BAB 3

PEMBAHASAN

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya untuk mengidentifikasikan masalah-masalah, dan hambatan-hambatan sehingga dapat diusulkan kebutuhan-kebutuhan untuk perbaikannya[4]. Tahap analisis merupakan tahap yang paling kritis dan sangat penting, karena kesalahan didalam tahap ini akan menyebabkan juga kesalahan di tahap selanjutnya.

3.1.1 Analisis Masalah

Analisis masalah merupakan sebuah Perkiraan dari permasalahan yang akan diuraikan dalam prosedur-prosedur membangun aplikasi. Analisis permasalahan dari prosedur yang ada, yaitu:

1. Proses pengolahan data pegawai menggunakan *Microsoft Word* dan pencatatan secara manual ke dalam buku besar, sehingga terjadi penumpukan kertas dan juga pembuatan laporan yang menjadi lambat.
2. Sering terjadi kesalahan-kesalahan atau ketidaktepatan data keuangan dalam hal proses perhitungan gaji pegawai.
3. Pengelolaan data gaji dianggap terlalu merepotkan, karena terlalu banyak melewati proses.

Dari pihak Balai Besar Pengembangan Budidaya Air Tawar Sukabumi itu sendiri memang telah lama memerlukan suatu sistem aplikasi yang dapat menyajikan informasi mengenai data-data kepegawaian. Karena tingkat kebutuhan akan informasi mengenai kepegawaian semakin meningkat sehingga dituntut agar dapat mempermudah sistemnya yaitu dengan

membuat suatu sistem aplikasi kepegawaian yang dapat menyajikan informasi.

3.1.2 Analisis Prosedur yang sedang berjalan

Analisis prosedur merupakan kegiatan menganalisis prosedur-prosedur kerja yang terjadi pada sistem yang sedang berjalan. Adapun hasil dari kegiatan analisis ini berupa gambaran nyata dari urutan kegiatan-kegiatan yang dilakukan oleh unit-unit organisasi khususnya dalam kegiatan-kegiatan yang berhubungan dengan pengolahan data.

3.1.2.1 Prosedur Penyimpanan data pegawai baru

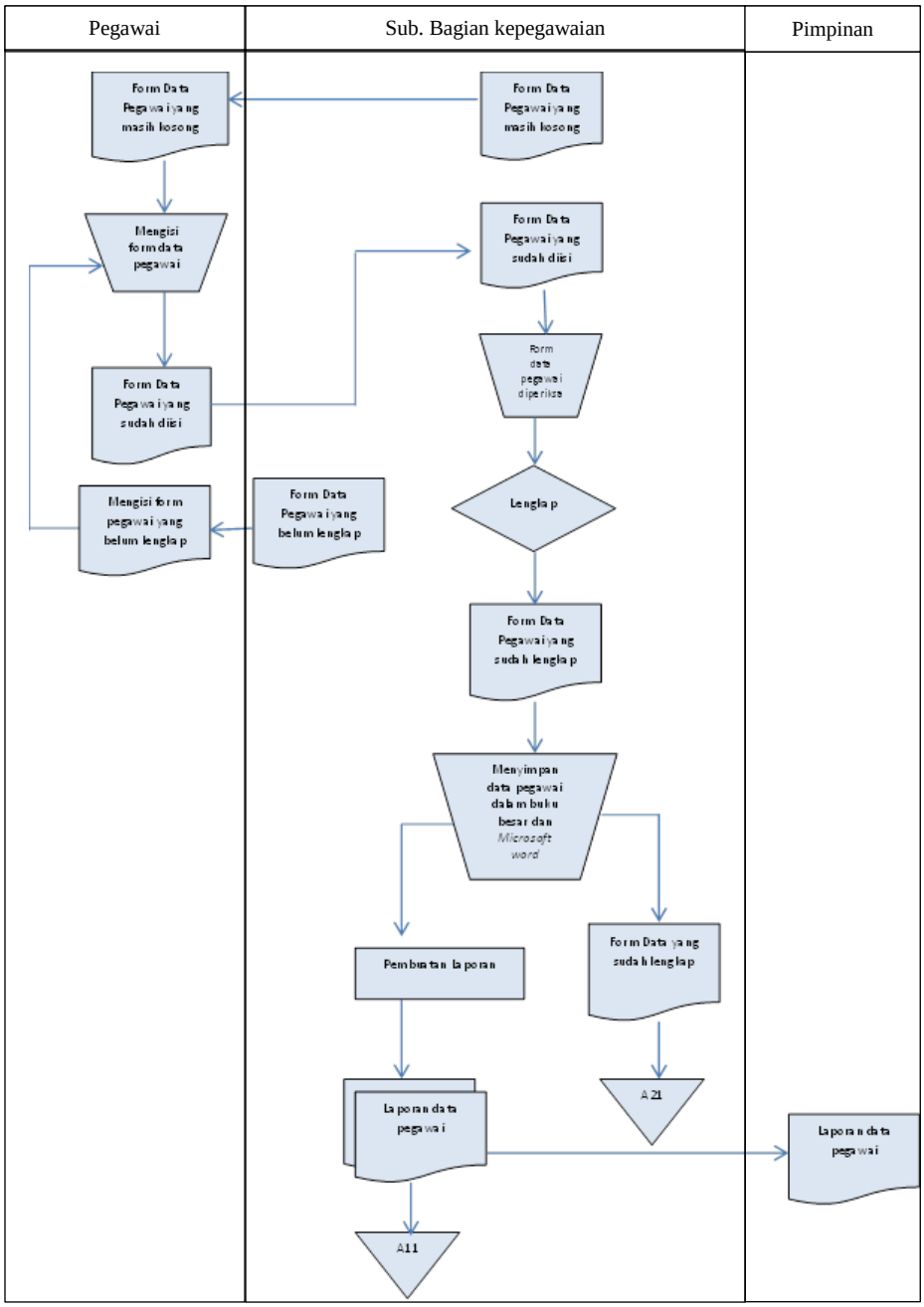
Sub bagian kepegawaian memberikan form data pegawai kepada pegawai.

Form data pegawai diisi

Semua data yang sudah diisi berupa kelengkapan data pegawai yang diperoleh dari pegawai diserahkan kepada bagian SDM lalu diperiksa, bila form data pegawai lengkap langsung dicatat dalam buku besar dan *Microsoft word* lalu dibuat laporannya, bila form data pegawai tidak lengkap maka dikembalikan lagi kepada pegawai.

Form data pegawai yang sudah lengkap diarsipkan.

Laporan data pegawai diserahkan kepada pimpinan dan diarsipkan di bagian kepegawaian.



