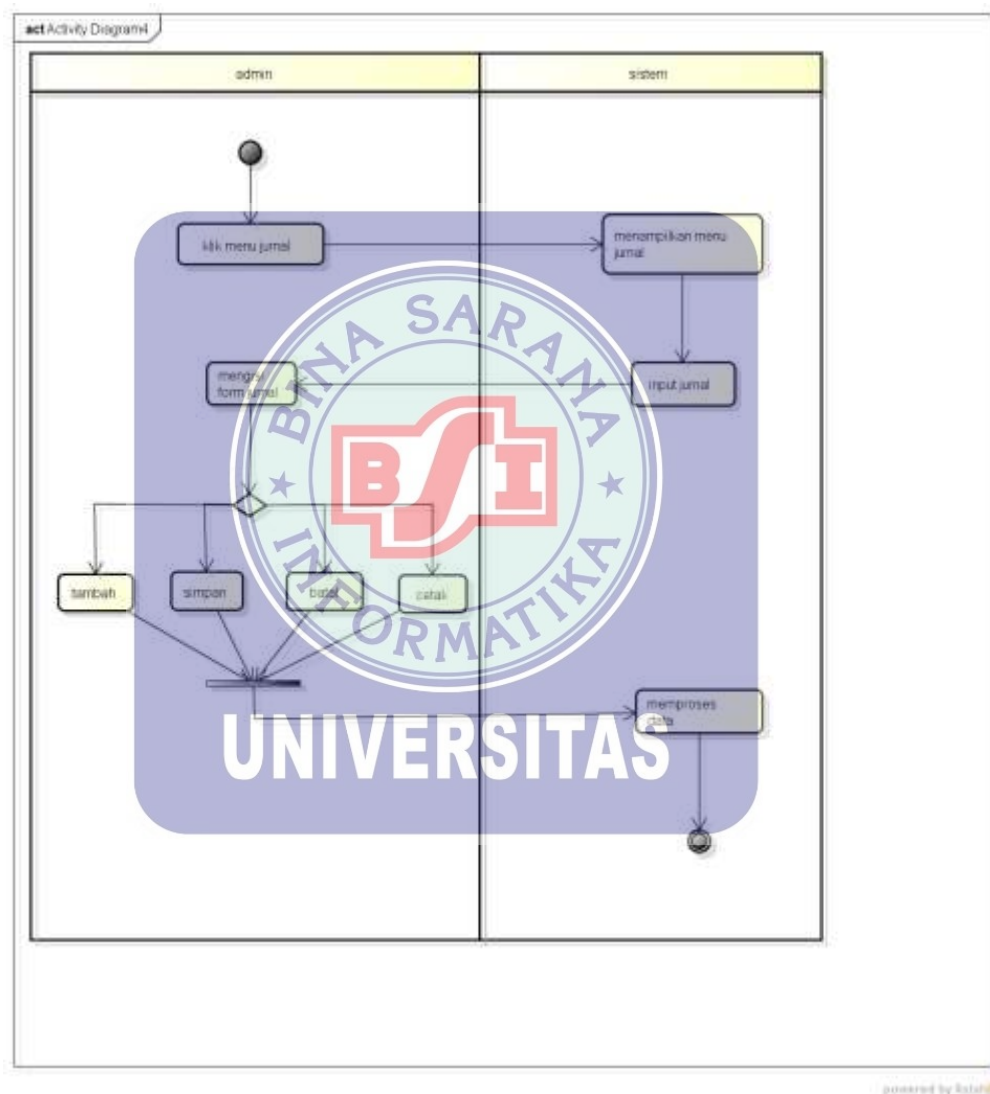
Sumber : Hasil penelitian (2019)

Gambar III.11. Activity Diagram Kas Keluar

Pada Gambar III.11 sistem informasi kas keluar akan memilih menu transaksi dan submenu kas keluar, lalu sistem akan menampilkan *form* kas keluar. Pengguna dapat

memilih tombol tambah, ubah, hapus, simpan dan cetak maka sistem akan memproses data kas keluar.

6. Activity Diagram Jurnal



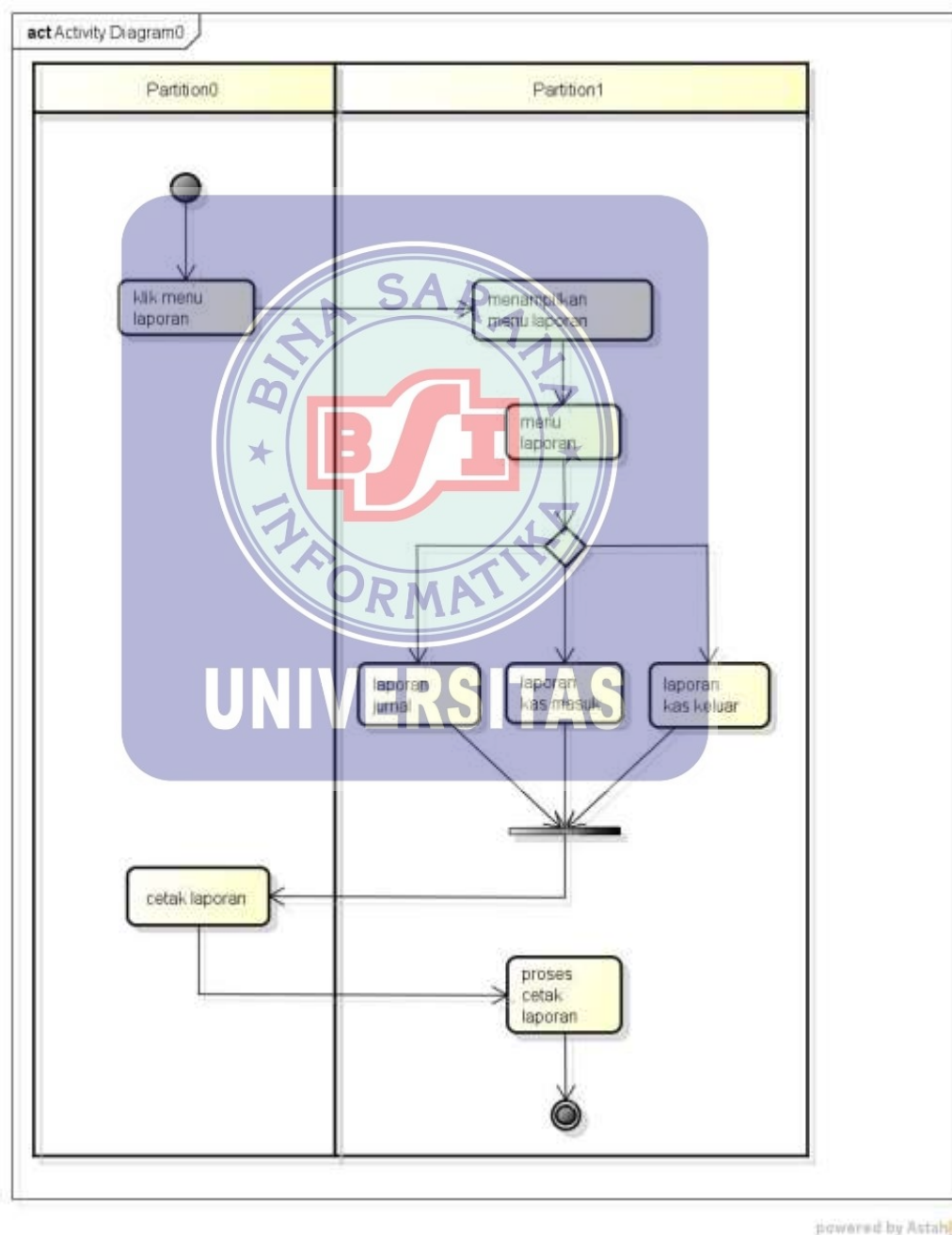
Sumber : Hasil penelitian (2019)

Gambar III.12. Activity Diagram Jurnal

Pada Gambar III.12 dapat dijelaskan apabila pengguna ingin menginput jurnal umum, pengguna dapat memilih menu jurnal lalu akan menampilkan submenu input jurnal, pengguna dapat memilih menu jurnal lalu akan menampilkan submenu input jurnal,

pengguna dapat mengisi form jurnal lalu memilih tombol tambah, simpan, batal, dan cetak setelah itu sistem akan memproses data jurnal.

7. Activity Diagram Laporan



Sumber : Hasil penelitian (2019)

Gambar III.13. Activity Diagram cetak laporan

Pada gambar III.13 pengguna dapat mencetak laporan yang diinginkan. Pengguna dapat mengklik menu cetak laporan lalu akan menampilkan submenu cetak laporan jurnal, cetak laporan kas masuk, cetak laporan kas keluar dan sistem akan memproses cetak laporan.

3.4. Desain

Desain adalah suatu sistem yang berlaku untuk segala jenis perancangan dimanatitik beratnya adalah melihat segala sesuatu persoalan tidak secara terpisah atau tersendiri, melainkan sebagai suatu kesatuan dimana satu masalah dengan lainnya saling terkait.

