

BAB III

PEMBAHASAN

3.1. Tinjauan Perusahaan

Alokasi Dana Desa (ADD) adalah dana yang bersumber dari Anggaran Pendapatan Belanja Daerah (APBD) yang dialokasikan dengan tujuan pemerataan kemampuan keuangan antar desa untuk mendanai kebutuhan desa dalam rangka penyelenggaraan pemerintahan dan pelaksanaan pembangunan serta pelayanan masyarakat. Alokasi Dana Desa (ADD) merupakan perolehan bagian Keuangan Desa dari kabupaten yang penyalurannya melalui kas Desa.

Berdasarkan dari APBD jumlah Dana ADD yang ada di Desa Ciasembaru Kecamatan Ciasem yaitu sebesar Rp.654.075.500,- dibagi menjadi 2 tahapan dengan ketentuan Pergub yaitu Tahap 1 sebesar 70% untuk penyelenggaraan pemerintahan desa, pelaksanaan pembangunan desa, pembinaan kemasyarakatan, dan pemberdayaan masyarakat desa. Tahap 2 sebesar 30% untuk kas operasional pemerintahan desa. Ada 4 (empat) bidang yang direalisasikan di Desa Ciasembaru Kecamatan Ciasem Kabupaten Subang, diantaranya Bidang Penyelenggaraan Pemerintahan Desa, Bidang Pelaksanaan Pembangunan Desa, Bidang Pembinaan Kemasyarakatan, Dan Bidang Pemberdayaan Masyarakat Desa.

Dalam hasil pelaksanaan kegiatan riset Tugas Akhir di Desa Ciasembaru terdapat aturan yang ditentukan, dalam hal ini struktur organisasi yang menggambarkan garis perintah dan penerimaan perintah serta fungsi-fungsi pelaksanaan tersebut sehingga semua menjadi lancar. Penulis akan menerangkan sejarah dan struktur di tempat penulis mengadakan riset Tugas Akhir.

1.1.1. Sejarah Perusahaan

Pada masa pemekaran tahun 1980 Pejabat Sementara (PJS) kepala desa dijabat oleh Misna Sutisna. Selanjutnya pada tahun 1983 diadakan pemilihan kepala Desa Devinitif hasil pemilihan tersebut terpilih yaitu Tamir Effendi untuk masa jabatan 8 tahun, pada tahun 1984 keadaan Desa Ciasembarusangat jauh tertinggal dibanding Desa-Desa lain yang berada di Kecamatan Ciasem. Jalan desa dan jalan gang keadaan masih tanah, Dusun-Dusun masih kumuh anak usia SD dan SMP banyak yang tidak tamat (*Drop Out*) Kantor Pemerintah Desa pun masih sewa bangunan, kemiskinan mencapai 60% dari jumlah penduduk.

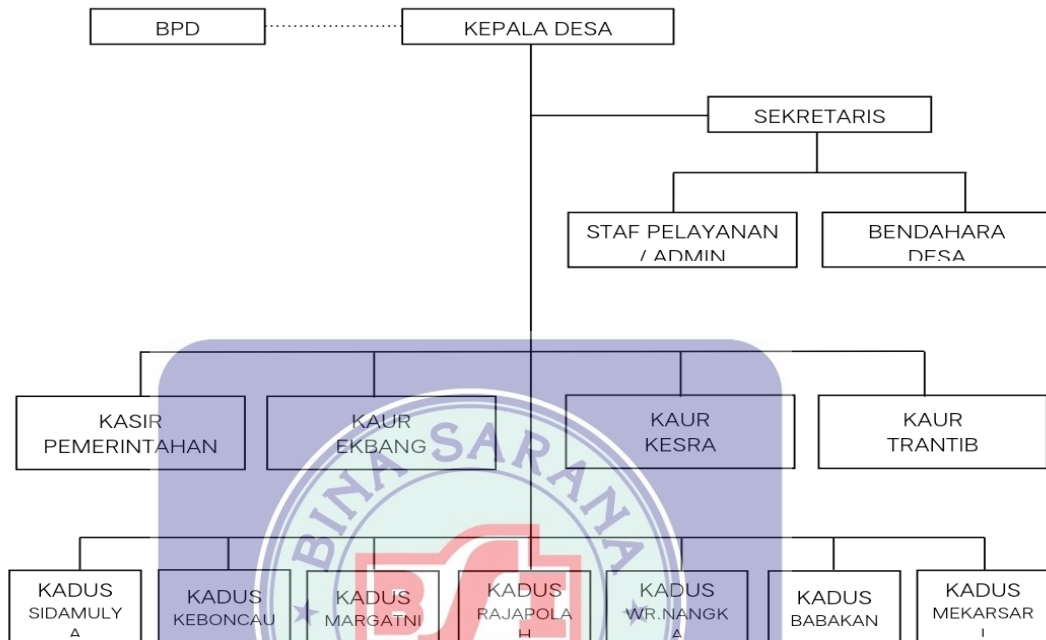
Sekarang berkat kerja keras Pemerintah Desa dan masyarakat dari tahun 1984 sampai dengan sekarang Desa Ciasembaru sudah banyak mengalami perubahan baik dibidang pembangunan infrastruktur jalan Desa dan jalan gang 90% sudah di hotmix meskipun sudah banyak yang rusak lagi. Begitu juga sarana Pendidikan baik SD ataupun Pendidikan Agama sudah banyak perbaikan meskipun belum 100%. Masalah kemiskinan dan pengangguran terus berkurang, anak usia SD dan SMP yang tidak tamat (*Drop Out*) sekarang sudah berkurang.

Masyarakat Desa Ciasembaru memiliki 7 Dusun serta 7 Rukun Warga (RW) dan 24 Rukun Tetangga (RT), diantara Dusun tersebut selain penduduk asli juga banyak penduduk pendatang. Dari berbagai aspek kehidupan pada Desa Ciasembaru memiliki beragam karakteristik penduduk berdasarkan usia, jenis kelamin, tingkat pendidikan dan mata pencaharian. Data mengenai kependudukan diperoleh melalui Buku Profil Desa. Desa Ciasembaru memiliki luas wilayah menurut penggunaannya yaitu 727,586 Ha, yang terdiri dari luas permukiman 153,736 Ha, luas pesawahan 570,000 Ha, luas TPU 3.080 Ha, luas perkantoran 0,350 Ha dan luas prasarana umum dan lainnya 0,420 Ha.

1.1.2. Struktur Organisasi Pemerintahan Desa

Struktur Organisasi Kantor Desa Ciasembaru Kecamatan Ciasem Kabupaten

Subang terlihat pada gambar berikut ini:



Sumber: Kantor Desa Ciasembaru.

Gambar III.1.
Struktur Organisasi Desa Ciasembaru

Adapun tugas pokok dan fungsi setiap jenjang dari struktur organisasi yang terdapat pada Kantor Desa Ciasembaru Kecamatan Ciasem Kabupaten Subang yaitu:

1. Badan Perwakilan Desa (BPD)

BPD mempunyai fungsi menetapkan peraturan desa bersama kepala desa, menampung dan menyalurkan aspirasi masyarakat.

a. Tugas pokok:

- 1) Membahas rancangan peraturan desa bersama kepala desa.
- 2) Membentuk panitia pemilihan kepala desa.

- 3) Melaksanakan pengawasan terhadap pelaksanaan peraturan desa dan peraturan kepala desa.
- 4) Mengusulkan, pengangkatan dan pemberhentian kepala desa.
- 5) Menggali, menampung, menghimpun, merumuskan dan menyalurkan aspirasi masyarakat.
- 6) Menyusun tata tertib BPD.

b. Fungsi:

- 1) Meminta keterangan kepada pemerintah desa.
- 2) Menyatakan pendapat Kewajiban.
- 3) Mengamalkan Pancasila, melaksanakan UUD 1945 dan mentaati segala peraturan perundang-undangan.
- 4) Melaksanakan kehidupan demokrasi dalam penyelenggaraan pemerintahan desa.
- 5) Mempertahankan dan memelihara hukum nasional serta keutuhan NKRI. Menyerap, menampung, menghimpun dan menindaklanjuti aspirasi masyarakat Memproses pemilihan kepala desa.
- 6) Mendahulukan kepentingan umum diatas kepentingan pribadi, kelompok dan golongan.
- 7) Menghormati nilai-nilai sosial budaya dan adat istiadat masyarakat setempat.
- 8) Menjaga norma dan etika dalam hubungan kerja dengan lembaga kemasyarakatan.
- 9) Memberikan persetujuan kerjasama antar Desa dalam Kabupaten maupun antar desa di luar Kabupaten.
- 10) Menggali, menampung, menghimpun, merumuskan dan menyalurkan aspirasi masyarakat

2. Kepala Desa

- a. Tugas pokok: menyelenggarakan pemerintahan Desa, melaksanakan pembangunan, pembinaan kemasyarakatan, dan pemberdayaan masyarakat.
- b. Fungsi:
 - 1) Menyelenggarakan pemerintahan Desa berdasarkan kebijakan yang ditetapkan berdasarkan BPD.
 - 2) Mengajukan rancangan peraturan Desa.
 - 3) Menetapkan peraturan-peraturan yang telah mendapatkan persetujuan bersama BPD.
 - 4) Menyusun dan mengajukan rancangan peraturan Desa mengenai APBD untuk dibahas dan ditetapkan bersama BPD.
 - 5) Membina kehidupan masyarakat Desa.
 - 6) Membina ekonomi Desa.
 - 7) Mengordinasikan pembangunan Desa secara partisipasi.
 - 8) Mewakili Desa di dalam dan diluar pengadilan dan dapat menunjuk kuasa hukum untuk mewakilinya sesuai dengan peraturan perundang-undangan.
 - 9) Melaksanakan wewenang lain sesuai dengan peraturan perundang-undangan.

3. 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) Penyiapan bantuan penyusunan Peraturan Desa.
- 4) Penyiapan bahan laporan penyelenggaraan Pemerintahan Desa.
- 5) Melaksanakan tugas Kepala Desa apabila Kepala Desa diberhentikan sementara.
- 6) Pengkoordinasian penyelenggaraan tugas-tugas urusan; dan
- 7) Pelaksanaan tugas lain yang diberikan oleh Kepala Desa.

4. Bendahara

- a. Tugas pokok: perbendaharaan meliputi perencanaan kas yang baik, pencegahan terjadinya kebocoran/penyimpangan, pencairan sumber pembiayaan yang peling murah, dan mengidari adanya dana yang menganggur (tidak terserap).
- b. Fungsi:
 - 1) Menerima, menyimpan, menatausahakan, dan membukukan uang/surat berharga dalam pengelolaan.
 - 2) Melakukan pengujian dan pembayaran berdasarkan perintah.
 - 3) Menolak perintah pembayaran apabila tidak memenuhi persyaratan untuk dibayarkan.
 - 4) Melakukan memungutan/pemotongan penerimaan negara dari pembayaran yang dilakukan.
 - 5) Menyetorkan pemotongan/pemungutan kewajiban ke kas negara.
 - 6) Mengelola rekening tempat penyimpanan; dan
 - 7) Menyampaikan laporan pertanggung jawaban (LPJ) kepada Kepala Desa.

5. Staf Pelayanan

- a. Tugas pokok: Membantu Kepala Desa dalam melaksanakan penyiapan bahan perumusan kebijakan teknis Penyusunan Program Keagamaan serta melaksanakan Program pemberdayaan masyarakat dan sosial kemasyarakatan.

b. 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, dan;
- 4) mengkoordinasikan kegiatan pemberdayaan masyarakat sesuai bidang tugasnya.

6. Kaur 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. Kaur Ekonomi Pembangunan (ekbang)

- a. Fungsi pokok: Membantu Kepala Desa dalam melaksanakan penyiapan bahan perumusan kebijakan teknis pengembangan ekonomi masyarakat dan potensi desa, pengelolaan administrasi pembangunan, pengelolaan pelayanan masyarakat serta penyiapan bahan usulan kegiatan dan pelaksanaan tugas pembantuan.
- b. Fungsi:
 - 1) Penyiapan bantuan-bantuan analisa & kajian perkembangan ekonomi masyarakat.
 - 2) Pelaksanaan kegiatan administrasi pembangunan.
 - 3) Pengelolaan tugas pembantuan; dan
 - 4) Pelaksanaan tugas lain yang diberikan oleh Kepala Desa.

8. Kaur Kesejahteraan Rakyat (kesra)

- a. Tugas pokok: 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 bahan untuk pelaksanaan program kegiatan keagamaan.
 - 2) Penyiapan dan pelaksanaan program perkembangan kehidupan beragama.
 - 3) Penyiapan bahan dan pelaksanaan program, pemberdayaan masyarakat dan sosial kemasyarakatan; dan
 - 4) Pelaksanaan tugas-tugas lain yang diberikan Kepala Desa.

9. Kaur ketertamandan Ketertiban (tramtib)

- a. Tugas pokok: Membantu kepala desa melaksanakan pembinaan ketentraman dan ketertiban umum serta pembinaan perlindungan masyarakat.

b. Fungsi:

- 1) Penyusunan dan pelaksanaan program kegiatan pembinaan ketentraman dan ketertiban umum.
- 2) Penyelenggaraan penegakan Peraturan Daerah, Peraturan Bupati, Keputusan Kepala Daerah dan Peraturan perundang-undangan lainnya di desa.
- 3) Pelaksanaan pembinaan dan perlindungan kepada masyarakat dan anggota LINMAS di Kelurahan.
- 4) Pelaksanaan penertiban terhadap gangguan social.
- 5) Pelaksanaan tugas lain yang diberikan oleh atasan sesuai dengan tugas dan fungsinya.

10. Kepala Dusun (kadus)

a. Tugas pokok:

- 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

Pada tinjauan kasus ini, penulis akan menjelaskan dan membahas mengenai pengelolaan kas dana desa diantaranya proses sistem berjalan, *activity diagram*, spesifikasi dokumen masukan, spesifikasi dokumen keluaran, permasalahan pokok, dan pemecahan masalah.