Gambar 3.1 Flowmap prosedur penyimapanan data pegawai baru kepegawaian BBPBAT Sukabumi yang sedang berjalan

Keterangan gambar:

A1: Arsip Laporan data pegawai

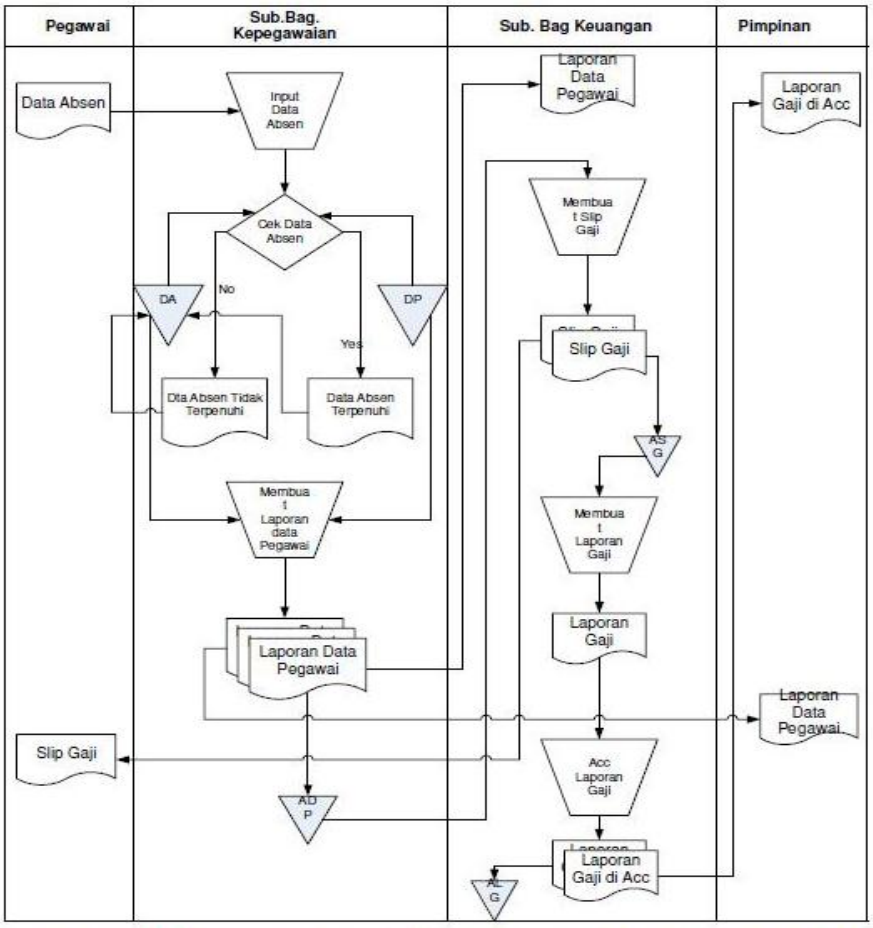
A2: Arsip form data pegawai

3.1.2.2 Prosedur Penggajian

prosedur Sistem kepegawaian di kantor Balai Besar Pengembangan Budidaya Air Tawar Sukabumi adalah seperti berikut:

1. Pegawai melakukan absen rutin di bagian kepegawaian.
2. Bagian kepegawaian menyesuaikan dengan data pegawai dan data absen yang ada.
3. Selanjutnya bagian kepegawaian merekap data absensi baik yang hadir maupun tidak
4. Kemudian bagian kepegawaian akan membuat laporan data pegawai yang menyesuaikan dengan data absen dan pegawai sebanyak 3 lembar, lembar pertama di serahkan kepada bagian keuangan, lembar kedua di serahkan kepada pimpinan,dan lembar ketiga disimpan bagian kepegawaian sebagai arsip.
5. Kemudian bagian keuangan akan menghitung gaji pegawai setelah disesuaikan dengan laporan data pegawai.
6. dengan Kemudian bagian keuangan akan membuat slip gaji sesuai laporan sebanyak 2 lembar, lembar pertama akan dijadikan arsip di bagian keuangan, dan lembar kedua akan diserahkan pada pegawai.
7. Selanjutnya bagian keuangan akan membuat laporan gaji berdasarkan dari arsip gaji di bagian keuangan.
8. kemudian laporan gaji tersebut akan di Acc oleh bagian keuangan sebanyak 2 lembar. Lembar pertama akan diberikan akan diberikan kepada pimpinan sedangkan lembar kedua akan diarsipkan dibagian keuangan.

Untuk Diagram Alir Dokumen (Flowmap) yang sedang berjalan dapat dilihat pada gambar 3.2 di bawah ini :



Gambar 3. 2 Flowmap prosedur penggajian kepegawaian BBPBAT Sukabumi yang sedang berjalan

Keterangan gambar :

DA : Data Absen

DP : Data Pegawai

ADP : Arsip Data Pegawai

ASG : Arsip Slip Gaji

ALP : Arsip Laporan Penggajian

3.1.3 Evaluasi Sistem Yang sedang Berjalan

Berdasarkan analisis sistem yang sedang berjalan di atas dapat disimpulkan bahwa perlu diadakannya perbaikan sistem yaitu :

1. Pembangunan aplikasi yang dapat melakukan pengolahan dan perhitungan gaji karyawan.
2. Mengatasi kelemahan-kelemahan yang terjadi pada sistem yang belum terkomputerisasi.
3. Menyediakan informasi data pegawai yang tepat dan akurat serta membuat laporan Gaji cepat akurat bagi pihak yang memerlukannya.

3.1.4 Aturan Bisnis

Analisis aturan bisnis dilakukan untuk menganalisis dan menyimpulkan proses bisnis apa saja yang terkait dan membatasi pada berjalannya suatu proses sehingga memberi gambaran jelas dan asumsi yang tepat dengan alur berlangsungnya suatu proses. Berikut adalah aturan bisnis yang terdapat di dalam bagian kepegawaian kantor Balai Besar Pengembangan Budidaya Air Tawar Sukabumi:

3.1.4.1 Aturan Bisnis Pencatatan data Pegawai

Pada proses pencatatan pegawai baru terdapat aturan-aturan bisnis, diantaranya adalah sebagai berikut :

1. Penerimaan pegawai dilakukan oleh tim penyeleksi yang sudah dibentuk oleh bagian kepegawaian.
2. Proses penerimaan pegawai baru yang dilakukan oleh tim penyeleksi meliputi persyaratan pelamar kerja, kriteria sesuai kebutuhan perusahaan, lulus uji dan seleksi praktek, serta *interview*.
3. Data pegawai baru akan diberikan kepada bagian kepegawaian setelah melalui proses penerimaan yang dilakukan oleh tim penyeleksi untuk proses pencatatan pegawai baru.

4. Bagian kepegawaian akan melakukan pencatatan data pegawai baru kemudian akan diperiksa kembali oleh kepala bagian kepegawaian.