3.4.1. Entity Relationship Diagram (ERD)

ERD adalah merupakan salah satu model yang digunakan untuk mendesain database dengan tujuan menggambarkan data yang berelasi pada sebuah database.

Berikut adalah rancangan pelaporan arus kas pada Kantor Desa Gempolsari :



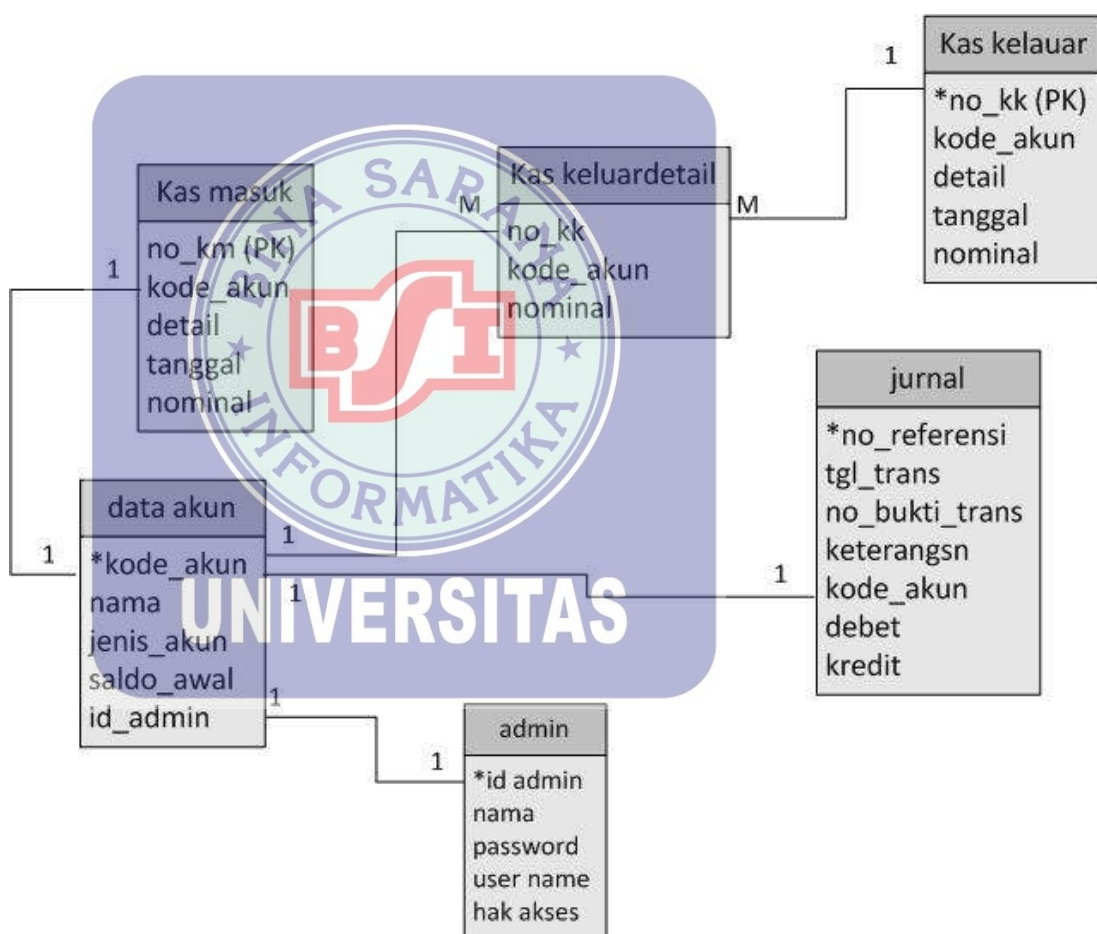
Sumber : Hasil penelitian (2019)

Gambar III.14. ERD

Pada gambar diatas menjelaskan proses desain dari pelaporan arus kas Desa Gempolsari.

4.4.2. Logical Record Structure (LRS)

LRS adalah representasi dari struktur record-record pada tabel yang terbentuk dari hasil antar himpunan entitas. Berikut adalah rancangan pelaporan arus kas pada Kantor Desa Gempolsari dengan menggunakan LRS.



Sumber : Hasil penelitian (2019)

Gambar III.15. LRS

pada gambar diatas menjelaskan hubungan antar entitas yang ada pada pelaporan arus kas Desa Gempolsari.

3.4.3. Spesifikasi File

Adapun spesifikasi *file* yang penulis gunakan dalam pelaporan arus kas Desa Gempolsari adalah sebagai berikut :

1. Spesifikasi File Data Akun

Nama	: Tabel Akun
Akronim	: tb_akun.sql
Fungsi	: Untuk mengolah data akun
Tipe File	: File Master
Organisasi File	: Index Sequential
Akses File	: Random
Record Size	: 85
Kunci Field	: Kode_Akun
Media File	: Harddisk
Software	: MySQL

Tabel III.3.

Tabel Data Master Akun

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Akun	kode_akun	Varchar	20	Primary key
2	Nama Akun	Nama_akun	Varchar	30	
3	Jenis Akun	Jenis_akun	Varchar	15	
4	Saldo Normal	Saldo_normal	Varchar	20	

2. Spesifikasi File Data Saldo Awal

Nama : Tabel Data Saldo Awal
 Akronim : tb_saldo_awal.sql
 Fungsi : Untuk mengolah data saldo awal
 Tipe File : *File Master*
 Organisasi File : Index Sequential
 Akses File : *Random*
 Record Size : 37
 Kunci Field : id
 Media File : Harddisk
 Software : MySQL

Tabel III.4.
Tabel Data Saldo Awal

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Nama Akun	Id	int	11	Primary key
2	Kode Akun	kode_akun	Varchar	15	
3	Nominal	Saldo_debet	int	11	

3. Spesifikasi *File* Data jurnal

Nama : Tabel Data jurnal
 Akronim : tb_jurnal.sql
 Fungsi : Untuk mengolah data jurnal
 Tipe File : *File Master*
 Organisasi File : Index Sequential
 Akses File : *Random*

Record Size : 126
Kunci Field : no_jurnal
Media File : Harddisk
Software : MySQL

Tabel III.5.

Tabel Data jurnal

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	No Jurnal	No_jurnal	Varchar	20	Primary key
2	Kode Akun	kode_akun	Varchar	14	
3	Tanggal	Tagnggal	Varchar	14	
4	Debit	Debit	Int	14	
5	Kredit	Kredit	Int	14	
6	Keterangan	keterangan	varchar	50	

4. Spesifikasi *File* Data Kas Masuk

Nama : Tabel data kas masuk
 Akronim : tb_kas_masuk.sql
 Fungsi : Untuk mengolah data kas masuk
 Tipe File : *File Master*
 Organisasi File : Index Sequential
 Akses File : *Random*
 Record Size : 121
 Kunci Field : no_km
 Media File : Harddisk

Software : MySQL

Tabel III.6.

Tabel Data Kas Masuk

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	No Kas Masuk	No_km	Varchar	20	Primary key
2	Kode Akun	kode_akun	Varchar	20	
3	Nominal	nominal	int	11	
4	Tanggal	tgl_km	Varchar	20	
5	Keterangan	keterangan	Varchar	50	

5. Spesifikasi File Data Kas Keluar

Nama : Tabel data kas keluar

Akronim : tb_kas_keluar.sql

Fungsi : Untuk mengolah data kas keluar

Tipe File : File Master

Organisasi File : Index Sequential

Akses File : Random

Record Size : 30

Kunci Field : no_kk

Media File : Harddisk

Software : MySQL

Tabel III.7.

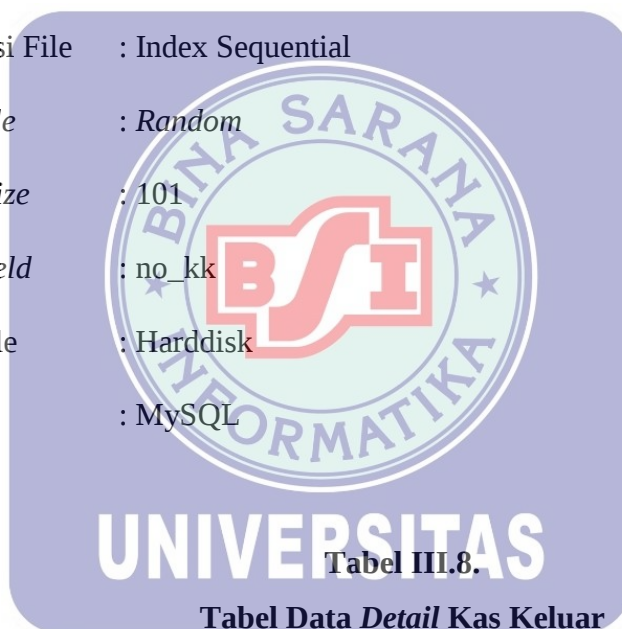
Tabel Data Kas Keluar

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
----	-------------	---------	------	---------	------------

