

BAB III

PEMBAHASAN

3.1. Tinjauan Perusahaan

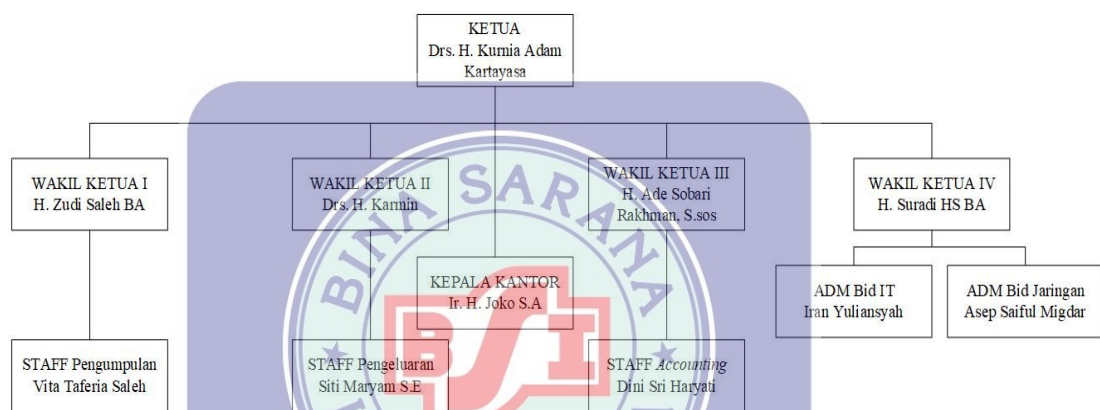
3.1.1. Sejarah Perusahaan

Badan Amil Zakat Nasional (BAZNAS) merupakan badan resmi dan satu-satunya yang dibentuk oleh pemerintah berdasarkan keputusan Presiden RI No.8 Tahun 2001 yang memiliki tugas dan fungsi menghimpun dan menyalurkan zakat, infaq, dan shadaqah (ZIS) pada tingkat nasional. Lahirnya Undang-Undang Nomor 23 Tahun 2011 tentang pengelolaan zakat semakin mengukuhkan peran BAZNAS sebagai lembaga yang berwenang melakukan pengelolaan zakat secara nasional. Dalam Undang-Undang tersebut, BAZNAS dinyatakan sebagai lembaga pemerintah nonstruktural yang bersifat mandiri dan bertanggung jawab kepada Presiden melalui menteri agama, dengan demikian, BAZNAS bersama pemerintah bertanggung jawab untuk mengawal pengelolaan zakat yang berasaskan: syariat islam, amanah, kemanfaatan, keadilan, kepastian hukum, terintegrasi dan akuntabilitas.

Pengelolaan zakat oleh lembaga awalnya hanya diatur oleh Keppres No 07/POIN/10/1968 tertanggal 31 Oktober 1968 tentang pengelolaan zakat nasional. Lembaga pengelola zakat saat itu hanya dilakukan terbatas di beberapa daerah saja seperti BAZIS DKI (1968), BAZIS Kaltim (1972), BAZIS Jawa Barat (1974) dan beberapa BUMN mendirikan lembaga zakat seperti BAMUIS BNI (1968). Lahirnya Undang-Undang Nomor 38 Tahun 1999 tentang Pengelolaan Zakat merupakan langkah awal pengelolaan zakat yang berlaku secara Nasional. Dalam Surat Keputusan ini disebutkan tugas dan fungsi BAZNAS yaitu untuk melakukan

penghimpunan dan pendayagunaan zakat. Dalam Undang-Undang tersebut diakui adanya dua jenis organisasi pengelola zakat yaitu Badan Amil Zakat (BAZ) yang dibentuk pemerintah dan Lembaga Amil Zakat (LAZ) yang dibentuk oleh masyarakat dan dikukuhkan oleh pemerintah. Adapun BAZ terdiri dari BAZNAS pusat, BAZ Propinsi, BAZ kota, BAZ Kecamatan.

3.1.2. Struktur Organisasi dan Fungsi



Sumber: Baznas Karawang

Gambar III.2.
Struktur Organisasi

Adapun tugas dan wewenang dari masing-masing bagian adalah sebagai berikut:

1. Ketua
 - a. Melaksanakan kebijakan BAZ dalam pengumpulan, penyaluran, dan pendayagunaan zakat;
 - b. Memimpin pelaksanaan program-program BAZ;
 - c. Merencanakan pengumpulan, penyaluran, dan pendayagunaan zakat;
 - d. Mempertanggung jawabkan pelaksanaan tugas kepada DPRD Tingkat Kabupaten dan Bupati atau wakil kota.

2. Wakil Ketua I

- a. Membantu ketua dalam menjalankan tugas dalam bidang pengumpulan
- b. Melaksanakan tugas lain yang diperintahkan atasan
- c. Mewakili ketua apabila ketua berhalangan dalam menjalankan bidang tugas pengumpulan

3. Wakil Ketua II

- a. Membantu ketua dalam menjalankan tugas dalam bidang pendistribusian dan pendayagunaan
- b. Melaksanakan tugas lain yang diperintahkan ketua dalam bidang pendistribusian dan pendayagunaan
- c. Mewakili ketua apabila ketua berhalangan dalam menjalankan bidang pendistribusian dan pendayagunaan

4. Wakil Ketua III

- a. Membantu ketua dalam menjalankan tugas dalam bidang perencanaan keuangan dan pelaporan
- b. Melaksanakan tugas lain yang diperintahkan ketua dalam bidang perencanaan keuangan dan pelaporan
- c. Mewakili ketua apabila ketua berhalangan dalam menjalankan bidang tugas pengumpulan

5. Wakil Ketua IV

- a. Membantu ketua dalam menjalankan tugas dalam bidang SDM kesekretariatan dan umum
- b. Melaksanakan tugas lain yang diperintahkan ketua dalam bidang SDM kesekretariatan dan umum

- c. Mewakili ketua apabila ketua berhalangan dalam menjalankan tugas dalam bidang SDM kesekretariatan dan umum
6. Kepala Kantor
- a. Melaksanakan pembinaan teknis, koordinasi dan merumuskan serta melaksanakan kebijakan
 - b. Mengkoordinasi serta memfasilitasi atas penyelenggaraan segala kegiatan
7. Staff Pengumpulan Keuangan
- a. Melakukan pendataan muzakki, harta, zakat dan lainnya
 - b. Melakukan penggalan zakat dan lainnya
 - c. Melakukan pengumpulan zakat dan lainnya, dan menyetorkan hasilnya ke bank yang ditunjuk serta menyampaikan tanda bukti penerimaan kepada bendahara
 - d. Mencatat dan membukukan hasil pengumpulan zakat dan lainnya
 - e. Mengkoordinasi kegiatan pengumpulan zakat dan lainnya.
8. Staff Keuangan Pengeluaran atau Pendistribusian
- a. Melaksanakan tata administrasi *mustahik*
 - b. Menyediakan bahan untuk pelaksanaan kegiatan BAZ serta mempersiapkan bahan laporan
9. Staff *Accounting*
- a. Menginput data jurnal akuntansi kedalam sitem
 - b. Membuat bukti transaksi
 - c. Membuat laporan
10. Admin IT
- a. Bertanggungjawab atas pemeliharaan, konfigurasi dan pengoperasian system computer

- b. Menjaga tingkat keamanan instalasi komputer

11. Admin Jaringan

- a. Menjaga dan mengoperasikan computer sistem atau jaringan
- b. Bertanggung jawab atas keamanan jaringan yang digunakan

3.2. Tinjauan Kasus

3.2.1. Proses Bisnis Sistem Berjalan

1. Proses Pengelolaan Dana Zakat, Infaq dan Shadaqah

a. Kas Masuk

Bagian penerimaan zakat membuat bukti setor zakat dan menyerahkan bukti setor zakat dan uang yang diterimanya ke bagian *accounting*, setelah menerima bukti setor zakat dan uang, *accounting* melakukan pengecekan terhadap bukti setor zakat, jika sudah sesuai maka bagian *accounting* menginput transaksi, jika ada hal yang tidak sesuai akan dikembalikan ke bagian penerimaan zakat, dan bagian penerimaan zakat merevisinya, setelah bagian penerimaan zakat merevisinya memberikannya bukti setor zakat dan uang ke bagian *accounting* dan bagian *accounting* menerima dan mengeceknya kembali jika sudah sesuai maka bagian *accounting* menginputnya ke *microsoft excel* dan membuat laporan penerimaan kas

b. Kas Keluar

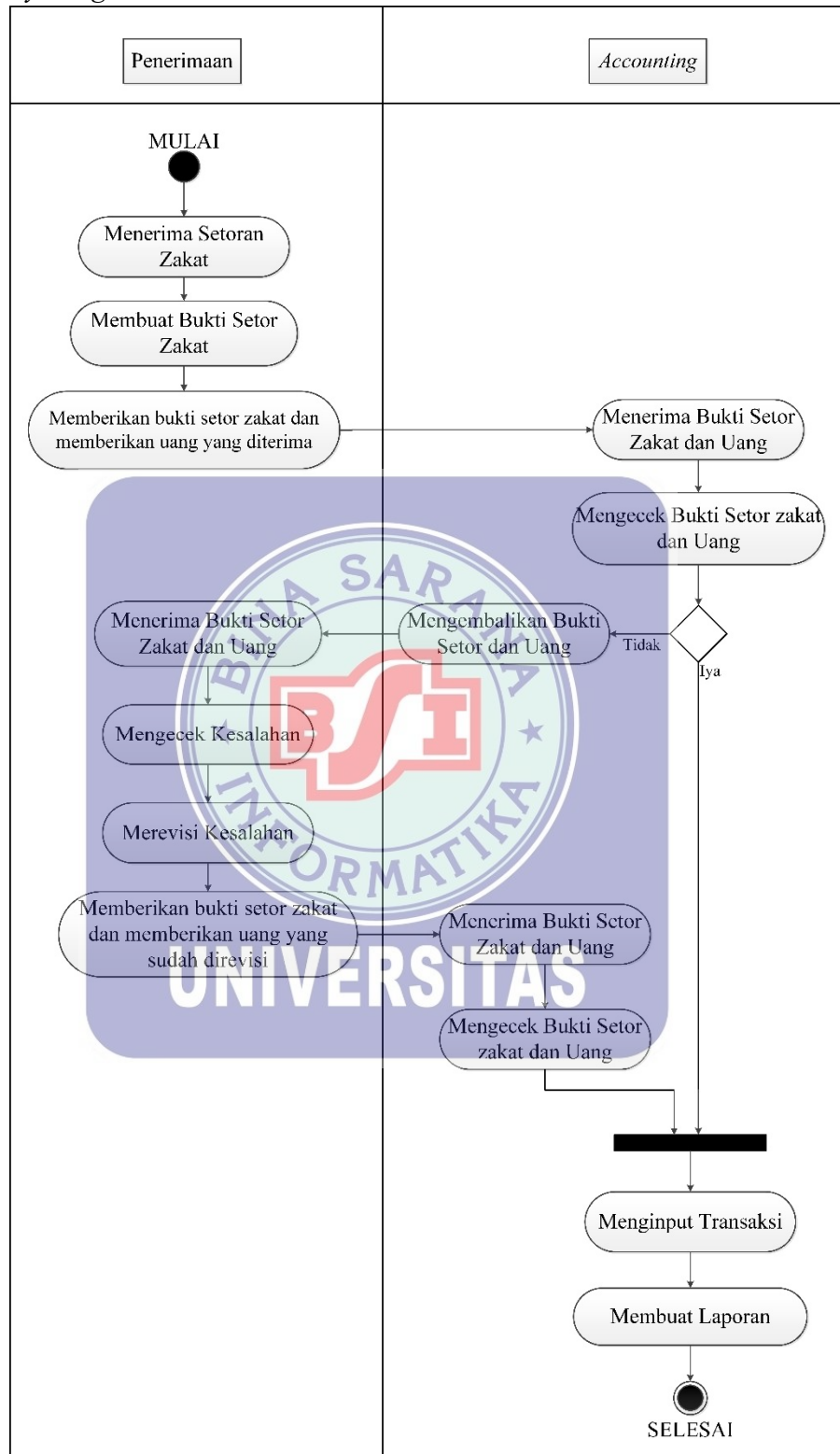
Bagian staff pendistribusian zakat memberikan formulir permohonan pendistribusian ke bagian *accounting*, bagian *accounting* mengecek permohonan dan membuat kwitansi dan menyerahkannya ke wakil ketua II bidang pendistribusian zakat, wakil ketua II menerima surat permohonan dan kwitansi jika setuju wakil ketua II akan menandatangani jika tidak wakil

ketua akan merevisi surat permohonan dan yang akan didistribusikan, jika sudah direvisi wakil ketua II menandatangani kwitansi dan menyerahkannya ke bagian *accounting*, bagian *accounting* menerima kwitansi yang sudah disetujui dan menyiapkan uang yang akan diberikan, setelah uang dan kwitansi sudah siap bagian *accounting* memberikannya ke bagian pendistribusian.



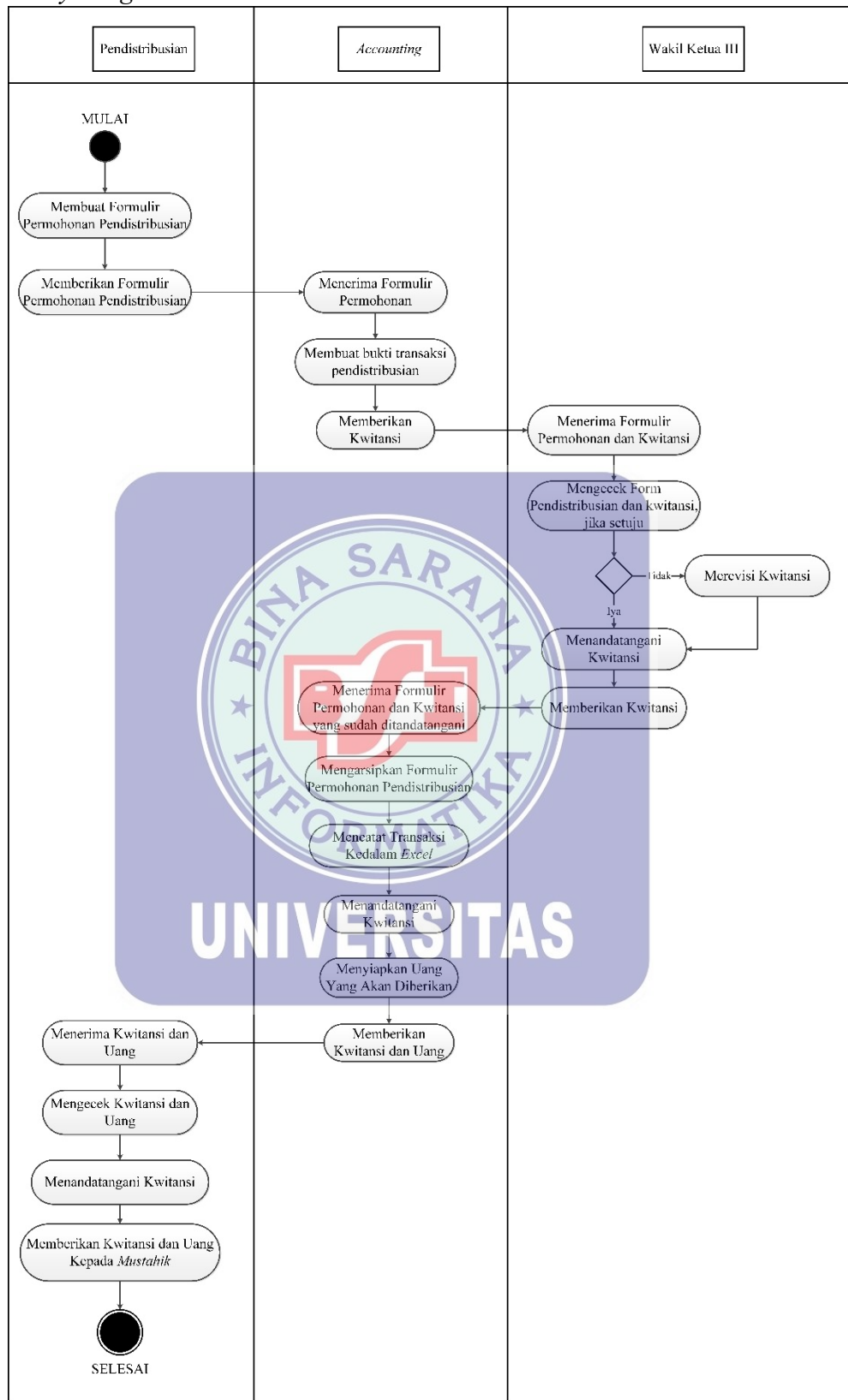
3.2.2. Activity Diagram

1. Activity Diagram Kas Masuk



Gambar III.3.
Activity Diagram Kas Masuk

2. Activity Diagram Kas Keluar



Gambar III.4.
Activity Diagram Kas Keluar

3.2.3. Dokumen Masukan

1. Nama Dokumen : Bukti Setor Zakat
- Fungsi : Bukti transaksi setiap muzakki membayar zakat
- Sumber : Bagian Penerimaan Zakat, Infaq dan Shadaqah
- Tujuan : Muzakki
- Media : Kertas
- Frekuensi : setiap pembayaran zakat
- Format : Lampiran – A.1.

2. Nama Dokumen : Bukti Setor Infak dan Sedekah
- Fungsi : Bukti transaksi setiap donatur membayar
- Sumber : Bagian Penerimaan Zakat,
- Tujuan : Muzakki
- Media : Kertas
- Frekuensi : setiap pembayaran Infaq dan Shadaqah
- Format : Lampiran – A.2.

3. Nama Dokumen : Form Permohonan
- Fungsi : Bukti pendaftaran permohonan bantuan
- Sumber : Bagian pendistribusian
- Tujuan : Wakil Ketua III
- Media : Kertas
- Frekuensi : Setiap permohonan bantuan
- Format : Lampiran – B.1.

3.2.4. Dokumen Keluaran

1. Nama Dokumen : Kwitansi
- Fungsi : Bukti transaksi kas keluar
- Sumber : Bagian *Accounting*
- Tujuan : Arsip dan Data
- Media : Kertas
- Frekuensi : Setiap transaksi kas keluar
- Format : Lampiran – B.2.