3.2.1. Proses Bisnis Sistem Berjalan

Dalam Tugas Akhir ini, proses sistem berjalan yang diambil pada kantor Desa Ciasembaru Kecamatan Ciasem Kabupaten Subang yaitu sistem pembuatan laporan pengelolaan kas yang sudah melalui beberapa proses. Proses bisnis sistem berjalan pembuatan laporan pengelolaan kas secara umum pada kantor Desa Ciasembaru adalah:

1. Proses Pengajuan Dana

Kas masuk bersumber dari alokasi dana desa yang diajukan setiap satu tahun sekali, Sekretaris Desa melakukan pengajuan dana dalam bentuk proposal yang berisikan rincian-rincian data tersebut kepada tim Dinas Pemberdayaan Masyarakat dan Desa (DPMD). Jika persyaratan tidak lengkap atau tidak

memenuhi persyaratan maka tim Dinas Pemberdayaan Masyarakat dan Desa (DPMD) mengembalikan proposal tersebut ke desa untuk melengkapi persyaratan yang kurang tetapi jika proposal sudah lengkap kemudian proposal tersebut disetujui oleh kepala kecamatan maka tim Dinas Pemberdayaan Masyarakat dan Desa (DPMD) mengirimkan proposal tersebut ke Kabupaten untuk diproses.

2. Proses Pencairan Dana

Dari pengajuan sampai kepencairan Dana berjarak 1 bulan hari kerja, pencairan dana ini melalui kepala desa dan bendahara untuk melakukan pencairan dana tersebut di Bank Bjb dengan mengisi slip penarikan yang sudah disediakan Bank Bjb. Alokasi dana desa tersebut di transfer ke rekening desa Ciasem sesuai aturan Pergub, penyaluran dana tersebut dilakukan paling lambat 7 hari kerja setelah dana tersebut diterima. Kemudian bendahara mencatat data penerimaan dana ke dalam kas masuk.

3. Proses Pengeluaran Dana

Pada prosedur ini, Bendahara desa memberikan dana dan rencana pengeluaran biaya kepada KAUR yang sudah disetujui Kepala Desa untuk biaya yang dibutuhkan. Kemudian bagian KAUR memberikan bukti pengeluaran kas berupa kwitansi kepada Bendahara Desa. Bendahara Desa kemudian mencatatnya kedalam buku kas keluar dan mengarsipkan kwitansi tersebut.

4. Proses Laporan Pertanggungjawaban

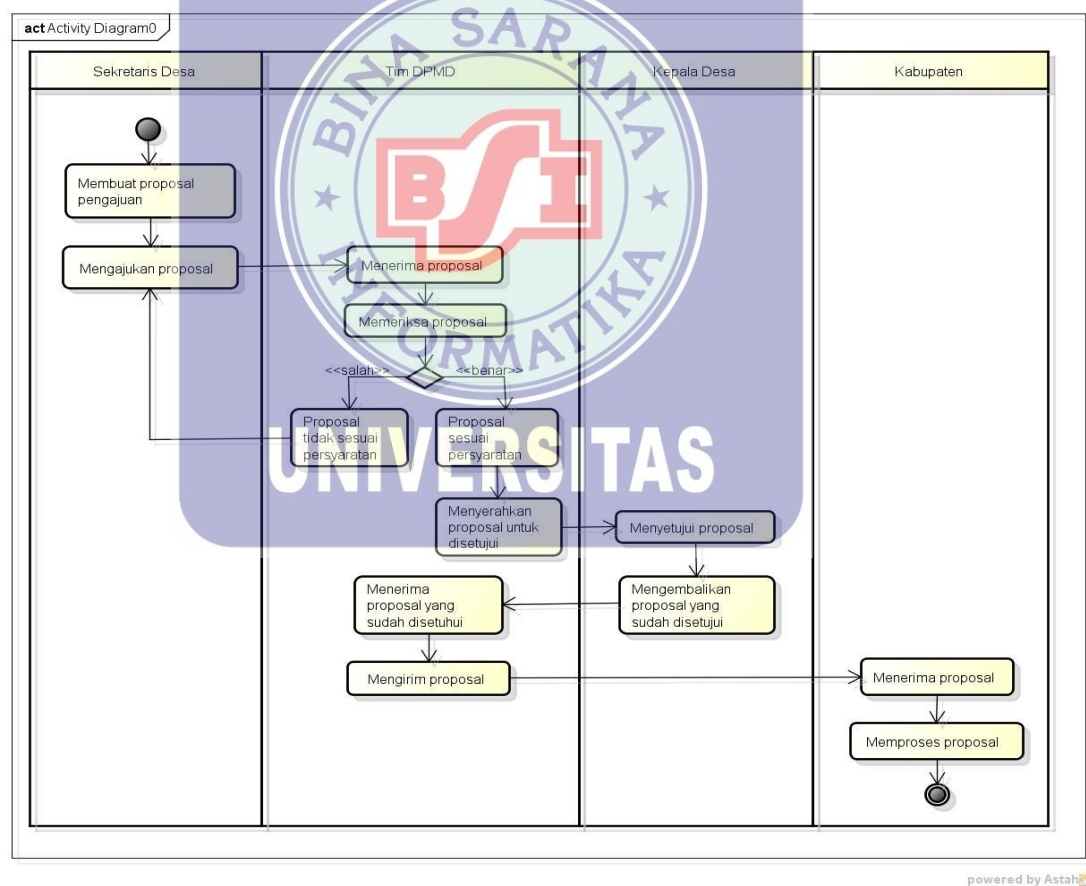
Bendahara Desa membuat laporan-laporan yang ada setiap bulannya berupa laporan buku kas umum (BKU), dan laporan pertanggungjawaban penerimaan dan pengeluaran. Laporan buku kas umum diketahui Kepala Desa, sedangkan laporan pertanggungjawaban penerimaan dan pengeluaran tersebut harus diverifikasi oleh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

pertanggungjawaban penerimaan dan pengeluaran tersebut kemudian bendahara menyerahkan laporan-laporan tersebut kepada kepala Desa dan dikembalikan kepada sekretaris desa untuk diarsipkan.

3.2.2. Activity Diagram Proses Sistem Berjalan

Activity Diagram digunakan untuk menggambarkan alur kerja pada sebuah sistem berjalan pengelolaan kas. Adapun penggambaran dari *activity diagram* yang menggambarkan sistem berjalan pengelolaan kas sebagai berikut:

1. Proses Pengajuan Dana

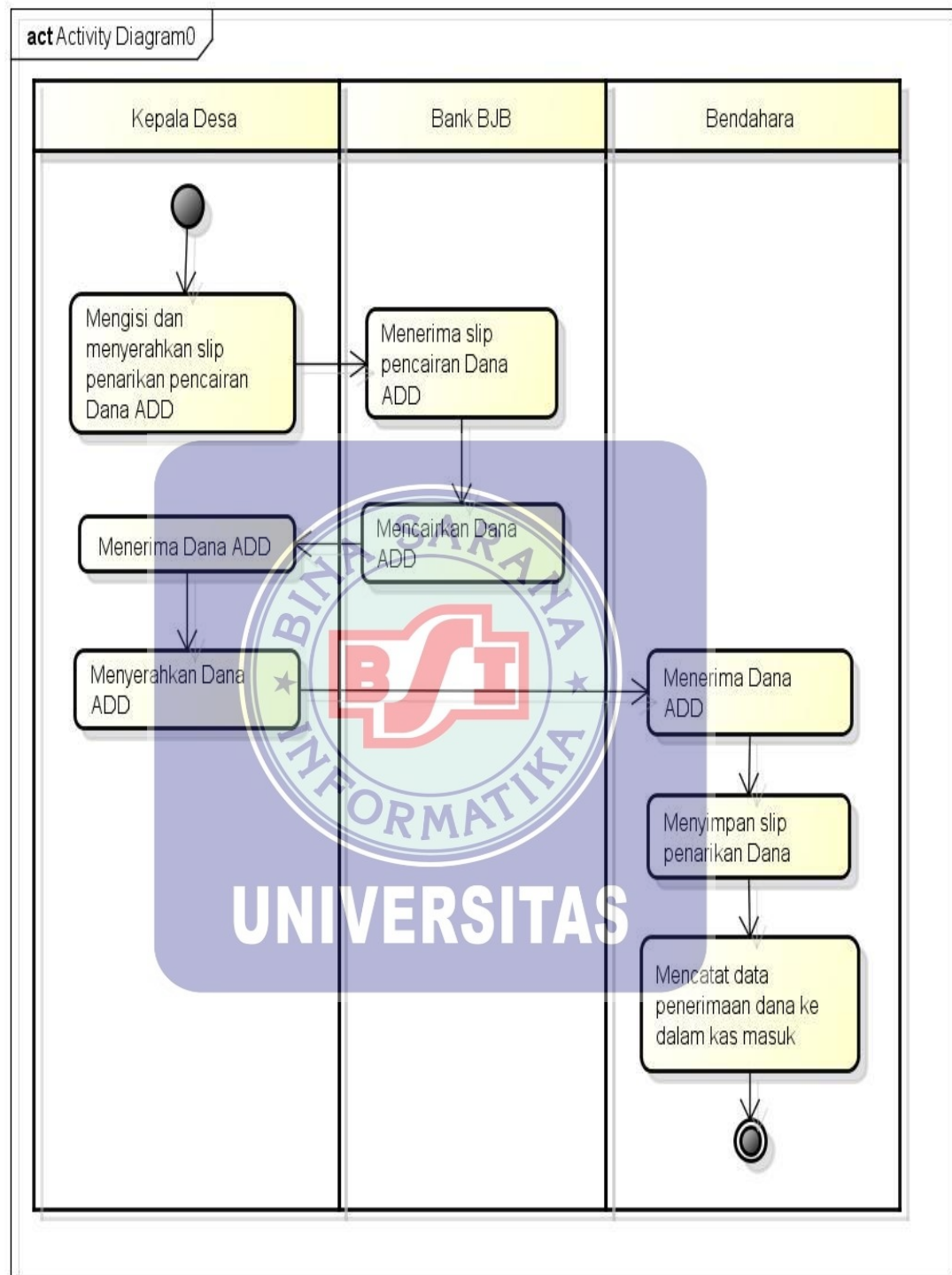


Sumber: (Hasil Penelitian, 2019)

Gambar III.2.

Activity Diagram Proses Pengajuan Dana

2. Proses Pencairan Dana



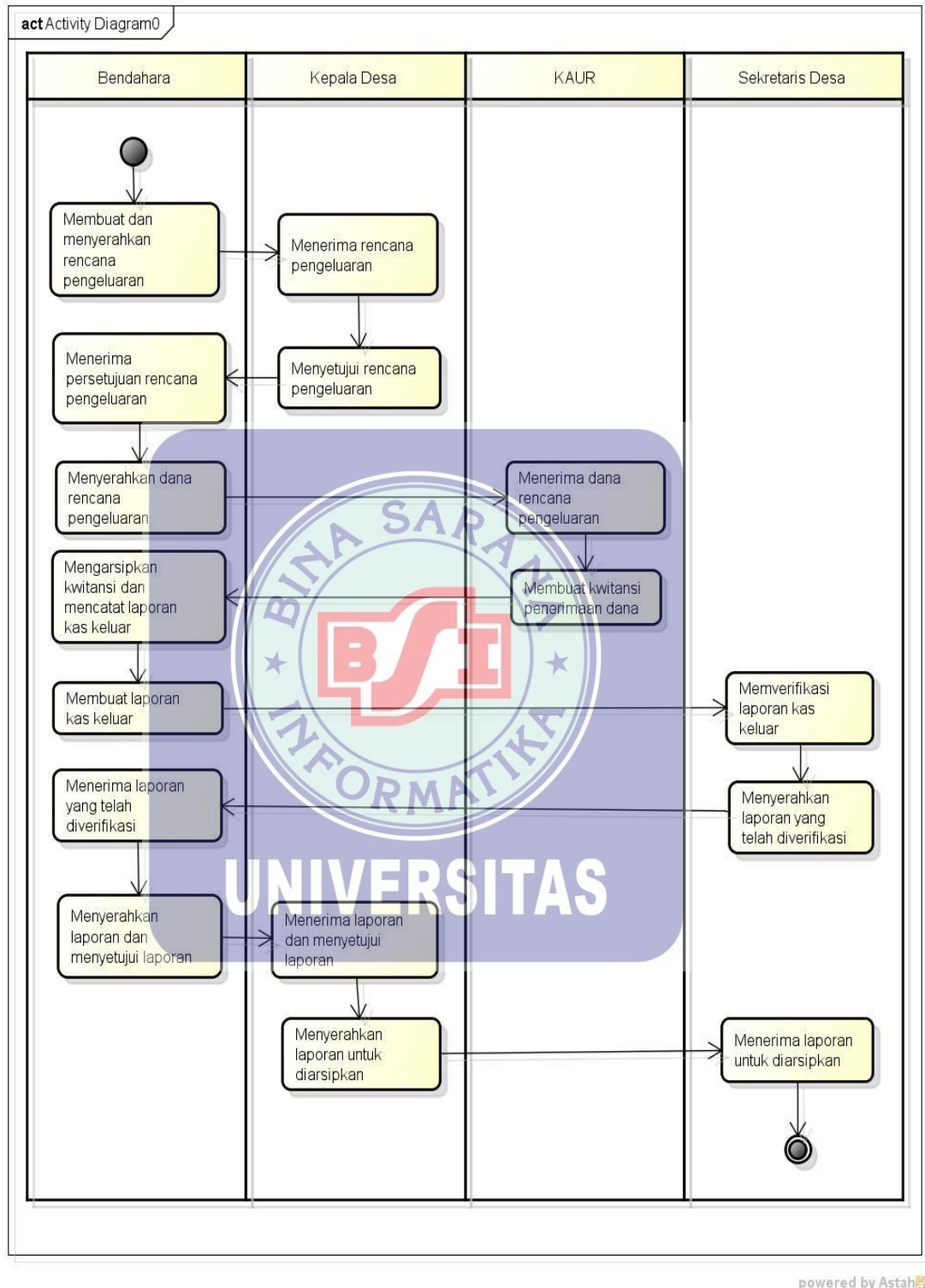
powered by Astah

Sumber: (Hasil Penelitian, 2019)

Gambar III.3.