1	No Kas Keluar	no_kk	Varchar	10	Primary key
2	Tanggal	Tgl_date	Varchar	20	

6. Spesifikasi File Data Detail Kas Keluar

Nama : Tabel Data Detail Kas Keluar
 Akronim : tb_detail_kas_keluar.sql
 Fungsi : Untuk mengolah data detail kas keluar
 Tipe File : File Master
 Organisasi File : Index Sequential
 Akses File : Random
 Record Size : 101
 Kunci Field : no_kk
 Media File : Harddisk
 Software : MySQL



No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	No Kas Keluar	no_kk	Varchar	20	Primary key
2	Kode Akun	Kode_akun	Varchar	20	
3	Nominal	Nominal	Int	11	
4	Keterangan	keterangan	Varchar	50	

7. Spesifikasi File login

Nama : Tabel Login

Akronim	: tb_login.sql
Fungsi	: Untuk menyimpan data pengguna
Tipe File	: <i>File Master</i>
Organisasi File	: Index Sequential
Akses File	: <i>Random</i>
Record Size	: 26
Kunci Field	: id_admin
Media File	: Harddisk
Software	: MySQL

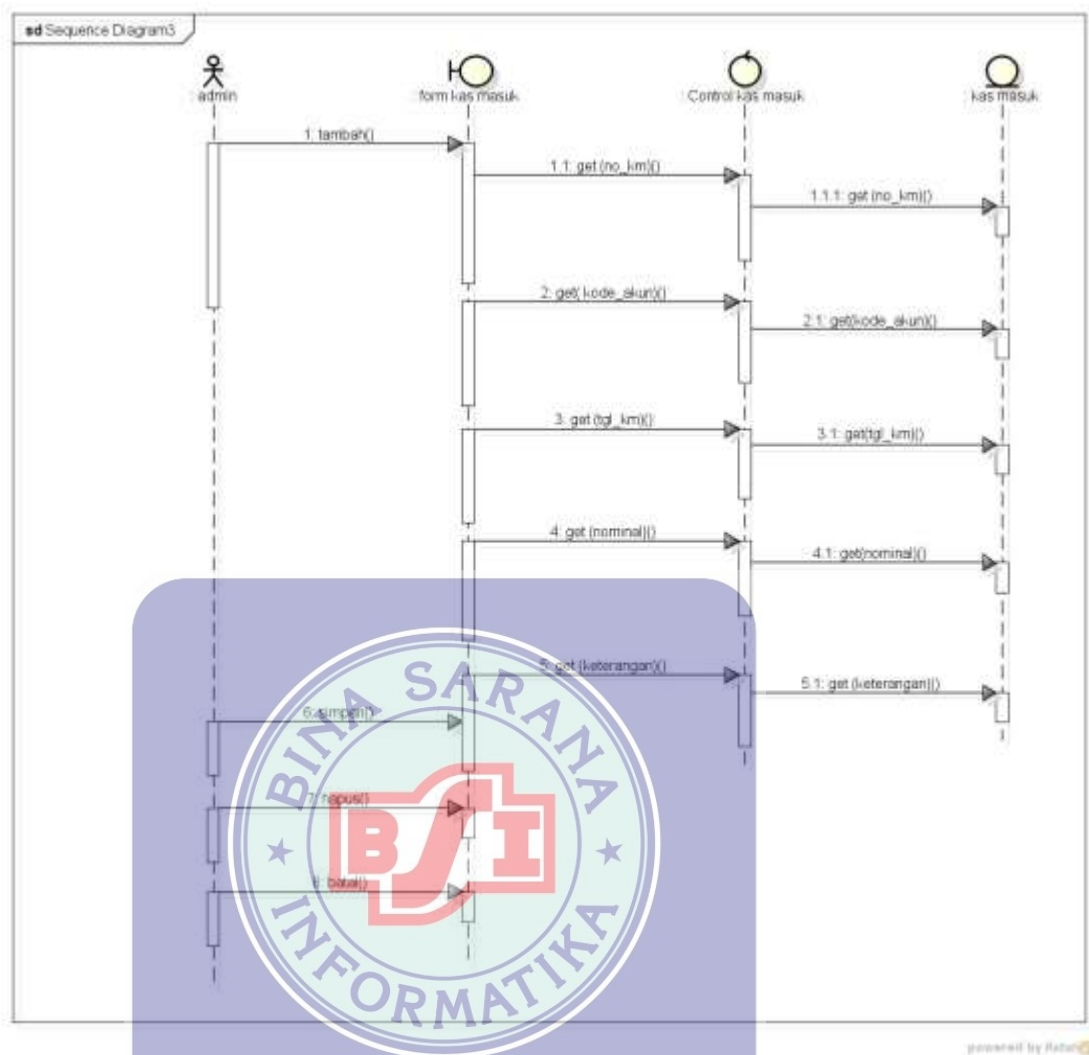
Tabel III.9:
Tabel Data User

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Username	id_admin	int	11	<i>Primary key</i>
2	Password	password	Varchar	15	

3.4.4. Sequence Diagram

Berikut ini adalah hasil urutan kegiatan menggunakan *sequence diagram* untuk perancangan sistem informasi akuntansi arus kas pada Desa Gempolsari.

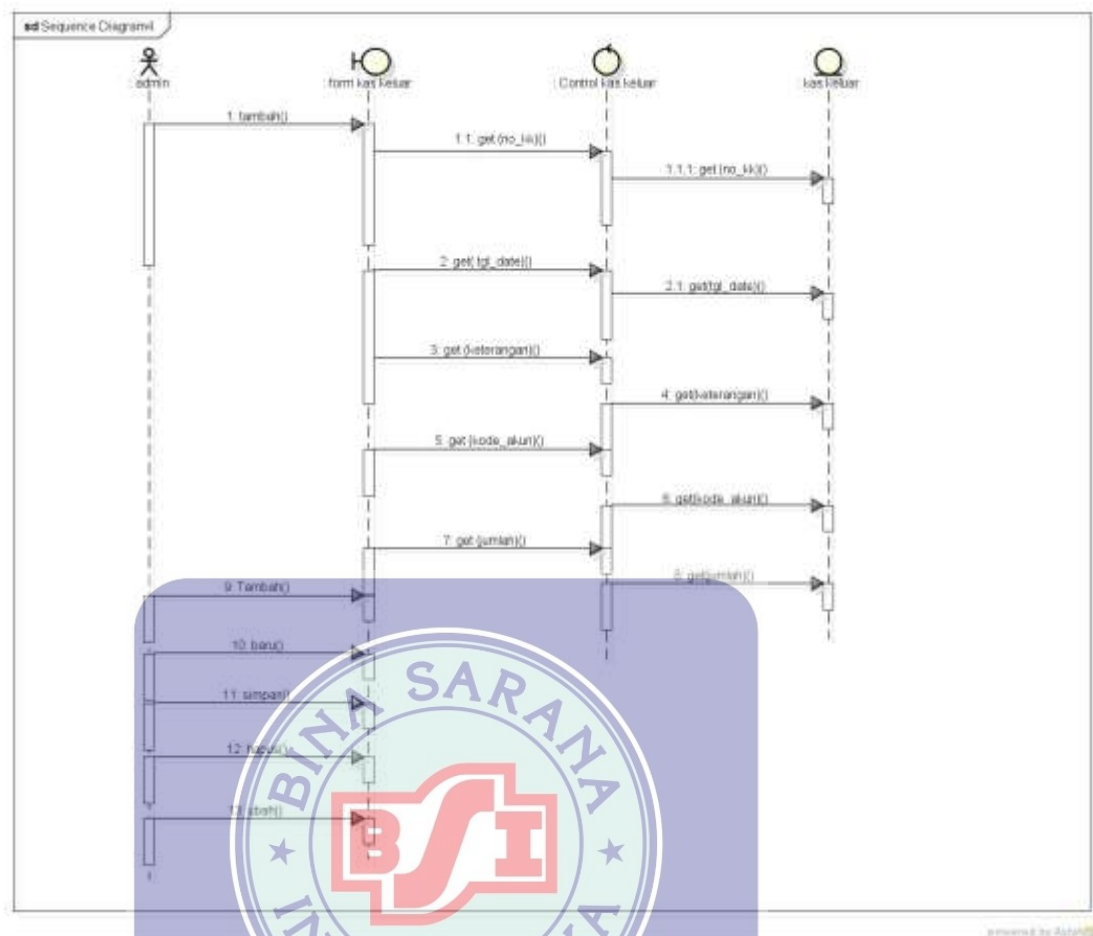
1. *Sequence Diagram* Kas masuk



Sumber : Hasil penelitian (2019)