3.2.5. Permasalahan Pokok

Berdasarkan hasil observasi yang dilakukan secara langsung mengenai pengelolaan dana zakat pada baznas karawang maka permasalahan pokok yang ditemui yaitu:

1. Pengelolaan dana zakat yang masih manual, seperti pencatatan jurnal dan transaksi kas masuk dan keluar menggunakan *microsift excel*.
2. kehilangan data pada saat pencatatan transaksi karena masih menumpuknya suatu pekerjaan pembuatan laporan.
3. Kurang efisiennya waktu yang digunakan karena harus menginput data transaksi beberapa kali.

3.2.6. Pemecahan Masalah

Berdasarkan Penjelasan permasalahan pokok yang ada pada baznas karawang, penulis memberikan pemecahan masalah atau solusi agar proses pengelolaan dana zakat, infaq dan shodaqah berjalan dengan baik dan lancar.

1. Penulis merancang sistem pengelolaan dana zakat, infaq dan shadaqah menggunakan *java* agar mempermudah proses pendataan muzakki, mustahik, penerimaan dana, pengeluaran dana maupun laporannya.
2. Penginputan transaksi lebih mudah dan tidak terjadinya penumpukan pekerjaan.
3. Pembuatan laporan menggunakan sistem agar lebih efektif dan efisien.

3.3. Analisis Kebutuhan Software

3.3.1. Analisa Kebutuhan

Dalam perancangan sistem usulan, yaitu menganalisa kebutuhan *software* yang akan digunakan. Berikut penguraian analisa kebutuhan dari sistem berjalan yang penulis usulkan dan rancang:

A. Ketua

A.1. Login

A.2. Mengelola Menu

A.3. Mengelola Data Pengguna

A.4. Melihat Laporan Transaksi

B. Wakil Ketua III

B.1. Login

B.2. Mengelola Menu

B.3. Melihat Laporan Transaksi

C. Accounting

C.1. Login

C.2. Mengelola Menu

C.3. Mengelola data muzakki

C.4. Mengelola data mustahik

C.5. Mengelola data perkiraan

C.6. Mengelola data kas masuk

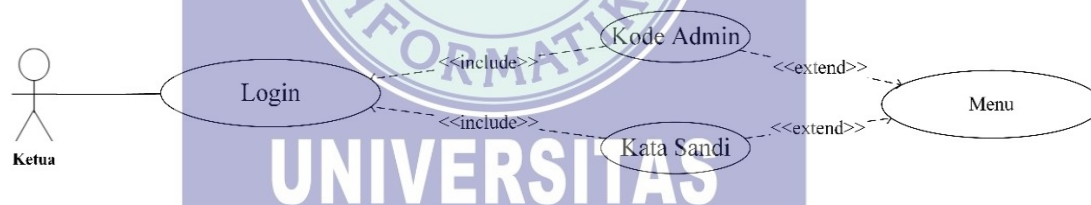
C.7. Mengelola data kas keluar

C.8. Mengelola laporan transaksi

3.3.2. Use Case Diagram

A. Ketua

A.1. Login Ketua



Gambar III.5.
Use Case Diagram Login Ketua

Tabel III.1.
Deskripsi Use Case Diagram Login Ketua

Use Case Narative Login	
Tujuan	Masuk kedalam sistem halaman utama menu
Deskripsi	Aktor masuk kedalam sistem halaman utama yang terdapat menu utama
Skenario Utama	

Aktor	Ketua
Kondisi Awal	Aktor berhasil melakukan <i>login</i>
Aksi Aktor	Reaksi Sistem
1. Aktor melakukan <i>login</i>	Sistem akan menampilkan halaman yang berisi masukkan kode admin dan kata sandi
Kondisi Akhir	Jika <i>login</i> berhasil dilakukan maka sistem akan menampilkan menu utama



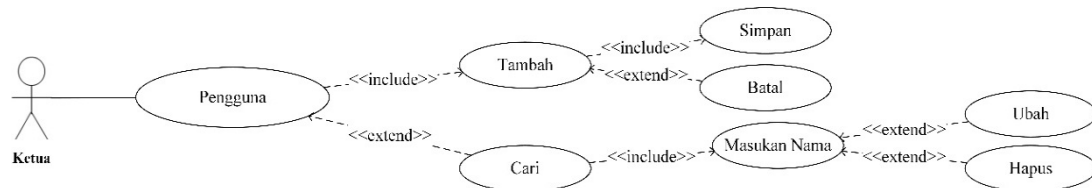
Gambar III.6.
Use Case Diagram Mengelola Menu Ketua

Tabel III.2.
Deskripsi Use Case Diagram Mengelola Menu Ketua

Use Case Narrative Menu Utama	
Tujuan	Masuk kedalam menu utama dan kedalam sistem jasa penyedia tenaga kerja.
Deskripsi	Sistem ini memungkinkan aktor untuk mengakses menu
Skenario Utama	
Aktor	Ketua
Kondisi Awal	Tampil Menu
Aksi Aktor	Reaksi Sistem
1. Aktor memilih menu pengguna	Sistem akan menampilkan <i>form</i> pengguna

2. Aktor memilih tombol Laporan Transaksi	Sistem akan menampilkan <i>form</i> Laporan Transaksi
Kondisi Akhir	Jika perintah sesuai maka sistem akan menampilkan seperti yang diinginkan oleh Aktor.

A.3. Mengelola Data Pengguna



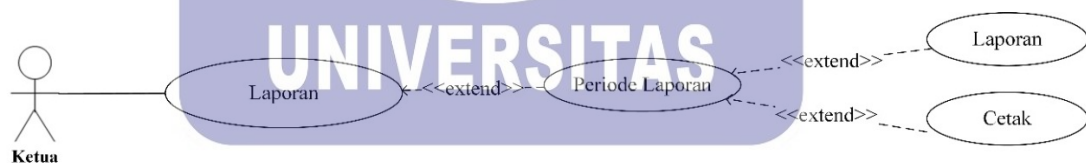
Gambar III.7.
Use Case Diagram Mengelola Data Pengguna

Tabel III.3.
Deskripsi Use Case Diagram Mengelola Data Pengguna

Use Case Narrative Pengguna	
Tujuan	<i>Accounting</i> dapat menambah, mencari, menyimpan, menghapus, mengubah dan membatalkan data admin
Deskripsi	Aktor masuk kedalam sistem halaman pengguna yang terdapat data pengguna
Skenario Utama	
Aktor	Ketua
Kondisi Awal	Aktor memilih menu pengguna
Aksi Aktor	Reaksi Sistem
1. Aktor memilih Tambah	Sistem akan menampilkan halaman form pengguna

2. Aktor memilih Cari	Sistem akan menampilkan data pengguna yang tersimpan
3. Aktor memilih Simpan	Sistem akan menyimpan data pengguna yang baru maupun yang telah diubah
4. Aktor memilih Hapus	Sistem akan menghapus data pengguna
5. Aktor memilih Batal	Sistem akan membatalkan semua proses yang berjalan
6. Aktor memilih Ubah	Sistem akan mengubah data pengguna yang tersimpan
Kondisi Akhir	Jika berhasil maka sistem akan menampilkan apa yang diperintahkan

A.4. Melihat Laporan Transaksi Ketua



Gambar III.8.
Use Case Diagram Laporan Transaksi Ketua

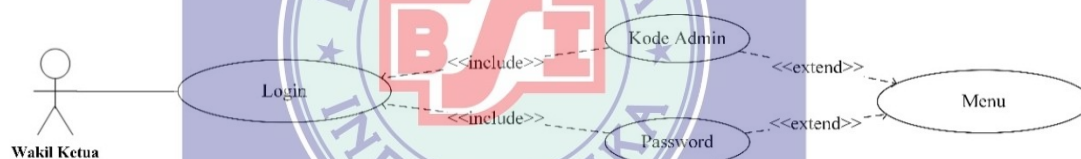
Tabel III.4.
Deskripsi Use Case Diagram Laporan Transaksi

Use Case Narative Laporan Transaksi	
Tujuan	Masuk kedalam sistem halaman Laporan
Deskripsi	Aktor masuk kedalam sistem halaman utama yang terdapat menu laporan

Skenario Utama	
Aktor	Ketua
Kondisi Awal	Aktor berhasil melakukan <i>login</i>
Aksi Aktor	Reaksi Sistem
1. Aktor melakukan <i>login</i>	Sistem akan menampilkan halaman laporan yang berisi masukkan periode
Kondisi Akhir	Jika periode telah diisi berhasil dilakukan maka sistem akan menampilkan Laporan Transaksi

B. Wakil Ketua III

B.1. Login



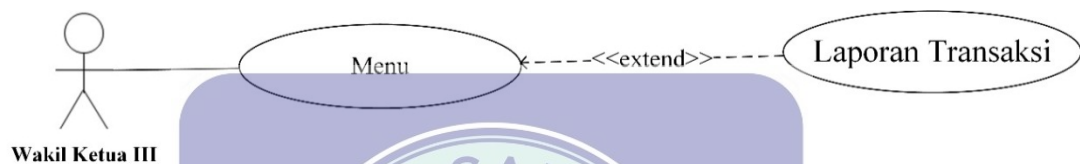
Gambar III.9.
Login Wakil Ketua III

Tabel III.5.
Deskripsi Use Case Diagram Login Wakil Ketua III

Use Case Narrative Login Wakil Ketua III	
Tujuan	Masuk kedalam sistem halaman utama menu
Deskripsi	Aktor masuk kedalam sistem halaman utama yang terdapat menu utama
Skenario Utama	
Aktor	Wakil Ketua III
Kondisi Awal	Aktor berhasil melakukan <i>login</i>

Aksi Aktor	Reaksi Sistem
1. Aktor melakukan <i>login</i>	Sistem akan menampilkan halaman yang berisi masukkan kode admin dan kata sandi
Kondisi Akhir	Jika <i>login</i> berhasil dilakukan maka sistem akan menampilkan menu utama

B.2. Mengelola Menu Wakil Ketua III

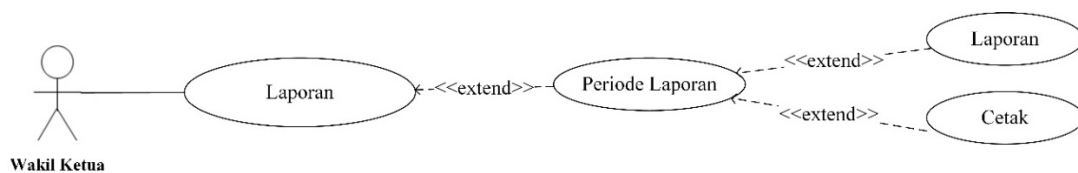


Gambar III.10.
Use Case Diagram Mengelola Menu Wakil Ketua III

Tabel III.6.
Deskripsi Use Case Diagram Mengelola Menu Wakil Ketua III
Use Case Narrative Menu Utama

Use Case Narrative Menu Utama	
Tujuan	Masuk kedalam menu utama dan kedalam sistem jasa penyedia tenaga kerja.
Deskripsi	Sistem ini memungkinkan aktor untuk mengakses menu
Skenario Utama	
Aktor	Ketua
Kondisi Awal	Tampil Menu
Aksi Aktor	Reaksi Sistem
1. Aktor memilih tombol Laporan Transaksi	Sistem akan menampilkan <i>form</i> Laporan Transaksi
Kondisi Akhir	Jika perintah sesuai maka sistem akan menampilkan seperti yang diinginkan oleh Aktor.

B.3. Mengelola Laporan Transaksi



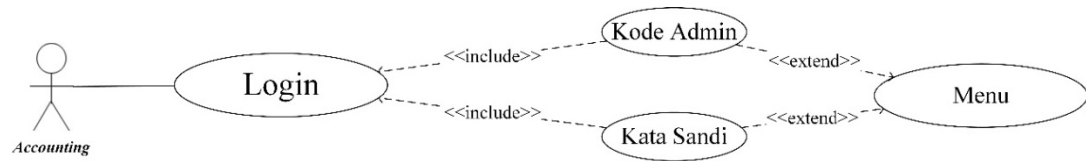
Gambar III.11.
Use Case Diagram Laporan Transaksi

Tabel III.7.
Deskripsi Use Case Diagram Laporan Transaksi
Use Case Narrative Laporan Transaksi

Use Case Narrative Laporan Transaksi	
Tujuan	Masuk kedalam sistem halaman Laporan
Deskripsi	Aktor masuk kedalam sistem halaman utama yang terdapat menu laporan
Skenario Utama	
Aktor	Wakil Ketua III
Kondisi Awal	Aktor berhasil melakukan <i>login</i>
Aksi Aktor	Reaksi Sistem
1. Aktor melakukan <i>login</i>	Sistem akan menampilkan halaman laporan yang berisi masukkan periode
Kondisi Akhir	Jika periode telah diisi berhasil dilakukan maka sistem akan menampilkan Laporan Transaksi

C. Accounting

C.1. Use Case Login

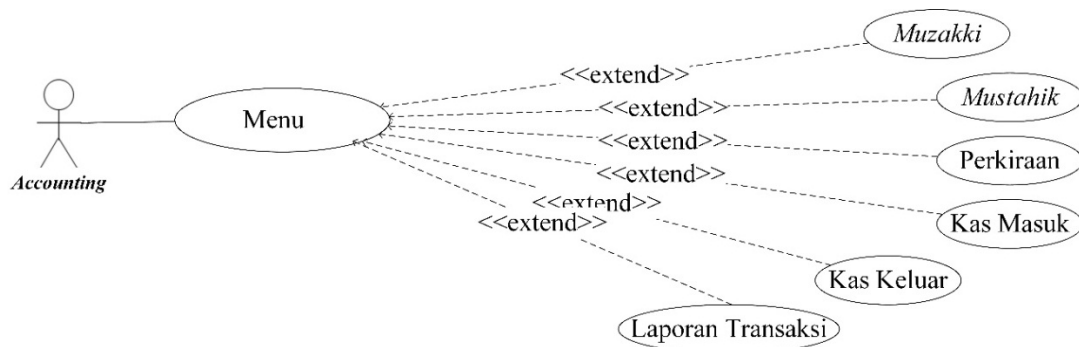


Gambar III.12.
Use Case Diagram Login

Tabel III.8.
Deskripsi Use Case Diagram Login Accounting
Use Case Narrative Login Accounting

Use Case Narrative Login Accounting	
Tujuan	Masuk kedalam sistem halaman utama menu
Deskripsi	Aktor masuk kedalam sistem halaman utama yang terdapat menu utama
Skenario Utama	
Aktor	<i>Accounting</i>
Kondisi Awal	Aktor berhasil melakukan <i>login</i>
Aksi Aktor	Reaksi Sistem
1. Aktor melakukan <i>login</i>	Sistem akan menampilkan halaman yang berisi masukkan kode admin dan kata sandi
Kondisi Akhir	Jika <i>login</i> berhasil dilakukan maka sistem akan menampilkan menu utama

C.2. Mengelola Menu



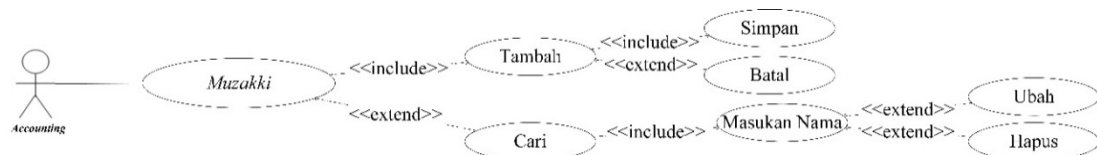
Gambar III.13.
Use Case Diagram Mengelola Menu Accounting

Tabel III.9.
Deskripsi Use Case Diagram Mengelola Menu Accounting

Use Case Narrative Menu Accounting	
Tujuan	Masuk kedalam menu utama dan kedalam sistem jasa penyedia tenaga kerja.
Deskripsi	Sistem ini memungkinkan aktor untuk mengakses menu
Skenario Utama	
Aktor	Ketua
Kondisi Awal	Tampil Menu
Aksi Aktor	Reaksi Sistem
1. Aktor memilih menu Muzakki	Sistem akan menampilkan <i>form</i> Muzakki
2. Aktor memilih menu Mustahik	Sistem akan menampilkan <i>form</i> Mustahik
3. Aktor memilih menu Perkiraan	Sistem akan menampilkan <i>form</i> Perkiraan
4. Aktor memilih menu Kas Masuk	Sistem akan menampilkan <i>form</i> Kas Masuk
5. Aktor memilih menu Kas Keluar	Sistem akan menampilkan <i>form</i> Kas Keluar

6. Aktor memilih menu Laporan Transaksi	Sistem akan menampilkan <i>form</i> Laporan Transaksi
Kondisi Akhir	Jika perintah sesuai maka sistem akan menampilkan seperti yang diinginkan oleh Aktor.