3.1.4.2 Aturan Bisnis Penggajian

Pada proses penggajian terdapat aturan-aturan bisnis, diantaranya adalah sebagai berikut :

1. Gaji pegawai diberikan setiap akhir bulan.
2. Rincian variabel penggajian pegawai terdiri dari gaji pokok, tunjangan-tunjangan, serta potongan.
3. Apabila pegawai ingin mengajukan pinjaman, maka harus dengan persetujuan atasan.
4. Pegawai yang mempunyai pinjaman akan mendapatkan potongan gaji dengan cicilan pinjaman yang sudah ditentukan.
5. Berikut adalah rincian komponen gaji
   a. Gaji Pokok = Upah pegawai sesuai perjanjian
   b. Tunjangan = Tunjangan Jabatan + Tunjangan Makan + Transport
   c. Potongan = Pinjaman, Asuransi
   d. Jumlah Gaji = ((Gaji Pokok + Tunjangan) - Potongan)

3.1.5 SKPL

3.1.5.1 Fungsional

Tabel 3.1 SKPL fungsional

No.	Nama	Nomor	Dsekripsi
1	*Interface* Login Petugas	SKPL-F-001	Login petugas untuk melakukan pengolahan data sesuai dengan username petugas.
2	Pengolahan Data	SKPL-F-002	Sistem dapat menambahkan, mengubah ataupun menghapus data karyawan, data golongan, data jabatan, data potongan dan data gaji.
3	Cetak Laporan	SKPL-F-002	Sistem dapat mencetak data karyawan, data golongan, data jabatan, data

			potongan dan data gaji.
4	*Save*	SKPL-F-004	Digunakan untuk menyimpan data pada Database

3.1.5.2 Non Fungsional

Tabel 3.2 SKPL Non Fungsional

No.	Nama	Nomor	Dsekripsi
1	*Interface*	SKPL-NF-001	Tampilan yang mudah untuk petugas untuk memenuhi keperluan tugasnya.
2	Operasi *software*	SKPL-NF-002	*Software* dapat beroperasi selama kantor beroperasi
3	Kecepatan penyampaian data	SKPL-NF-003	Tidak terjadi keterlambatan penyampaian data.

3.1.6 Analisis Kebutuhan Non Fungsional

Analisis non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

3.1.3.1 Analisis Perangkat Keras

Pembangunan sistem kepegawaian di kantor Balai Besar Pengembangan Budidaya Air Tawar Sukabumi terutama penggunaan perangkat keras yang akan mendukung pembuatan sistem dari awal pembuatan hingga pengujian sistem.

Adapun perangkat keras yang dibutuhkan yaitu dengan spesifikasi yang dapat dilihat di tabel 3.3.

Tabel 3.3 Spesifikasi Perangkat Keras Kebutuhan Sistem

No	Peangkat Keras	Keadaan yang sekarang	Keadaan yang dibutuhkan
1	Prosessor	Intel Pentium core i3 2.2 Ghz	Intel Pentium 4 2.0 Ghz
2	Monitor	Monitor LCD 19 inch	Monitor VGA 15 inch
3	VGA	VGA card on-board 128 MB	VGA card on-board 128 MB
4	Memori	Memori DDR3 2 GB	Memori DDR3 1 Gb
5	*Hard Disk*	160GB (Kapasitas bebas minimal 100 GB)	Kapasitas bebas minimal 2 GB
6	*Optical Drive*	DVD-R 52x	CD-R
7	*Keyboard*	*Keyboard*	*Keyboard*
8	*Mouse*	*Mouse*	*Mouse*
9	*Printer*	*Printer* multifungsi warna	*Printer* hitam putih

Setelah evaluasi mengenai spesifikasi perangkat keras di kantor Balai Besar Pengembangan Budidaya Air Tawar Sukabumi dengan membandingkan kebutuhan spesifikasi minimal untuk pembuatan sistem, dapat disimpulkan spesifikasi perangkat keras di bagian kepegawaian sudah memenuhi kebutuhan perangkat keras dari sistem kepegawaian.

3.1.3.2 Analisis Perangkat Lunak

Pembangunan sistem kepegawaian di kantor Balai kepegawaian Besar Pengembangan Budidaya Air Tawar Sukabumi, selain kebutuhan perangkat keras yang harus diperhatikan juga harus mengevaluasi mengenai pemilihan perangkat lunak untuk mendukung sistem yang dibangun. Adapun spesifikasi perangkat lunak yang dibutuhkan dalam pembangunan sistem kepegawaian dapat dilihat pada tabel 3.4

Tabel 3.4 Spesifikasi Kebutuhan Perangkat Lunak Sistem Kepegawaian

NO	Perangkat Lunak	Keadaan yang sekarang	Keadaan yang dibutuhkan
1	Sistem Operasi	Windows 7 Ultimate	Windows XP SP3
2	Basis Data	Microsft access 2010	Microsft access 2010

.

Setelah evaluasi mengenai spesifikasi perangkat lunak di kantor Balai Besar Pengembangan Budidaya Air Tawar Sukabumi dengan membandingkan kebutuhan spesifikasi minimal untuk pembuatan sistem, dapat disimpulkan spesifikasi perangkat lunak di bagian kepegawaian sudah memenuhi kebutuhan perangkat lunak dari sistem kepegawaian.

3.1.3.3 Analisis Kebutuhan Pengguna (*user*)

Suatu sistem informasi akan berjalan optimal apabila ditunjang oleh perangkat pikir yang memiliki kemampuan dalam menjalankan suatu sistem informasi.

Adapun pengguna (*user*) yang dimiliki saat ini hanya satu petugas di bagian kepegawaian, dapat dilihat pada tabel 3.5

Tabel 3.5 pengguna (*user*) yang dimiliki saat ini

Pengguna	Tanggung Jawab	Pendidikan	Kemampuan	Pengalaman
Petugas	- Mengelola data pegawai dan data gaji. - Mencetak laporan data karyawan dan gaji.	Diploma 3 (D3)	Mampu menggunakan perangkat lunak *Office Suite* dalam menjalankan tugasnya dengan bantuan sistem yang diberikan.	-

Sedangkan pengguna (*user*) yang dibutuhkan untuk menjalankan sistem kepegawaian ini dapat dilihat pada tabel 3.6

Tabel 3.6 pengguna (*user*) yang dibutuhkan

Pengguna	Tanggung Jawab	Pendidikan	Kemampuan	Pengalaman
Petugas	- Mengelola data pegawai, data jabatan, data golongan dan data gaji. - Mencetak laporan data karyawan dan gaji.	Diploma 3 (D3)	Mampu menggunakan aplikasi dari sistem informasi yang dibangun untuk pengolahan kepegawaian.	-

Melihat perbandingan antara pengguna yang ada dengan pengguna sistem yang dibutuhkan untuk menjalankan aplikasi yang dibangun maka dapat disimpulkan bahwa diperlukan adanya pelatihan dan *tutorial* khusus dalam penggunaan sistem terhadap calon pengguna dari sistem informasi kepegawaian. Pelatihan perangkat pikir yang ada bertujuan agar pengguna sistem dapat sebaik mungkin menggunakan aplikasi yang dibangun secara optimal.