Gambar III.16. Sequence Diagram Kas Masuk

2. Sequence Diagram Kas Keluar

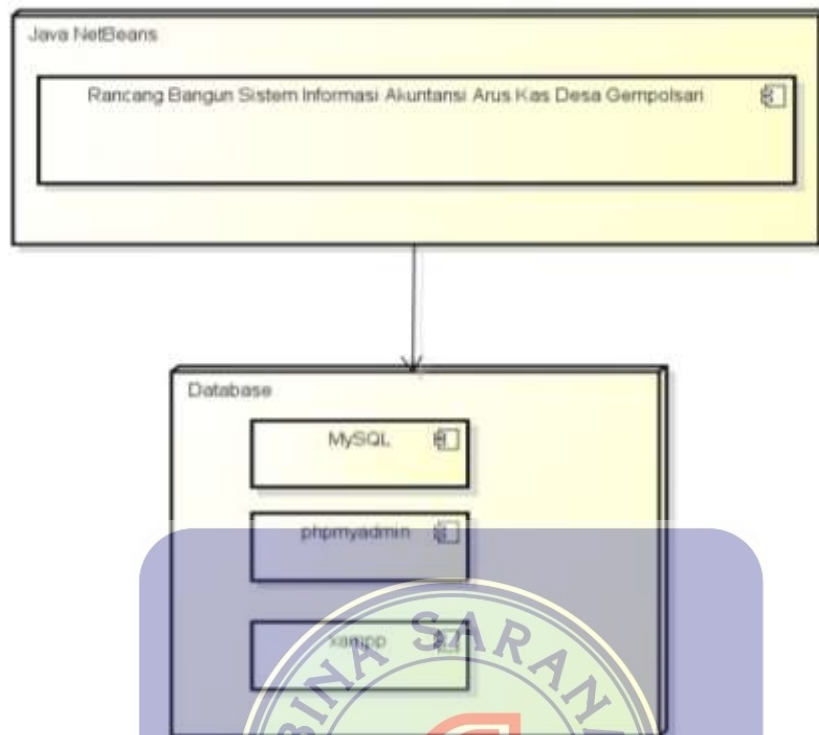


Sumber : Hasil penelitian (2019)

Gambar III.17. Sequence Diagram Kas Keluar

3.4.5. Deployment Diagram

Untuk menjalankan perancangan sistem informasi akuntansi laporan arus kas pada Desa Gempolsari ini memerlukan konfigurasi sistem komputer. Berikut ini adalah hasil penggambaran konfigurasi menggunakan deployment diagram.



Sumber : Hasil penelitian (2019)

Gambar III.18. Deployment Diagram

3.4.6. Use Interface

Setelah melakukan serangkaian kegiatan perancangan, sampai pada waktunya untuk merancang tampilan antar muka untuk sistem informasi akuntansi laporan arus kas pada Kantor Desa Gempolsari Kecamatan Patokbeusi Kabupaten Subang. Berikut ini adalah hasil rancangan tampilan antar muka atau user interface.

1. Menu Login



Sumber : Hasil penelitian (2019)

Gambar III.19. Rancangan Tampilan Login

2. Menu Utama



Sumber : Hasil penelitian (2019)

Gambar III.20. Rancangan Tampilan Menu Utama

3. Master Akun

Master Akun

Baru Hapus

Kode Akun	Nama	Jenis Akun	Saldo Normal
AK0001	Anggran Dana D...	Debet	125000000

Kode Akun

Nama Akun

Jenis Akun

Saldo Normal

Simpan

Sumber : Hasil penelitian (2019)

Gambar III.21. Rancangan Tampilan Master Akun

4. Saldo Awal

SALDO AWAL

Hapus

Kode Akun	Saldo Debet
AK0001	125000000

Kode Akun Cari

Nama Akun

Nominal

Simpan

Sumber : Hasil penelitian (2019)

Gambar III.22. Rancangan Tampilan Saldo Awal

5. Kas Masuk

No Kas Masuk

Kode Akun

Tanggal

Nominal

Keterangan

Baru Simpan Hapus

NO Kas Masuk	tanggal	Kode Akun	Nominal	Keterangan
--------------	---------	-----------	---------	------------

Sumber : Hasil penelitian (2019)

Gambar III.23. Rancangan Tampilan Kas Masuk

UNIVERSITAS

6. Kas Keluar

No Kas Keluar: Tanggal: 2019/07/08

Keterangan:

Kode Akun:

Jumlah:

Buttons: Baru, Tambah, Ubah, Hapus

No Kas Keluar	Kode Akun	Jumlah	Tanggal	Keterangan

Simpan & Cetak

Total jLabel7

Sumber : Hasil penelitian (2019)

Gambar III.24. Rancangan Tampilan Kas Keluar

7. Menu Jurnal

Jurnal

Kode Akun: Tanggal Jurnal: 2019-07-08 Kredit:

No Jurnal: Debit: Keterangan:

Buttons: Tambah, Simpan, Batal

No Jurnal	Kode_akun	Tanggal	Debit	Kredit	Keterangan

Jumlah DatajLabel8

CETAK

Sumber : Hasil penelitian (2019)

Gambar III.25. Rancangan Tampilan Jurnal

3.5. Implementasi

Implementasi merupakan perealisasiian dari perancangan sistem informasi akuntansi laporan arus kas pada Kantor Desa Gempolsari abupaten Subang. Implementasi ini menjelaskan tentang code generation untuk transaksi bisnis program, pengujian menggunakan black box testing, spesifikasi hardware dan software.

3.5.1. Code Generation

Berikut ini adalah hasil code generation Pada Kantor Desa Gempolsari Kecamatan Patokbeusi Kabupatne Subang:

1. Menu transaksi kas masuk

```
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

import java.sql.Connection;
import java.sql.Statement;
import java.text.DateFormat;
```

```

import java.util.Calendar;

import java.util.Locale;

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author User
 */
public class kas_in extends javax.swing.JFrame {

    String kode,penerima,keterangan;

    java.util.Date tglsekarang = new java.util.Date();

    private final SimpleDateFormat smpdtfmt = new
SimpleDateFormat("dd/MM/yyyy", Locale.getDefault());

    //diatas adalah pengaturan format penulisan, bisa diubah sesuai keinginan.

    private final String tanggal = smpdtfmt.format(tglsekarang);

    int jml;

    Date tgl;

    ResultSet resultSet;

    Statement statement;

    String kode_akun, ubah;

```

```

int count, jml_awal;

double jml_total;

koneksi conn;

DefaultTableModel tb;

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

/**
 * Creates new form Data_Anggaran
 */

alurkas.koneksi1 konek = new alurkas.koneksi1();

public kas_in() {
    initComponents();
    conn=new koneksi();
    setLocationRelativeTo(this);
    tabelmodel();

    DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

    Calendar cal = Calendar.getInstance();

    DTtgl.setText(dateFormat.format(cal.getTime()));

    txtkode.setEditable(false);

    setform(false);

    cekstok();

    SelectKategori();

    txtkode.hide();
}

```

```

private void SelectKategori(){
    try {
        Connection conn = konek.openkoneksi();

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

        java.sql.ResultSet rs = stm.executeQuery("SELECT * FROM
master_akun WHERE saldo_normal='Debet'");

        cmbkode.addItem("Pilih");

        while(rs.next()){
            cmbkode.addItem(rs.getString("nama_akun"));
        }
        konek.closekoneksi();
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error " + e);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(kas_in.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

private int cekstok(){
    int stok = 0;

    String row_kode = txtkode.getText();

    String total ="SELECT * FROM saldo_awal WHERE
kode_akun='"+row_kode+"'";

```

```

try {

    Connection conn = konek.openkoneksi();

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

    resultSet = stm.executeQuery(total);

    while(resultSet.next()){

        stok = Integer.parseInt(resultSet.getString("saldo_debet"));

    }

} catch (Exception ex) {

    JOptionPane.showMessageDialog(null, ex.getMessage());

    System.out.println("'" + ex.getMessage());

}

return stok;

}

public void Updatesaldo(String ubah){

    int stokbaru = 0;

    int total;

    String row_kode = txtkode.getText();

    String row_txtnominal = txtnominal.getText();

    stokbaru = cekstok() + Integer.parseInt(txtnominal.getText());

    String sql="UPDATE saldo_awal SET saldo_debet=" + stokbaru + "

where kode_akun = " + row_kode + """;

```

```

        conn.simpanData(sql);
    }

    /**
     * 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() {

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        txtnominal = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();

        jLabel2 = new javax.swing.JLabel();

        jScrollPane2 = new javax.swing.JScrollPane();
        txtket = new javax.swing.JTextArea();

        cmbkode = new javax.swing.JComboBox<>();
        txtkode = new javax.swing.JTextField();

        jLabel4 = new javax.swing.JLabel();
        txtno = new javax.swing.JTextField();
        DTtgl = new javax.swing.JTextField();

```

```

btnhapus = new javax.swing.JButton();

btnsimpan = new javax.swing.JButton();

btnbaru = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

tblanggaran          =          new          javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

```

```

jPanel1.setBackground(new java.awt.Color(255, 204, 204));

jPanel1.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel1.setText("Kode Akun");
jLabel5.setText("Keterangan");
jLabel3.setText("Nominal");
jLabel2.setText("Tanggal");
txtket.setColumns(20);
txtket.setRows(5);

jScrollPane2.setViewportView(txtket);

cmbkode.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbkodeItemStateChanged(evt);
    }
});

cmbkode.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        cmbkodeActionPerformed(evt);
    }
});

txtkode.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtkodeActionPerformed(evt);
    }
});

jLabel4.setText("No Kas Masuk");
DTtgl.setEditable(false);
DTtgl.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        DTtglActionPerformed(evt);
    }
});

btnhapus.setText("Hapus");

btnhapus.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnhapusActionPerformed(evt);
    }
});

btnsimpan.setText("Simpan");

```

```

btnsimpan.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnsimpanActionPerformed(evt);
    }
});