Activity Diagram Proses Pencairan Dana

3. Proses Pengeluaran Dana

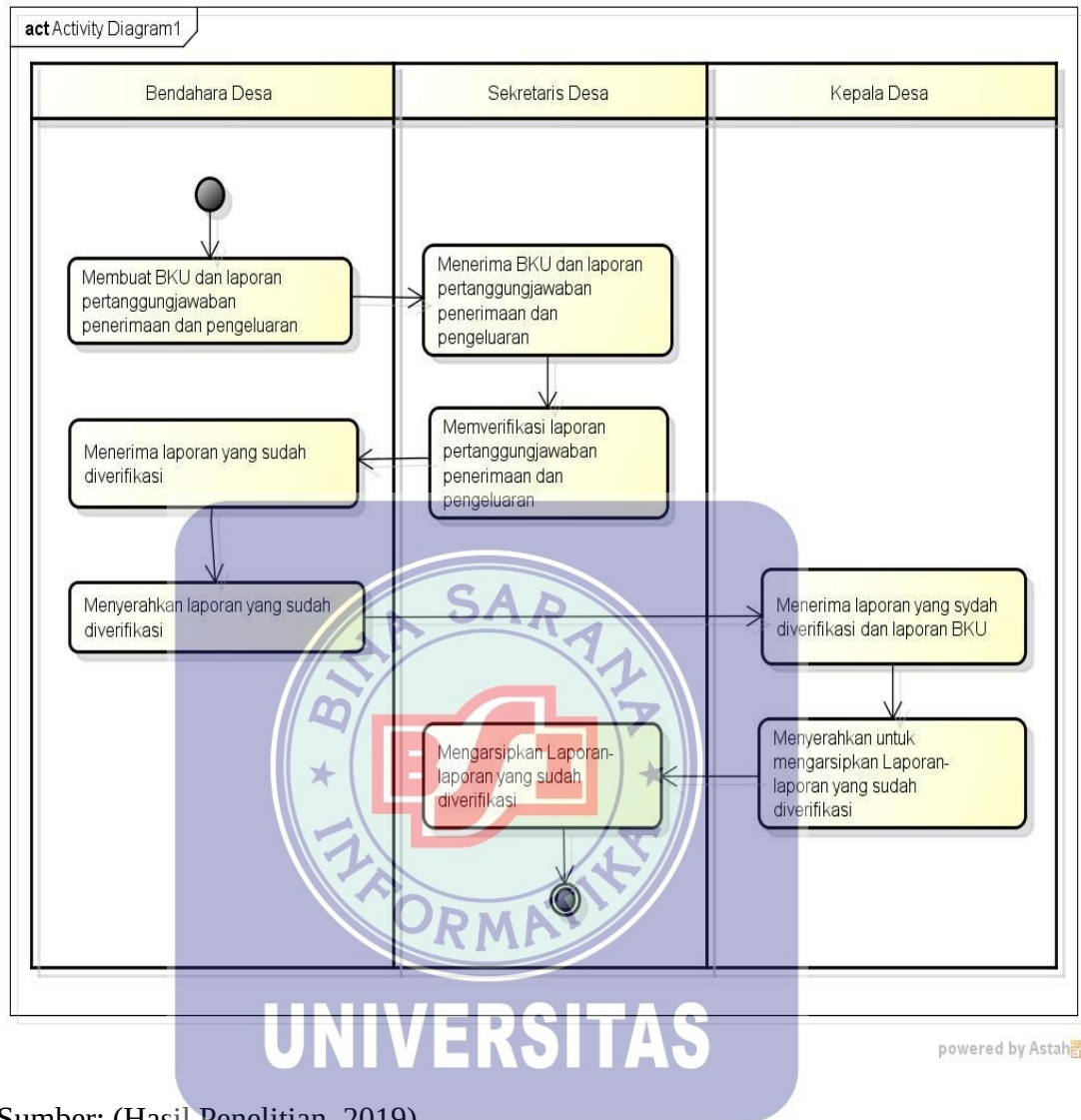


Sumber: (Hasil Penelitian, 2019)

Gambar III.4.

Activity Diagram Proses Pengeluaran Dana

4. Proses Laporan Pertanggungjawaban



Sumber: (Hasil Penelitian, 2019)

Gambar III.5.

Activity Diagram Proses Laporan Pertanggungjawaban

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 Pengeluaran

Nama Dokumen	: Kwitansi pengeluaran
Fungsinya	: Sebagai bukti dari pengeluaran kas keluar
Sumber	: KAUR
Tujuan	: Bendahara
Jumlah	: Satu lembar
Media	: Kertas
Frekuensi	: Setiap terjadi pengeluaran kas keluar
Lampiran	: A.1.

2. Slip Penarikan

Nama Dokumen	: Slip penarikan
Fungsinya	: Sebagai bukti dari penarikan dana
Sumber	: Bank Bjb
Tujuan	: Bendahara Desa
Jumlah	: Satu lembar
Media	: Kertas
Frekuensi	: Setiap terjadi pengeluaran dana
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
Tujuan	: Kepala Desa
Jumlah	: Satu lembar
Media	: Kertas
Frekuensi	: Setiap terjadi perencanaan anggaran biaya
Lampiran	: B.1.

2. Pengajuan Dana

Nama Dokumen	: Proposal
Fungsinya	: Sebagai persyaratan pengajuan dana
Sumber	: Sekretaris
Tujuan	: Tim DPMD
Jumlah	: Enam lembar
Media	: Kertas
Frekuensi	: Setiap terjadi pengajuan dana dalam waktu 1 tahun 2 kali
Lampiran	: B.2.

3. Buku Kas Umum (BKU)

Nama Dokumen	: Buku Kas Umum
Fungsinya	: Sebagai laporan dari kas masuk dan kas keluar
Sumber	: Bendahara Desa
Tujuan	: Kepala Desa

Jumlah : Satu lembar

Media : Kertas

Frekuensi : Setiap akhir bulan

Lampiran : B.3.

4. Laporan Pertanggungjawaban

Nama : Laporan Pertanggungjawaban

Fungsinya : Sebagai laporan pertanggungjawaban atas penerimaan kas

dan pengeluaran kas

Sumber : Bendahara Desa

Tujuan : Kepala Desa

Jumlah : Satu lembar

Media : Kertas

Frekuensi : Setiap akhir bulan

Lampiran : B.4.



3.2.5. Permasalahan Pokok

Dari hasil pengamatan serta uraian yang telah penulis paparkan di atas, dapat diambil kesimpulan bahwa dalam proses pelaporan pengelolaan kas yang terjadi di Kantor Desa Ciasembaru masih dilakukan secara konvensional. Dengan dilakukan proses pengolahan data dalam penerima kas yang masih secara manual dimana dalam melakukan pembuatan laporan pengelolaan kas 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 ke Kabupaten.

2. Sering terjadi kekeliruan dalam pengisian data sehingga data kurang akurat.
3. Proses pencatatan data membutuhkan waktu lama sehingga memperlambat kinerja pembuatan laporan.
4. Membutuhkan ruang penyimpanan data yang luas dan aman untuk pengarsipan dokumen dana kas keluar dan kas masuk sehingga meminimalisasi kecurangan dalam dokumen.
5. Tidak adanya Buku khusus kas masuk dan keluar sehingga kurang efektif dalam pembuatan laporan.

3.2.6. Pemecahan Masalah

Untuk memecahkan masalah yang ada di Kantor Desa Ciasembaru, 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 mengefisienkan proses kerja agar lebih baik dan cepat.

Selain itu, apabila sebuah desa 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 Ciasembaru. Menggunakan bahasa pemrograman *java* berbasis desktop *client server* dengan editor *netbeans* dan *database mysql*, kemudian *local server XAMPP*. 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 *Software* akan menjelaskan kebutuhan-kebutuhan yang akan digunakan dalam rancangan sistem. Penelitian dilakukan pada Desa Ciasembaru menghasilkan hasil penelitian, pengolahan data yang dilakukan untuk dapat menyediakan informasi sebagai pendukung suatu laporan keuangan masih memerlukan waktu yang lama. Sehingga informasi yang dihasilkan pun belum secara maksimal dapat digunakan sebagai penunjang keputusan bagian keuangan.

3.3.1. Analisa Kebutuhan

Kebutuhan *Software* yang menjadi *interface* yang menghubungkan antara *user* dan program. Adapun Analisa Kebutuhan dari hasil tinjauan kasus untuk sebuah rancangan Aplikasi pengelolaan Kas Desa Ciasembaru.

A. Bendahara Desa

- A.1. Bendahara Desa dapat login
- A.2. Bendahara Desa dapat mengelola data Karyawan
- A.3. Bendahara Desa dapat mengelola data akun
- A.4. Bendahara Desa dapat mengelola transaksi kas masuk

A.5. Bendahara Desa dapat mengelola transaksi kas keluar

A.6. Bendahara Desa dapat mencari data jurnal

B. Sekretaris Desa

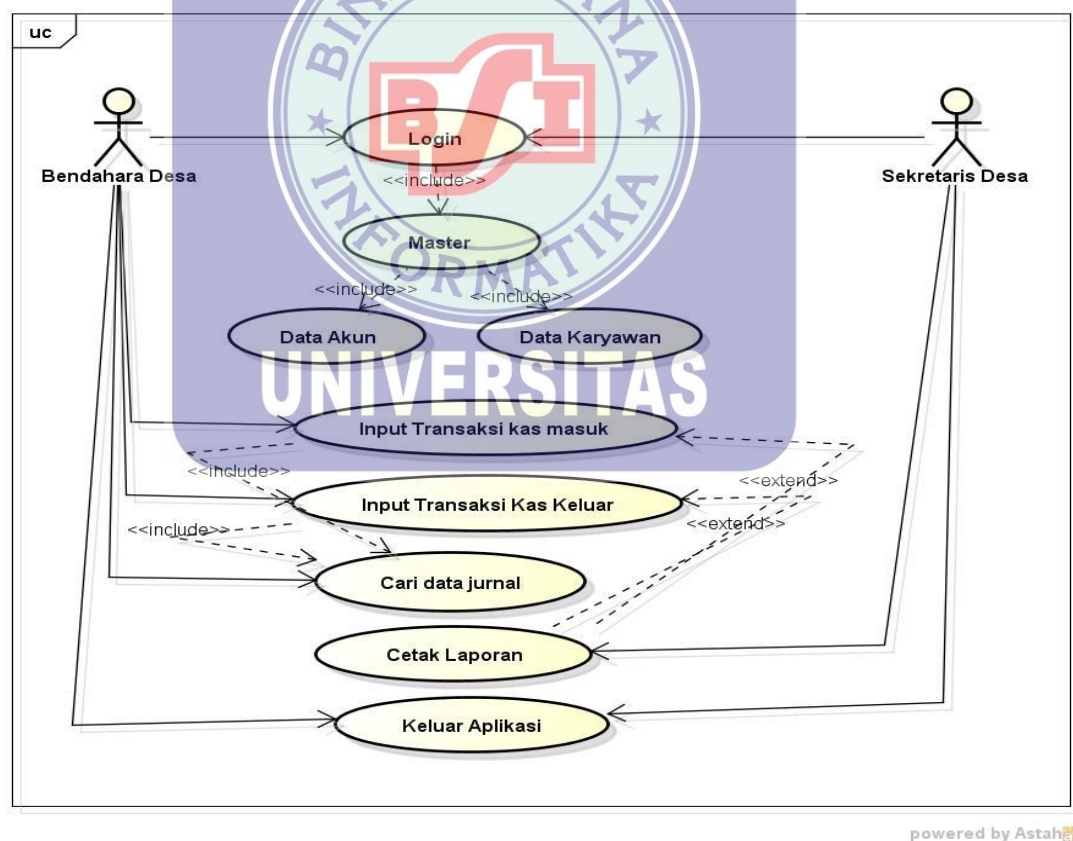
B.1. Sekretaris Desa dapat login

B.2. Sekretaris Desa dapat mencetak laporan

3.3.2. Use Case Diagram

Use case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

1. Use Case Diagram Sistem informasi akuntansi pelaporan pengelolaan kas



Sumber: Hasil Rancangan (2019)

Gambar III.6.

Use Case Diagram Proses Sistem Pengelolaan Kas

Tabel III.1.

Deskripsi Use Case Diagram Sistem Login

Use Case Narative sistem informasi <i>Login</i>	
Tujuan	Bendahara dapat <i>login</i>
Deskripsi	Sistem ini memungkinkan aktor untuk melakukan <i>login</i> dan masuk ke menu utama
Skenario Utama	
Aktor	Bendahara
Kondisi Awal	Aktor membuka aplikasi halaman <i>login</i>
Aksi Aktor	Reaksi Sistem
1. Aktor memilih submenu <i>Login</i> 2. Aktor Memasukan <i>Username</i> dan <i>Password</i>	Sistem akan menampilkan <i>form Login</i> Sistem akan menampilkan Menu Utama
Kondisi Akhir	Jika perintah sesuai maka sistem akan masuk kedalam menu utama dan aktor dapat melakukan aktivitas didalam sistem.

Tabel III.2.

Deskripsi Use Case Diagram Sistem Data Akun

Use Case Narative sistem informasi Data Akun	
Tujuan	Bendahara dapat data akun
Deskripsi	Sistem ini memungkinkan aktor untuk melakukan data akun
Skenario Utama	
Aktor	Bendahara
Kondisi Awal	Aktor membuka aplikasi halaman data akun
Aksi Aktor	Reaksi Sistem
1. Aktor memilih submenu data akun	Sistem akan menampilkan <i>form</i> data akun
2. Aktor memilih submenu tambah	Sistem akan menginput data akun baru
3. Aktor memilih submenu simpan	Sistem akan menyimpan data kedalam <i>database</i>
4. Aktor akan memilih submenu batal	Sistem akan menampilkan <i>form</i> awal
5. Aktor akan memilih submenu edit	Sistem akan mengedit data sesuai dengan keinginan aktor
6. Aktor akan memilih submenu hapus	Sistem akan menghapus data akun yang dipilih oleh aktor
Kondisi Akhir	Jika perintah sesuai maka sistem akan menampilkan sistem sesuai keinginan.

Tabel III.3.

Deskripsi Use Case Diagram Sistem Data Karyawan

Use Case Narative sistem informasi Data Karyawan	
Tujuan	Bendahara dapat data karyawan
Deskripsi	Sistem ini memungkinkan aktor untuk melakukan data karyawan
Skenario Utama	
Aktor	Bendahara
Kondisi Awal	Aktor membuka aplikasi halaman data karyawan
Aksi Aktor	Reaksi Sistem
1. Aktor memilih submenu data karyawan	Sistem akan menampilkan <i>form</i> data karyawan
2. Aktor memilih submenu tambah	Sistem akan menginput data karyawan baru
3. Aktor memilih submenu simpan	Sistem akan menyimpan data kedalam <i>database</i>
4. Aktor akan memilih submenu batal	Sistem akan menampilkan <i>form</i> awal
5. Aktor akan memilih submenu edit	Sistem akan mengedit data sesuai dengan keinginan aktor
6. Aktor akan memilih submenu hapus	Sistem akan menghapus data karyawan yang dipilih oleh aktor
Kondisi Akhir	Jika perintah sesuai maka sistem akan menampilkan sistem sesuai keinginan.