C.3. Mengelola data *muzakki*



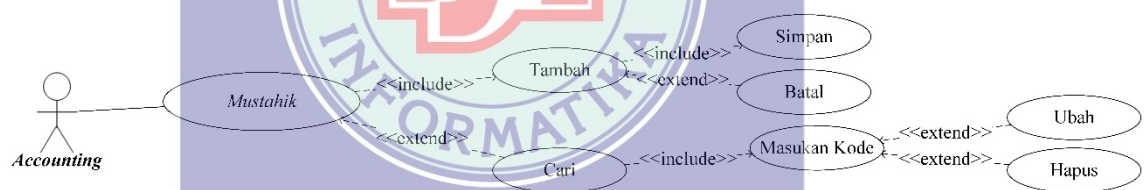
Gambar III.14.
Use Case Diagram Mengelola data muzakki

Tabel III.10.
Deskripsi Use Case Diagram muzakki

Use Case Narrative Muzakki	
Tujuan	<i>Accounting</i> dapat menambah, mencari, menyimpan, menghapus, mengubah dan membatalkan data muzakki
Deskripsi	Aktor masuk kedalam sistem halaman muzakki yang terdapat data muzakki
Skenario Utama	
Aktor	<i>Accounting</i>
Kondisi Awal	Aktor memilih menu muzakki
Aksi Aktor	Reaksi Sistem
1. Aktor memilih Tambah	Sistem akan menampilkan halaman <i>form</i> muzakki
2. Aktor memilih Cari	Sistem akan menampilkan data muzakki yang tersimpan

3. Aktor memilih Simpan	Sistem akan menyimpan data muzakki yang baru maupun yang telah diubah
4. Aktor memilih Hapus	Sistem akan menghapus data muzakki
5. Aktor memilih Batal	Sistem akan membatalkan semua proses yang berjalan
6. Aktor memilih Ubah	Sistem akan mengubah data muzakki yang tersimpan
Kondisi Akhir	Jika berhasil maka sistem akan menampilkan apa yang diperintahkan

C.4. Mengelola data Mustahik



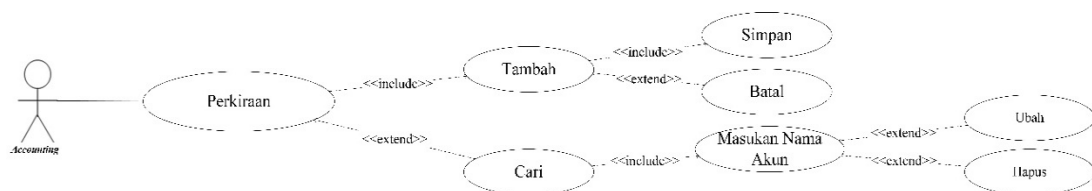
Gambar III.15.
Use Case Diagram Mengelola data mustahik

Tabel III.11.
Deskripsi Use Case Diagram mustahik

Use Case Narrative Mustahik	
Tujuan	<i>Accounting</i> dapat menambah, mencari, menyimpan, menghapus mengubah dan membatalkan data mustahik
Deskripsi	Aktor masuk kedalam sistem halaman mustahik yang terdapat data mustahik
Skenario Utama	

Aktor	<i>Accounting</i>
Kondisi Awal	Aktor memilih menu mustahik
Aksi Aktor	Reaksi Sistem
1. Aktor memilih Tambah	Sistem akan menampilkan halaman <i>form</i> mustahik
2. Aktor memilih Cari	Sistem akan menampilkan data mustahik yang tersimpan
3. Aktor memilih Simpan	Sistem akan menyimpan data mustahik yang baru maupun yang telah diubah
4. Aktor memilih Hapus	Sistem akan menghapus data mustahik
5. Aktor memilih Batal	Sistem akan membatalkan semua proses yang berjalan
6. Aktor memilih Ubah	Sistem akan mengubah data mustahik yang tersimpan
Kondisi Akhir	Jika berhasil maka sistem akan menampilkan apa yang diperintahkan

C.5. Mengelola data perkiraan

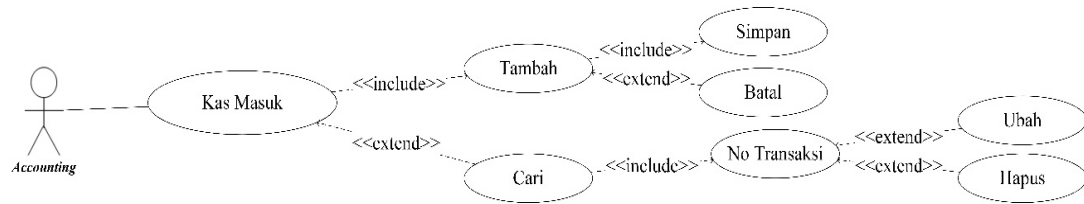


Gambar III.16.
Use Case Diagram Mengelola data perkiraan

Tabel III.12.
Deskripsi Use Case Diagram Perkiraan
Use Case Narative Perkiraan

Tujuan		<i>Accounting</i> dapat menambah, mencari, menyimpan, menghapus, mengubah dan membatalkan data master akun
Deskripsi		Aktor masuk kedalam sistem halaman master akun yang terdapat data master akun
Skenario Utama		
Aktor		<i>Accounting</i>
Kondisi Awal		Aktor memilih menu perkiraan
Aksi Aktor		Reaksi Sistem
1. Aktor memilih Tambah		Sistem akan menampilkan halaman form perkiraan
2. Aktor memilih Cari		Sistem akan menampilkan data perkiraan yang tersimpan
3. Aktor memilih Simpan		Sistem akan menyimpan data Perkiraan yang baru maupun yang telah diubah
4. Aktor memilih Hapus		Sistem akan menghapus data perkiraan
5. Aktor memilih Batal		Sistem akan membatalkan semua proses yang berjalan
6. Aktor memilih Ubah		Sistem akan mengubah data perkiraan yang tersimpan
Kondisi Akhir		Jika berhasil maka sistem akan menampilkan apa yang diperintahkan

C.6. Mengelola data Kas Masuk



Gambar III.17.
Use Case Diagram Mengelola Data Kas Masuk

Tabel III.13.
Deskripsi Use Case Diagram Kas Masuk

Use Case Narative Kas Masuk	
Tujuan	Accounting dapat menambah, mencari, menyimpan, menghapus, mengubah dan membatalkan data admin
Deskripsi	Aktor masuk kedalam sistem halaman kas masuk yang terdapat data admin
Skenario Utama	
Aktor	Accounting
Kondisi Awal	Aktor memilih menu Kas Masuk
Aksi Aktor	Reaksi Sistem
1. Aktor memilih Tambah	Sistem akan menampilkan halaman form admin
2. Aktor memilih Cari	Sistem akan menampilkan data kas masuk yang tersimpan
3. Aktor memilih Simpan	Sistem akan menyimpan data kas masuk yang baru maupun yang telah diubah
4. Aktor memilih	Sistem akan menghapus data kas masuk

Hapus	
5. Aktor memilih Batal	Sistem akan membatalkan semua proses yang berjalan
6. Aktor memilih Ubah	Sistem akan mengubah data kas masuk yang tersimpan
Kondisi Akhir	Jika berhasil maka sistem akan menampilkan apa yang diperintahkan

C.7. Mengelola data Kas Keluar



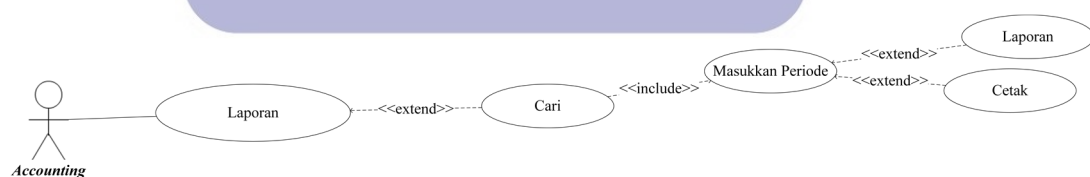
Gambar III.18.
Use Case Diagram Mengelola Data Kas Keluar

Tabel III.14.
Deskripsi Use Case Diagram Kas Keluar
Use Case Narative Kas Keluar

Tujuan	<i>Accounting</i> dapat menambah, mencari, menyimpan, menghapus, mengubah dan membatalkan data kas keluar
Deskripsi	Aktor masuk kedalam sistem halaman kas keluar yang terdapat data kas keluar
Skenario Utama	
Aktor	<i>Accounting</i>
Kondisi Awal	Aktor memilih menu Kas Keluar
Aksi Aktor	Reaksi Sistem

1. Aktor memilih Tambah	Sistem akan menampilkan halaman form admin
2. Aktor memilih Cari	Sistem akan menampilkan data kas keluar yang tersimpan
3. Aktor memilih Simpan	Sistem akan menyimpan data kas keluar yang baru maupun yang telah diubah
4. Aktor memilih Hapus	Sistem akan menghapus data kas keluar
5. Aktor memilih Batal	Sistem akan membatalkan semua proses yang berjalan
6. Aktor memilih Ubah	Sistem akan mengubah data kas keluar yang tersimpan
Kondisi Akhir	Jika berhasil maka sistem akan menampilkan apa yang diperintahkan

C.8. Mengelola Laporan Transaksi



Gambar III.19.
Use Case Diagram Laporan Transaksi Accounting

Tabel III.15.
Deskripsi Laporan Transaksi
Use Case Narative Laporan Transaksi

Use Case Narative Laporan Transaksi	
Tujuan	Masuk kedalam sistem halaman Laporan

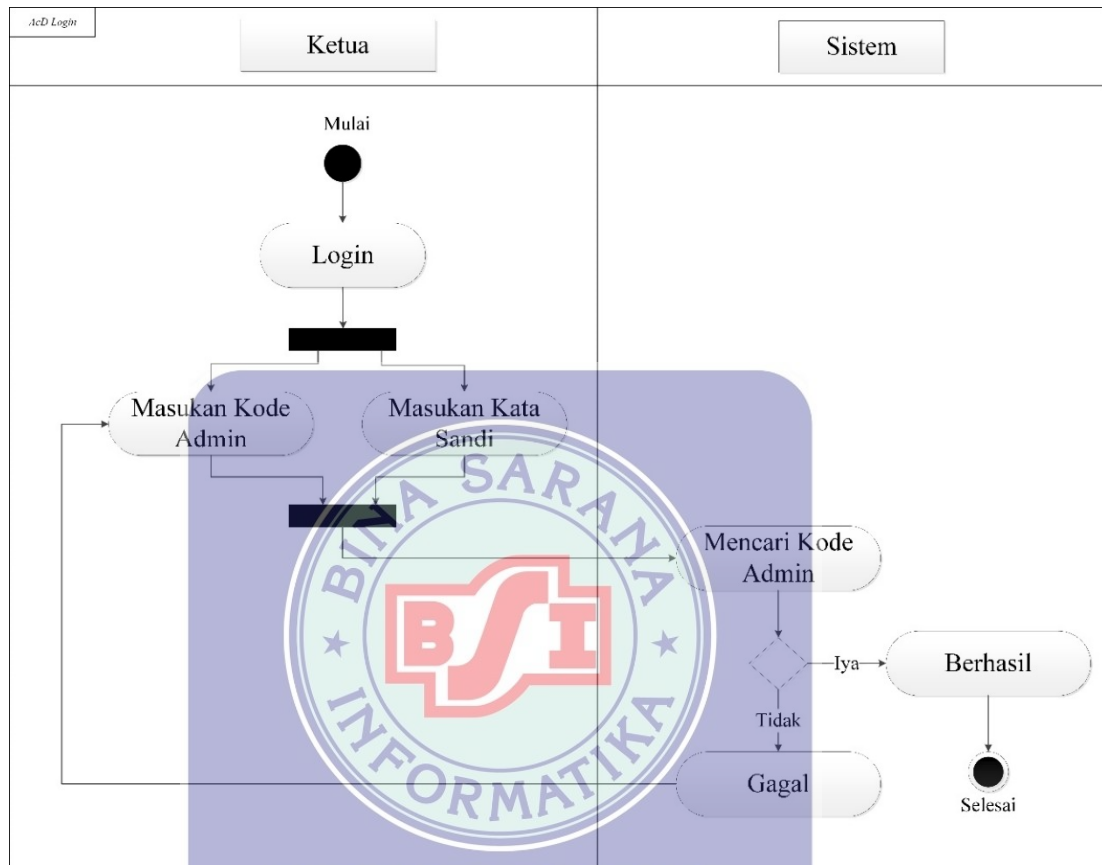
Deskripsi	Aktor masuk kedalam sistem halaman utama yang terdapat menu laporan
Skenario Utama	
Aktor	<i>Accounting</i>
Kondisi Awal	Tampil menu
Aksi Aktor	Reaksi Sistem
1. Aktor memilih menu Laporan Transaksi	Sistem akan menampilkan halaman laporan yang berisi masukkan periode
Kondisi Akhir	Jika periode telah diisi berhasil dilakukan maka sistem akan menampilkan Laporan Transaksi