```

```

btnbaru.setText("Baru");

```

```

btnbaru.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnbaruActionPerformed(evt);
    }
});

```

```

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(

```

```

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

.addComponent(jLabel1)

.addComponent(jLabel2)

.addComponent(jLabel3)

.addComponent(jLabel5)

.addComponent(jLabel4))

.addGap(57, 57, 57)

```

```

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)

```

```

.addGroup(jPanel1Layout.createSequentialGroup())

```

```

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)

```

```

.addGroup(jPanel1Layout.createSequentialGroup())

```

```

.addComponent(btnbaru)

```

```

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

```

```

.addComponent(btnsimpan)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELA
TED)

```

```

.addComponent(btnhapus))

```

```

        .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE,                238,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(txtnominal,
javax.swing.GroupLayout.PREFERRED_SIZE,                218,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

.addGroup(jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.TRAILING, false)

.addComponent(DTtgl,
javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(txtno,
javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(cmbkode,
javax.swing.GroupLayout.Alignment.LEADING,            0,            150,
Short.MAX_VALUE))

.addGap(18, 18, 18)

.addComponent(txtkode,
javax.swing.GroupLayout.PREFERRED_SIZE,                58,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

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

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,

jPanel1Layout.createSequentialGroup()

.addGap(26, 26, 26)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel4)

.addComponent(txtno,

javax.swing.GroupLayout.PREFERRED_SIZE,

javax.swing.GroupLayout.DEFAULT_SIZE,

javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

```

```

        .addComponent(cmbkode,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtkode))

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

```

```

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Ali
        gnment.BASELINE)
        .addComponent(jLabel2)
        .addComponent(DTtgl,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(9, 9, 9)

```

```

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Ali
        gnment.LEADING)
        .addComponent(jLabel3)
        .addComponent(txtnominal,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))

```

```
.addGap(15, 15, 15)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Ali  
gnment.LEADING)
```

```
.addComponent(jLabel5)
```

```
.addComponent(jScrollPane2,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE,
```

72,

```
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addGap(18, 18, 18)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Ali  
gnment.BASELINE)
```

```
.addComponent(btnbaru)
```

```
.addComponent(btnsimpan)
```

```
.addComponent(btnhapus))
```

```
.addContainerGap())
```

```
);
```

```
tblanggaran.setModel(new javax.swing.table.DefaultTableModel(  
new Object [][] {
```

```
new Object [][] {
```

```
{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"
}

));

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

jScrollPane1.setViewportView(tblanggaran);

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(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addGap()
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 477, Short.MAX_VALUE)
        .addGap())

```

```

);

layout.setVerticalGroup(

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

    .addGroup(layout.createSequentialGroup()

        .addComponent(jPanel1,

javax.swing.GroupLayout.PREFERRED_SIZE,

javax.swing.GroupLayout.DEFAULT_SIZE,

javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(18, 18, 18)

        .addComponent(jScrollPane1,

javax.swing.GroupLayout.PREFERRED_SIZE, 150,

javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,

Short.MAX_VALUE))

    );

pack();

}

// </editor-fold>

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

    // TODO add your handling code here:

    if (txtket.getText().equals("") || txtnominal.getText().equals("")) {

```

```

        JOptionPane.showMessageDialog(this,"Semuah Data Harus Di Isi");
    }

    else{

        Updatesaldo(ubah);

        simpan();

        tabelmodel();

        bersih();
    }
}

private void btnhapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    hapus(txtno.getText());
    tabelmodel();
    setform(true);
}

private void tblanggaranMouseClicked(java.awt.event.MouseEvent evt) {
    int row = tblanggaran.getSelectedRow();

    String row_id = (tblanggaran.getModel().getValueAt(row, 0).toString());

    txtno.setText(row_id);

    btnhapus.setEnabled(true);
}

```

```
}
```

```
private void btnbaruActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    if (btnbaru.getText().equals("Batal")) {
```

```
        setform(false);
```

```
        bersih();
```

```
    }
```

```
    else{
```

```
        setform(true);
```

```
        setkode();
```

```
    }
```

```
}
```

```
private void cmbkodeItemStateChanged(java.awt.event.ItemEvent evt) {
```

```
    String nm_kategori = cmbkode.getSelectedItem().toString();
```

```
    if(!nm_kategori.equals("")){
```

```
        try {
```

```
            Connection conn = konek.openkoneksi();
```

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

```
            java.sql.ResultSet sql = stm.executeQuery("SELECT kode_akun  
FROM master_akun WHERE nama_akun='"+nm_kategori+"'");
```

```
            if(sql.next()){
```

```
                txtkode.setText(sql.getString("kode_akun"));
```

```
            }
```

```

        konek.closekoneksi();

    } catch (SQLException e) {

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

    } catch (ClassNotFoundException ex) {

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

    }

}

}else{

    txtkode.setText("");

    } // TODO add your handling code here:

}

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

    // TODO add your handling code here:

}

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

    // TODO add your handling code here:

}

private void DTtglActionPerformed(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(kas_in.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

```

```

java.util.logging.Logger.getLogger(kas_in.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

```

```

java.util.logging.Logger.getLogger(kas_in.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```

java.util.logging.Logger.getLogger(kas_in.class.getName()).log(java.util.logg
ing.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 kas_in().setVisible(true);

```

```

    }

```

```

});

```

```

}

```

```

// Variables declaration - do not modify

```

```

private javax.swing.JTextField DTtgl;

private javax.swing.JButton btnbaru;

private javax.swing.JButton btnhapus;

private javax.swing.JButton btnsimpan;

private javax.swing.JComboBox<String> cmbkode;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JTable tblanggaran;

private javax.swing.JTextArea txtket;

private javax.swing.JTextField txtkode;

private javax.swing.JTextField txtno;

private javax.swing.JTextField txtnominal;

// End of variables declaration

```

```

public int simpan(){

    String sql="insert into kas_in "+

    "values"+

    "("+txtno.getText()+", " +

    """+DTtgl.getText()+", "+

```

```

        ""+txtkode.getText()+"", "+
        ""+txtnominal.getText()+"", "+
        ""+txtket.getText()+" " + ")";

    conn.simpanData(sql);

    return 0;
}

public int hapus(String kd){
    String sql="delete from kas_in where no_km = '"+kd+"'";
    conn.simpanData(sql);
    return 0;
}

public void tabelmodel(){
    tb =new DefaultTableModel();
    tb.addColumn("NO Kas Masuk");
    tb.addColumn("tanggal");
    tb.addColumn("Kode Akun");
    tb.addColumn("Nominal");
    tb.addColumn("Keterangan");

    tblanggaran.setModel(tb);

    try {
        ResultSet res = conn.ambilData("select *from kas_in ");
        while (res.next()){

```

```

        tb.addRow(new
Object[]{res.getString(1),res.getString(2),res.getString(3),res.getString(4),res.
getString(5)});
    }

    tblanggaran.setModel(tb);

    } catch (SQLException ex) {

        JOptionPane.showMessageDialog(this, ex);
    }
}

void setform(boolean flag){

    txtket.setEditable(flag);
    txtnominal.setEditable(flag);
    DTtgl.setEnabled(flag);
    btnsimpan.setEnabled(flag);
    btnhapus.setEnabled(false);

    if (flag==true) {

        btnbaru.setText("Batal");

    }

    else{

        btnbaru.setText("Baru");

    }

}

void bersih(){

    txtkode.setText("");

```

```

txtket.setText("");

txtnominal.setText("");

DTtgl.setText("");
}

void setkode(){

    try {

        ResultSet res = conn.ambilData("select * from kas_in order by no_km
desc");

        if (res.next()) {

            String nofak = res.getString("no_km").substring(2);

            String AN = "" + (Integer.parseInt(nofak) + 1);

            String Nol = "";

            if(AN.length()==1)
            {Nol = "000";}

            else if(AN.length()==2)

            {Nol = "00";}

            else if(AN.length()==3)

            {Nol = "0";}

            else if(AN.length()==4)

            {Nol = "";}

            txtno.setText("KM" + Nol + AN);

```

```

        } else {

            txtno.setText("KM0001");

        }

    }catch(Exception e){

        JOptionPane.showMessageDialog(null, e);

    }

}

}

```

2. Menu transsaksi kas keluar

```

import com.mysql.jdbc.Statement;
import java.io.File;
import java.sql.Connection;
import java.util.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;


import java.util.List;

import java.util.Map;

import java.util.HashMap;

```

```

import java.util.Locale;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;


import net.sf.jasperreports.engine.JRException;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.engine.design.JasperDesign;

import net.sf.jasperreports.engine.xml.JRXmlLoader;

import net.sf.jasperreports.view.JasperViewer;

```

```

/*

* To change this template, choose Tools | Templates
* and open the template in the editor.

*/

/**

*

* @author User