Tabel III.4.

Deskripsi Use Case Diagram Sistem Jurnal

Use Case Narative sistem informasi Jurnal	
Tujuan	Bendahara dapat mencari jurnal
Deskripsi	Sistem ini memungkinkan aktor untuk melakukan jurnal
Skenario Utama	
Aktor	Bendahara
Kondisi Awal	Aktor membuka aplikasi halaman jurnal
Aksi Aktor	Reaksi Sistem
1. Aktor memilih submenu jurnal	Sistem akan menampilkan <i>form</i> jurnal
2. Aktor memilih submenu cari	Sistem akan menampilkan sesuai dengan pencarian
Kondisi Akhir	Jika perintah sesuai maka sistem akan menampilkan sistem sesuai keinginan.

Tabel III.5.

Deskripsi Use Case Diagram Sistem Transaksi

Use Case Narative sistem informasi Transaksi	
Tujuan	Bendahara dapat transaksi
Deskripsi	Sistem ini memungkinkan aktor untuk melakukan transaksi
Skenario Utama	
Aktor	Bendahara
Kondisi Awal	Aktor membuka aplikasi halaman transaksi
Aksi Aktor	Reaksi Sistem
1. Aktor memilih submenu transaksi	Sistem akan menampilkan <i>form</i> transaksi
2. Aktor memilih submenu baru	Sistem akan menginput transaksi baru
3. Aktor memilih submenu simpan	Sistem akan menyimpan data kedalam database
4. Aktor akan memilih submenu batal	Sistem akan menampilkan <i>form</i> awal
5. Aktor akan memilih submenu hapus	Sistem akan menghapus transaksi yang dipilih oleh aktor
Kondisi Akhir	Jika perintah sesuai maka sistem akan menampilkan sistem sesuai keinginan.

Tabel III.6.

Deskripsi Use Case Diagram Sistem Laporan

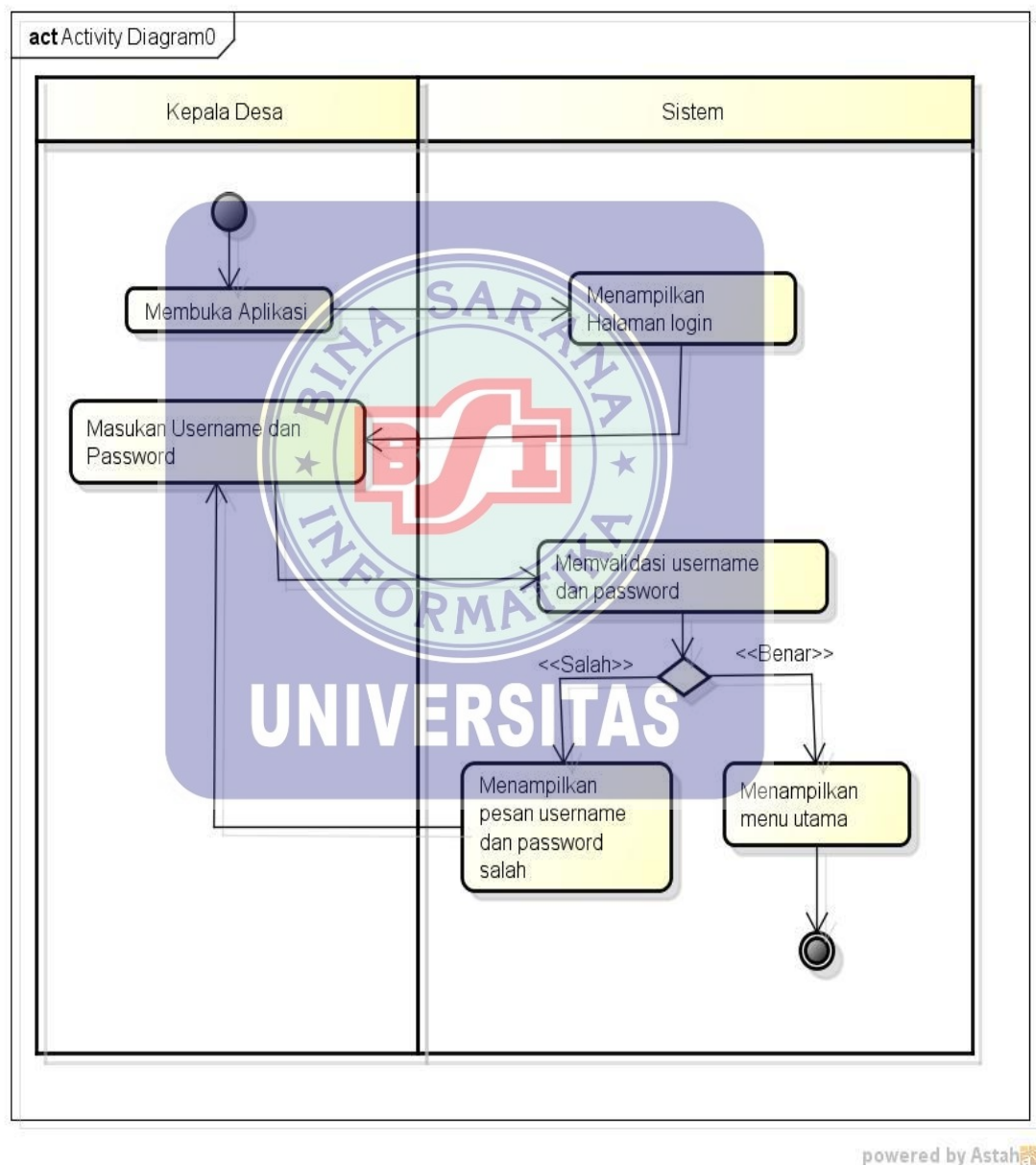
Use Case Narative sistem informasi Laporan	
Tujuan	Bendahara dapat cetak laporan
Deskripsi	Sistem ini memungkinkan aktor untuk melakukan cetak laporan
Skenario Utama	
Aktor	Bendahara
Kondisi Awal	Aktor membuka aplikasi halaman cetak laporan
Aksi Aktor	Reaksi Sistem
1. Aktor memilih submenu laporan	Sistem akan menampilkan <i>form</i> laporan
2. Aktor memilih submenu cetak	Sistem akan menampilkan laporan sesuai pilihan
Kondisi Akhir	Jika perintah sesuai maka sistem akan menampilkan sistem sesuai keinginan.

1.3.3. Activity Diagram Sistem Usulan

Activity Diagram adalah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses.

1. Activity Diagram Sistem Usulan Login

Berikut ini adalah proses bendahara menjalankan proses login:



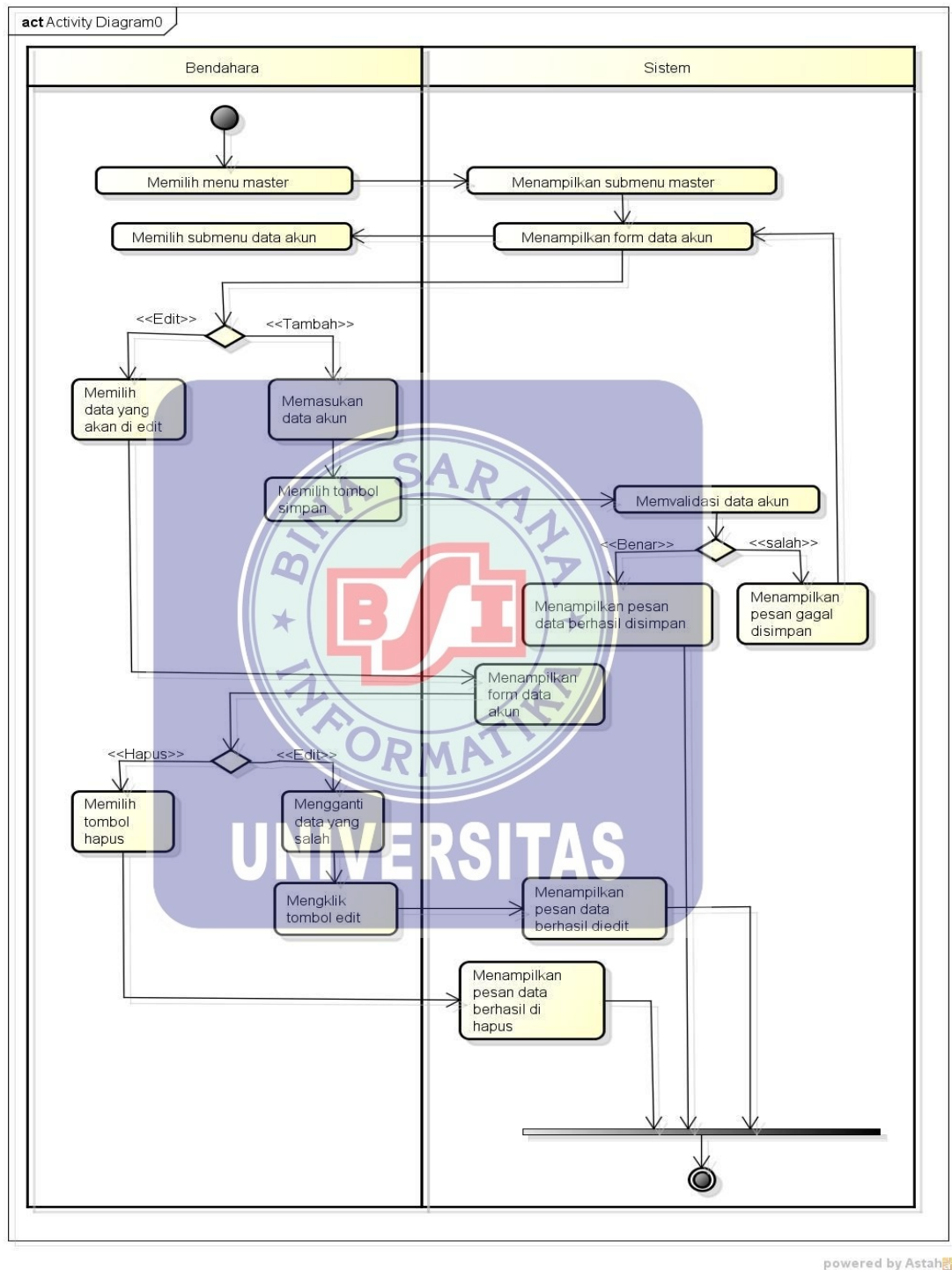
Sumber: Hasil Rancangan (2019)

Gambar III.7.

Activity Diagram Sistem Usulan login

2. Activity Diagram Sistem Usulan Data Akun

Berikut ini adalah proses bendahara masuk ke Data Akun:



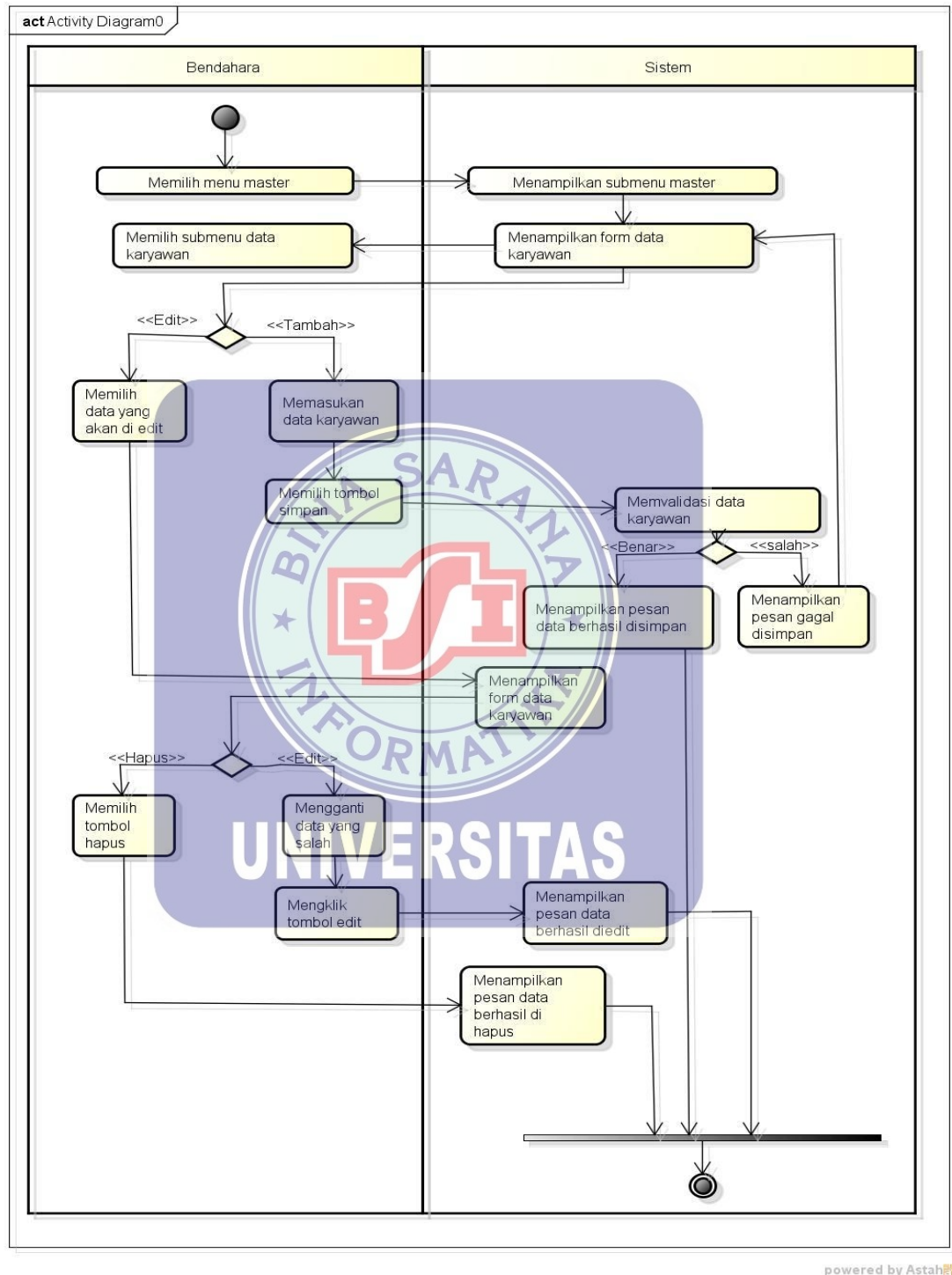
Sumber: Hasil Rancangan (2019)

Gambar III.8.