3.3.2. Activity Diagram

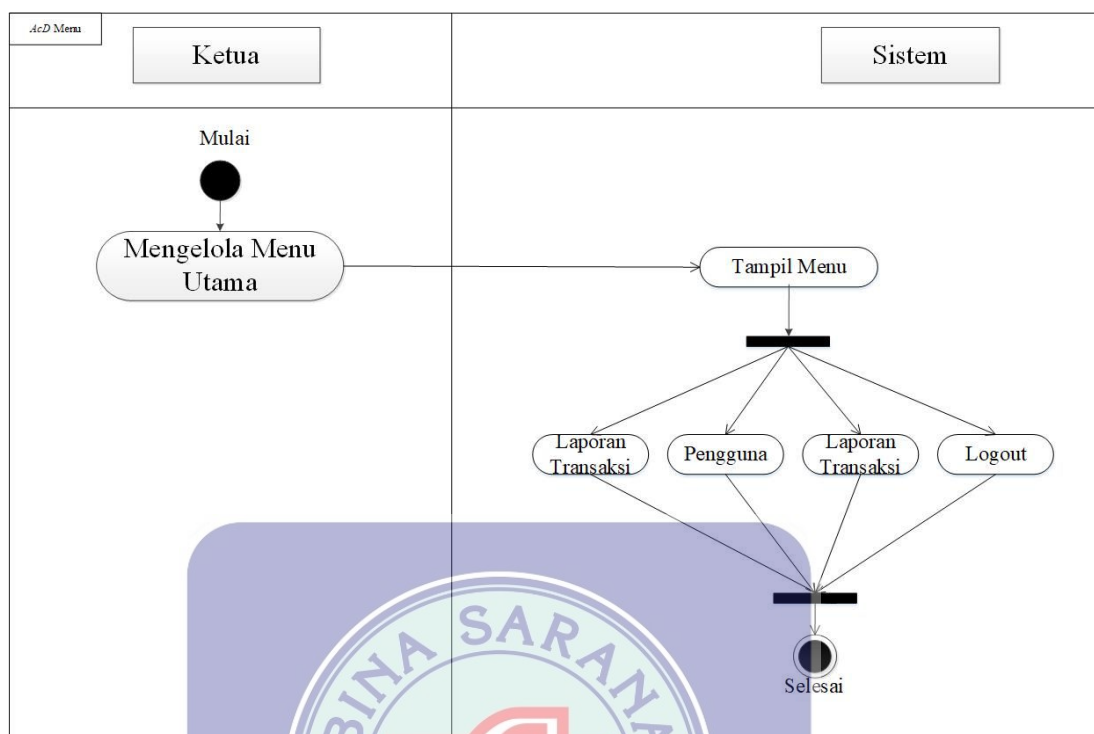
A. Ketua

A.1. Login



Gambar III.20.
Activity Diagram Login Ketua

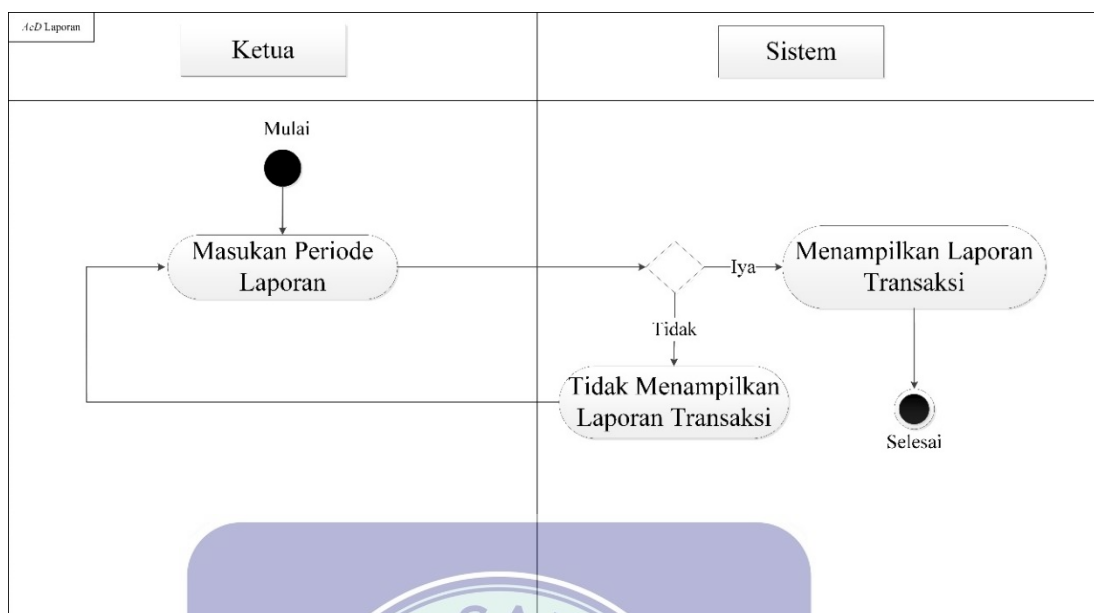
A.2. Mengelola Menu



Gambar III.21★
Activity Diagram Mengelola Menu

UNIVERSITAS

A.4. Melihat Laporan Transaksi

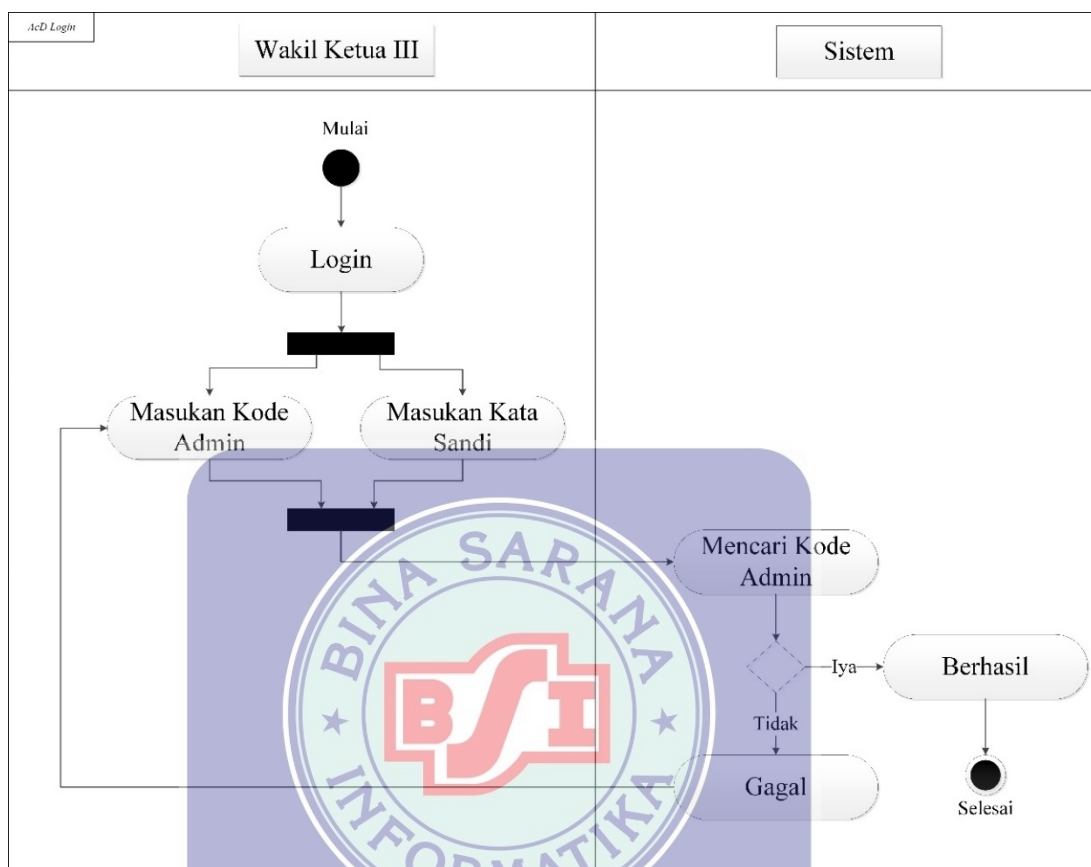


Gambar III.23.
Activity Diagram Laporan Transaksi



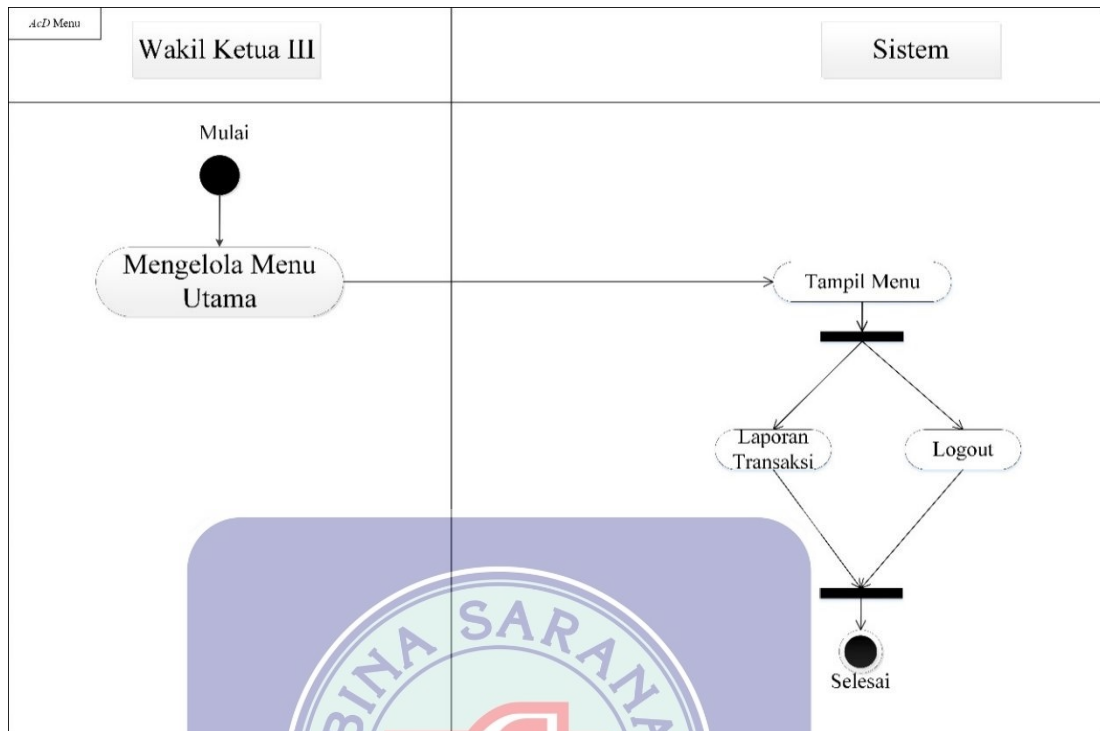
B. Wakil Ketua III

B.1. Login



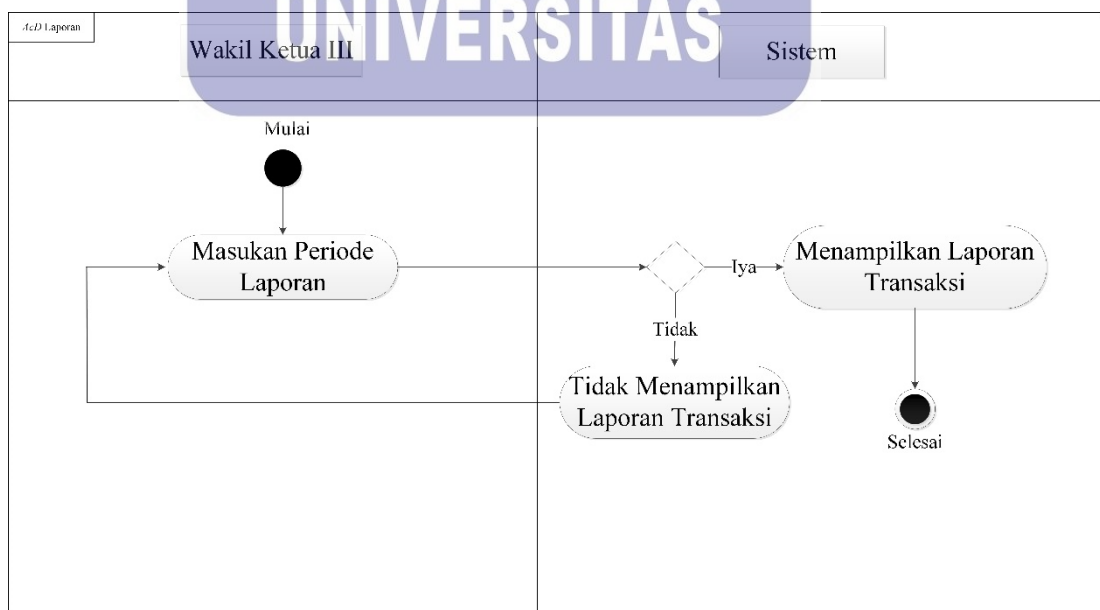
Gambar III.24.
Activity Diagram Login Wakil Ketua III