```

```

*/

public class Data_Pengeluaran_Kas extends javax.swing.JFrame {

    java.util.Date tglsekarang = new java.util.Date();

    private final SimpleDateFormat smpdtfmt = new
SimpleDateFormat("dd/MM/yyyy", Locale.getDefault());

    //diatas adalah pengaturan format penulisan, bisa diubah sesuai keinginan.

    private final String tanggal = smpdtfmt.format(tglsekarang);

    JasperDesign jasperDesign1;

    JasperPrint jasperPrint1;

    JasperReport jasperReport1;

    Map<String, Object> param = new HashMap<String, Object>();
koneksi conn= new koneksi();
String kode,penerima,keterangan;

int jml;

Date tgl;

DefaultTableModel tb;

/**

 * Creates new form Data_Anggaran

 */

alurkas.koneksi1 konek = new alurkas.koneksi1();

public Data_Pengeluaran_Kas() {

    initComponents();

    tabelmodel();

```

```

        setLocationRelativeTo(this);

        DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd");

        Calendar cal = Calendar.getInstance();

        DTtgl.setText(dateFormat.format(cal.getTime()));

        txtkode.setEditable(false);

        setform(false);

        SelectKategori();

        txtkodee.hide();
    }

    private void SelectKategori(){
        try {
            Connection conn = konek.openkoneksi();
            java.sql.Statement stm = conn.createStatement();
            java.sql.ResultSet rs = stm.executeQuery("SELECT * FROM master_akun
            WHERE saldo_normal=Debet");

            cmbkode.addItem("Pilih");

            while(rs.next()){
                cmbkode.addItem(rs.getString("nama_akun"));
            }

            konek.closekoneksi();
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "Error " + e);
        } catch (ClassNotFoundException ex) {

```

```

        Logger.getLogger(kas_in.class.getName()).log(Level.SEVERE, null, 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() {

    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    txtkode = new javax.swing.JTextField();
    DTtgl = new javax.swing.JTextField();

    jPanel2 = new javax.swing.JPanel();
    btnbaru = new javax.swing.JButton();
    btnsimpan = new javax.swing.JButton();
    btnubah = new javax.swing.JButton();
    btnhapus = new javax.swing.JButton();

    jScrollPane1 = new javax.swing.JScrollPane();
    tblanggaran = new javax.swing.JTable();

    jPanel3 = new javax.swing.JPanel();

```

```

jLabel5 = new javax.swing.JLabel();

txtjml = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

jScrollPane2 = new javax.swing.JScrollPane();

txtket = new javax.swing.JTextArea();

jLabel6 = new javax.swing.JLabel();

cmbkode = new javax.swing.JComboBox<>();

txtkodee = new javax.swing.JTextField();

jPanel4 = new javax.swing.JPanel();

jLabel4 = new javax.swing.JLabel();

txttotal = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

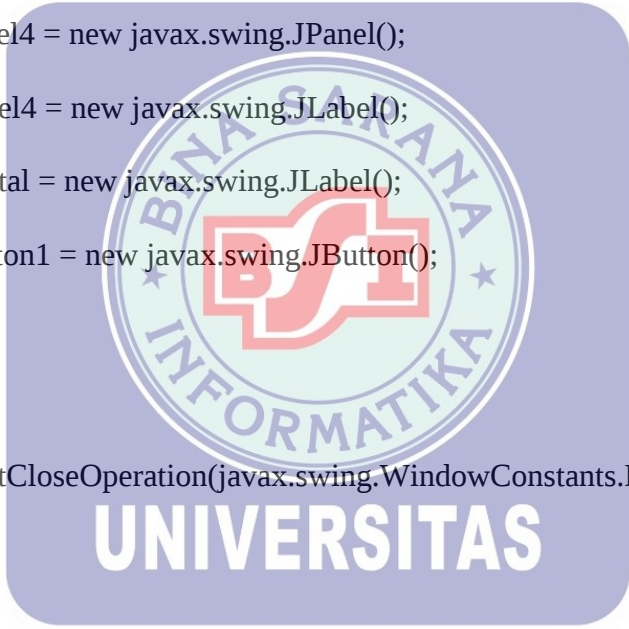
setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));

jLabel1.setText("No Kas Keluar");

jLabel2.setText("Tanggal");

```



```

DTtgl.setEditable(false);

DTtgl.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        DTtglActionPerformed(evt);

    }

});

```

```

        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(57, 57, 57)
                    .addComponent(txtkode,
                        javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel2)
                )
        );
    }

    /**
     * The method actionPerformed() is called when the user clicks on the
     * button. It calls the method actionPerformed() of the class
     * DTtglActionPerformed.
     */
    private void DTtglActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }
}

```

```

        .addGap(18, 18, 18)

        .addComponent(DTtgl)

        .addGap(255, 255, 255))

    );

    JPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel1Layout.createSequentialGroup()

            .addContainerGap()

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(jLabel1)

                .addComponent(txtkode,

                    javax.swing.GroupLayout.PREFERRED_SIZE,

                    javax.swing.GroupLayout.DEFAULT_SIZE,

                    javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(jLabel2)

                .addComponent(DTtgl,

                    javax.swing.GroupLayout.PREFERRED_SIZE,

                    javax.swing.GroupLayout.DEFAULT_SIZE,

                    javax.swing.GroupLayout.PREFERRED_SIZE))

            .addContainerGap(19, Short.MAX_VALUE))

        );

```

```
jPanel2.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.bo
rder.BevelBorder.RAISED));
```

```
btnbaru.setText("Baru");
```

```
btnbaru.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        btnbaruActionPerformed(evt);
```

```
    }
```

```
});
```

```
btnsimpan.setText("Tambah");
```

```
btnsimpan.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        btnsimpanActionPerformed(evt);
```

```
    }
```

```
});
```

```
btnubah.setText("Ubah");
```

```
btnubah.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        btnubahActionPerformed(evt);
```

```
    }
```

```
});
```

```

btnhapus.setText("Hapus");

btnhapus.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        btnhapusActionPerformed(evt);

    }

});

```

```

        javax.swing.GroupLayout jPanel2Layout = new
        javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(

            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addGroup(jPanel2Layout.createSequentialGroup()

                    .add(jPanel2Layout.createParallelGroup(

                        javax.swing.GroupLayout.Alignment.LEADING)

                            .addComponent(btnbaru)

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

                            .addComponent(btnsimpan)

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

                            .addComponent(btnubah)

```

```

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

.addComponent(btnhapus)

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

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(btnbaru)

.addComponent(btnsimpan)

.addComponent(btnubah)

.addComponent(btnhapus))

.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}

    },

    new String [] {

        "Title 1", "Title 2", "Title 3", "Title 4"

    }

));

tblanggaran.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        tblanggaranMouseClicked(evt);

    }

});

jScrollPane1.setViewportView(tblanggaran);

jPanel3.setBackground(new java.awt.Color(255, 204, 204));

jPanel3.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel5.setText("Keterangan");

jLabel3.setText("Jumlah");

txtket.setColumns(20);

txtket.setRows(5);

```

```
jScrollPane2.setViewportViewView(txtket);
```

```
jLabel6.setText("Kode Akun");
```

```
cmbkode.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbkodeItemStateChanged(evt);
    }
});
```

```
cmbkode.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbkodeActionPerformed(evt);
    }
});
```

```
txtkodee.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtkodeeActionPerformed(evt);
    }
});
```

```
javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
```

```
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
NG)
```

```
.addGroup(jPanel3Layout.createSequentialGroup())
```

```
.addContainerGap()
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
ent.LEADING)
```

```
.addGroup(jPanel3Layout.createSequentialGroup())
```

```
.addComponent(jLabel5)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
)
```

```
.addComponent(jScrollPane2,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE, 345,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addGroup(jPanel3Layout.createSequentialGroup())
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
ent.LEADING)
```

```
.addComponent(jLabel6)
```

```
.addComponent(jLabel3))
```

```
.addGap(18, 18, 18)
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(jPanel3Layout.createSequentialGroup()
```

```
.addComponent(txtjml,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE, 218,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE)
```

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

```
.addComponent(txtkodee,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE, 58,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addComponent(cmbkode,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE, 184,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE))))
```

```
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
```

```
Short.MAX_VALUE))
```

```
);
```

```
jPanel3Layout.setVerticalGroup(
```

```
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(jPanel3Layout.createSequentialGroup()
```

```
.addContainerGap()
```

```
.addGroup(jPanel3Layout.createParallelGroup(GroupLayout.Alignment.LEADING)
```

```
    .addComponent(jLabel5)
```

```
    .addComponent(jScrollPane2,
        javax.swing.GroupLayout.PREFERRED_SIZE, 68,
        javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
    .addGap(19, 19, 19)
```

```
.addGroup(jPanel3Layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jLabel6)
```

```
    .addComponent(cmbkode,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(13, 13, 13)
```

```
.addGroup(jPanel3Layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(txtjml,
```

```
        javax.swing.GroupLayout.PREFERRED_SIZE,
```

```
        javax.swing.GroupLayout.DEFAULT_SIZE,
```

```
        javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(jLabel3)
```

```

        .addComponent(txtkodee))

        .addGap(40, 40, 40))

    );

    jPanel4.setBackground(new java.awt.Color(0, 0, 0));

    jLabel4.setForeground(new java.awt.Color(255, 255, 0));
    jLabel4.setText("Total");

    txttotal.setForeground(new java.awt.Color(255, 255, 0));
    txttotal.setText("jLabel7");

    javax.swing.GroupLayout jPanel4Layout = new
    javax.swing.GroupLayout(jPanel4);
    jPanel4.setLayout(jPanel4Layout);
    jPanel4Layout.setHorizontalGroup(

    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel4Layout.createSequentialGroup()

            .addContainerGap()

            .addComponent(jLabel4)

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

            )