Activity Diagram Sistem Usulan Data Akun

3. Activity Diagram Sistem Usulan Data Karyawan

Berikut ini adalah proses bendahara masuk ke Data Karyawan:



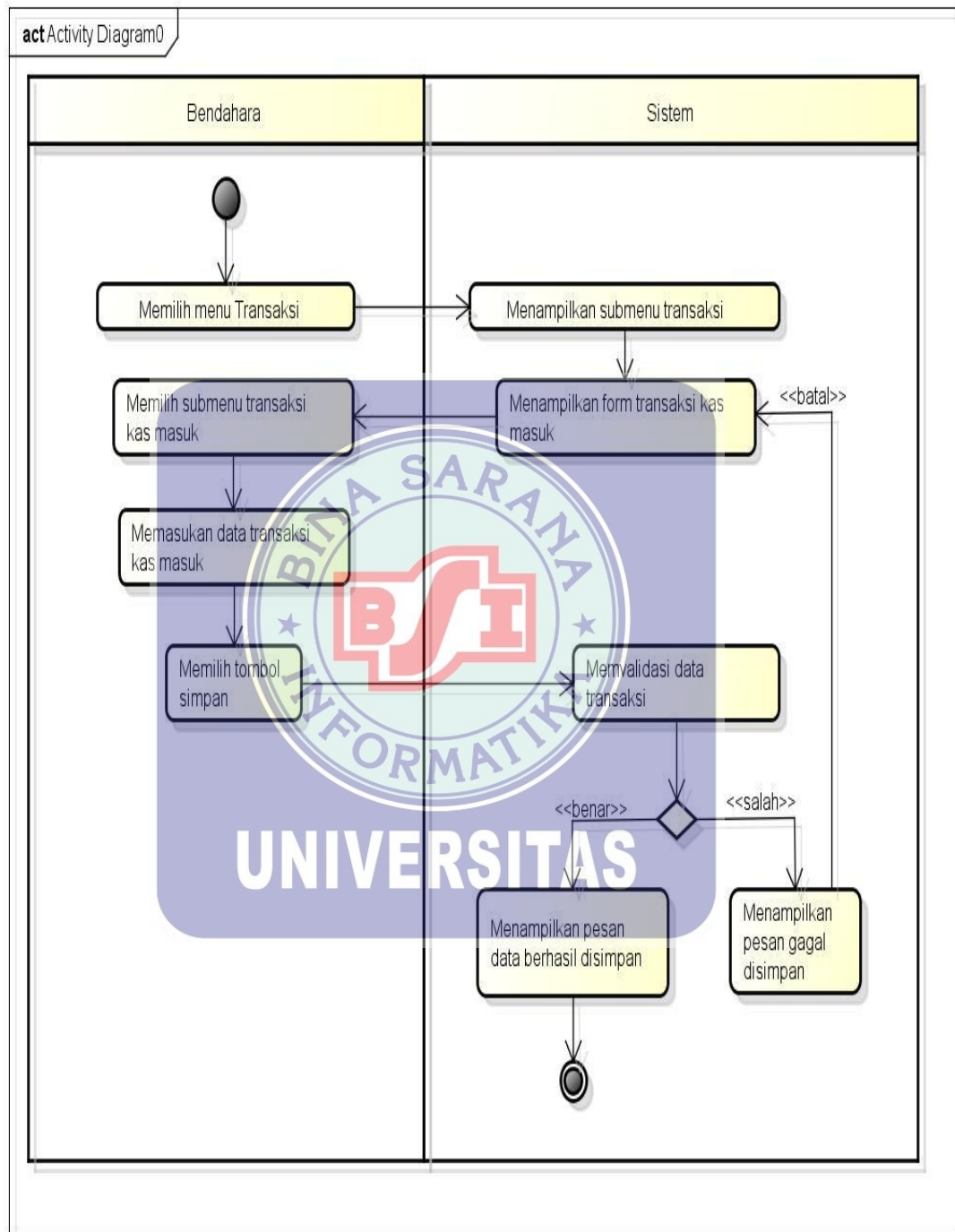
Sumber: Hasil Rancangan (2019)

Gambar III.9.

Activity Diagram Sistem Usulan Data Karyawan

4. Activity Diagram Sistem Usulan Transaksi Kas Masuk

Berikut ini adalah proses bendahara masuk kesubmenu Transaksi Kas Masuk:



powered by Astah

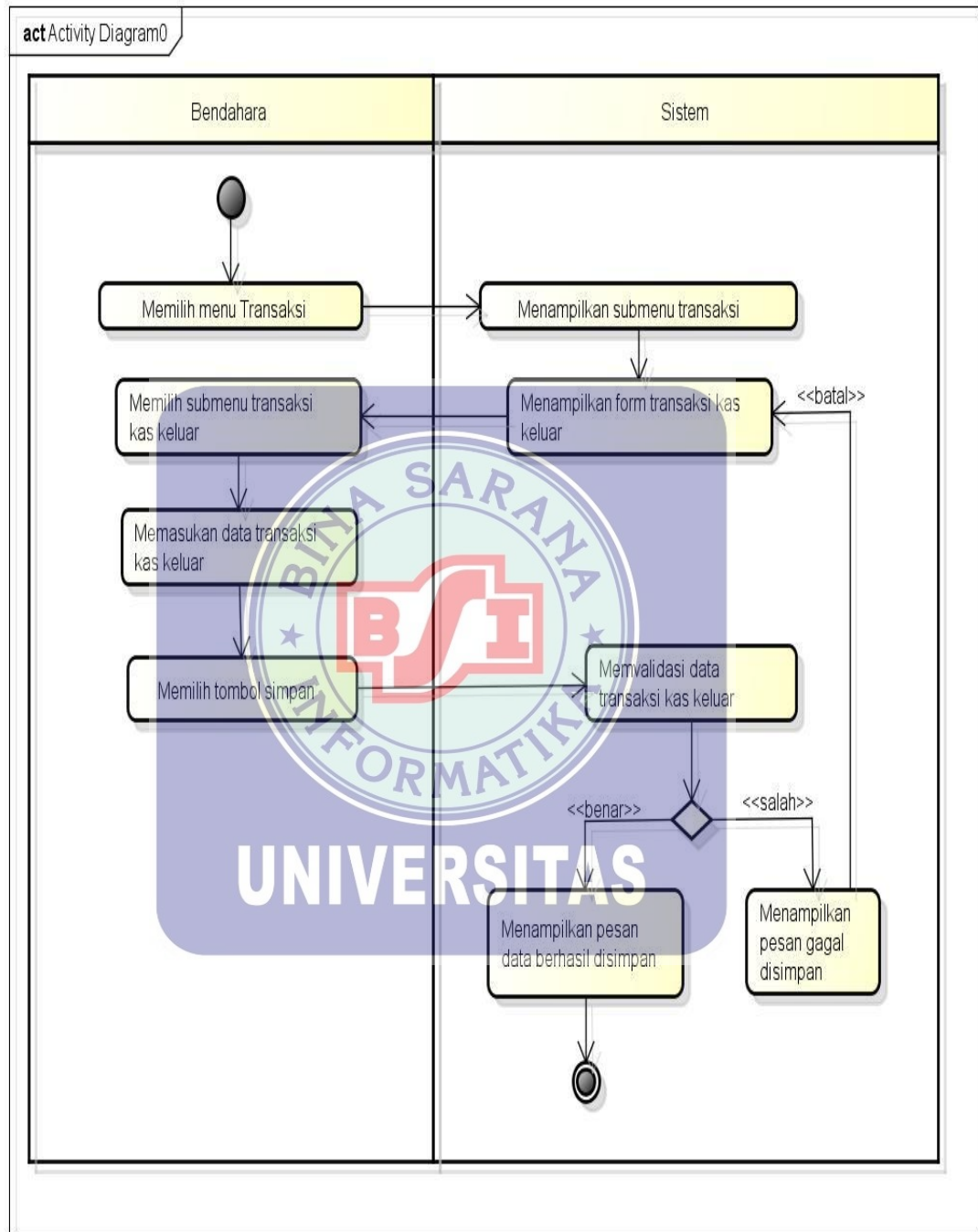
Sumber: Hasil Rancangan (2019)

Gambar III.10.

Activity Diagram Sistem Usulan Transaksi Kas Masuk

5. Activity Diagram Sistem Usulan Transaksi Kas Keluar

Berikut ini adalah proses bendahara masuk ke submenu Transaksi Kas Keluar:



powered by Astah

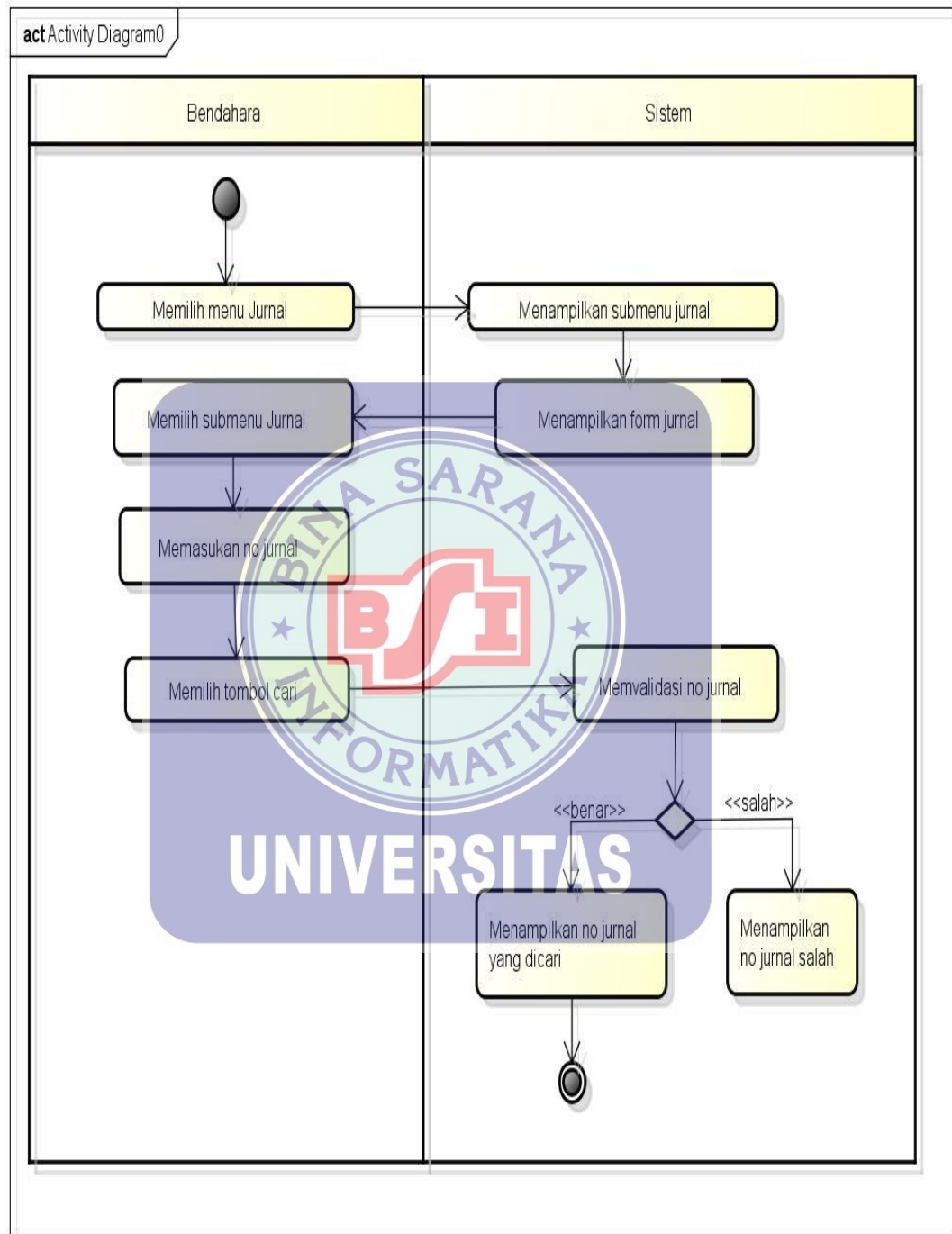
Sumber: Hasil Rancangan (2019)

Gambar III.11.