B.2. Mengelola Menu



Gambar III.25.
Activity Diagram Menu Wakil Ketua III

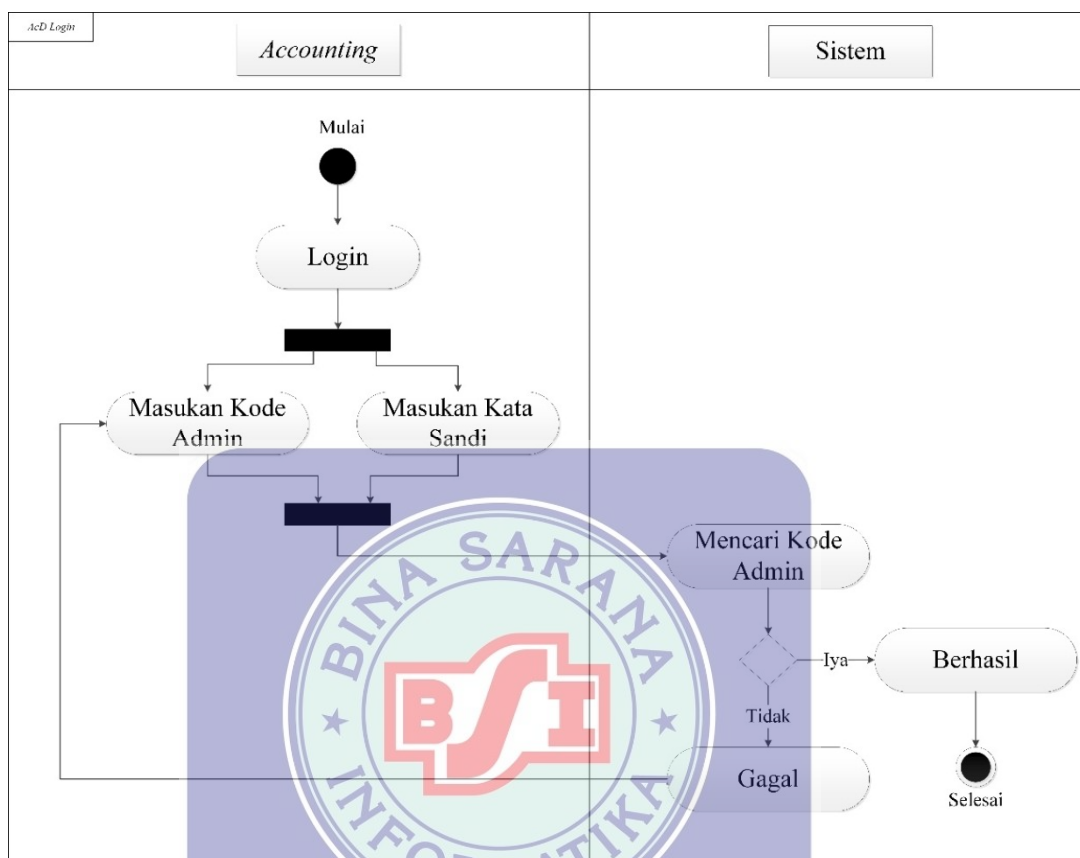
B.3. Melihat Laporan Transaksi



Gambar III.26.
Activity Diagram Laporan Transaksi

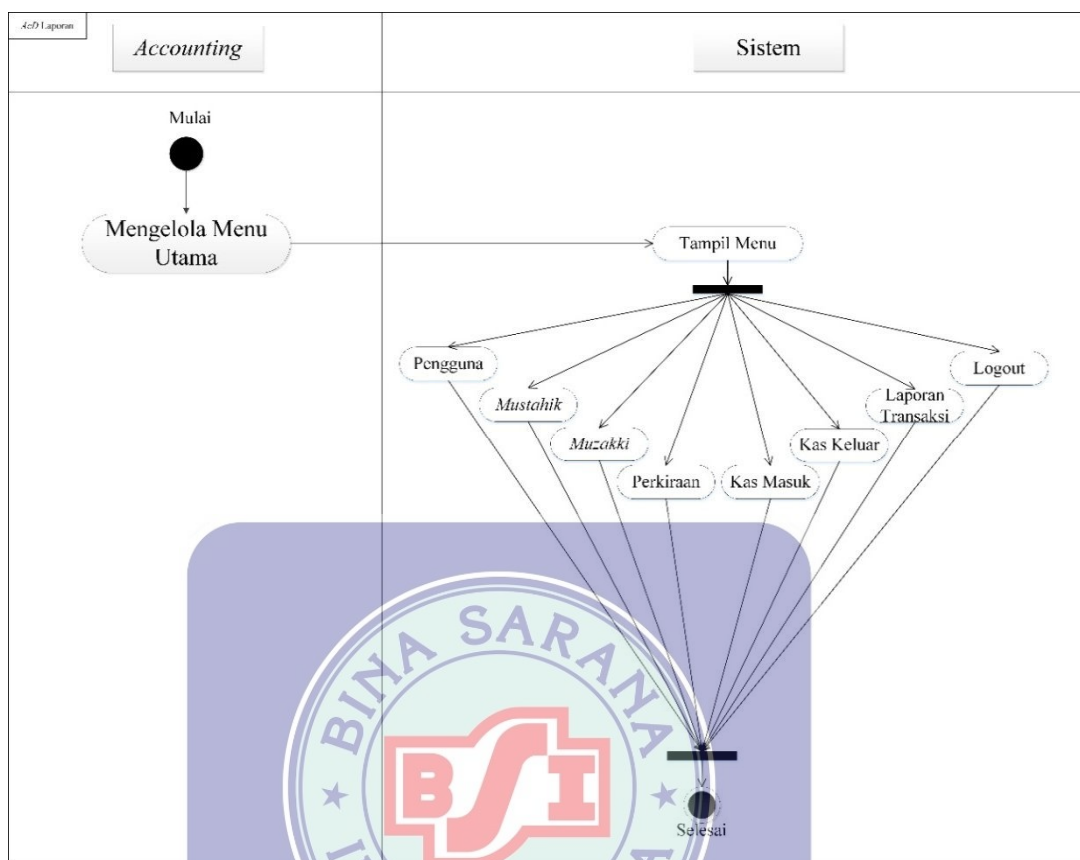
C. Accounting

C.1. Login



Gambar III.27.
Activity Diagram Login Accounting

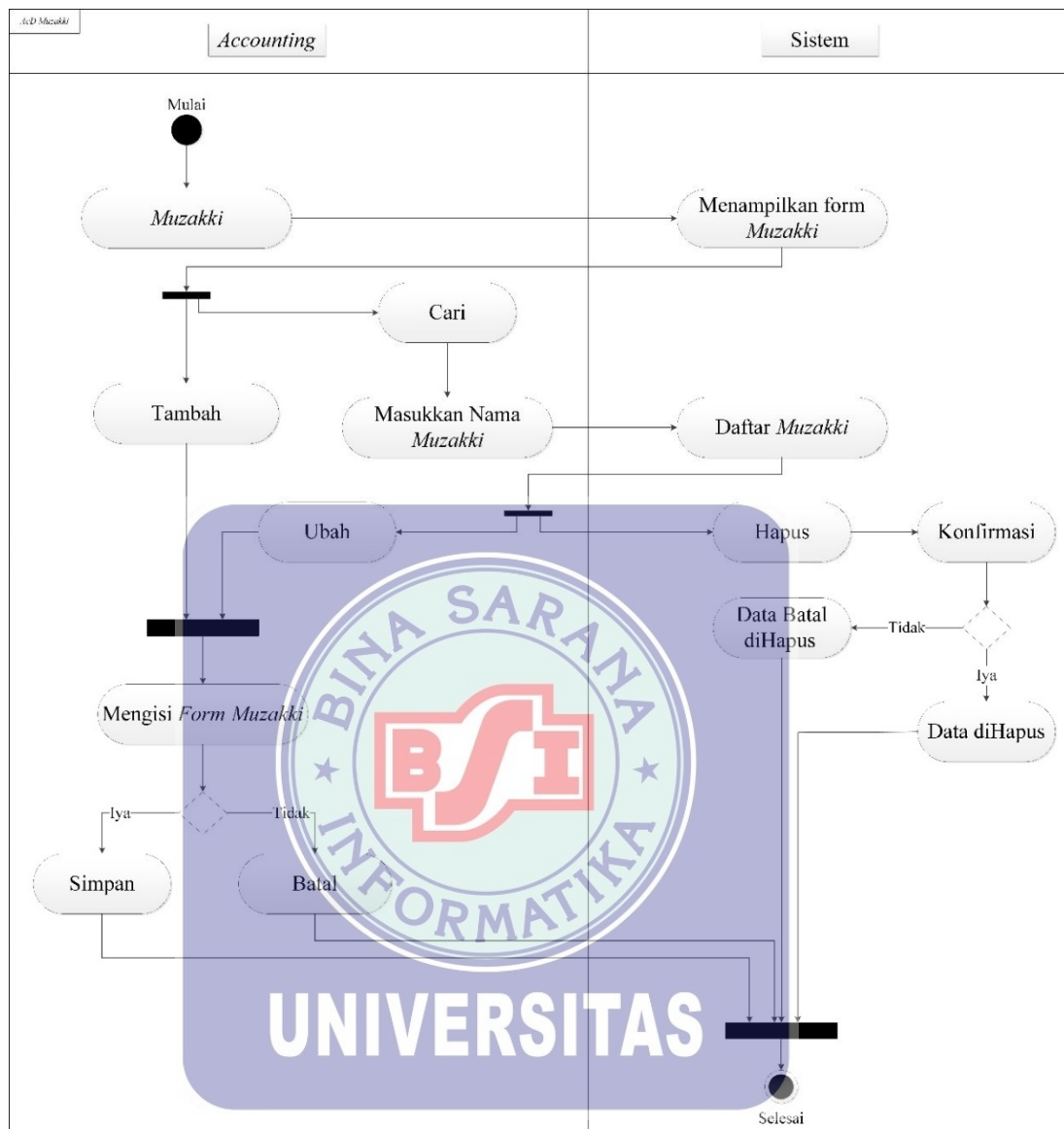
C.2. Mengelola Menu



Gambar III.28.
Activity Diagram Menu

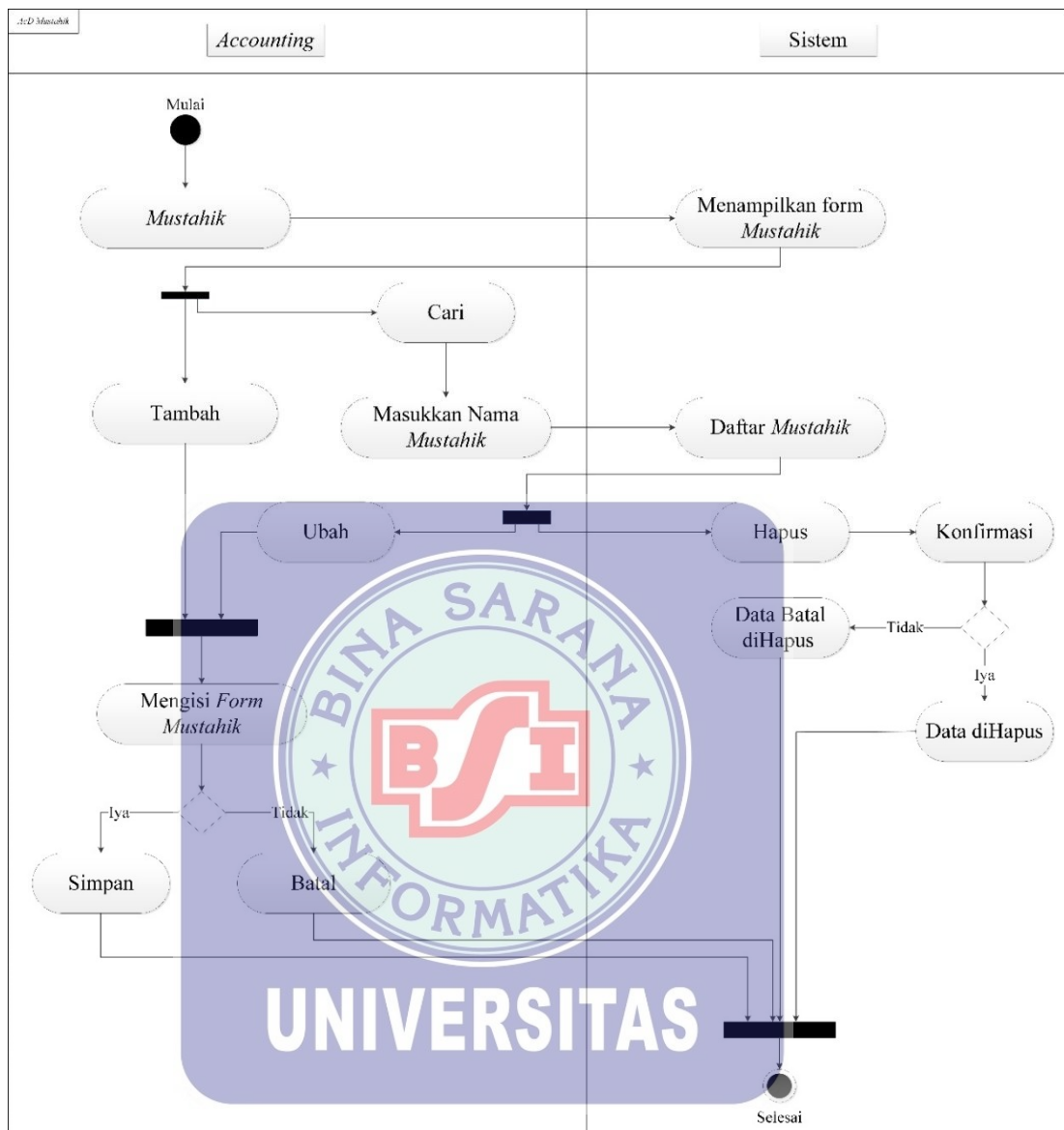
UNIVERSITAS

C.3. Mengelola Data Muzakki



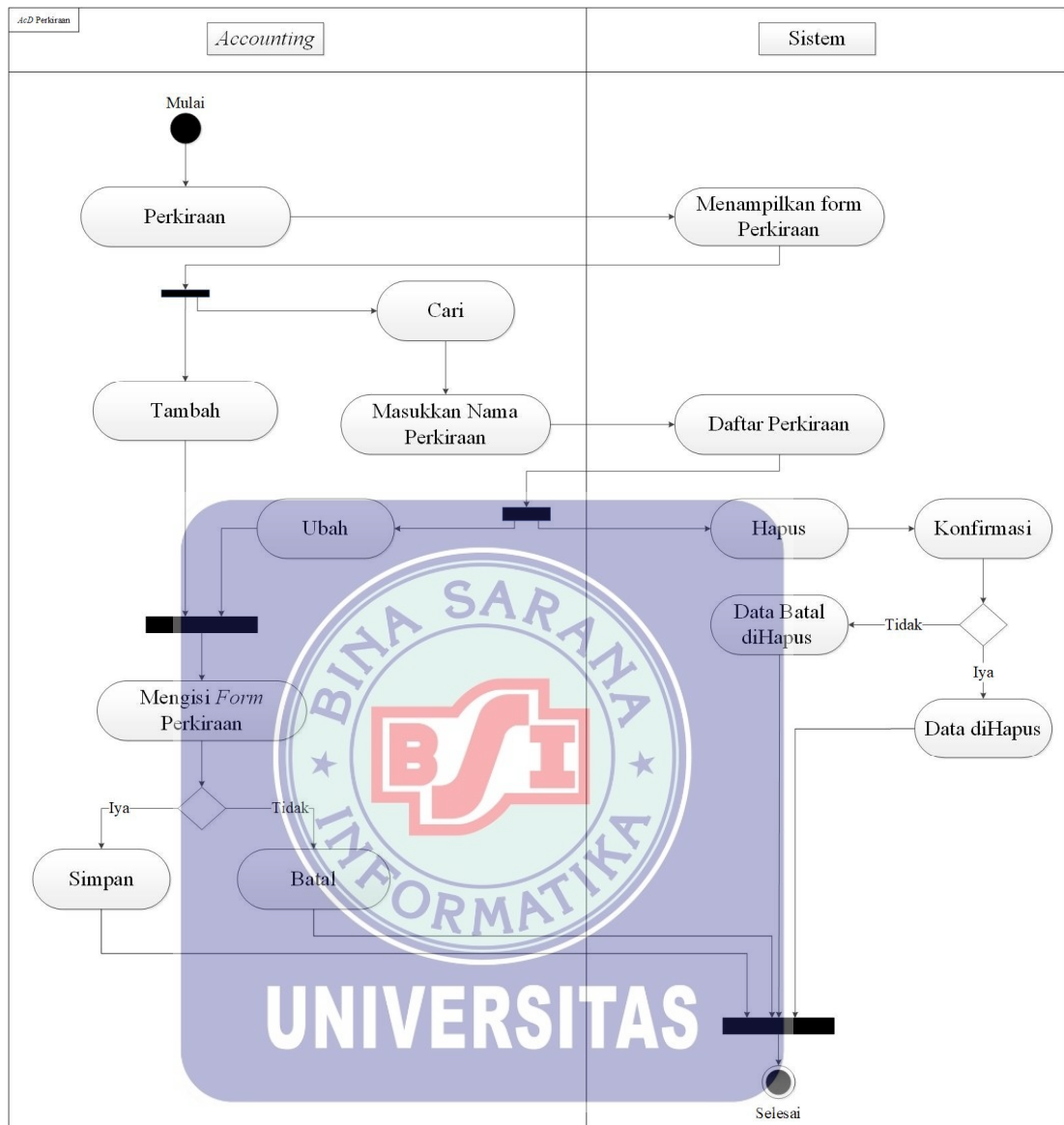
Gambar III.29.
Activity Diagram Muzakki

C.4. Mengelola Data Mustahik



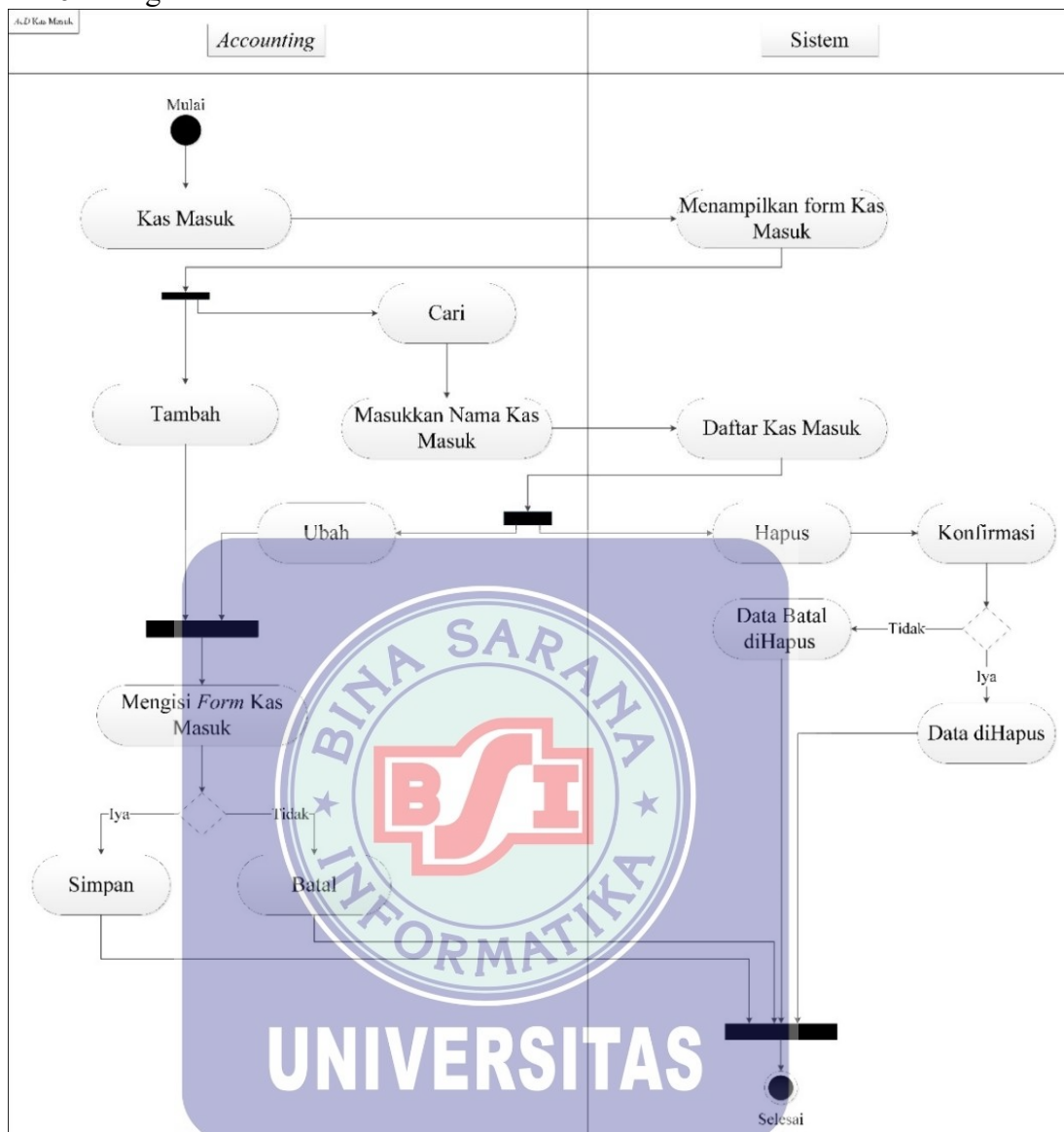
Gambar III.30.
Activity Diagram Mustahik

A.5. Mengelola data perkiraan



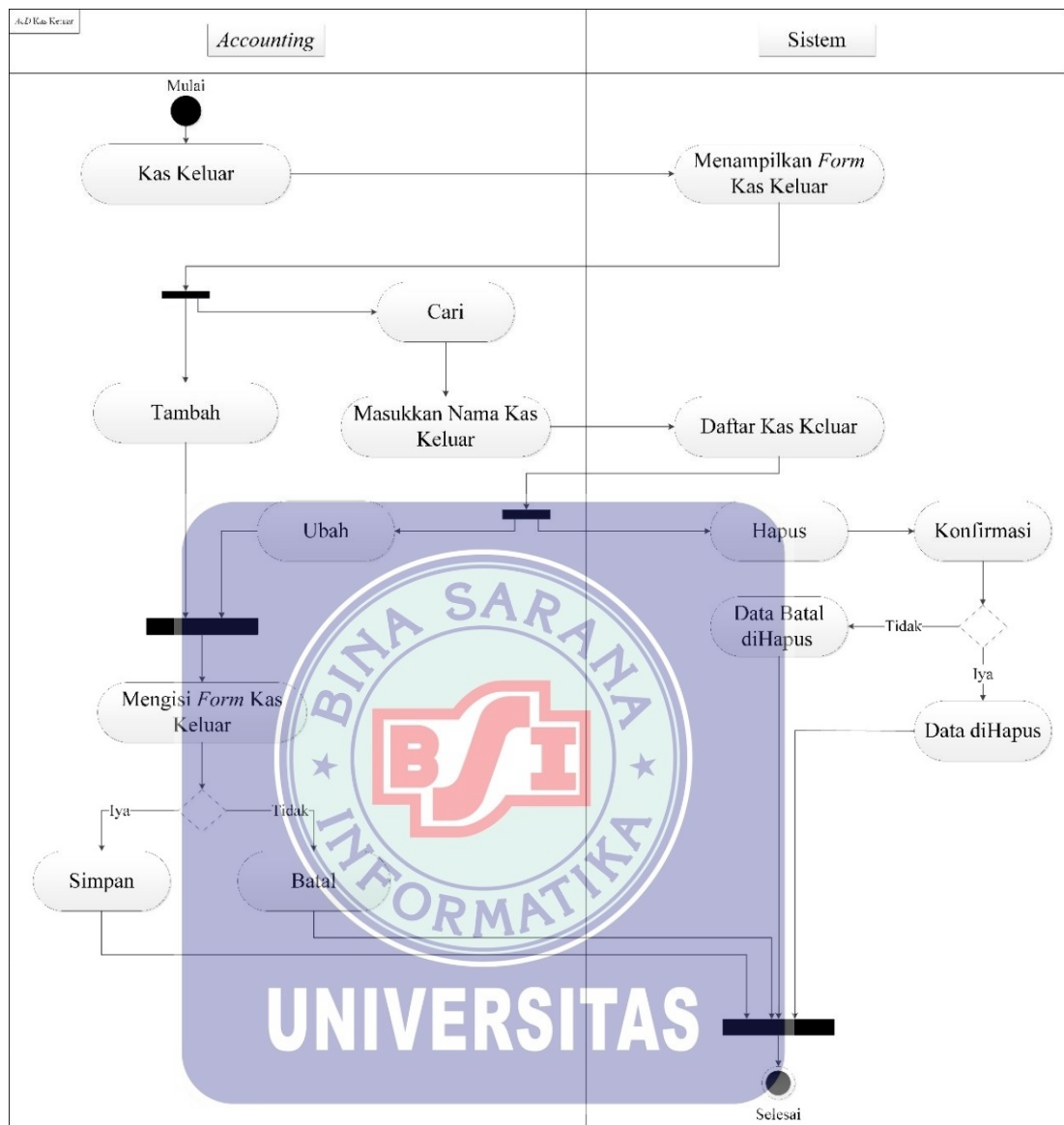
Gambar III.31.
Activity Diagram Perkiraan

A.6. Mengelola Kas Masuk



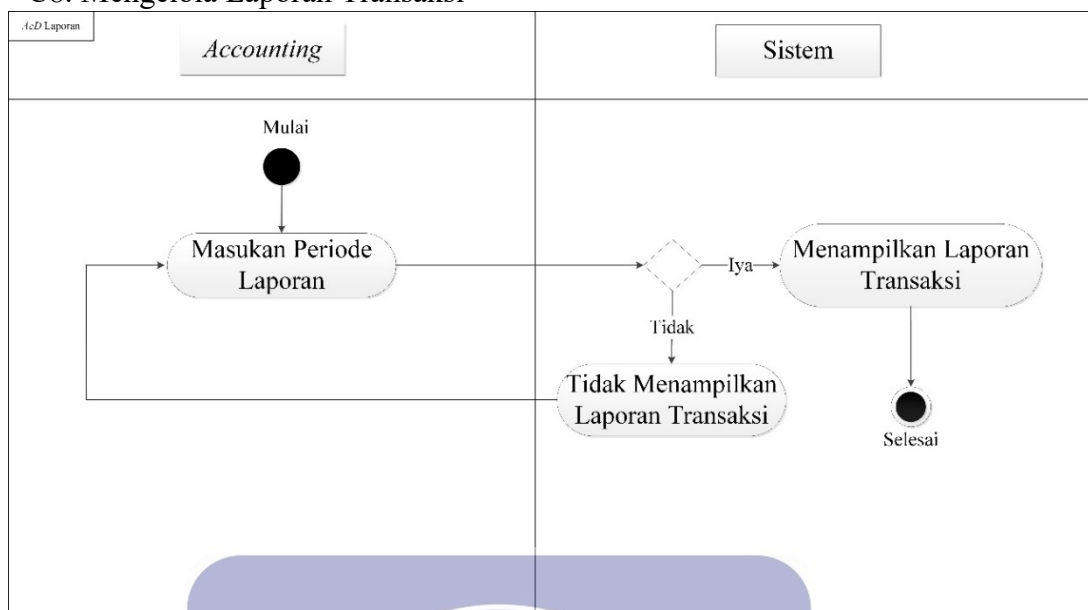
Gambar III.32.
Activity Diagram Kas Masuk

C7. Mengelola Kas Keluar



Gambar III.33.
Activity Diagram Kas Keluar

C8. Mengelola Laporan Transaksi



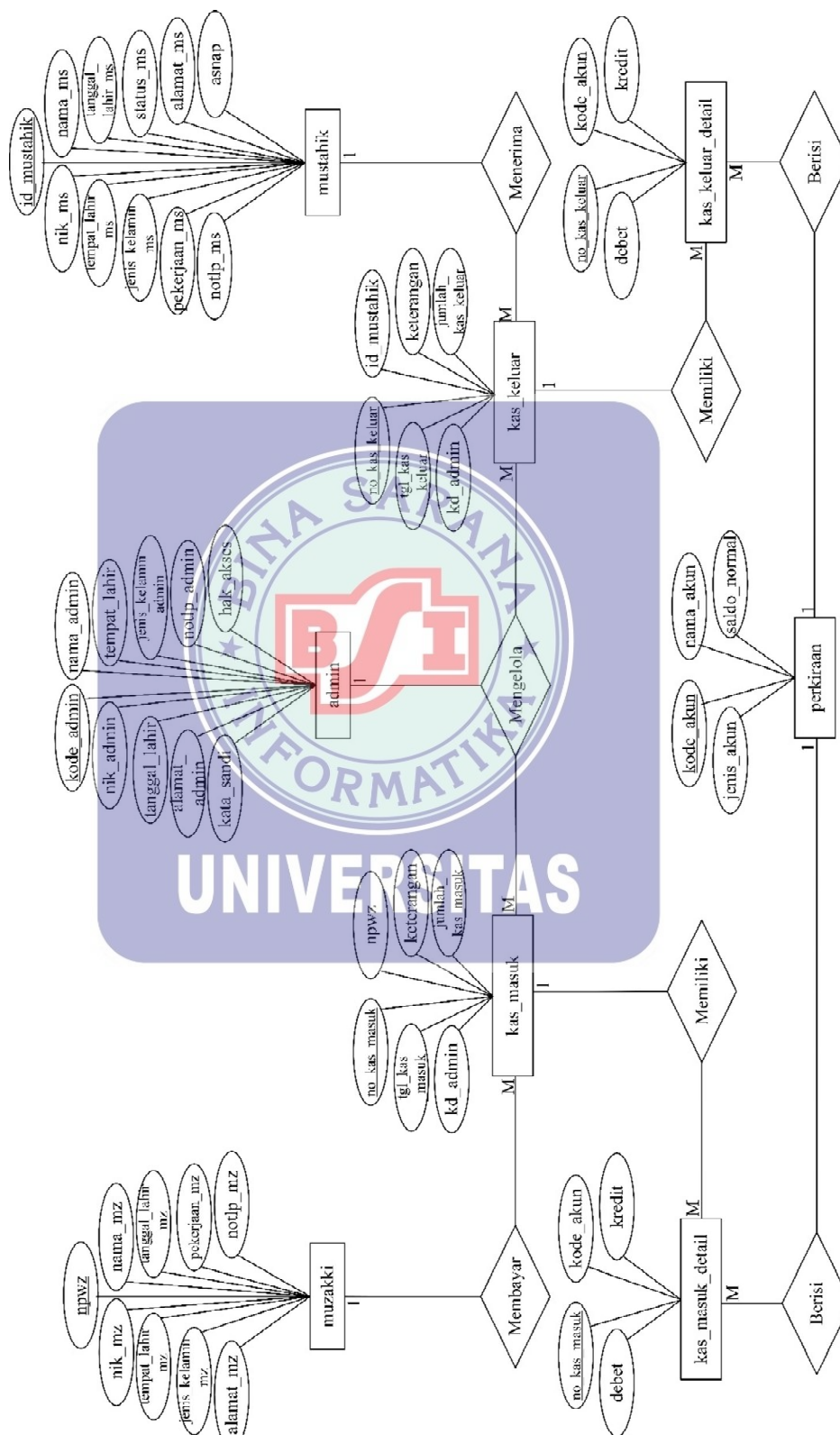
Gambar III.34.
Activity Diagram Mengelola Laporan Transaksi