```

```

        .addComponent(txttotal)

        .addContainerGap(138, Short.MAX_VALUE))

    );

    jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    NG)

        .addGroup(jPanel4Layout.createSequentialGroup())

            .addGap(42, 42, 42)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel4)

            .addComponent(txttotal))

        .addContainerGap(44, Short.MAX_VALUE))

    );

```



```

jButton1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jButton1.setText("Simpan & Cetak");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(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)

            .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

                .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(jPanel4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

        .addComponent(jScrollPane1,
        javax.swing.GroupLayout.Alignment.TRAILING,
        javax.swing.GroupLayout.DEFAULT_SIZE, 734, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jPanel3,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jPanel2,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, 0)

```

```

        .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 113, Short.MAX_VALUE)

        .addGap(0, 0, 0)

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

        .addComponent(jPanel4,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(jButton1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

        .addGap(0, 0, 0))
);

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

```

```

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

    if (txtjml.getText().equals("") || txtket.getText().equals("") || DTtgl.getText() == null)
    {

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

    }

    else{

```

```

        tambah();
    bersih();
}

}

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

    hapus();
    bersih();
}

private void btnubahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (txtjml.getText().equals("") || txtket.getText().equals("") || DTtgl.getText() == null)
    {
        JOptionPane.showMessageDialog(this, "Semuah Data Harus Di Isi");
    }
    else{
        ubah();
    }
    bersih();
}

}

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

```

// TODO add your handling code here:

```
int baris=tblanggaran.getSelectedRow();

txtjml.setText(tblanggaran.getModel().getValueAt(baris,2).toString());

txtket.setText(tblanggaran.getModel().getValueAt(baris,1).toString());

}
```

```
private void btnbaruActionPerformed(java.awt.event.ActionEvent evt) {
if (btnbaru.getText().equals("Baru")){
    setform(true);
    bersih();
    Setkode();
    txttotal.setText("0");
    tabelmodel();
    btnbaru.setText("Batal");
}
else{
    btnbaru.setText("Baru");
    bersih();
    Setkode();
    txttotal.setText("0");
    tabelmodel();
}
}
```



```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    String row_tgl = DTtgl.getText();

    String row_kode = txtkode.getText();

    String ket = txtket.getText();

    String kodee = txtkodee.getText();

    String id, kode, kete;

    Integer no_kk = 0, jumlah, stok, not_found, empty = 0;

    DefaultTableModel model = (DefaultTableModel) tblanggaran.getModel();
    int rowCount = model.getRowCount();

    if(rowCount > 0 && !"".equals(row_tgl) && !"".equals(row_kode)){

        //----- Memasukan pada tabel transaksi lihat [trbarang_] dan
mengeluarkan id terakhir

        try {

            Connection conn = konek.openkoneksi();

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

            stm.executeUpdate("INSERT INTO kas_out (no_kk, tgl_date)
VALUES ('" + row_kode + "', '" + row_tgl + "')");

            konek.closekoneksi();

        } catch (SQLException e) {

```

```

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

    } catch (ClassNotFoundException ex) {

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

    }

```



```

for (int i = 0; i < rowCount; i++) {
    not_found = 0;
    stok = 0;
    row_kode = (tblanggaran.getModel().getValueAt(i, 0).toString());
    kodee = (tblanggaran.getModel().getValueAt(i, 1).toString());
    jumlah = Integer.parseInt((String)
tblanggaran.getModel().getValueAt(i, 2));

```

```

    kete =( tblanggaran.getModel().getValueAt(i, 4).toString());

```

```

//----- Mengurangi stok dengan data jumlah

try {

    Connection conn = konek.openkoneksi();

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

```

```

        java.sql.ResultSet sql = stm.executeQuery("SELECT saldo_debet
FROM saldo_awal WHERE kode_akun = " + kodee + "");

```

```

        sql.next();

```

```

        sql.last();

```

```

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

```

```

            stok = (sql.getInt("saldo_debet") - jumlah);

```

```

        } else {

```

```

            not_found = 1;

```

```

        }

```

```

        konek.closekoneksi();

```

```

    } catch (SQLException e) {

```

```

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

```

```

    } catch (ClassNotFoundException ex) {

```

```

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

```

```

    }

```

```

        if(not_found == 0){

```

```

            //----- Mengupdate jumlah stok barang

```

```

            try {

```

```

                Connection conn = konek.openkoneksi();

```

```

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

```

```

        stm.executeUpdate("UPDATE saldo_awal SET saldo_debet=" +
        stok + " WHERE kode_akun = " + kodee + "");

        konek.closekoneksi();

    } catch (SQLException e) {

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

    } catch (ClassNotFoundException ex) {

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

```

```

    }

    //----- Memasukan pada table transaksi detail
    try {
        Connection conn = konek.openkoneksi();
        java.sql.Statement stm = conn.createStatement();

        stm.executeUpdate("INSERT INTO kas_out_detail(no_kk,
        kode_akun, nominal, keterangan) VALUES (" + row_kode + ", " + kodee + ", "
        + jumlah + ", " + kete + ")");

        empty = 1;

        konek.closekoneksi();

    } catch (SQLException e) {

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

    } catch (ClassNotFoundException ex) {

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

```

```

    }else{

        JOptionPane.showMessageDialog(null, "Sistem tidak menemukan
barang dengan kode = " + kodee , "Barang " + kodee + " Gagal Disimpan",
JOptionPane.ERROR_MESSAGE);

    }

```

```

//----- Opsi jika terdapat barang yang belum satupun di masukan
if(empty == 0){

```

```

    try {

        Connection conn = konek.openkoneksi();
        java.sql.Statement stm = conn.createStatement();
        stm.executeUpdate("DELETE FROM kas_out WHERE no_kk = "
+ row_kode + "");
        konek.closekoneksi();
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error " + e);
    } catch (ClassNotFoundException ex) {

```

```

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

    }

```

```

    }else{

    }

}

```

```

        JOptionPane.showMessageDialog(null, "Berhasil menyimpan data
transaksi");

```

```

try {

    HashMap hash = new HashMap();

    hash.put("no_keluar", txtkode.getText());


    File file = new File("src/report/nota_anggarankeluar.jrxml");

    JasperDesign jasperDesign = JRXmlLoader.load(file);

    JasperReport                                jasperReport                                =
JasperCompileManager.compileReport(jasperDesign);

    JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport,
hash, konek.openkoneksi());

    JasperViewer.viewReport(jasperPrint, false);
} catch (ClassNotFoundException | JRException e) {
    JOptionPane.showMessageDialog(null, "Error " + e);
}

this.hide();
} else {

    JOptionPane.showMessageDialog(null, "Terdapat inputan yang kosong.");

}

}

```

```

private void cmbkodeItemStateChanged(java.awt.event.ItemEvent evt) {

    String nm_kategori = cmbkode.getSelectedItem().toString();

    if(!nm_kategori.equals("")){

        try {

```

```

        Connection conn = konek.openkoneksi();

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

        java.sql.ResultSet sql = stm.executeQuery("SELECT kode_akun FROM
master_akun WHERE nama_akun='"+nm_kategori+"'");

        if(sql.next()){

            txtkodee.setText(sql.getString("kode_akun"));

        }

        konek.closekoneksi();

    } catch (SQLException e) {

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

    } catch (ClassNotFoundException ex) {

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

    }

    }else{

        txtkode.setText("");

    }    // TODO add your handling code here:

}

```

```

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

    // TODO add your handling code here:

}

```

```

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

    // TODO add your handling code here:

```

```

    }

    private void DTtglActionPerformed(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(Data_Pengeluaran_Kas.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Data_Pengeluaran_Kas.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Data_Pengeluaran_Kas.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Data_Pengeluaran_Kas.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);

    }

//</editor-fold>

//</editor-fold>

/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        new Data_Pengeluaran_Kas().setVisible(true);

```

```

    }

    });

}

// Variables declaration - do not modify

private javax.swing.JTextField DTtgl;

private javax.swing.JButton btnbaru;

private javax.swing.JButton btnhapus;

private javax.swing.JButton btnsimpan;

private javax.swing.JButton btnubah;

private javax.swing.JComboBox<String> cmbkode;

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel3;

private javax.swing.JPanel jPanel4;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JTable tblanggaran;

private javax.swing.JTextField txtjml;

```

```

private javax.swing.JTextArea txtket;

private javax.swing.JTextField txtkode;

private javax.swing.JTextField txtkodee;

private javax.swing.JLabel txttotal;

// End of variables declaration

```

```

public int simpan(){
    try{
        String sql="insert into kas_out "+
        "values"+
        "("+txtkode.getText()+", "+
        """+DTtgl.getText()+", "+
        """+txtket.getText()+"" "+
        ")";
        conn.simpanData(sql);
    }
    catch(Exception e){
        System.out.printf(null, e);
    }
    return 0;
}

public int simpansetill(){
    for (int x = 0; x < tblanggaran.getRowCount(); x++) {
        try {
            String sql="insert into kas_out_detail "+
            "values"+