Activity Diagram Sistem Usulan Transaksi Kas Keluar

6. Activity Diagram Sistem Usulan Jurnal

Berikut ini adalah proses bendahara masuk kedata Jurnal:



powered by Astah

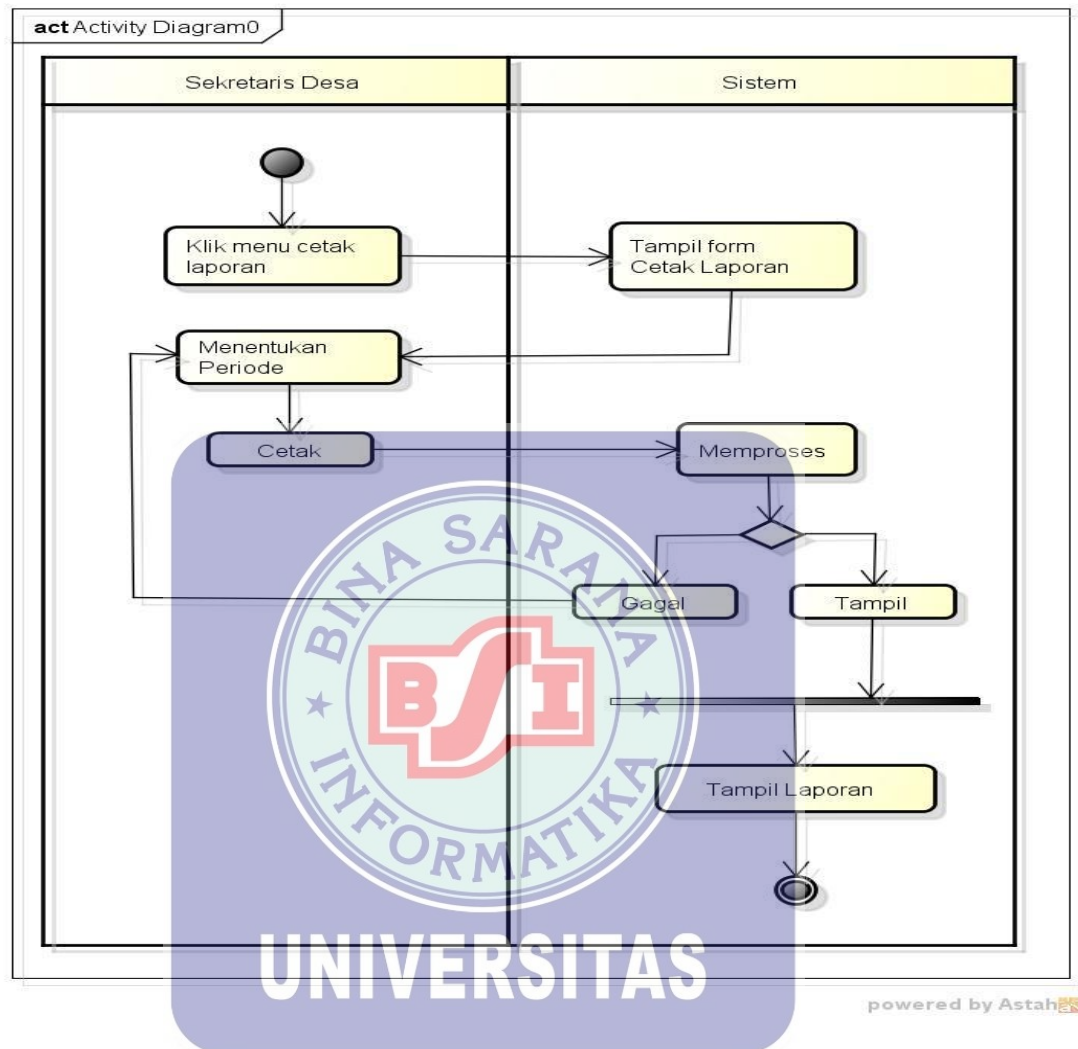
Sumber: Hasil Rancangan (2019)

Gambar III.12.

Activity Diagram Sistem Usulan Jurnal

7. Activity Diagram Sistem Usulan Cetak Laporan

Berikut ini adalah proses sekretaris masuk ke Cetak Laporan:



Gambar III.12.

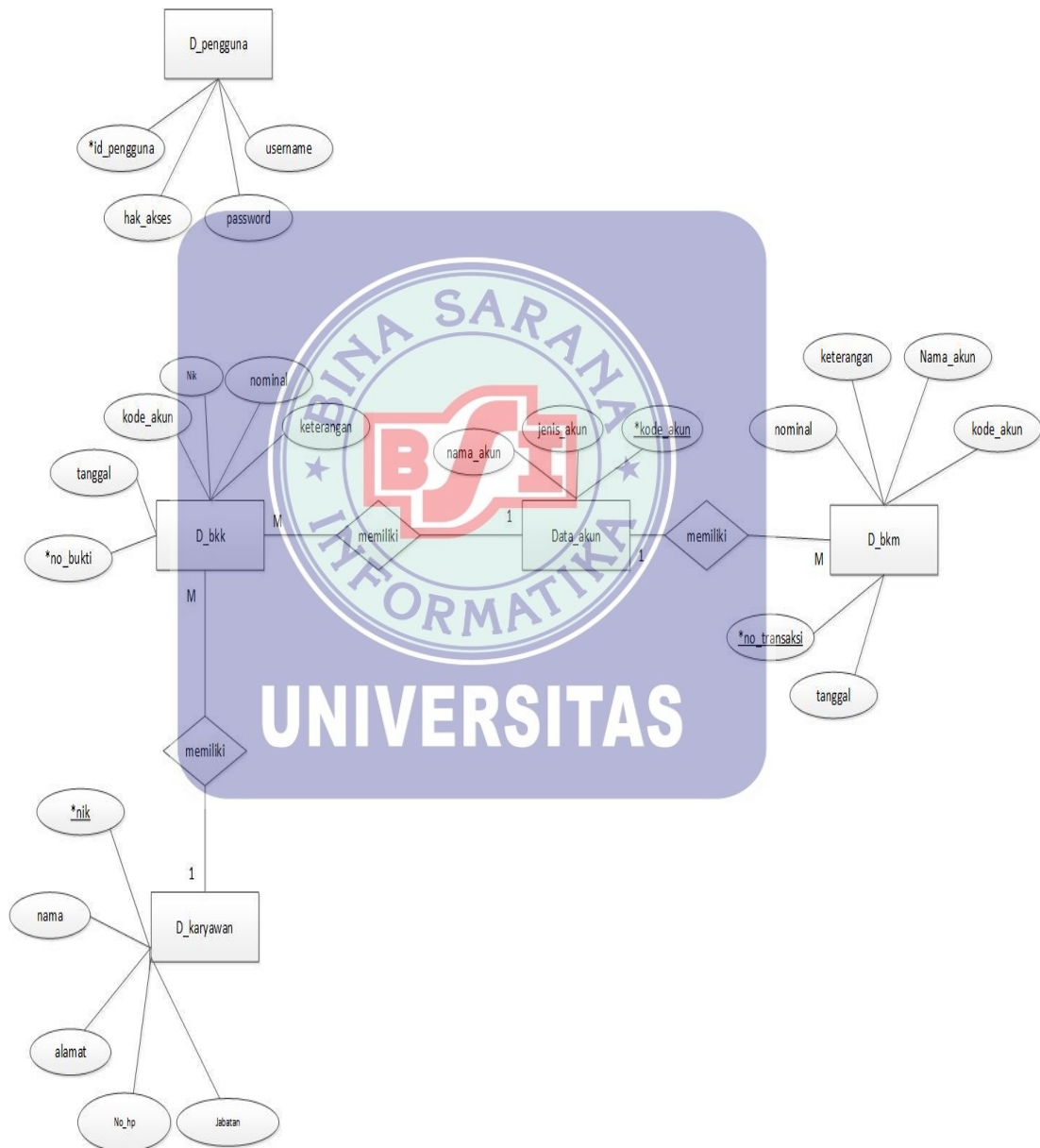
Activity Diagram Sistem Usulan Cetak Laporan

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 Ciasembaru:



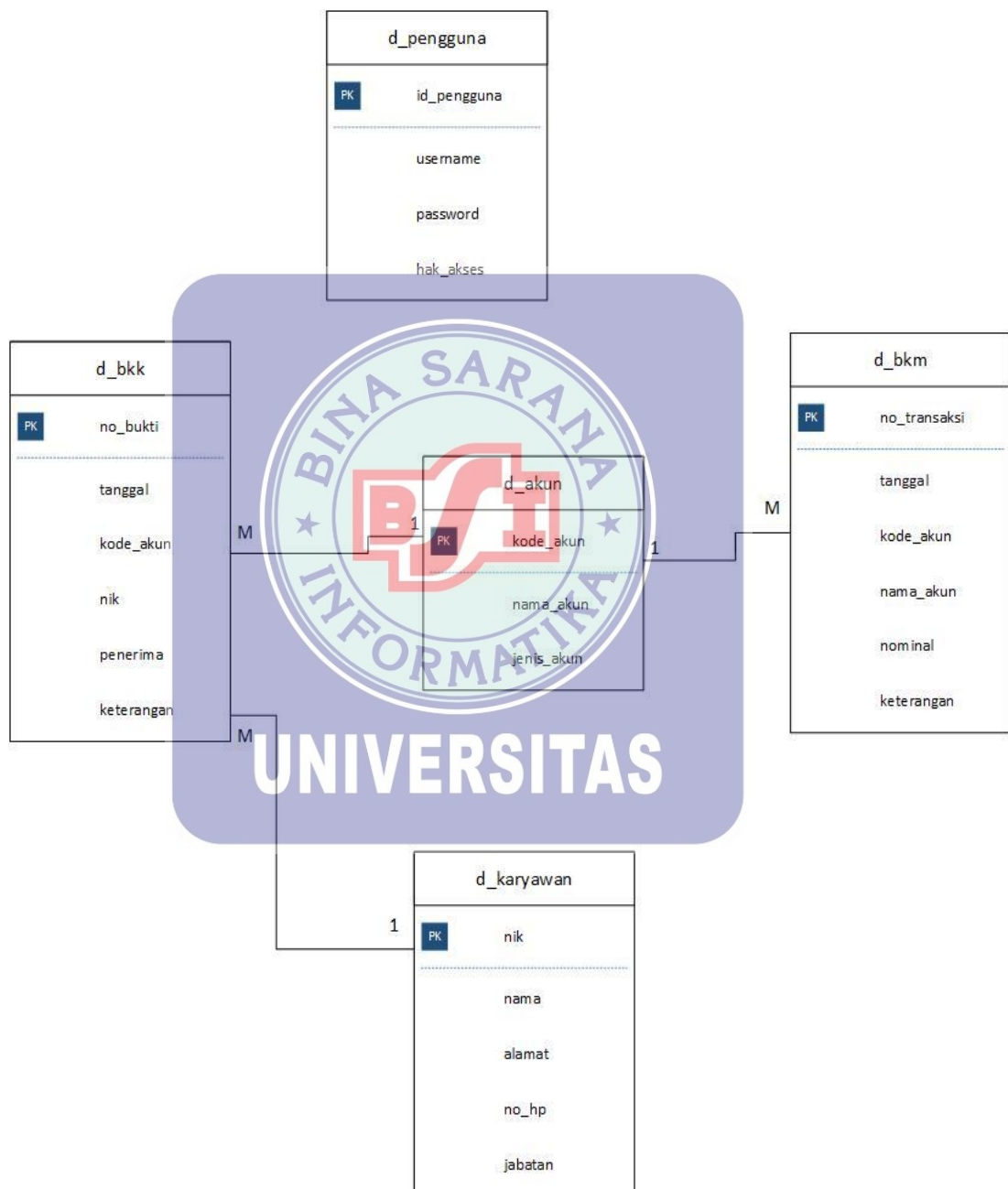
Sumber: Hasil Rancangan (2019)

Gambar III.13.

Entity Relationship Diagram Pengelolaan Kas

3.4.2. 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 Ciasembaru dengan menggunakan LRS.



Sumber: Hasil Rancangan (2019)

Gambar III.14.

Logical Record Structure Pengelolaan Kas

3.4.3. Spesifikasi File

Adapun spesifikasi file yang penulis gunakan dalam pelaporan kas aruskas pada Kantor Desa Ciasembaru adalah sebagai berikut :

1. Spesifikasi File Pengguna

Nama Database : pdb_erry

Nama File : Tabel Pengguna

Akronim : d_pengguna.sql

Fungsi : Untuk mengolah login

Tipe File : *File master*

Akses File : *Random*

Record Size : 75 Byte

Field Key : id_pengguna

Tabel III.12.

Tabel Pengguna

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Id pengguna	id_pengguna	int	10	Primary Key
2	Nama lengkap	nama_lengkap	Varchar	20	
3	User name	user_name	Varchar	10	
4	Password	password	Varchar	35	
5	Hak akses	hak_akses	Enum	-	

2. Spesifikasi File Data Akun

Nama Database : pdb_erry

Nama File : Tabel Data Akun

Akronim : d_akun.sql

Fungsi : Untuk mengolah data akun

Tipe File : *File Master*

Akses File : *Random*

Record Size : 45Byte

Field Key : Kode_akn

Tabel III.7.

Tabel Data Akun

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Kode Akun	Kode_akn	Varchar	10	Primary Key
2	Nama Akun	Nama_akn	Varchar	20	
3	Jenis Akun	Jenis_akn	Varchar	15	

3. Spesifikasi File Transaksi Kas Keluar

Nama Database : pdb_erry

Nama File : Tabel Transaksi Kas Keluar

Akronim : d_bkk.sql

Fungsi : Untuk mengolah Transaksi Kas Keluar

Tipe File : File Transaksi

Akses File : Random

Record Size : 91 Byte

Field Key : no_bukti

Tabel III.8.

Tabel Transaksi Kas Keluar

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Nomor Bukti	no_bukti	Varchar	15	Primary Key
2	Tanggal	tanggal	Date	-	
3	Kode akun	kode_akun	Varchar	10	
4	NIK	nik	Varchar	15	
5	Penerima	penerima	Varchar	25	
6	Total Dana	total_dana	Int	11	
7	Keterangan	keterangan	Varchar	15	

4. Spesifikasi File Transaksi Kas Masuk

Nama Database : pdb_erry

Nama File : Tabel Transaksi Kas Masuk

Akronim : d_bkm.sql

Fungsi : Untuk mengolah Transaksi Kas Masuk

Tipe File : *File Transaksi*

Akses File : *Random*

Record Size : 70Byte

Field Key : no_transaksi

Tabel III.9.

Tabel Transaksi Kas Masuk

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Nomor Transaksi	no_transaksi	Varchar	15	Primary Key
2	Tanggal	tanggal	Date	-	
3	Kode akun	kode_akun	Varchar	15	
4	Nominal	nominal	int	15	
5	Keterangan	Keterangan	Varchar	25	

5. Spesifikasi File Jurnal

Nama Database : pdb_erry

Nama File : Tabel Jurnal

Akronim : d_jurnal.sql

Fungsi : Untuk mengolah Jurnal

Tipe File : *File Transaksi*

Akses File : *Random*

Record Size : 88Byte

Field Key : no_jurnal

Tabel III.10.

Tabel Jurnal Umum

No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	No jurnal	no_jurnal	<i>varchar</i>	15	<i>Primary Key</i>
2	Tanggal	tanggal	<i>date</i>	-	
3	Id transaksi	id_trans	<i>varchar</i>	11	
4	Kode akun	kode_akun	<i>varchar</i>	15	
5	Debit	debit	<i>Int</i>	11	
6	Kredit	kredit	<i>Int</i>	11	
7	Keterangan	keterangan	<i>varchar</i>	25	

6. Spesifikasi File Data Karyawan

Nama Database : pdb_erry

Nama File : Tabel Data Karyawan

Akronim : d_karyawan.sql

Fungsi : Untuk mengolah Data Karyawan

Tipe File : *File Master*

Akses File : *Random*

Record Size : 76Byte

Field Key : Nik

Tabel III.11.