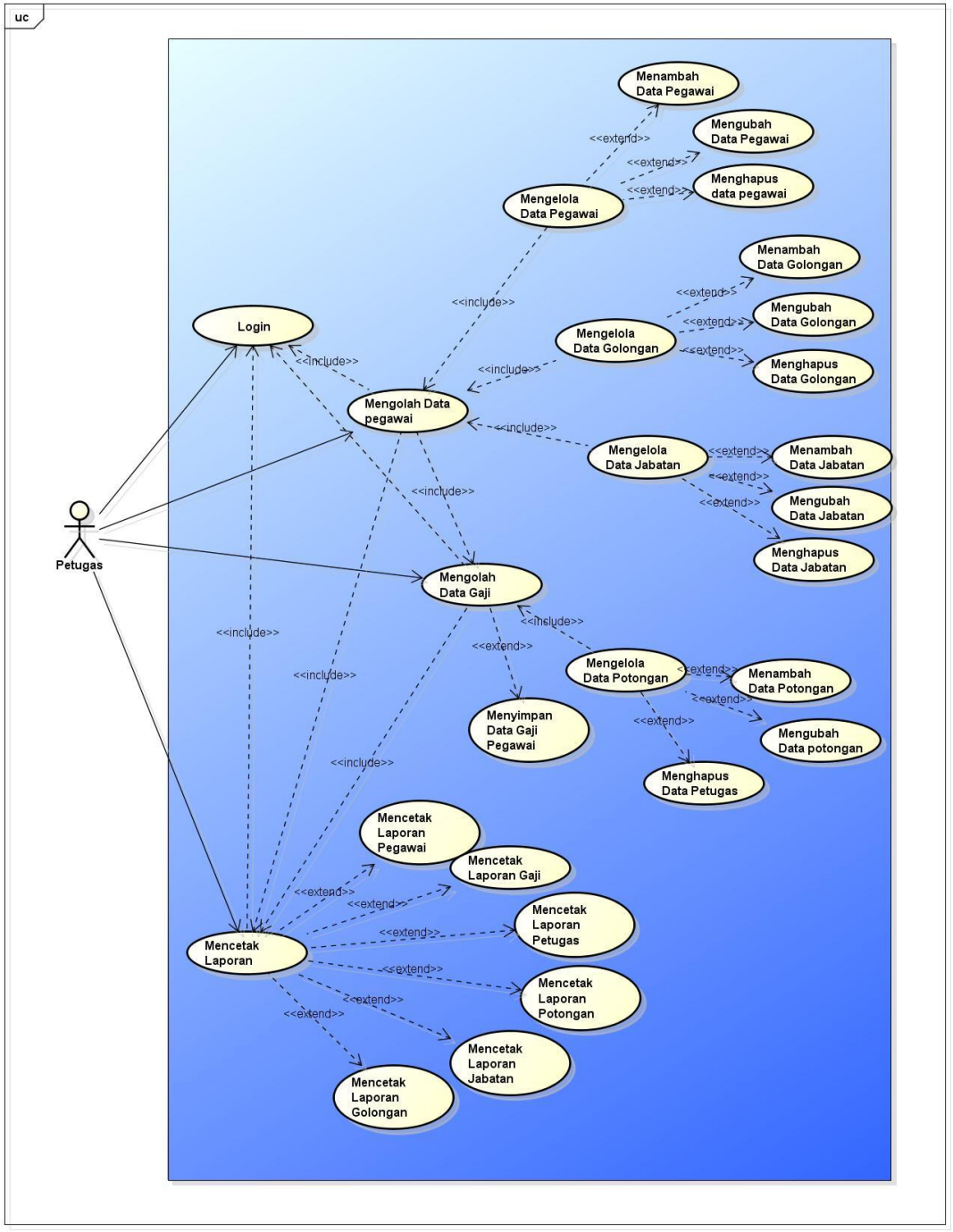
3.1.7 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sistem dan menjelaskan kebutuhan yang diperlukan agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan sistem kepegawaian.

3.1.7.1 Use Case Diagram

Use Case Diagram memperlihatkan hubungan diantara aktor dan *use case*. Aktor merepresentasikan seorang *user* atau subsistem lain yang akan berinteraksi dengan sistem. Sedangkan *use case* merupakan urutan kejadian yang menggambarkan interaksi antara *user* dengan

sistem. Fungsionalitas sistem didefinisikan ke dalam use case dari sudut eksternal sistem yang berguna untuk uji kelayakan sistem.



Gambar 3.3 *Use Case Diagram*

3.1.7.2 Definisi aktor

Tabel 3.7 Definisi aktor

No	Aktor	Deskripsi
1	Petugas	Orang yang bertugas dan memiliki hak akses untuk melakukan pengolahan data karyawan, data golongan, data jabatan, data potongan dan data gaji

3.1.7.3 Definisi use case

Tabel 3.8 Definisi *Use Case*

No	*Use Case*	Deskripsi
1	Login	Proses untuk melakukan login petugas.
2	Menambah data pegawai	Proses menambah data pegawai ke dalam *database*
3	Mengubah data pegawai	Proses mengubah data pegawai dari *database*
4	Menghapus data pegawai	Proses menghapus data pegawai dari *database*
5	Menambah data golongan	Proses menambah data golongan ke dalam *database*
6	Mengubah data golongan	Proses mengubah data golongan dari *database*
7	Menghapus data golongan	Proses menghapus data golongan dari *database*
8	Menambah data jabatan	Proses menambah data jabatan ke dalam *database*
9	Mengubah data jabatan	Proses mengubah data jabatan dari *database*
10	Menghapus data jabatan	Proses menghapus data jabatan dari

		database
11	Menambah data potongan	Proses menambah data potongan ke dalam *database*
12	Mengubah data potongan	Proses mengubah data potongan dari *database*
13	Menghapus data potongan	Proses menghapus data potongan dari *database*
14	Menyimpan data gaji	Proses menyimpan data gaji ke dalam *database*
15	Cetak Laporan pegawai	Proses mencetak laporan pegawai dari data pegawai yang ada pada aplikasi.
16	Cetak Laporan golongan	Proses mencetak laporan golongan dari data pegawai yang ada pada aplikasi.
17	Cetak Laporan jabatan	Proses mencetak laporan jabatan dari data pegawai yang ada pada aplikasi.
18	Cetak Laporan potongan	Proses mencetak laporan potongan dari data pegawai yang ada pada aplikasi.
19	Cetak Laporan gaji	Proses mencetak laporan gaji dari data pegawai yang ada pada aplikasi.