UNIVERSITAS

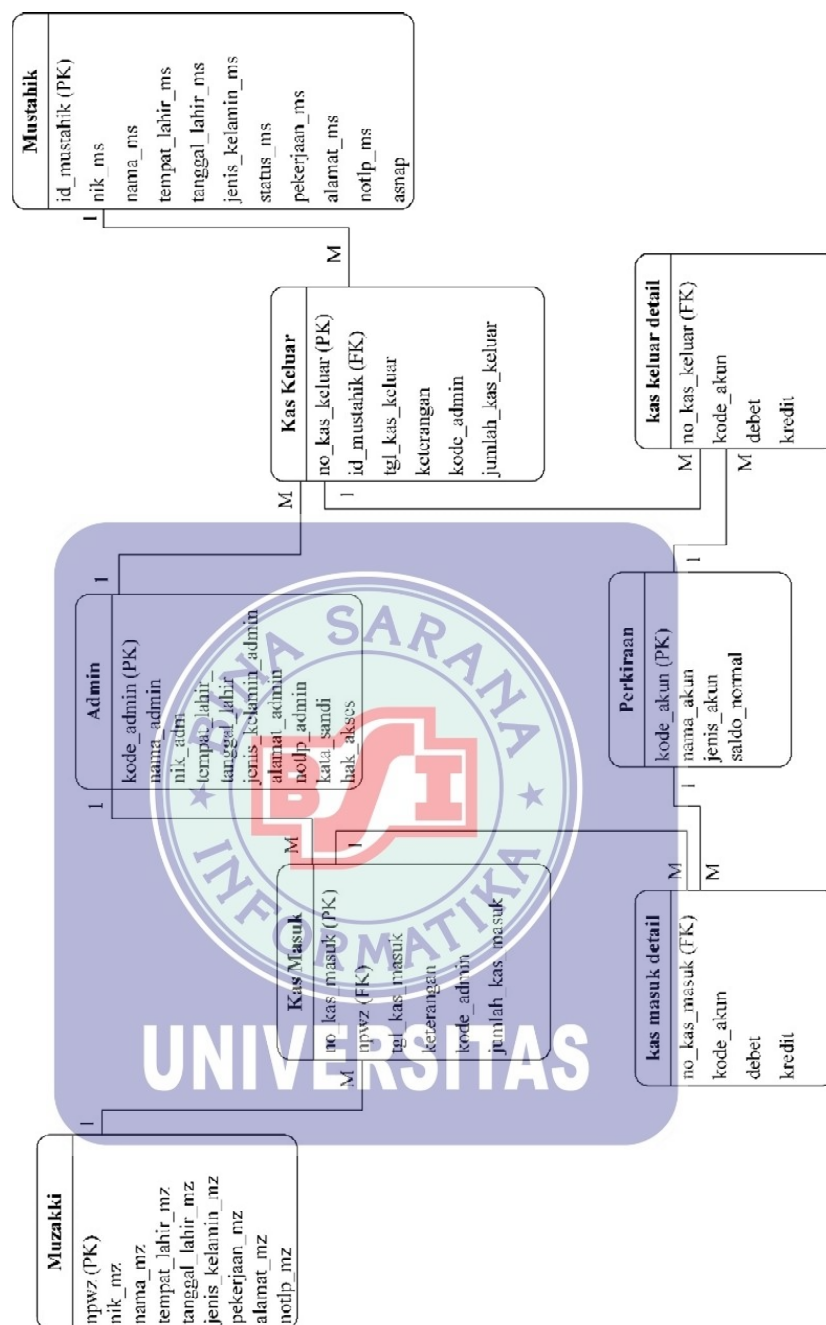
3.4 Desain

3.4.1. Entity Relationship Diagram (ERD)



Gambar III.35.
Entity Relationship Diagram (ERD)

3.4.2. Logical Record Structure (LRS)



Gambar III.36.
Logical Record Structure (LRS)

3.4.3. Spesifikasi *File*

1. Spesifikasi *File* Data Muzakki

Nama *Database* : muzakki

Nama *File* : Muzakki

Tipe *File* : *File* Data Master

Akses *File* : *Random*

Panjang *Record* : 279 Karakter

Kunci *Field* : npwz

Tabel III.16.
Spesifikasi *File* Data Muzakki

No	Elemen Data	Akronim	Jenis	Panjang	Keterangan
1	Nomor Peserta Wajib Zakat	npwz	Char	8	Primary Key
2	NIK	nik_mz	Char	16	
3	Nama	nama_mz	Char	25	
4	Tempat Lahir	tempat_lahir_mz	Varchar	35	
5	Tanggal Lahir	tanggal_lahir_mz	Date		
6	Jenis Kelamin	jenis_kelamin_mz	Varchar	10	
7	Alamat	alamat_mz	Varchar	150	
8	No Telpn	notlp_mz	Varchar	15	
9	Pekerjaan	pekerjaan_mz	Varchar	20	

2. Spesifikasi *File* Data Mustahik

Nama *Database* : mustahik

Nama *File* : Mustahik

Tipe *File* : *File* Data Master

Akses *File* : *Random*

Panjang *Record* : 314 Karakter

Kunci *Field* : id_mustahik

Tabel III.17.
Spesifikasi File Data Mustahik

No	Elemen Data	Akronim	Jenis	Panjang	Keterangan
1	Identitas Mustahik	<u>id_mustahik</u>	Char	8	Primary Key
2	NIK	nik_ms	Char	16	
3	Nama	nama_ms	Char	25	
4	Tempat Lahir	tempat_lahir_ms	Varchar	35	
5	Tanggal Lahir	tanggal_lahir_ms	Date		
6	Jenis Kelamin	jenis_kelamin_ms	Varchar	10	
7	Status	status_ms	Varchar	15	
8	Pekerjaan	pekerjaan_ms	Varchar	20	
9	Alamat	alamat_ms	Varchar	150	
10	No Telpon	notlp_ms	Varchar	15	
11	Asnap	asnap	Varchar	20	

3. Spesifikasi File Data Pengguna

Nama Database : admin

Nama File : Admin

Tipe File : File Data Master

Akses File : Random

Panjang Record : 286 Karakter

Kunci Field : kd_admin

Tabel III.18.
Spesifikasi File Data Admin

No	Elemen Data	Akronim	Jenis	Panjang	Keterangan
1	Kode Admin	<u>kd_admin</u>	Char	7	Primary Key
2	Nama	nama_admin	Varchar	25	
3	NIK	nik_admin	Int	16	
4	Tempat Lahir	tempat_lahir	Varchar	35	
5	Tanggal Lahir	tanggal_lahir	Date		
6	Jenis Kelamin	jenis_kelamin_admin	Varchar	10	
7	Alamat	alamat_admin	Varchar	150	
8	No Telpon	notlp_admin	Int	15	
9	Kata Sandi	kata_sandi	Varchar	8	
10	Hak Akses	hak_akses	Varchar	20	

4. Spesifikasi *File* Data Perkiraan

Nama *Database* : perkiraan

Nama *File* : Perkiraan

Tipe *File* : *File* Data Master

Akses *File* : *Random*

Panjang *Record* : 66 Karakter

Kunci *Field* : kode_akun

Tabel III.19.
Spesifikasi *File* Data Perkiraan

No	Elemen Data	Akronim	Jenis	Panjang	Keterangan
1	Kode Akun	kode_akun	Char	5	Primary Key
2	Nama Akun	nama_akun	Varchar	30	
3	Jenis Akun	jenis_akun	Varchar	25	
4	Saldo Normal	saldo_normal	Char	6	

5. Spesifikasi *File* Data Kas Masuk

Nama *Database* : kasmasuk

Nama *File* : Kas Masuk

Tipe *File* : *File* Data Transaksi

Akses *File* : *Random*

Panjang *Record* : 123 Karakter

Kunci *Field* : no_kas_masuk

Tabel III.20.
Spesifikasi File Data Kas Masuk

No	Elemen Data	Akronim	Jenis	Panjang	Keterangan
1	No Transaksi Kas Masuk	no_kas_masuk	<i>Char</i>	8	<i>Primary Key</i>
2	Nomor Pokok Wajib Pajak	npwz	<i>Char</i>	8	<i>Foreign Key</i>
3	Tanggal Kas Masuk	tgl_kas_masuk	<i>Date</i>		
4	Keterangan Kas Masuk	keterangan	<i>Varchar</i>	100	
5	Kode Admin	kd_admin	<i>Char</i>	7	<i>Foreign Key</i>
6	Jumlah Kas Keluar	jumlah_kas_keluar	<i>Double</i>		

6. Spesifikasi File Data Kas Keluar

Nama *Database* : kaskeluar
 Nama *File* : Kas Keluar
 Tipe *File* : File Data Transaksi
 Akses *File* : *Random*
 Panjang *Record* : 123 Karakter
 Kunci *Field* : no_kas_keluar

Tabel III.21.
Spesifikasi File Data Kas Keluar

No	Elemen Data	Akronim	Jenis	Panjang	Keterangan
1	No Transaksi Kas Keluar	no_kas_keluar	<i>Char</i>	8	<i>Primary Key</i>
2	Id Mustahik	id_mustahik	<i>Char</i>	8	<i>Foreign Key</i>
3	Tanggal Kas Keluar	tgl_kas_keluar	<i>Date</i>		
4	Keterangan Kas Keluar	keterangan	<i>Varchar</i>	100	
5	Kode Admin	kd_admin	<i>Char</i>	7	<i>Foreign Key</i>
6	Jumlah Kas Keluar	jumlah_kas_keluar	<i>Double</i>		

7. Spesifikasi File Data Kas Masuk Detail

Nama *Database* : kas_masuk_detail
 Nama *File* : Kas Masuk Detail

Tipe File : *File Data Transaksi*
Akses File : *Random*
Panjang Record : 13 Karakter
Kunci Field : -

Tabel III.22.
Spesifikasi *File Data Kas Keluar*

No	Elemen Data	Akronim	Jenis	Panjang	Keterangan
1	No Transaksi Kas Masuk	no_kas_masuk	Char	8	
2	Kode Akun	Kode akun	Char	5	
3	Debet	debet	Double		
4	Kredit	kredit	Double		

8. Spesifikasi *File Data Kas Keluar Detail*

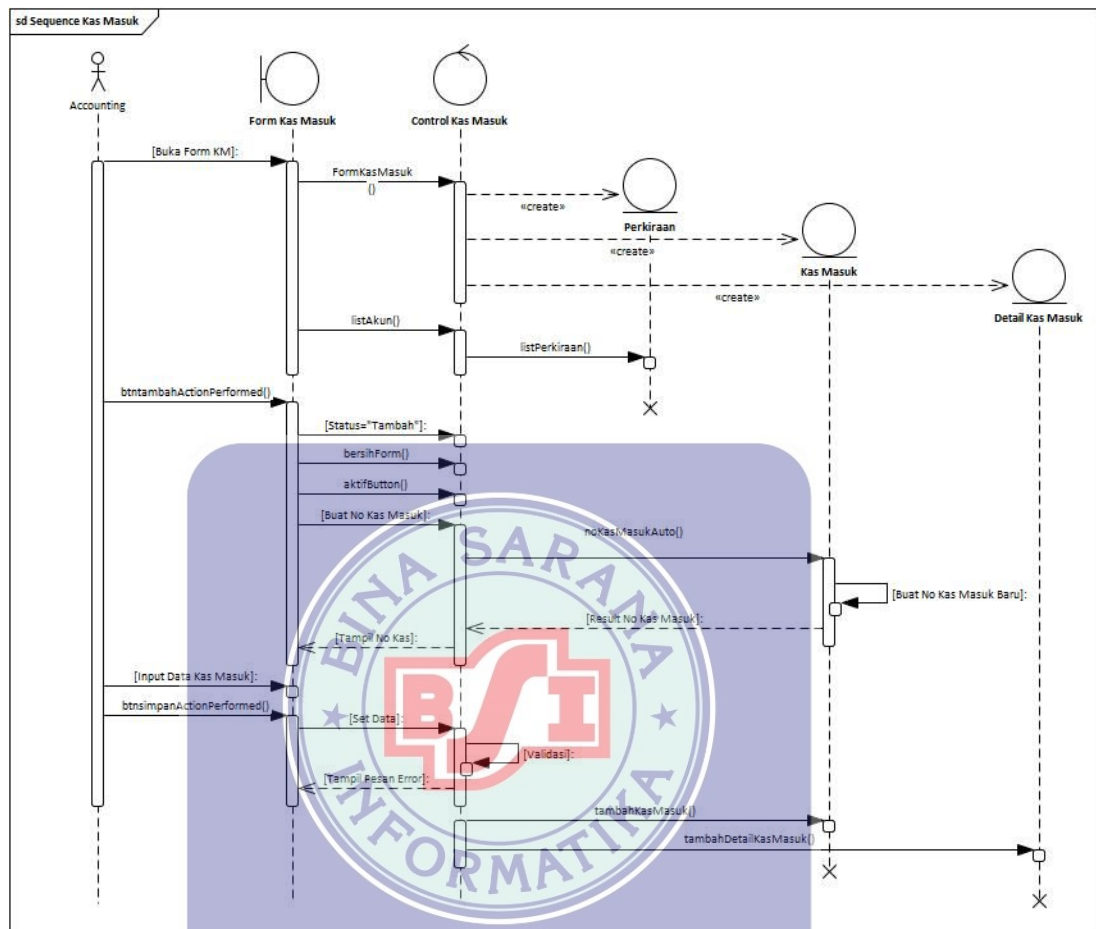
Nama Database : kas_keluar_detail
Nama File : Kas Keluar Detail
Tipe File : *File Data Transaksi*
Akses File : *Random*
Panjang Record : 13 Karakter
Kunci Field : -

Tabel III.23.
Spesifikasi *File Data Kas Keluar*

No	Elemen Data	Akronim	Jenis	Panjang	Keterangan
1	No Transaksi Kas Keluar	no_kas_keluar	Char	8	
2	Kode Akun	Kode akun	Char	5	
3	Debet	debet	Double		
4	Kredit	kredit	Double		

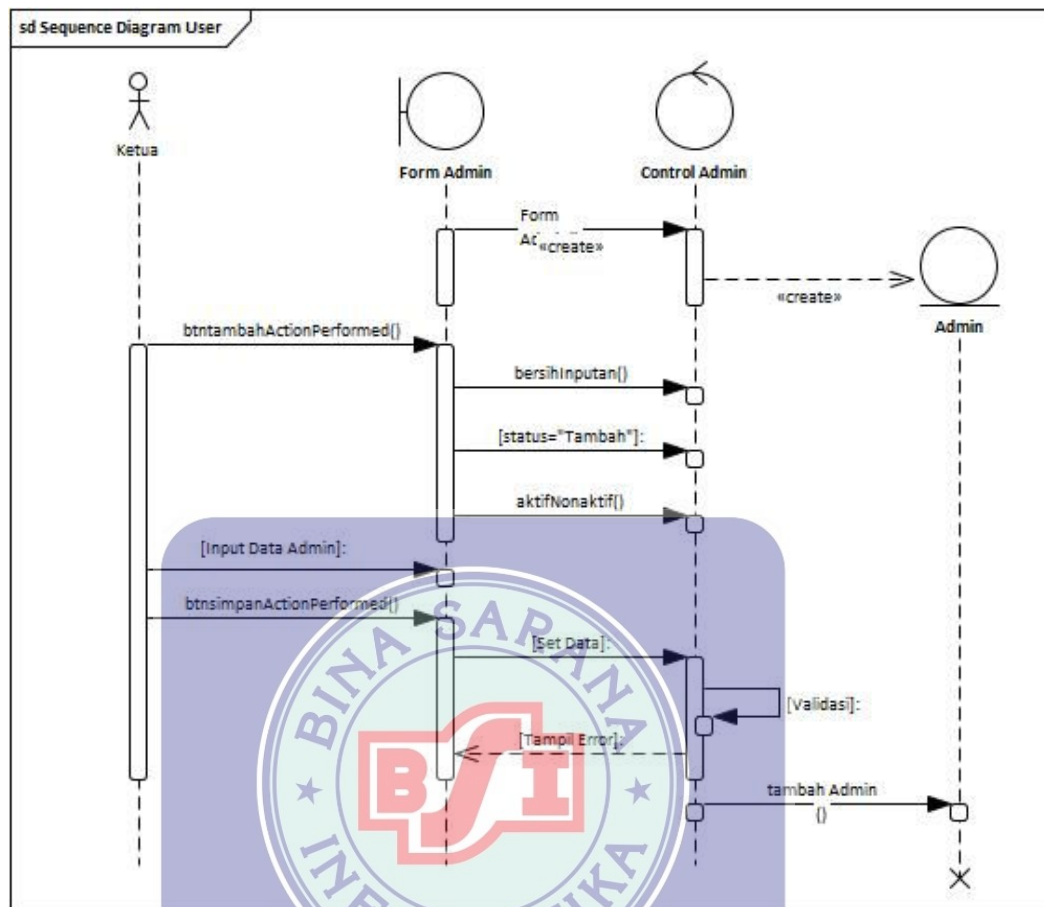
3.4.4. Sequence Diagram

B. Sequence Diagram Kas Masuk



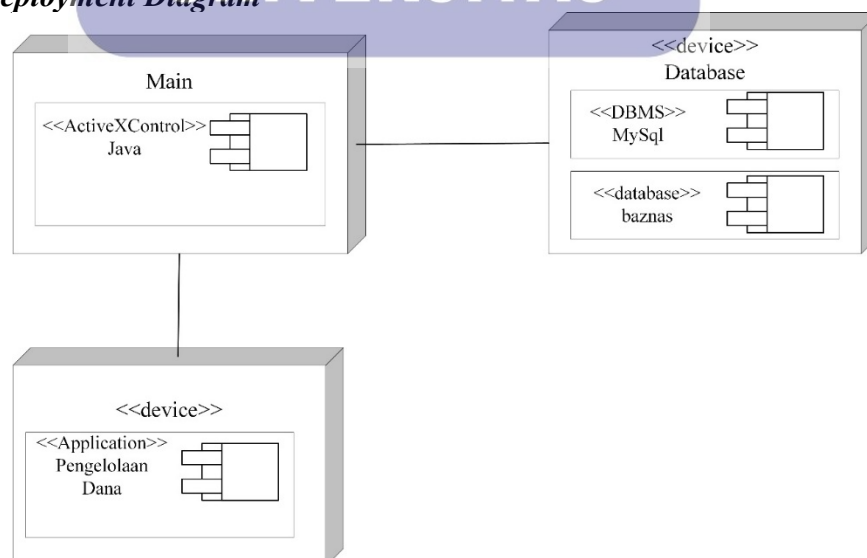
Gambar III.37.
Sequence Diagram Kas Masuk