Tabel Data Karyawan

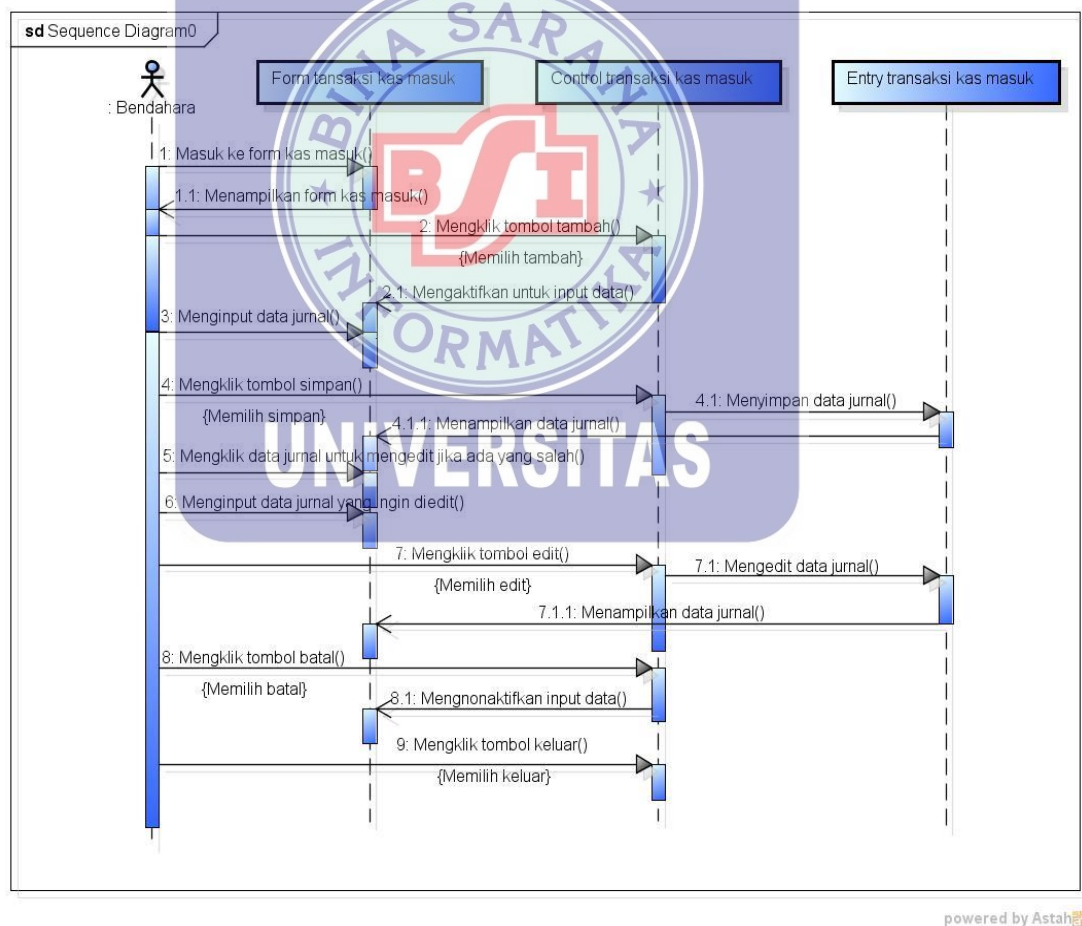
No	Elemen Data	Akronim	Tipe	Panjang	Keterangan
1	Nik	nik	<i>Varchar</i>	15	<i>Primary Key</i>
2	Nama Karyawan	nama	<i>Varchar</i>	25	
3	Alamat	alamat	<i>Varchar</i>	25	
4	Nomor Handphone	no_hp	<i>int</i>	11	

1.4.4. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian untuk menghasilkan output tertentu. *Sequence Diagram* digunakan untuk memperlihatkan aliran fungsionalitas yang meliputi tiap obyek pada Sistem Informasi Akuntansi pengelolaan Kas Pada Desa Ciasembaru. Adapun *Sequence Diagram* pada pengelolaan kas sebagai berikut:

1. Sequence Diagram Transaksi Kas Masuk

Berikut ini adalah proses untuk masuk ke Transaksi Kas Masuk



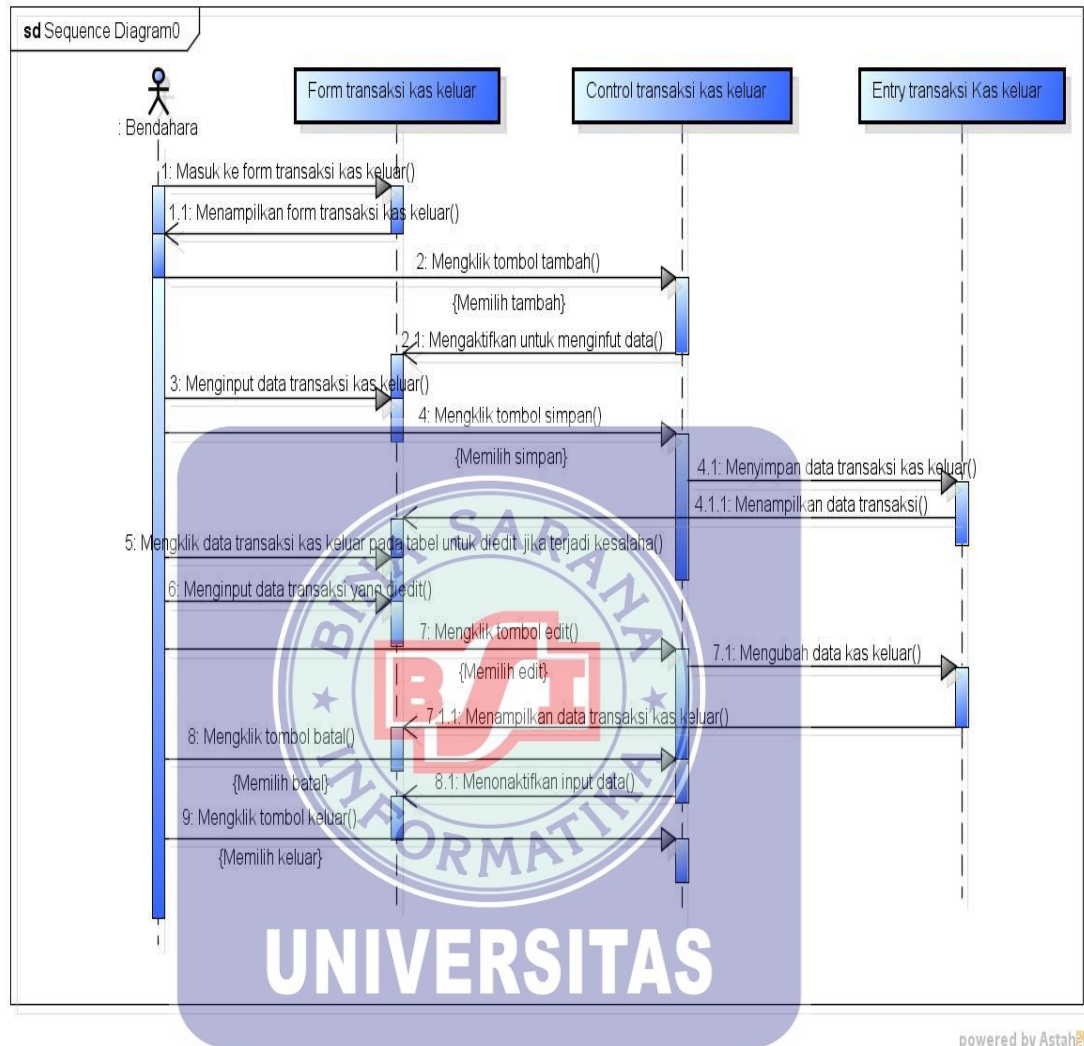
Sumber: Hasil Rancangan (2019)

Gambar III.16.

Sequence Diagram Sistem Usulan Transaksi Kas Masuk

2. Sequence Diagram Transaksi Kas Keluar

Berikut ini adalah proses untuk masuk ke Transaksi Kas Keluar

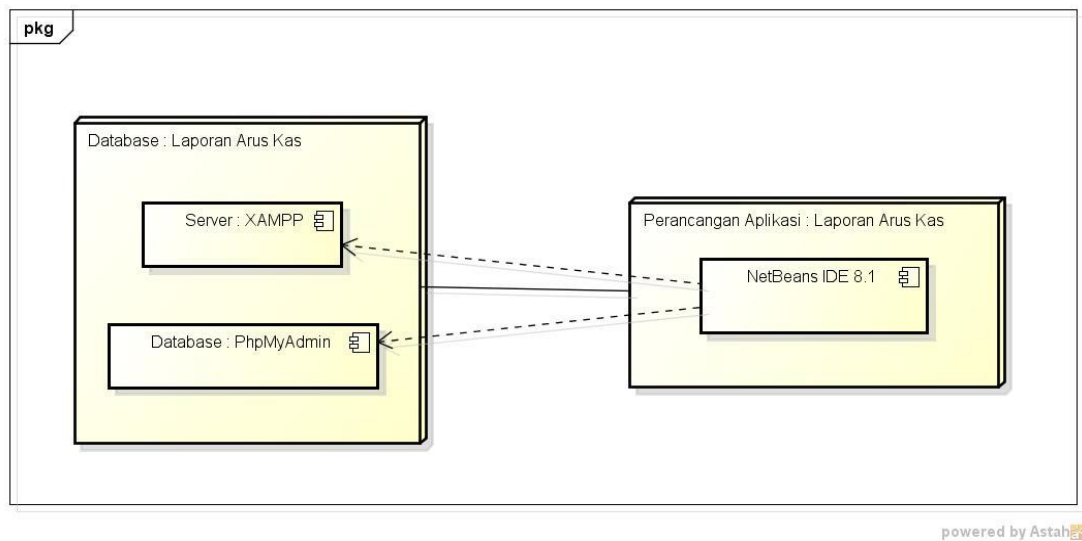


Gambar III.17.

Sequence Diagram Sistem Usulan Transaksi Kas Keluar

3.4.5. Deployment Diagram

Deployment Diagram adalah menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi pada sebuah komputer server dimana di dalamnya sudah terdapat php server dan MySQL sebagai DBMS. Rancangan *Deployment Diagram* dalam sistem laporan keuangan pada Desa Ciasembaru adalah sebagai berikut :

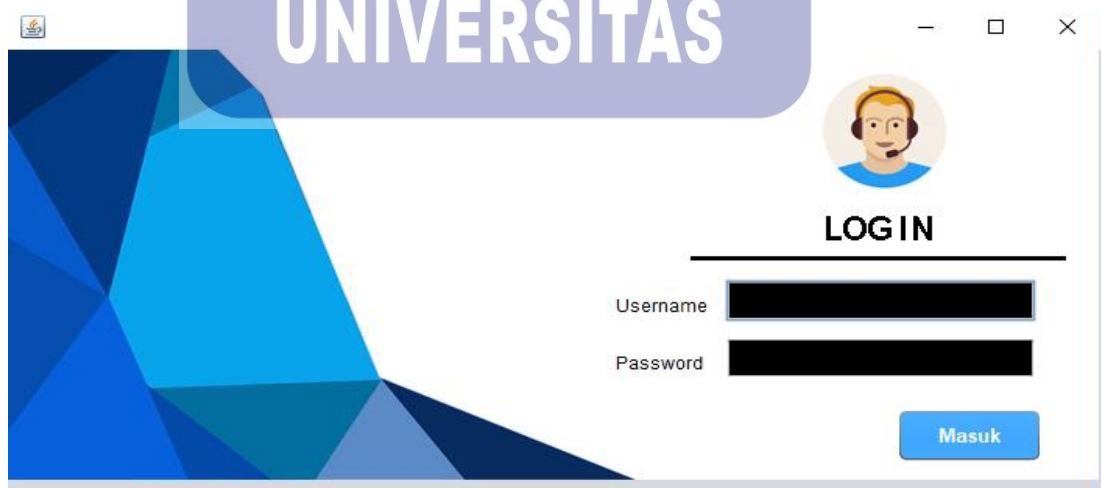


Sumber: Hasil Rancangan (2019)

Gambar III.18.
Deployment Diagram Pengelolaan Kas

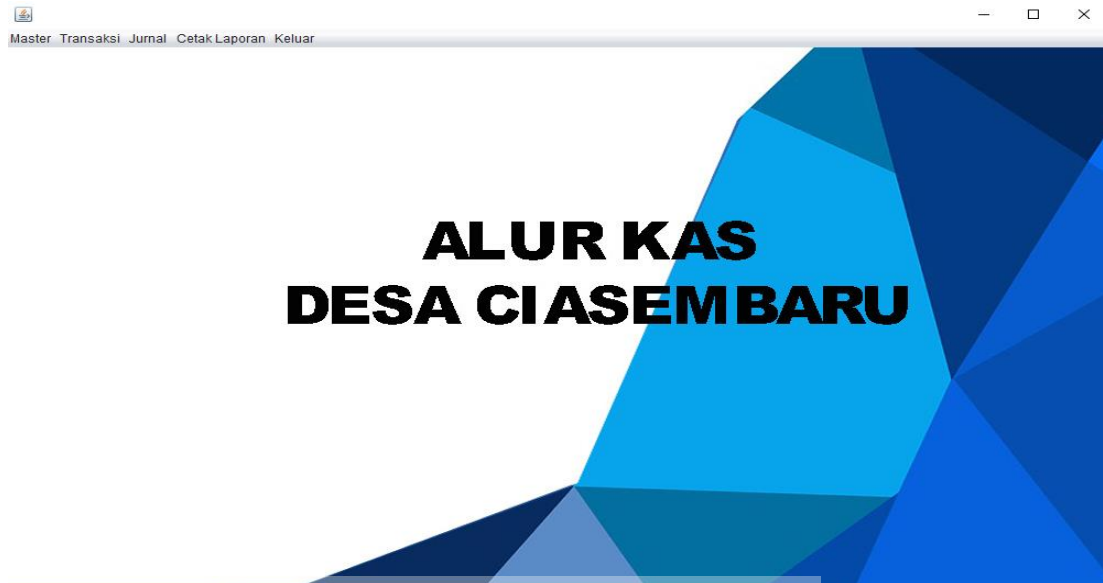
3.4.6. User Interface

Berikut ini merupakan tampilan perancangan sistem pelaporan pengelolaan kas desapada Kantor Desa Ciasembaru, yaitu:



Sumber: Hasil Rancangan (2019)

Gambar III.19.
Rancangan Tampilan Menu Login



Sumber: Hasil Rancangan (2019)

Gambar III.20.