3.1.7.4 *Use Case Scenario*

3.1.7.4.1 Skenario Use Case Login

Tabel 3.9 Skenario *Use Case Login*

Nama Use Case	*Login*
Aktor	*User* (Petugas)
Tujuan	Masuk ke dalam sistem
Deskripsi	Memasukkan *username* dan *password* untuk masuk ke dalam system
Kondisi Awal	Petugas berada di halaman *login*.
Skenario Utama	
Aktor	**Sistem**
1. Mengisi *Username* dan *Password*.	
	2. Mengecek *valid* atau tidaknya data Memasukkan.
Skenario Alternative (jika Gagal)	
Aktor	**Sistem**
1. Mengisi *Username* dan *Password.*	
	2. Mengecek *valid* atau tidaknya data inputan
	3. Menampilkan pesan kesalahan bahwa *login* tidak *valid*.
4. Menerima pesan kesalahan	
5. Mengisi *Username* dan *Password*.	
	6. Mengecek validasi data
Kondisi Akhir	Login petugas berhasil

3.1.7.4.2 Skenario *Use Case* Menambah Data pegawai

Tabel 3. 10 Skenario *Use Case* Menambah Data pegawai

Nama Use Case	Menambah Data Pegawai
Aktor	*User* (Petugas)
Tujuan	Menambah data pegawai
Deskripsi	Memasukkan data pegawai baru ke dalam sistem yang akan disimpan dalam *database*
Kondisi Awal	Petugas berada di menu Data pegawai

Skenario Utama	
Aktor	**Sistem**
1. Mengisi Data Pegawai.	
	2. Mencatat Data pegawai.
	3. Validasi *database*
	4. Cek kesamaan data.
	5. Menyimpan Data Pegawai ke *database*
	6. Menampilkan pesan bahwa data berhasil disimpan.

Skenario Alternative (jika Gagal)	
Aktor	**Sistem**
1. Mengisi Data Pegawai.	
	2. Mencatat Data pegawai.
	3. Validasi *database*
	4. Cek kesamaan data.
	5. Menampilkan pesan bahwa data sudah ada.
6. Mengisi data pegawai.	
	7. Mencatat Data jabatan.
	8. Validasi *database*
	9. Cek kesamaan Data
	10. Menyimpan Data pegawai ke *database*
	11. Menampilkan pesan bahwa data berhasil disimpan.

Kondisi Akhir	Data pegawai baru berhasil disimpan

3.1.7.4.3 Skenario *Use Case edit* Data pegawai

Tabel 3.11 *Skenario Use Case Edit Data pegawai*

Nama Use Case	*Edit* Data Pegawai
Aktor	*User* (Petugas)
Tujuan	Mengubah data pegawai yang ada
Deskripsi	Mengubah data pegawai lama yang ada di dalam sistem yang telah disimpan dalam *database*
Kondisi Awal	Petugas berada di halaman data pegawai
Skenario Utama	
Aktor	**Sistem**
1. Memilih data yang akan diubah yang tersedia di menu data pegawai.	
2. Mengisi data pegawai yang baru.	
	3. Mencatat Data Pegawai yang baru.
	4. Validasi *database*
	5. Cek kesamaan data
	6. Menyimpan data pegawai yang baru ke dalam *database*
	7. Menampilkan pesan bahwa data berhasil disimpan.
Skenario Alternative (Jika Gagal)	
Aktor	**Sistem**
1. Memilih data yang akan diubah yang tersedia di menu pegawai.	
2. Mengisi data pegawai yang baru.	
	3. Mencatat Data Pegawai yang baru.
	4. Validasi *database*
	5. Cek kesamaan data pegawai
	6. Menampilkan pesan bahwa data sudah ada.
7. Mengisi Data Pegawai yang baru..	
	8. Mencatat Data pegawai yang baru.

	9. Validasi *database*
	10. Cek kesamaan data.
	11. Menyimpan Data Pegawai yang baru ke dalam *database*
	12. Menampilkan pesan bahwa data berhasil disimpan.
Kondisi Akhir	Data pegawai berhasil diubah

3.1.7.4.4 Skenario *Use Case* hapus Data pegawai

Tabel 3. 12 Skenario *Use Case* Hapus Data Pegawai

Nama Use Case	Hapus Data Pegawai
Aktor	*User* (Petugas)
Tujuan	Menghapus data pegawai yang tersedia
Deskripsi	Menghapus data pegawai yang ada di dalam sistem yang telah disimpan dalam *database*
Kondisi Awal	Petugas berada di halaman data pegawai
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih data yang akan dihapus yang tersedia di menu pegawai.	
2. Menghapus data pegawai	
	3. Menampilkan pesan konfirmasi hapus
	4. Validasi ke *Database*
	5. Menghapus data pegawai yang dipilih
	6. Menampilkan pesan bahwa data sudah dihapus
Kondisi Akhir	Data pegawai Berhasil Dihapus dari *database*

3.1.7.4.5 Skenario *Use Case* Menambah Data golongan

Tabel 3. 13 Skenario *Use Case* Menambah Data golongan

Nama Use Case	Menambah Data golongan
Aktor	*User* (Petugas)
Tujuan	Menambah data golongan
Deskripsi	Memasukkan data golongan baru ke dalam sistem yang akan

	disimpan dalam *database*
Kondisi Awal	Petugas berada di menu Data golongan

Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Mengisi Data Golongan.	
	2. Mencatat Data golongan.
	3. Validasi *database*
	4. Cek kesamaan data.
	5. Menyimpan Data Golongan ke *database*
	6. Menampilkan pesan bahwa data berhasil disimpan.
Skenario Alternative (jika gagal)	
Aktor	**Sistem**
1. Mengisi Data Golongan.	
	2. Mencatat Data golongan.
	3. Validasi *database*
	4. Cek kesamaan data.
	5. Menampilkan pesan bahwa data sudah ada.
6. Mengisi data masukkan.	
	7. Mencatat Data golongan.
	8. Validasi *database*
	9. Cek kesamaan data.
	10. Menyimpan Data golongan ke *database*
	11. Menampilkan pesan bahwa data berhasil disimpan.