```

```

"(null,"+tblanggaran.getValueAt(x, 0).toString()+", "+
"""+tblanggaran.getValueAt(x, 1).toString()+", "+
"""+tblanggaran.getValueAt(x, 2).toString()+"" "+
");

conn.simpanData(sql);

ResultSet rs2=conn.ambilData("SELECT SisaAnggaran from tbl_jurnal
group by kode_jurnal "
+ "having kode_jurnal=max(kode_jurnal) order by kode_jurnal desc
");

rs2.next();
int jml=rs2.getInt(1);
String sql2="insert into tbl_jurnal "+
"values"+
"(null, "+
"""+tblanggaran.getValueAt(x, 0).toString()+", "+
"NULL, "+
"""+DTtgl.getText()+"',0', "+
"""+tblanggaran.getValueAt(x,
2).toString()+""
, ""+tblanggaran.getValueAt(x, 1).toString()+""
+ " ,("+jml+"-"+tblanggaran.getValueAt(x, 2).toString()+")"+
");

conn.simpanData(sql2);
} catch (SQLException ex) {

```

```

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

    }

}

return 0;

}

```

```

public void tambah(){
    Object[] data = new Object[6];
    data[0] = txtkode.getText();
    data[1] = txtkodee.getText();
    data[2] = txtjml.getText();
    data[3] = DTtgl.getText();
    data[4] = txtket.getText();
    tb.addRow(data);
    int subtotal=0;

    for (int x = 0; x < tblanggaran.getRowCount(); x++) {
        subtotal = subtotal + Integer.parseInt(tblanggaran.getValueAt(x,
2).toString().replace(".", ""));
        txttotal.setText(""+subtotal);
    }
}

public void ubah(){
    Object qty=txtjml.getText();

```

```

        Object ket=txtket.getText();

        int baris =tblanggaran.getSelectedRow();

        tblanggaran.setValueAt(qty, baris, 2);

        tblanggaran.setValueAt(ket, baris, 3);

        int subtotal=0;

        for (int x = 0; x < tblanggaran.getRowCount(); x++) {

            subtotal = subtotal + Integer.parseInt(tblanggaran.getValueAt(x,
2).toString().replace(".", ""));

            txttotal.setText(""+subtotal);

        }

    }

    public void hapus(){

        int baris =tblanggaran.getSelectedRow();

        tb.removeRow(baris);int subtotal=0;

        for (int x = 0; x < tblanggaran.getRowCount(); x++) {

            subtotal = subtotal + Integer.parseInt(tblanggaran.getValueAt(x,
2).toString().replace(".", ""));

            txttotal.setText(""+subtotal);

        }

    }

    public int hapus(String kd){

        String sql="delete from barang where kode = '"+kd+"'";

        conn.simpanData(sql);

        return 0;

    }

```

```

public void tabelmodel(){
    tb =new DefaultTableModel();
    tb.addColumn("No Kas Keluar");
    tb.addColumn("Kode Akun");
    tb.addColumn("Jumlah");
    tb.addColumn("Tanggal");
    tb.addColumn("Keterangan");
    tblanggaran.setModel(tb);
}

void setform(boolean flag){
    txtjml.setEditable(flag);
    txtket.setEditable(flag);
}

void bersih(){
    txtjml.setText("");
    txtket.setText("");

    cmbkode.setSelectedItem("PILIH");

    // DTtgl.setDate(null);

}

public void Setkode(){
    try {

        ResultSet res  = conn.ambilData("select * from kas_out order by no_kk
desc");

```

```

if (res.next()) {

    String nofak = res.getString("no_kk").substring(2);

    String AN = "" + (Integer.parseInt(nofak) + 1);

    String Nol = "";

    if(AN.length()==1)

        {Nol = "000";}

    else if(AN.length()==2)

        {Nol = "00";}

    else if(AN.length()==3)

        {Nol = "0";}

    else if(AN.length()==4)

        {Nol = "";}

    txtkode.setText("KK" + Nol + AN);

} else {
    txtkode.setText("KK0001");

}

}

}

```

3.5.2. Black Box Testing

Pengujian unit berfungsi untuk menguji kelayakan aplikasi yang telah dibuat. Black box testing digunakan sebagai teknik dalam pengujian unit. Pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa kelayakan fungsional sistem informasi akuntansi laporan arus kas pada Desa Gempolsari Kecamatan Patobeusi Kabupaten Subang.

1. Pengujian halaman login

Tabel III.10.

Hasil Pengujian Black Box Testing Halaman Login

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Keterangan
1	Jika salah satu kolom dikosongkan	password: (kosong)	aplikasi menolak akses dan menampilkan pesan “password harus diisi”	Sesuai harapan	Valid
2	Kolom terisi dan sesuai (username dan password sesuai)	username: (selvi) password: (123456)	Aplikasi menerima akses dan mengantarkan pengguna ke halaman utama.	Sesuai Harapan	Valid

Sumber : Hasil penelitian (2019)

2. Pengujian halaman master akun

Tabel III.11.

Hasil Pengujian Black Box Testing Halaman Master Akun

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Keterangan
----	--------------------	-----------	-----------------------	-----------------	------------

1	Jika salah satu kolom dikosongkan	Jenis akun: (kosong)	aplikasi menolak akses dan menampilkan pesan “semua data harus diisi”	Sesuai harapan	Valid
2	Jika semua kolom terisai	Kolom: Terisi	Aplikasi menerima akses dan menyimpan data serta menampilkan “data tersimpan”	Sesuai Harapan	Valid

Sumber : Hasil penelitian (2019)

3. Pengujian halaman saldo awal

Tabel III.12.

Hasil Pengujian Black Box Testing Halaman saldo awal

No	Skenario Pengujian	<i>Test case</i>	Hasil yang diharapkan	Hasil pengujian	Keterangan
1	Jika salah satu kolom dikosongkan	nominal: (kosong)	aplikasi menolak akses dan menampilkan pesan “semua data harus diisi”	Sesuai harapan	Valid
2	Jika semua kolom terisai	Kolom: Terisi	Aplikasi menerima akses dan menyimpan data	Sesuai Harapan	Valid

Sumber : Hasil penelitian (2019)

4. Pengujian halaman kas masuk

Tabel III.13.

Hasil Pengujian Black Box Testing Halaman Kas Masuk

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Keterangan
1	Jika salah satu kolom dikosongkan	tanggal: (kosong)	aplikasi menolak akses dan menampilkan pesan “semua data harus diisi”	Sesuai harapan	Valid
2	Jika semua kolom terisi	Kolom: Terisi	Aplikasi menerima akses dan menyimpan data serta menampilkan	Sesuai Harapan	Valid

Sumber : Hasil penelitian (2019)

5. Pengujian halaman kas keluar

Tabel III.14.

Hasil Pengujian Black Box Testing Halaman Kas Keluar

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Keterangan
1	Jika salah satu kolom dikosongkan	Jumlah: (kosong)	aplikasi menolak akses dan menampilkan pesan “semua data harus diisi”	Sesuai harapan	Valid
2	Jika semua kolom terisi	Kolom: Terisi	Aplikasi menerima akses dan menyimpan data serta menampilkan	Sesuai Harapan	Valid

Sumber : Hasil penelitian (2019)

6. Pengujian halaman Jurnal

Tabel III.14.

Hasil Pengujian Black Box Testing Halaman Jurnal

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Keterangan
1	Jika salah satu kolom dikosongkan	No jurnal: (kosong)	aplikasi menolak akses dan menampilkan pesan “semua data harus diisi”	Sesuai harapan	Valid
2	Jika semua kolom terisi	Kolom: Terisi	Aplikasi menerima akses dan menyimpan data serta menampilkan	Sesuai Harapan	Valid

Sumber : Hasil penelitian (2019)

3.5.3. Spesifikasi *Hardware* dan *Software*

Spesifikasi kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*software*) yang diperlukan untuk mengoperasikan sistem informasi akuntansi laporan arus kas pada Desa Gempolsari Kecamatan Patobeusi Kabupaten Subang. Berikut ini adalah uraian dari kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*software*).

1. Perangkat Keras (*Hardware*)

Kebutuhan perangkat keras (*hardware*) yang diperlukan untuk mengimplementasikan sistem informasi akuntansi laporan arus kas pada Desa Gempolsari, yaitu:

1. Processor : 2.0 Ghz
2. Harddisk : 32 Gb
3. Memory : 2 Gb

4. VGA : 128 Mb
 5. Monitor : 15’’
 6. Mouse dan keyboard
 7. Printer
2. Perangkat Lunak (Software)

Spesifikasi perangkat lunak (software) yang dibutuhkan untuk mendukung sistem informasi akuntansi laporan arus kas pada Desa Gempolsari adalah sebagai berikut:

1. Sistem Operasi : Microsoft Windows 7
2. Sistem Aplikasi : Java NetBeans
3. *Data base* : Phpmyadmin
4. *Connector Aplikasi* : XAMPP