Rancangan Tampilan Menu Utama

Kode Akun	Nama Akun	Jenis Akun
1-0001	Bayar Pajak	Harta
5-0001	Biaya Transport	Beban

Sumber: Hasil Rancangan (2019)

Gambar III.21.

Rancangan Tampilan Data Akun

DATA KARYAWAN

NIK Alamat

Nama No HP

Jabatan

Cari NIK

Nik	Nama	Alamat	No HP	Jabatan
123	Andin	Ciasem	628765	Sekretaris
456	Erry	subang	628987	Bendahara
567	Hopip	Kebon cau	628456	Kepala Desa

Sumber: Hasil Rancangan (2019)

Gambar III.22.

Rancangan Tampilan Data Karyawan

KAS MASUK

No Kas Masuk

Kode Akun

Tanggal

Nominal

Keterangan

NO Kas Masuk	Tanggal	Kode Akun	Nominal	Keterangan
KM0001	2019-08-10	30001	392445300	Alokasi Dana ...

Sumber: Hasil Rancangan (2019)

Gambar III.23.

Rancangan Tampilan Transaksi Kas Masuk

NO Ka...	Tanggal	Kode A...	NIK	Peneri...	Nominal	Ketera...
KK0001	2019-0...	50002	567	Hopip	29332...	gajih

Sumber: Hasil Rancangan (2019)

Gambar III.24.

Rancangan Tampilan Transaksi Kas Keluar

No Jurnal	Tanggal	ID Transaksi	Kode Akun	Nama Akun	debit	kredit	Keterangan
15	2019-08-10	KM0001	1101	Kas	300000000	0	Alokasi Da...
16	2019-08-10	KM0001	30001	Alokasi Da...	0	300000000	Alokasi Da...
17	2019-08-10	KK0001	1101	Kas	0	290000000	Biaya Gajih
18	2019-08-10	KK0001	50002	Tunjangan ...	290000000	0	Biaya Gajih

Sumber: Hasil Rancangan (2019)

Gambar III.25.

Rancangan Tampilan Jurnal

Sumber: Hasil Rancangan (2019)

Gambar III.26.

Rancangan Tampilan Cetak Laporan

3.5.1. Generasi Kode (Code Generation)

Pengkodean merupakan proses perancangan desain dengan menggunakan bahasa pemrograman *java* guna menunjang rancang bangun sistem, penulis akan menggunakan aplikasi *NetbeansIDE* 8.1 berbasis dekstop untuk mempermudah pengembangan *software*. Berikut ini adalah pengkodean (*coding*) data transaksi kas keluar pada Desa Ciasembaru:

```
package allurkas;

import java.io.File;

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.HashMap;
import java.util.Locale;
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;
    @erry
public class kas_out extends javax.swing.JDialog {
    String kode,penerima,keterangan;
    java.util.Date tglsekarang = new java.util.Date();
        private final SimpleDateFormat smpdtfmt = new
SimpleDateFormat("dd/MM/yyyy", Locale.getDefault());
        private final String tanggal = smpdtfmt.format(tglsekarang);
    int jml;
    Date tgl;
    ResultSet resultSet;
    Statement statement;
    String kode_akun, ubah;
    int count, jml_awal;

```

```

        JOptionPane.showMessageDialog(null, "Error " + e);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(kas_in.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void SelectKategori() {
    try {
        Connection conn = konek.openkoneksi();
        java.sql.Statement stm = conn.createStatement();
        java.sql.ResultSet rs = stm.executeQuery("SELECT * FROM d_karyawan
ORDER BY nik");
        cmbkode1.addItem("Pilih");
        while(rs.next()){
            cmbkode1.addItem(rs.getString("nik")); }
        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() {
    jLabel6 = new javax.swing.JLabel();
    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<String>();
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();
no_jurnal = new javax.swing.JTextField();
kredit = new javax.swing.JTextField();
jLabel7 = new javax.swing.JLabel();
cmbkode1 = new javax.swing.JComboBox<String>();
nik = new javax.swing.JTextField();
jLabel8 = new javax.swing.JLabel();
ppenerima = new javax.swing.JTextField();
jTextField1 = new javax.swing.JTextField();
jScrollPane1 = new javax.swing.JScrollPane();
tblanggaran = new javax.swing.JTable();
setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
jLabel6.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel6.setText("KAS KELUAR");
jPanel1.setBackground(new java.awt.Color(51, 153, 255));
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 Keluar");
DTtgl.setEditable(false);
DTtgl.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        DTtglActionPerformed(evt); } });
btnbaru.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnbaruActionPerformed(evt); } });
kredit.setText("0");
jLabel7.setText("NIK");
cmbkode1.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        cmbkode1ItemStateChanged(evt);    } });
cmbkode1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmbkode1ActionPerformed(evt);    } });
nik.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        nikActionPerformed(evt); } });
jLabel8.setText("Penerima");
ppenerima.setEditable(false);
javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

```

```

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

    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
                    .addComponent(jLabel1)
                    .addComponent(jLabel4)
                    .addComponent(jLabel2)
                    .addComponent(jLabel7))
                    .addGap(57, 57, 57)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(btnbaru)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(btnsimpan)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(btnhapus)
                        .addGap(0, 0, Short.MAX_VALUE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
TRAILING, false)
                            .addComponent(cmbkode1, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .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))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

                .addGroup(jPanel1Layout.createSequentialGroup())
                .addComponent(txtkode,
javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(kredit,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup())
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(no_jurnal,
javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(jPanel1Layout.createSequentialGroup())
                .addComponent(nik,
javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(0, 0, Short.MAX_VALUE)))
        .addContainerGap()))
        .addGroup(jPanel1Layout.createSequentialGroup())
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

                .addComponent(jLabel3)
                .addComponent(jLabel5)
                .addComponent(jLabel8))

```



```

        .addGap(71, 71, 71)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
            .addComponent(txtnominal,
javax.swing.GroupLayout.PREFERRED_SIZE, 204,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 224,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(ppenerima,
javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 72,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap(49, 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)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
                    .addComponent(txtno,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(no_jurnal,
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)
        .addComponent(kredit, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
        .addComponent(jLabel2)
        .addComponent(DTtgl, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
        .addComponent(jLabel7)
        .addComponent(cmbkode1,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(nik))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
    .addComponent(jLabel8)
    .addComponent(ppenerima,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
.addGap(14, 14, 14)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
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.Alignment.
LEADING)
    .addComponent(jLabel5)
    .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 48,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

```

```

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
    .addComponent(btnbaru)
    .addComponent(btnsimpan)
    .addComponent(btnhapus))
.addGap(54, 54, 54))
);
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);
javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .add(btnbaru)
            .add(btnsimpan)
            .add(btnhapus))
        .addGap(54, 54, 54))
    .addGroup(layout.createSequentialGroup()
        .add(tblanggaran)
        .addGap(54, 54, 54))
);

```

```

        .addComponent(jScrollPane1,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 410, Short.MAX_VALUE)
        .addGroup(layout.createSequentialGroup()
            .addGap(138, 138, 138)
            .addComponent(jLabel6)))
        .addContainerGap()
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel6)
                .addGap(18, 18, 18)
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(18, 18, 18)
                .addComponent(jScrollPane1,
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*FROM d_akun
WHERE kode_akun='"+nm_kategori+"'");
        if(sql.next()){
            txtkode.setText(sql.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);
    }
} 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:
}

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

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 {
        simpan();
        simpanjurnal();
        tabelmodel();
        JOptionPane.showMessageDialog(null, "Berhasil menyimpan data
transaksi");
    }
}

```

```

try {
    HashMap hash = new HashMap();
    hash.put("no", txtno.getText());
    File file = new File("src/Laporan/bukti_kas_out.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();
bersih();
bersih();
}

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();
        setkodejurnal();
    }
}

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 cmbkode1ItemStateChanged(java.awt.event.ItemEvent evt) {
        // TODO add your handling code here:
    }
    private void cmbkode1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String nm_kategori = cmbkode1.getSelectedItem().toString();
        if(!nm_kategori.equals("")){
            try {
                Connection conn = konek.openkoneksi();
                java.sql.Statement stm = conn.createStatement();
                java.sql.ResultSet sql = stm.executeQuery("SELECT * FROM
d_karyawan WHERE nik='"+nm_kategori+"'");
                if(sql.next()){
                    ppenerima.setText(sql.getString("nama"));
                }
                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("");
        }
    }
    private void nikActionPerformed(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 (InstantiationException ex) {
    java.util.logging.Logger.getLogger(kas_out.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(kas_out.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(kas_out.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
}
//</editor-fold>

/* Create and display the dialog */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        kas_out dialog = new kas_out(new javax.swing.JFrame(), true);
        dialog.addWindowListener(new java.awt.event.WindowAdapter() {
            @Override
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
    }
});

```

```

        }
    });
    dialog.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.JComboBox<String> cmbkod;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField kredit;
private javax.swing.JTextField nik;
private javax.swing.JTextField no_jurnal;
private javax.swing.JTextField ppenerima;
private javax.swing.JTable tblanggaran;
private javax.swing.JTextArea txtket;
private javax.swing.JTextField txtno;
private javax.swing.JTextField txtnominal;

// End of variables declaration
public int simpan(){
    String sql="insert into d_bkk "+
        "values"+
        "("+txtno.getText()+", " +
        """+DTtgl.getText()+", "+
        """+cmbkode.getSelectedItem()+", "+

```

```

        ""+cmbkode1.getSelectedItemAt()+", "+
        ""+ppenerima.getText()+", "+
        ""+txtnominal.getText()+", "+
        ""+txtket.getText()+" "+ ")";
    conn.simpanData(sql);
    return 0;
}

public int simpanjurnal(){
    String sql="insert into d_jurnal "+
        "values"+
        "("+no_jurnal.getText()+", " +
        ""+DTtgl.getText()+", "+
        ""+txtno.getText()+", "+
        ""+cmbkode.getSelectedItemAt()+", "+
        ""+txtkode.getText()+", "+
        ""+kredit.getText()+", "+
        ""+txtnominal.getText()+", "+
        ""+txtket.getText()+" "+ ")";
    conn.simpanData(sql);
    return 0;
}

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

public void tabelmodel(){
    tb =new DefaultTableModel();
    tb.addColumn("NO Kas Keluar");
    tb.addColumn("Tanggal");
    tb.addColumn("Kode Akun");
    tb.addColumn("NIK");
    tb.addColumn("Penerima");

```

```

        tb.addColumn("Nominal");
    void bersih(){
        txtkode.setText("");
        txtket.setText("");
        txtnominal.setText("");
        txtno.setText("KK0001");
    }
    }catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
}

void setkodejurnal(){
    try {
        ResultSet res = conn.ambilData("select * from d_jurnal order by no_jurnal
desc");
        if (res.next()) {
            String nofak = res.getString("no_jurnal").substring(3);
            String AN = "" + (Integer.parseInt(nofak) + 1);

            String Nol = "";
            {Nol = "00";}
            else if(AN.length()==2)
            {Nol = "0";}
            else if(AN.length()==3)
            {Nol = "";}

            no_jurnal.setText("JRN" + Nol + AN);
        } else {
            no_jurnal.setText("JRN001");
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }
}
}

```

3.5.2. Blackbox Testing

Blackbox testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Adapun *Blackbox testing* pada pengelolaan kas sebagai berikut:

Tabel III.13.

Tabel Blackbox Testing

No	Skenario Pengujian	Test Case	Hasil Yang di Harapkan	Hasil Pengujian	Kesimpulan
1	Klik tombol SIMPAN ketika sedang mengisi data di textfield	Kode Akun : secara otomatis akan terisi TGL : secara otomatis akan terisi Kode akun, Kode kas masuk, Nama akun, Jenis akun, nominal, keterangan	Sistem akan menampilkan "Berhasil menyimpan data transaksi"	Sesuai Harapan	Valid
2	Klik tombol HAPUS ketika sedang mengisi data di textfield	Kode Akun : secara otomatis akan terisi TGL : secara otomatis akan terisi Kode akun, Kode kas masuk, Nama akun, Jenis akun, nominal, keterangan	Sistem akan menampilkan "Anda yakin ingin menghapus data ini"	Sesuai Harapan	Valid
3	Klik tombol BATAL	Kode Akun : secara otomatis akan terisi TGL : secara otomatis akan terisi Kode akun, Kode kas masuk, Nama akun, Jenis akun, nominal, keterangan	Tombol batal tidak akan berfungsi jika data pada textfield Data akun, data karyawan tidak di isi	Sesuai Harapan	Valid
4	Klik tombol KELUAR	Kode Akun : secara otomatis akan terisi TGL : secara otomatis akan terisi Kode akun, Kode kas masuk, Nama akun, Jenis akun, nominal, keterangan	Jika di klik tombol KELUAR akan kembali ke tampilan login	Sesuai Harapan	Valid
5	Klik tombol CETAK	Kode Akun : secara otomatis akan terisi TGL : secara otomatis akan terisi Kode akun, Kode kas masuk, Nama akun, Jenis akun, nominal, keterangan	Tombol batal tidak akan berfungsi jika data pada textfield Data akun, data karyawan, transaksi kas masuk dan kas keluar tidak di isi	Sesuai Harapan	Valid

Sumber: Hasil Rancangan (2019)

3.5.3. Spesifikasi *Hardware* dan *Software*

Komponen perangkat keras adalah terdiri dari unsur-unsur yang terdiri dari perangkat keras komputer yang digunakan untuk membantu proses kerja manusia (*brainware*) dan bersifat fisik.

Spesifikasi perangkat keras minimum yang dibutuhkan adalah sebagai berikut:

- a. Processor : Intel(R) Celeron (R) CPU N3350 @1.10Ghz
- b. Memory (RAM) : 2 GB
- c. Monitor : 14"
- d. Harddisk : 20 GB
- e. Mouse : PS/2
- f. Keyboard : 86 Keys
- g. Printer : Cannon E410

Komponen perangkat lunak adalah serangkaian unsur-unsur yang terdiri dari beberapa perangkat lunak program komputer yang digunakan untuk membantu proses kerja manusia (*brainware*) dan sifat non fisik. Adapun spesifikasi perangkat lunak yang dibutuhkan adalah sebagai berikut:

- a. Sistem Operasi : Microsoft Windows 10 Pro
- b. Sistem Aplikasi : *NetBeans* IDE 8.1
- c. Database : *MySQL*
- d. Program Pendukung : *Phpmyadmin*
- e. Connector Aplikasi : XAMPP