Kondisi Akhir	Data golongan baru berhasil disimpan

3.1.7.4.6 Skenario *Use Case edit* Data golongan

Tabel 3.14 *Skenario Use Case Edit Data golongan*

Nama Use Case	*Edit* Data golongan
Aktor	*User* (Petugas)
Tujuan	Mengubah data golongan yang ada
Deskripsi	Mengubah data golongan lama yang ada di dalam sistem yang telah disimpan dalam *database*

Kondisi Awal	Petugas berada di halaman data golongan
Skenario Utama	
Aktor	**Sistem**
1. Memilih data yang akan diubah yang tersedia di menu golongan.	
2. Mengisi data golongan yang baru.	
	3. Mencatat Data Golongan yang baru
	4. Validasi *database*
	5. Cek kesamaan data.
	6. Menyimpan Data golongan yang baru ke *database*
	7. Menampilkan pesan bahwa data berhasil disimpan.
Skenario Alternative (jika Gagal)	
Aktor	**Sistem**
1. Memilih data yang akan diubah yang tersedia di menu golongan.	
2. Mengisi data golongan yang baru.	
	3. Mencatat Data Golongan yang baru
	4. Validasi *database*
	5. Cek kesamaan data.
	6. Menampilkan pesan bahwa data sudah ada.
7. Mengisi data golongan yang baru.	
	8. Mencatat Data Golongan yang baru
	9. Validasi *database*
	10. Cek kesamaan data
	11. Menyimpan Data Golongan yang baru ke *database*
	12. Menampilkan pesan bahwa data berhasil disimpan.
Kondisi akhir	Data golongan berhasil diubah.

3.1.7.4.7 Skenario *Use Case* hapus Data golongan

Tabel 3. 15 *Skenario Use Case Hapus Data Golongan*

Nama Use Case	Hapus Data Golongan
Aktor	*User* (Petugas)
Tujuan	Menghapus data golongan yang tersedia

Deskripsi	Menghapus data golongan yang ada di dalam sistem yang telah disimpan dalam *database*
Kondisi Awal	Petugas berada di halaman data golongan
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih data yang akan dihapus yang tersedia di menu golongan.	
2. Menghapus data golongan	
	3. Menampilkan pesan konfirmasi hapus
	4. Validasi ke *Database*
	5. Menghapus data golongan yang dipilih
	6. Menampilkan pesan bahwa data sudah dihapus
Kondisi Akhir	Data golongan Berhasil Dihapus dari *database*

3.1.7.4.8 Skenario *Use Case* menambah Data jabatan

Tabel 3.16 Skenario *Use Case* Menambah Data jabatan

Nama Use Case	Menambah Data jabatan
Aktor	*User* (Petugas)
Tujuan	Menambah data jabatan
Deskripsi	Memasukkan data jabatan baru ke dalam sistem yang akan disimpan dalam *database*
Kondisi Awal	Petugas berada di menu Data jabatan
Skenario Utama	
Aktor	**Sistem**
1. Mengisi Data Jabatan.	
	2. Mencatat Data jabatan.
	3. Validasi *database*
	4. Cek kesamaan data.
	5. Menyimpan Data Jabatan ke *database*
	6. Menampilkan pesan bahwa data berhasil disimpan.
Skenario Alternative (jika gagal)	
Aktor	**Sistem**
1. Mengisi Data Jabatan.	

	2. Mencatat Data jabatan.
	3. Validasi *database*
	4. Cek kesamaan data.
	5. Menampilkan pesan bahwa data sudah ada.
6. Mengisi data jabatan.	
	7. Mencatat Data jabatan.
	8. Validasi *database*
	9. Cek kesamaan data.
	10. Menyimpan Data jabatan yanmg baru ke *database*
	11. Menampilkan pesan bahwa data berhasil disimpan.
Kondisi Akhir	Data jabatan baru berhasil disimpan.

3.1.7.4.9 Skenario *Use Case edit* Data jabatan

Tabel 3.17 Skenario *Use Case Edit* Data jabatan

Nama Use Case	*Edit* Data jabatan
Aktor	*User* (Petugas)
Tujuan	Mengubah data jabatan yang ada
Deskripsi	Mengubah data jabatan lama yang ada di dalam sistem yang telah disimpan dalam *database*
Kondisi Awal	Petugas berada di halaman data jabatan
Skenario Utama	
Aktor	**Sistem**
1. Memilih data yang akan diubah yang tersedia di menu jabatan.	
2. Mengisi data jabatan yang baru.	
	3. Mencatat Data Jabatan yang baru.
	4. Validasi *database*
	5. Cek kesamaan data.
	6. Menyimpan Data jabatan yang baru ke dalam *database*
	7. Menampilkan pesan bahwa data berhasil disimpan.
Skenario Alternative (jika gagal)	

Aktor	Sistem
1. Memilih data yang akan diubah yang tersedia di menu jabatan.	
2. Mengisi data jabatan yang baru.	
	3. Mencatat Data Jabatan yang baru.
	4. Validasi *database*
	5. Cek kesamaan data.
	6. Menampilkan pesan bahwa data sudah ada.
7. Mengisi data jabatan yang baru	
	8. Mencatat Data jabatan yang baru
	9. Validasi *database*
	10. Cek kesamaan data.
	11. Menyimpan Data Jabatan yang baru ke *database*
	12. Menampilkan pesan bahwa data berhasil disimpan.
Kondisi akhir	Data jabatan berhasil diubah

3.1.7.4.10 Skenario *Use Case* hapus Data jabatan

Tabel 3. 18 Skenario *Use Case* Hapus Data Jabatan

Nama Use Case	Hapus Data jabatan
Aktor	*User* (Petugas)
Tujuan	Menghapus data jabatan yang tersedia
Deskripsi	Menghapus data jabatan yang ada di dalam sistem yang telah disimpan dalam *database*
Kondisi Awal	Petugas berada di halaman data jabatan
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih data yang akan dihapus yang tersedia di menu jabatan.	
2. Menghapus data jabatan	
	3. Menampilkan pesan konfirmasi hapus
	4. Validasi ke *Database*
	5. Menghapus data jabatan yang dipilih
	6. Menampilkan pesan bahwa data sudah

	dihapus
Kondisi Akhir	Data jabatan Berhasil Dihapus dari *database*

3.1.7.4.11 Skenario *Use Case* Menambah Data potongan

Tabel 3. 19 Skenario *Use Case* Menambah Data potongan

Nama Use Case	Menambah Data potongan
Aktor	*User* (Petugas)
Tujuan	Menambah data potongan
Deskripsi	Memasukkan data potongan baru ke dalam sistem yang akan disimpan dalam *database*
Kondisi Awal	Petugas berada di menu Data potongan

Skenario Utama	
Aktor	**Sistem**
1. Mengisi Data Potongan.	
	2. Mencatat Data potongan.
	3. Validasi *database*
	4. Cek kesamaan data
	5. Menyimpan Data Potongan ke *database*
	6. Menampilkan pesan bahwa data berhasil disimpan.