B. Sequence Diagram Pengguna



Gambar III.38.
Sequence Diagram Pengguna

3.4.5. Deployment Diagram



Gambar III.39.
Deployment Diagram

3.4.6. User Interface

A. Login



LOGIN



BAZNAS
Badan Amil Zakat Nasional

Masukkan Kode Admin dan Kata Sandi Anda

Kode Admin

Kata Sandi

Gambar III.40.
Form Login

B. Menu



Badan Amil Zakat Nasional

Master Data Transaksi Laporan Logout



BAZNAS
Badan Amil Zakat Nasional

UNIVERSITAS

Gambar III.41.
Form Menu

C. Pengguna

Badan Amil Zakat Nasional
Master Data Laporan Logout

Form Admin

Input Data

ID Pengguna	Nama Pengguna	Hak Akses	Password
ADM-001	Kurnia Adam Kartajasa	Ketua	12345678
ADM-002	H.A Sobari Rachman	Wakil Ketua	87654321
ADM-003	Dini Sri Haryati	Accounting	00000

Tambah Simpan Hapus Ubah Batal

Gambar III.42.
Form Admin

D. Muzakki

Badan Amil Zakat Nasional
Master Data Transaksi Laporan Logout

Form Muzakki

Input Data

Npzw	Nik Muzakki	Nama Muzakki	Alamat Muzakki
MZK00001	321519450960001	Khofifah	Dusun Rawadukuh Rt.010/004Pasir T...
MZK00002	3215055901000002	Fidia Zahra	Dusun Kopo Rt.006/002 Kel.KlariKec....
MZK00003	3215293103990003	Rizki Ruspanda	Kp.Krajan Timur Rt.002/001 Kel.Cika...
MZK00004	3215014403950002	Siti Azizatul Fitriah	Santiong Utara Rt.002/017 Kel.Naga...
MZK00005	3215041601970001	Adi Purnama	Dusun I KP.Babakan Cikunju Rt.011/...
MZK00006	3216145907930003	Rosmiati	Cikelor Rt.015/005 Kel.Amansari Kec...
MZK00007	3215050404980008	Ahmad Purnama Hanif	Perum Terangsari Rt.012/007 Kel.Cib...
MZK00008	3215137107000001	Dinda Muslyanti	Perum BIP Blok DA 40/04 Rt.005/01...
MZK00009	3215206508960002	Ghina Rafidha Aziz	Dusun Tempuran Rt.006/002 Kel.Tem...
MZK00010	3215050202980009	Soni Perubi	Dusun Walahar I Rt.002/001 Kel.Wal...

Tambah Simpan Hapus Ubah Batal

Gambar III.43.
Form Muzakki

E. Mustahik

Badan Amil Zakat Nasional
Master Data Transaksi Laporan Logout

BAZNAS
Badan Amil Zakat Nasional

Form Mustahik

Input Data

ID Mustahik	Nik Mustahik	Nama Mustahik	Alamat Mustahik
MST00001	3215132112970001	Triyono Sonjaya	Kp. Krajan Tengah Rt.001/002 Kel.Ci...
MST00002	3215050812990005	Ami Priyan Saputra	Dusun Walahe II Rt.007/002Kel.Wal...
MST00003	3204324810990012	Eliza Triani Oktavia	Perum Griya Citra Persada T.49 Rt.0...
MST00004	3215010207000001	Nur Mauluddin Pahrevi	Jl.Cakradireja No.12 Rt.001/011 Kel...
MST00005	3215304211950001	Samania Anisa	Dusun Cibadar Rt.008/003Kel. Cipta...

Tambah Simpan Hapus Ubah Batal

Gambar III.44.

Form Mustahik

F. Transaksi Perkiraan

Badan Amil Zakat Nasional
Master Data Transaksi Laporan Logout

BAZNAS
Badan Amil Zakat Nasional

Form Perkiraan

Kode Akun
Nama Akun
Jenis Akun
Saldo Normal

Cari Nama Akun

Kode Akun	Nama Akun	Jenis Akun	Saldo Normal
10100	Aktiva Lancar	Aktiva Lancar	Debet
10101	Kas	Aktiva Lancar	Debet
10200	Aktiva tetap	Aktiva Tetap	Debet
10210	Gedung	Aktiva Tetap	Debet
10211	Akumulasi Penyusutan Gedung	Aktiva Lancar	Debet
10220	Kendaraan Operasional	Aktiva Tetap	Debet
10221	Akumulasi Penyusutan Kendaraan	Aktiva Lancar	Debet
10230	Tanah	Aktiva Tetap	Debet
10240	Peralatan Kantor	Aktiva Lancar	Debet
10241	Akumulasi Penyusutan Peralatan	Aktiva Lancar	Debet

Tambah Simpan Edit Batal Hapus

Gambar III.45.

Form Perkiraan

G. Kas Masuk

Badan Amil Zakat Nasional
Master Data Transaksi Laporan Logout

Form Penerimaan Kas



BAZNAS
Badan Amil Zakat Nasional

Input Data

No KM

Tanggal KM

Npwt

Akun Debet

Jumlah Kas

Akun Kredit

Keterangan

Gambar III.46.
Form Kas Masuk

H. Kas Keluar

Badan Amil Zakat Nasional
Master Data Transaksi Laporan Logout

Form Pengeluaran Kas



BAZNAS
Badan Amil Zakat Nasional

Input Data

No KK

Tanggal KK

Id Mustahik

Akun Debet

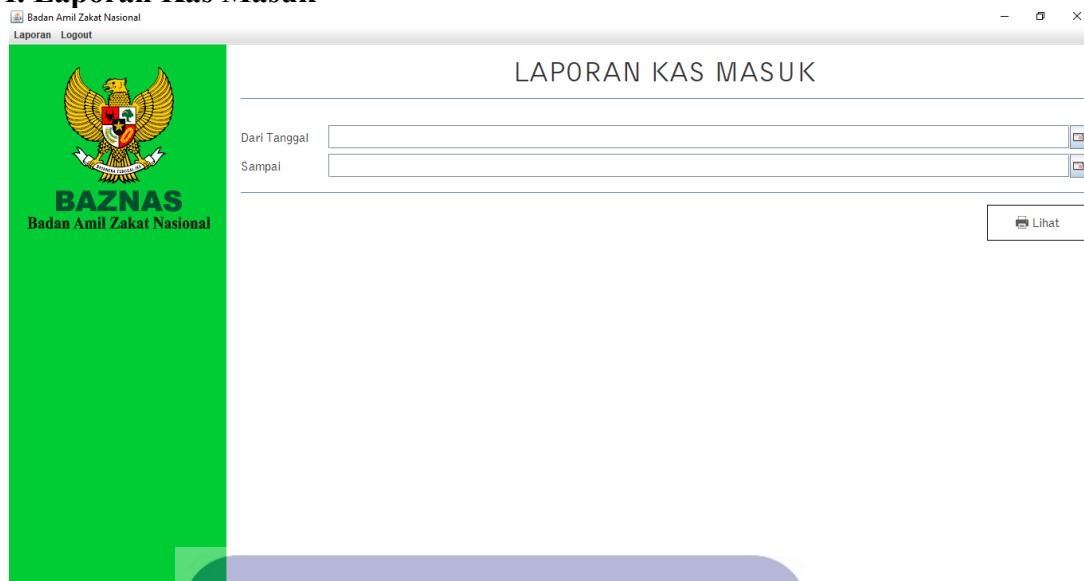
Jumlah Kas

Akun Kredit

Keterangan

Gambar III.47.
Form Kas Keluar

I. Laporan Kas Masuk



Badan Amil Zakat Nasional
Laporan Logout

BAZNAS
Badan Amil Zakat Nasional

LAPORAN KAS MASUK

Dari Tanggal

Sampai

Lihat

Gambar III.48.
Laporan Kas Masuk

J. Laporan Kas Keluar



Badan Amil Zakat Nasional
Laporan Logout

BAZNAS
Badan Amil Zakat Nasional

LAPORAN KAS KELUAR

Dari Tanggal

Sampai

Lihat

Gambar III.49.
Laporan Kas Keluar

3.5. Implementasi

3.5.1. Code Generation

A. Form Kas Masuk

```
package form;
```

```
import java.awt.Dimension;
import java.awt.Toolkit;
import java.text.SimpleDateFormat;
import java.util.List;
import java.util.ListIterator;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import model.KasMasuk;
import model.DetailKasMasuk;
import model.Perkiraan;
import java.awt.Font;
import java.awt.event.KeyEvent;
import java.text.NumberFormat;
import java.util.Date;
import java.util.Locale;
import javax.swing.table.DefaultTableModel;
import main.Baznas;
import model.Muzakki;
```

```
public class FormKasMasuk extends javax.swing.JFrame {
```

```
    /**
     * Creates new form FormKasMasuk
     */
```

```
    String status = "";
    String no_kas_masuk = "";
```

```
    NumberFormat nf = NumberFormat.getCurrencyInstance(new Locale("id", "ID"));
```

```
    KasMasuk kasMasuk = new KasMasuk();
    DetailKasMasuk detailKasMasuk = new DetailKasMasuk();
    Perkiraan perkiraan = new Perkiraan();
    Muzakki muzakki = new Muzakki();
```

```
    DefaultComboBoxModel tm1 = null;
    DefaultComboBoxModel tm2 = null;
```

```
    DefaultTableModel model = null;
```

```
    public FormKasMasuk() {
        initComponents();
    }
```

```

Toolkit tk = Toolkit.getDefaultToolkit();
Dimension d = tk.getScreenSize();
int x, y;
x = (int) ((d.getWidth() - getSize().width) / 2);
y = (int) ((d.getHeight() - getSize().height) / 2);

tm1 = (DefaultComboBoxModel) takun1.getModel();
tm2 = (DefaultComboBoxModel) takun2.getModel();

model = (DefaultTableModel) tblkasmasuk.getModel();
tblkasmasuk.getTableHeader().setFont(new Font("BellGothic BT", 1, 16));
tblkasmasuk.setRowHeight(30);

setLocation(x, y);

daftarKasMasuk();
bersihInputan();
daftarAkun();
aktifNonaktif();
MyTab.setSelectedIndex(1);
}

private void daftarKasMasuk() {
    List<KasMasuk> listKasMasuk = kasMasuk.daftarKasMasuk();
    ListIterator listKas = listKasMasuk.listIterator();

    model.setRowCount(0);
    tblkasmasuk.repaint();
    String No = "";

    while (listKas.hasNext()) {
        kasMasuk = (KasMasuk) listKas.next();
        muzakki = muzakki.cariMuzakki(kasMasuk.getNpwz());
        List<DetailKasMasuk> listDetail =
detailKasMasuk.daftarDetailKas(kasMasuk.getNoKasMasuk());
        ListIterator lIterator = listDetail.listIterator();
        Object[] data = null;
        while (lIterator.hasNext()) {
            detailKasMasuk = (DetailKasMasuk) lIterator.next();
            perkiraan = new
Perkiraan().cariKodeAkun(detailKasMasuk.getKodeAkun());
            if (No.equalsIgnoreCase(kasMasuk.getNoKasMasuk())) {
                Object[] jdata = {
                    "",
                    "",
                    "",
                    "",
                    "",
                    ""
                };
            }
        }
    }
}

```

```

        "",
        detailKasMasuk.getKodeAkun(),
        perkiraan.getNamaAkun(),
        nf.format(detailKasMasuk.getDebet()),
        nf.format(detailKasMasuk.getKredit())
    };
    data = jdata;
} else {
    Object[] jdata = {
        kasMasuk.getNoKasMasuk(),
        kasMasuk.getTglKasMasuk(),
        kasMasuk.getNp wz(),
        muzakki.getNamaMz(),
        nf.format(kasMasuk.getJumlahKasMasuk()),
        kasMasuk.getKeterangan(),
        detailKasMasuk.getKodeAkun(),
        perkiraan.getNamaAkun(),
        nf.format(detailKasMasuk.getDebet()),
        nf.format(detailKasMasuk.getKredit())
    };
    data = jdata;
}
model.addRow(data);
No = kasMasuk.getNoKasMasuk();
}
}

private void daftarKasMasuk(Date tgl1, Date tgl2) {
    List<KasMasuk> listKasMasuk = kasMasuk.cariKasMasuk(new
    java.sql.Date(tgl1.getTime()), new java.sql.Date(tgl2.getTime()));
    ListIterator li = listKasMasuk.listIterator();

    model.setRowCount(0);
    tblkasmasuk.repaint();
    String No = "";

    double totaldebet = 0, totalcredit = 0;

    while (li.hasNext()) {
        kasMasuk = (KasMasuk) li.next();
        muzakki = muzakki.cariMuzakki(kasMasuk.getNp wz());
        List<DetailKasMasuk> listDetail =
        detailKasMasuk.daftarDetailKas(kasMasuk.getNoKasMasuk());
        ListIterator lIterator = listDetail.listIterator();
        Object[] data = null;
        while (lIterator.hasNext()) {

```

```

        detailKasMasuk = (DetailKasMasuk) Iterator.next();
        perkiraan = new
Perkiraan().cariKodeAkun(detailKasMasuk.getKodeAkun());
        if (No.equalsIgnoreCase(kasMasuk.getNoKasMasuk())) {
            Object[] jdata = {
                "",
                "",
                "",
                "",
                "",
                "",
                "",
                detailKasMasuk.getKodeAkun(),
                perkiraan.getNamaAkun(),
                nf.format(detailKasMasuk.getDebet()),
                nf.format(detailKasMasuk.getKredit())
            };
            data = jdata;
        } else {
            Object[] jdata = {
                kasMasuk.getNoKasMasuk(),
                kasMasuk.getTglKasMasuk(),
                kasMasuk.getNp wz(),
                muzakki.getNamaMz(),
                kasMasuk.getKeterangan(),
                nf.format(kasMasuk.getJumlahKasMasuk()),
                detailKasMasuk.getKodeAkun(),
                perkiraan.getNamaAkun(),
                nf.format(detailKasMasuk.getDebet()),
                nf.format(detailKasMasuk.getKredit())
            };
            data = jdata;
        }
        model.addRow(data);
        No = kasMasuk.getNoKasMasuk();
    }
}

private void cariMuzakki() {
    if (!"".equals(tnpwz.getText())) {
        muzakki = muzakki.cariMuzakki(tnpwz.getText());
        if (muzakki.getNamaMz() != null) {
            tnama_muzakki.setText(muzakki.getNamaMz());
            takunl.requestFocus();
        } else {
            JOptionPane.showMessageDialog(rootPane, "Muzakki yang Anda cari
tidak ditemukan!", "Error", JOptionPane.ERROR_MESSAGE);
            tnpwz.setText("");
        }
    }
}

```

```

        tnama_muzakki.setText("");
    }
} else {
    tnama_muzakki.setText("");
}
}

private void daftarAkun() {
    List<Perkiraan> lp = perkiraan.DaftarPerkiraan();
    ListIterator li = lp.listIterator();
    tm1.removeAllElements();
    tm2.removeAllElements();

    tm1.addElement("");
    tm2.addElement("");

    while (li.hasNext()) {
        Perkiraan pkr = (Perkiraan) li.next();
        tm1.addElement(pkr.getKodeAkun());
        tm2.addElement(pkr.getKodeAkun());
    }
}

private void aktifNonaktif() {
    btntambah.setEnabled(!"".equals(this.status));
    btnsimpan.setEnabled(!"".equals(this.status));
    btnbatal.setEnabled(!"".equals(this.status));

    tno_kas_masuk.setEditable(false);
    ttanggal_kas_masuk.setEditable(false);
    tketerangan.setEditable(!"".equals(this.status));
    takun1.setEnabled(!"".equals(this.status));
    takun2.setEnabled(!"".equals(this.status));

    tnpwz.setEditable(!"".equals(this.status));
    MyTab.setEnabledAt(0, !"".equals(this.status));
    MyTab.setEnabledAt(1, "".equals(this.status));
}

private void bersihInputan() {
    tno_kas_masuk.setText("");
    SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");

    ttanggal_kas_masuk.setText(sdf.format(new java.util.Date()));
    tketerangan.setText("");
    takun1.setSelectedItem("");
    takun2.setSelectedItem("");
    tnpwz.setText("");
}