Skenario Alternative (jika gagal)	
Aktor	**Sistem**
1. Mengisi Data Potongan yang baru.	
	2. Mencatat Data potongan.
	3. Validasi *database*
	4. Cek kesamaan data
	5. Menampilkan pesan bahwa data sudah ada.
6. Mengisi Data potongan yang baru.	
	7. Mencatat Data potongan yang baru.
	8. Validasi *database*
	9. Cek kesamaan data

	10. Menyimpan Data potongan ke *Database*
	11. Menampilkan pesan bahwa data berhasil disimpan.
Kondisi Akhir	Data potongan baru berhasil disimpan

3.1.7.4.12 Skenario *Use Case edit* Data potongan

Tabel 3. 20 *Skenario Use Case Edit Data potongan*

Nama Use Case	*Edit* Data potongan
Aktor	*User* (Petugas)
Tujuan	Mengubah data potongan yang ada
Deskripsi	Mengubah data potongan lama yang ada di dalam sistem yang telah disimpan dalam *database*
Kondisi Awal	Petugas berada di halaman data potongan

Skenario Utama	
Aktor	**Sistem**
1. Memilih data yang akan diubah yang tersedia di menu potongan.	
2. Mengisi data potongan yang baru.	
	3. Mencatat Data Potongan yang baru.
	4. Validasi *database*
	5. Cek kesamaan data.
	6. Menyimpan Data Potongan yang baru ke *database*
	7. Menampilkan pesan bahwa data berhasil disimpan.

Skenario Alternative (jika gagal)	
Aktor	**Sistem**
1. Memilih data yang akan diubah yang tersedia di menu potongan.	
2. Mengisi data potongan yang baru.	
	3. Mencatat Data potongan yang baru
	4. Validasi *database*
	5. Cek kesamaan data
	6. Menampilkan pesan bahwa data sudah ada.
7. Mengisi data potongan yang baru	

	8. Mencatat Data potongan yang baru.
	9. Validasi *database*
	10. Cek kesamaan data.
	11. Menyimpan Data Potongan yang baru ke *database*
	12. Menampilkan pesan bahwa data berhasil disimpan.
Kondisi Akhir	Data potongan berhasil diubah

3.1.7.4.13 Skenario *Use Case* hapus Data potongan

Tabel 3. 21 *Skenario Use Case Hapus Data Potongan*

Nama Use Case	Hapus Data potongan
Aktor	*User* (Petugas)
Tujuan	Menghapus data potongan yang tersedia
Deskripsi	Menghapus data potongan yang ada di dalam sistem yang telah disimpan dalam *database*
Kondisi Awal	Petugas berada di halaman data potongan
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih data yang akan dihapus yang tersedia di menu potongan.	
2. Menghapus data potongan	
	3. Menampilkan pesan konfirmasi hapus
	4. Validasi ke *Database*
	5. Menghapus data potongan yang dipilih
	6. Menampilkan pesan bahwa data sudah dihapus
Kondisi Akhir	Data potongan Berhasil Dihapus dari *database*

3.1.7.4.14 Skenario Use Case menyimpan data gaji

Tabel 3. 22 *Skenario Use Case menyimpan* Data gaji

Nama Use Case	Menyimpan Data Gaji
Aktor	*User* (Petugas)
Tujuan	Menyimpan dat gaji pegawai
Deskripsi	Menyimpan data gaji yang di masukkan ke dalam sistem dan

	disimpan ke dalam *database*
Kondisi Awal	Petugas berada di halaman penggajian

Skenario Utama	
Aktor	**Sistem**
1. Mengisi NIP di data gaji.	
	2. Mencatat NIP pada data gaji
	3. Menyesuaikan dengan data NIP dari data pegawai yang ada di *database*
	4. Mengisi secara otomatis inputan lainnya.
5. Mengisi besaran potongan	
	6. Cek kelengkapan data
	7. Mengkalkulasi gaji dan besaran potongan.
8. Memilih Simpan.	
	9. Mengecek apakah pegawai dengan NIP yang diisi sudah mendapat gaji bulan ini.
	10. Menampilkan pesan bahwa data berhasil disimpan.
Skenario Alternative (jika gagal)	
Aktor	**Sistem**
1. Mengisi isian NIP.	
	2. Mencatat NIP pada data gaji
	3. Menyesuaikan dengan data NIP dari data pegawai yang ada di *database*
	4. Menampilkan pesan kesalahan Bahwa NIP tidak sesuai.
5. Mengisi form isian NIP.	
	6. Menyesuaikan dengan data NIP yang ada di *database*
	7. Mengisi secara otomatis inputan lainnya
8. Memasukkan besaran potongan	
	9. Cek kelengkapan data
	10. Menampilkan pesan kesalahan bahwa data tidak lengkap
11. Memasukkan besaran potongan	
	12. Cek kelengkapan data

	13. Mengkalkulasi gaji dan besaran potongan.
14. Memilih Simpan.	
	15. Mengecek apakah pegawai dengan NIP yang diisi sudah mendapat gaji bulan ini.
	16. Menampilkan pesan bahwa pegawai dengan NIP yang diisi sudah mendapat gaji bulan ini.
17. Menampilkan pesan bahwa pegawai dengan NIP yang diisi sudah mendapat gaji bulan ini	
	18. Mengecek apakah pegawai dengan NIP yang diisi sudah mendapat gaji bulan ini.
	19. Menampilkan pesan bahwa data berhasil disimpan.
Kondisi Akhir	Data gaji berhasil disimpan

3.1.7.4.15 Skenario *Use Case* Cetak Laporan Data pegawai

Tabel 3. 23 Skenario *Use Case* Cetak Laporan data pegawai

Nama Use Case	Cetak Laporan data pegawai
Aktor	*User* (Petugas)
Tujuan	mencetak laporan dari data pegawai
Deskripsi	Mencetak data pegawai yang ada di dalam sistem yang sudah disimpan dalam *database*
Kondisi Awal	Petugas berada di halaman cetak laporan
Skenario Utama	
Aktor	**Sistem**
1. Memilih data pegawai yang akan dicetak.	
	2. Menampilkan menu *print preview* dari data pegawai.
3. Mengatur hasil yang akan dicetak	
	4. Konfirmasi cetak Laporan pegawai
	5. Mencetak Laporan data pegawai.
Kondisi Akhir	Laporan data pegawai berhasil dicetak

3.1.7.4.16 Skenario *Use Case* Cetak Laporan Data golongan

Tabel 3. 24 Skenario *Use Case* Cetak Laporan data golongan

Nama Use Case	Cetak Laporan data golongan
Aktor	*User* (Petugas)
Tujuan	mencetak laporan dari data golongan
Deskripsi	Mencetak data golongan yang ada di dalam sistem yang sudah disimpan dalam *database*
Kondisi Awal	Petugas berada di halaman cetak laporan
Skenario Utama	
Aktor	**Sistem**
1. Pilih Menu Cetak Laporan Golongan.	
	2. Menampilkan menu *print preview* dari data golongan.
3. Mengatur hasil yang akan dicetak	
	4. Konfirmasi cetak Laporan golongan.
	5. Mencetak Laporan data golongan
Kondisi Akhir	Laporan data golongan berhasil dicetak