```

```

tnm_akun1.setText("");
tnm_akun2.setText("");

}

/**
 * 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() {

    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jLabel1 = new javax.swing.JLabel();
    jSeparator1 = new javax.swing.JSeparator();
    MyTab = new javax.swing.JTabbedPane();
    jPanel1 = new javax.swing.JPanel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    tnpwz = new javax.swing.JTextField();
    tno_kas_masuk = new javax.swing.JTextField();
    ttanggal_kas_masuk = new javax.swing.JTextField();
    takun1 = new javax.swing.JComboBox<>();
    tketerangan = new javax.swing.JTextField();
    takun2 = new javax.swing.JComboBox<>();
    tnm_akun2 = new javax.swing.JTextField();
    tjumlah_kas = new javax.swing.JTextField();
    jLabel8 = new javax.swing.JLabel();
    tnm_akun1 = new javax.swing.JTextField();
    jLabel9 = new javax.swing.JLabel();
    jLabel10 = new javax.swing.JLabel();
    tnama_muzakki = new javax.swing.JTextField();
    btncari_muzakki = new javax.swing.JButton();
    jPanel2 = new javax.swing.JPanel();
    jScrollPane2 = new javax.swing.JScrollPane();
    tblkasmasuk = new javax.swing.JTable();
    tgl2 = new com.toedter.calendar.JDateChooser();
    tgl1 = new com.toedter.calendar.JDateChooser();
    jLabel12 = new javax.swing.JLabel();
    btncari = new javax.swing.JButton();
    btnbatal = new javax.swing.JButton();
    btnsimpan = new javax.swing.JButton();
    btntambah = new javax.swing.JButton();

```

```

jTable1.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"
    }
));
jScrollPane1.setViewportViewView(jTable1);

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Form Jurnal Umum");

jLabel1.setFont(new java.awt.Font("BellGothic BT", 0, 36)); // NOI18N
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("Form Penerimaan Kas");

MyTab.setFont(new java.awt.Font("BellGothic BT", 0, 18)); // NOI18N

jLabel3.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
jLabel3.setText("No KM");

jLabel4.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
jLabel4.setText("Tanggal KM");

jLabel7.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
jLabel7.setText("Np wz");

jLabel5.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
jLabel5.setText("Keterangan");

tnp wz.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
tnp wz.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        tnp wzKeyPressed(evt);
    }
});

tno_kas_masuk.setEditable(false);
tno_kas_masuk.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N

ttanggal_kas_masuk.setEditable(false);
ttanggal_kas_masuk.setFont(new java.awt.Font("BellGothic BT", 0, 15)); //
NOI18N

```

```

takun1.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
takun1.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        takun1ItemStateChanged(evt);
    }
});

```

```

tketerangan.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N

```

```

takun2.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
takun2.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        takun2ItemStateChanged(evt);
    }
});

```

```

tnm_akun2.setEditable(false);
tnm_akun2.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
tnm_akun2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tnm_akun2ActionPerformed(evt);
    }
});

```

```

tjumlah_kas.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N

```

```

jLabel8.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
jLabel8.setText("Jumlah Kas");

```

```

tnm_akun1.setEditable(false);
tnm_akun1.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
tnm_akun1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tnm_akun1ActionPerformed(evt);
    }
});

```

```

jLabel9.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
jLabel9.setText("Akun Kredit");

```

```

jLabel10.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
jLabel10.setText("Akun Debet");

```

```

tnama_muzakki.setEditable(false);
tnama_muzakki.setFont(new java.awt.Font("BellGothic BT", 0, 15)); //
NOI18N

```

```

        btncari_muzakki.setFont(new java.awt.Font("BellGothic BT", 0, 15)); //
        NOI18N
        btncari_muzakki.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/form/image/cari.png"))); //
        NOI18N
        btncari_muzakki.setPreferredSize(new java.awt.Dimension(49, 26));
        btncari_muzakki.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btncari_muzakkiActionPerformed(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(10, 10, 10)

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel3)
                .addComponent(jLabel4)
                .addComponent(jLabel5)
                .addComponent(jLabel7)
                .addComponent(jLabel10)
                .addComponent(jLabel8)
                .addComponent(jLabel9))
                .addGap(22, 22, 22)

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(tno_kas_masuk,
                    javax.swing.GroupLayout.PREFERRED_SIZE, 102,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(takun1, javax.swing.GroupLayout.PREFERRED_SIZE,
                    218, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(tnm_akun1,
                    javax.swing.GroupLayout.PREFERRED_SIZE, 381,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(tjumlah_kas,
                    javax.swing.GroupLayout.PREFERRED_SIZE, 218,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(takun2, javax.swing.GroupLayout.PREFERRED_SIZE,
                    190, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(tnm_akun2,
javax.swing.GroupLayout.PREFERRED_SIZE, 381,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tketerangan,
javax.swing.GroupLayout.PREFERRED_SIZE, 324,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tnama_muzakki,
javax.swing.GroupLayout.PREFERRED_SIZE, 283,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(jPanel1Layout.createSequentialGroup())

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
TRAILING, false)
        .addComponent(tnpwz,
javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(ttanggal_kas_masuk,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 160, Short.MAX_VALUE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(btncari_muzakki,
javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(437, Short.MAX_VALUE))
    );
    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
        .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(37, 37, 37)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addComponent(jLabel3,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tno_kas_masuk,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(6, 6, 6)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(ttanggal_kas_masuk,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(6, 6, 6)
            .addComponent(jLabel7,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel1Layout.createSequentialGroup()

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

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addComponent(btncari_muzakki,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tnpwz,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(37, 37, 37)
            .addComponent(jLabel10,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel1Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(tnama_muzakki,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(takun1,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(tnm_akun1,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)

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

```



```

        .addComponent(jLabel12)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(tgl1, javax.swing.GroupLayout.PREFERRED_SIZE,
        188, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(tgl2, javax.swing.GroupLayout.PREFERRED_SIZE,
        188, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(btncari,
        javax.swing.GroupLayout.PREFERRED_SIZE, 37,
        javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap()
    );
    jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
            .addGap(23, 23, 23)

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
CENTER)
            .addComponent(btncari, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(tgl2, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(tgl1, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel12))
            .addGap(18, 18, 18)
            .addComponent(jScrollPane2,
javax.swing.GroupLayout.DEFAULT_SIZE, 333, Short.MAX_VALUE)
            .addContainerGap()
        );

    jPanel2Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new
java.awt.Component[] {btncari, tgl2});

    MyTab.addTab("Data", jPanel2);

    MyTab.setSelectedIndex(1);

    btnbatal.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N

```

```

        btnbatal.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/form/image/batal.png"))); //
NOI18N
        btnbatal.setText("Batal");
        btnbatal.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnbatalActionPerformed(evt);
            }
        });

        btnsimpan.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
        btnsimpan.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/form/image/simpan.png"))); //
NOI18N
        btnsimpan.setText("Simpan");
        btnsimpan.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnsimpanActionPerformed(evt);
            }
        });

        btntambah.setFont(new java.awt.Font("BellGothic BT", 0, 15)); // NOI18N
        btntambah.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/form/image/tambah2.png"))); //
NOI18N
        btntambah.setText("Tambah");
        btntambah.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btntambahActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(MyTab,
javax.swing.GroupLayout.Alignment.TRAILING)
                        .addComponent(jSeparator1)
                        .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

```

```

        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(btntambah,
javax.swing.GroupLayout.PREFERRED_SIZE, 115,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(btnsimpan,
javax.swing.GroupLayout.PREFERRED_SIZE, 119,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(btnbatal,
javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap())
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addGap(15, 15, 15)
        .addComponent(jLabel1)
        .addGap(18, 18, 18)
        .addComponent(jSeparator1,
javax.swing.GroupLayout.PREFERRED_SIZE, 2,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(MyTab, javax.swing.GroupLayout.PREFERRED_SIZE,
452, Short.MAX_VALUE)

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

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.CENTE
R)
        .addComponent(btntambah,
javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnsimpan,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnbatal,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap())
    );

    layout.linkSize(javax.swing.SwingConstants.VERTICAL, new
java.awt.Component[] {btnbatal, btnsimpan, btntambah});

```

```

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

private void btntambahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.status = "Tambah";
    this.no_kas_masuk = "";
    bersihInputan();
    aktifNonaktif();
    tno_kas_masuk.setText(kasMasuk.buatNoKasMasuk());
    tnpwz.requestFocus();
    MyTab.setSelectedIndex(0);
}

private void btnsimpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if ("".equals(tno_kas_masuk.getText())
        || "".equals(ttanggal_kas_masuk.getText())
        || "".equals(tketerangan.getText())
        || "".equals(tnpwz.getText())
        || "".equals(tjumlah_kas.getText())
        || "".equals(takun1.getSelectedItem().toString())
        || "".equals(takun2.getSelectedItem().toString())) {
        JOptionPane.showMessageDialog(rootPane, "Mohon isi dulu semua data",
            "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }

    if (!Baznas.validasiInput(tjumlah_kas.getText(), "[0..9\\.]\\{1,7}\\$")) {
        tjumlah_kas.requestFocus();
        JOptionPane.showMessageDialog(rootPane, "Jumlah kas hanya boleh berupa
            angka", "Galat", JOptionPane.ERROR_MESSAGE);
        return;
    }

    kasMasuk.setNoKasMasuk(tno_kas_masuk.getText());
    kasMasuk.setTglKasMasuk(new java.sql.Date(new java.util.Date().getTime()));
    kasMasuk.setKeterangan(tketerangan.getText());
    kasMasuk.setNpwz(tnpwz.getText());
    kasMasuk.setJumlahKasMasuk(Double.parseDouble(tjumlah_kas.getText()));
    kasMasuk.setKdAdmin(Baznas.kd_admin);

    if ("Tambah".equals(status)) {
        String[] Akun;
        if (kasMasuk.tambahKasMasuk() > 0) {
            detailKasMasuk.setNoKasMasuk(tno_kas_masuk.getText());

```

```

detailKasMasuk.setKodeAkun(takun1.getSelectedItem().toString());
detailKasMasuk.setDebet(Double.parseDouble(tjumlah_kas.getText()));
detailKasMasuk.setKredit(0);
detailKasMasuk.tambahDetailKasMasuk();

detailKasMasuk.setKodeAkun(takun2.getSelectedItem().toString());
detailKasMasuk.setDebet(0);
detailKasMasuk.setKredit(Double.parseDouble(tjumlah_kas.getText()));
detailKasMasuk.tambahDetailKasMasuk();

JOptionPane.showMessageDialog(rootPane, "Data kas masuk berhasil
disimpan!", "Info", JOptionPane.INFORMATION_MESSAGE);
    } else {
        JOptionPane.showMessageDialog(rootPane, "Data kas masuk gagal
disimpan.\n" + kasMasuk.getPesanError(), "Error",
JOptionPane.ERROR_MESSAGE);
        return;
    }
}

this.status = "";
this.no_kas_masuk = "";
bersihInputan();
daftarKasMasuk();
aktifNonaktif();
MyTab.setSelectedIndex(1);
}

private void btnbatalActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.status = "";
    this.no_kas_masuk = "";
    bersihInputan();
    aktifNonaktif();
    MyTab.setSelectedIndex(1);
}

private void tnm_akun2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void btncariActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (tgl1.getDate() == null || tgl2.getDate() == null) {
        daftarKasMasuk();
    }
}

```

```

    } else {
        daftarKasMasuk(tgl1.getDate(), tgl2.getDate());
    }
}

private void tnm_akun1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void takun1ItemStateChanged(java.awt.event.ItemEvent evt) {
    // TODO add your handling code here:
    try {
        String akun = takun1.getSelectedItem().toString();
        perkiraan = perkiraan.cariKodeAkun(akun);
        tnm_akun1.setText(perkiraan.getNamaAkun());
    } catch (Exception ex) {

    }

}

private void takun2ItemStateChanged(java.awt.event.ItemEvent evt) {
    // TODO add your handling code here:
    try {
        String akun = takun2.getSelectedItem().toString();
        perkiraan = perkiraan.cariKodeAkun(akun);
        tnm_akun2.setText(perkiraan.getNamaAkun());
    } catch (Exception ex) {

    }

}

private void tnpwzKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if(evt.getKeyCode() == KeyEvent.VK_ENTER){
        cariMuzakki();
    }
}

private void btncari_muzakkiActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    tnpwz.setText(new FormCariMuzakki(this, true).cariMuzakki());
    cariMuzakki();
}

/**
 * @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 ("Windows".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

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

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

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

java.util.logging.Logger.getLogger(FormKasMasuk.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
    }
    //</editor-fold>
    //</editor-fold>
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new FormKasMasuk().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JTabbedPane MyTab;
private javax.swing.JButton btnbatal;
private javax.swing.JButton btncari;
private javax.swing.JButton btncari_muzakki;

```

```

private javax.swing.JButton btnsimpan;
private javax.swing.JButton btntambah;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JTable jTable1;
private javax.swing.JComboBox<String> takun1;
private javax.swing.JComboBox<String> takun2;
private javax.swing.JTable tblkasmasuk;
private com.toedter.calendar.JDateChooser tgl1;
private com.toedter.calendar.JDateChooser tgl2;
private javax.swing.JTextField tjumlah_kas;
private javax.swing.JTextField tketerangan;
private javax.swing.JTextField tnama_muzakki;
private javax.swing.JTextField tnm_akun1;
private javax.swing.JTextField tnm_akun2;
private javax.swing.JTextField tno_kas_masuk;
private javax.swing.JTextField tnpwz;
private javax.swing.JTextField ttanggal_kas_masuk;
// End of variables declaration
}

```

3.5.2. Blackbox Testing

A. Form Login Accounting

Tabel III.24.
Blackbox Testing Form Login Accounting

No	Skenario Pengujian	Test Case	Tujuan Hasil	Hasil Pengujian	Kesimpulan
1	Kode Admin dan Kata Sandi tidak diisi kemudian klik tombol <i>Login</i>	Kode Admin : (Kosong) Kata Sandi :	Sistem akan menolak dan menampilkan pesan "Kode	Sesuai Harapan	<i>Valid</i>

		(Kosong)	Admin dan Kata Sandi tidak boleh kosong”		
2	Mengisi kode admin sesuai dengan <i>database</i> dan mengisi kata sandi tidak sesuai kemudian klik tombol <i>Login</i>	Kode Admin : (ADM-003) Kata Sandi : (11111)	Sistem akan menolak dan menampilkan pesan “Kata Sandi Salah”	Sesuai Harapan	<i>Valid</i>
3	Mengisi kode admin tidak sesuai dengan <i>database</i> dan mengisi kata sandi sesuai kemudian klik tombol <i>Login</i>	Kode Admin : (AB000) Kata Sandi : (00000)	Sistem akan menolak dan menampilkan pesan “Kode Admin Salah”	Sesuai Harapan	<i>Valid</i>
4	Mengisi kode admin dan kata sandi sesuai dengan <i>database</i> kemudian klik tombol <i>Login</i>	Kode Admin : (ADM-003) Kata Sandi : (00000)	Sistem akan menolak dan menampilkan pesan “Selamat Datang Dini Sri Haryati, Anda Login Sebagai Accounting”	Sesuai Harapan	<i>Valid</i>

B. *Form* Transaksi Kas Masuk

Tabel III.25.
***Blackbox Testing Form* Transaksi Kas Masuk**

No	Skenario Pengujian	<i>Tes Case</i>	Tujuan Hasil	Hasil Pengujian	Kesimpulan
1	<i>Form</i> Transaksi dikosongkan	No Transaksi : (KM00001) Tanggal : (01/07/2019) NPWZ : (Kosong) Akun Debet : (Kosong) Jumlah Kas : (Kosong) Akun Kredit : (Kosong) Keterangan : (Kosong)	Sistem akan menolak akses dan menampilkan pesan “Data Belum Terisi”	Sesuai Harapan	<i>Valid</i>
2	<i>Form</i> di isi sesuai dengan transaksi	No Transaksi : (KM00001) Tanggal : (01/07/2019) NPWZ : (MZX00001) (Khofifah) Akun Debet : (10101) (Kas) Jumlah Kas : (Rp.125.000) Akun Kredit : (40100) (Penerimaan	Sistem akan menerima akses dan menampilkan pesan “data berhasil disimpan”	Sesuai harapan	<i>Valid</i>

		ZIS) Keterangan : (Setor Zakat)			
--	--	---------------------------------------	--	--	--

3. Form Transaksi Kas Keluar

Tabel III.26.
Blackbox Testing Form Transaksi Kas Keluar

No	Skenario Pengujian	Tes Case	Tujuan Hasil	Hasil Pengujian	Kesimpulan
1	Form Transaksi dikosongkan	No Transaksi : (KK00001) Tanggal : (01/07/2019) ID Mustahik : (Kosong) Akun Debet : (Kosong) Jumlah Kas : (Kosong) Akun Kredit : (Kosong) Keterangan : (Kosong)	Sistem akan menolak akses dan menampilkan pesan “Data Belum Terisi”	Sesuai Harapan	<i>Valid</i>
2	Form di isi sesuai dengan transaksi	No Transaksi : (KM00001) Tanggal : (01/07/2019) ID Mustahik : (MST00002) (Ami Priyan Saputra) Akun Debet : (50100)	Sistem akan menerima akses dan menampilkan pesan “data berhasil disimpan”	Sesuai harapan	<i>Valid</i>

		(Penyaluran ZIS) Jumlah Kas : (Rp.150.000) Akun Kredit : (10101) (Kas) Keterangan : (Penunggu Pasien)			
--	--	---	--	--	--

3.5.3. Spesifikasi *Hardware* dan *Software*

Tabel III.27.
Spesifikasi *Hardware* dan *Software*

Kebutuhan	Keterangan
Sistem	Windows 10
Processor	Intel Celereon N4000, up to 2.6GHz
RAM	4 GB
Harddisk	1 TB
Monitor	14"
Keyboard	PC/AT Enhanced PS/2 Keyboard (101/102-Key)
Printer	Epson L360
Mouse	Logitech M165
Software	Netbeans IDE 8.1 Bahasa Pemrograman : Java Netbeans Aplikasi Pendukung : Xampp v3.2.1 Database : MySQL Anti Virus : Smadav 2019