3.1.7.4.17 Skenario *Use Case* Cetak Laporan Data jabatan

Tabel 3. 25 Skenario *Use Case* Cetak Laporan data jabatan

Nama Use Case	Cetak Laporan data jabatan
Aktor	*User* (Petugas)
Tujuan	mencetak laporan dari data jabatan
Deskripsi	Mencetak data jabatan yang ada di dalam sistem yang sudah disimpan dalam *database*
Kondisi Awal	Petugas berada di halaman cetak laporan
Skenario Utama	
Aktor	**Sistem**
1. Pilih Menu Cetak Laporan Jabatan.	
	2. Menampilkan menu *print preview* dari data jabatan.
3. Mengatur hasil yang akan dicetak	

	4. Konfirmasi cetak Laporan jabatan
	5. Mencetak Laporan data jabatan
Kondisi Akhir	Laporan data jabatan berhasil dicetak

3.1.7.4.18 Skenario *Use Case* Cetak Laporan Data potongan

Tabel 3. 26 Skenario *Use Case* Cetak Laporan data potongan

Nama Use Case	Cetak Laporan data potongan
Aktor	*User* (Petugas)
Tujuan	mencetak laporan dari data potongan
Deskripsi	Mencetak data potongan yang ada di dalam sistem yang sudah disimpan dalam *database*
Kondisi Awal	Petugas berada di halaman cetak laporan
Skenario Utama	
Aktor	**Sistem**
1. Pilih Menu Cetak Laporan Potongan.	
	2. Menampilkan menu *print preview* dari data potongan.
3. Mengatur hasil yang akan dicetak	
	4. Konfirmasi cetak Laporan potongan.
	5. Mencetak Laporan data potongan
Kondisi Akhir	Laporan data potongan berhasil dicetak

3.1.7.4.19 Skenario *Use Case* Cetak Laporan Data gaji

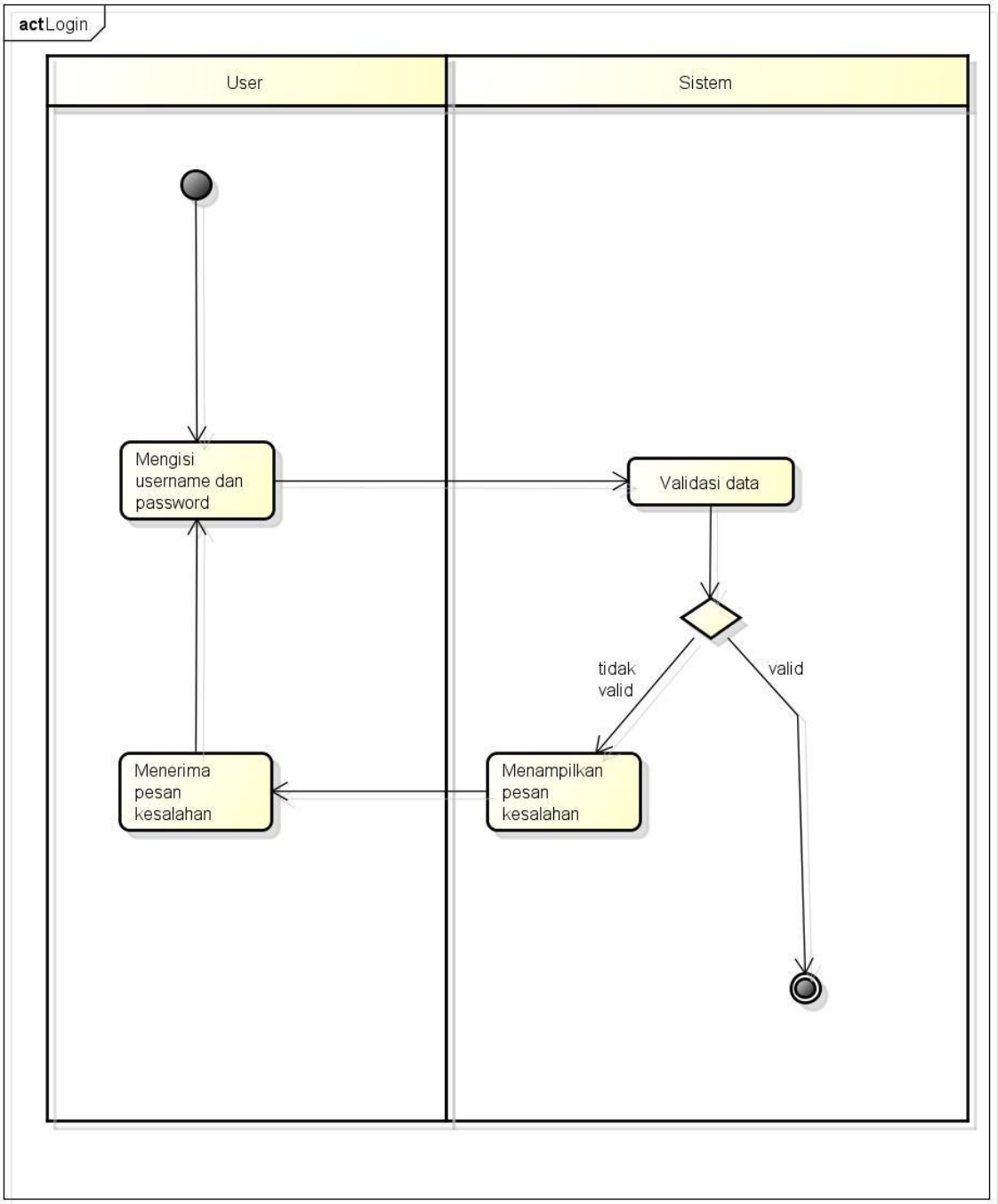
Tabel 3. 27 Skenario *Use Case* Cetak Laporan data gaji

Nama Use Case	Cetak Laporan data gaji
Aktor	*User* (Petugas)
Tujuan	mencetak laporan dari data gaji
Deskripsi	Mencetak data gaji yang ada di dalam sistem yang sudah disimpan dalam *database*
Kondisi Awal	Petugas berada di halaman cetak laporan
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pilih Menu Cetak Laporan Gaji.	
	2. Menampilkan menu *print preview*

	dari data potongan.
3. Mengatur hasil yang akan dicetak	
	4. Konfirmasi cetak Laporan gaji
	5. Mencetak Laporan gaji
Kondisi Akhir	Lapotan data gaji berhasil dicetak

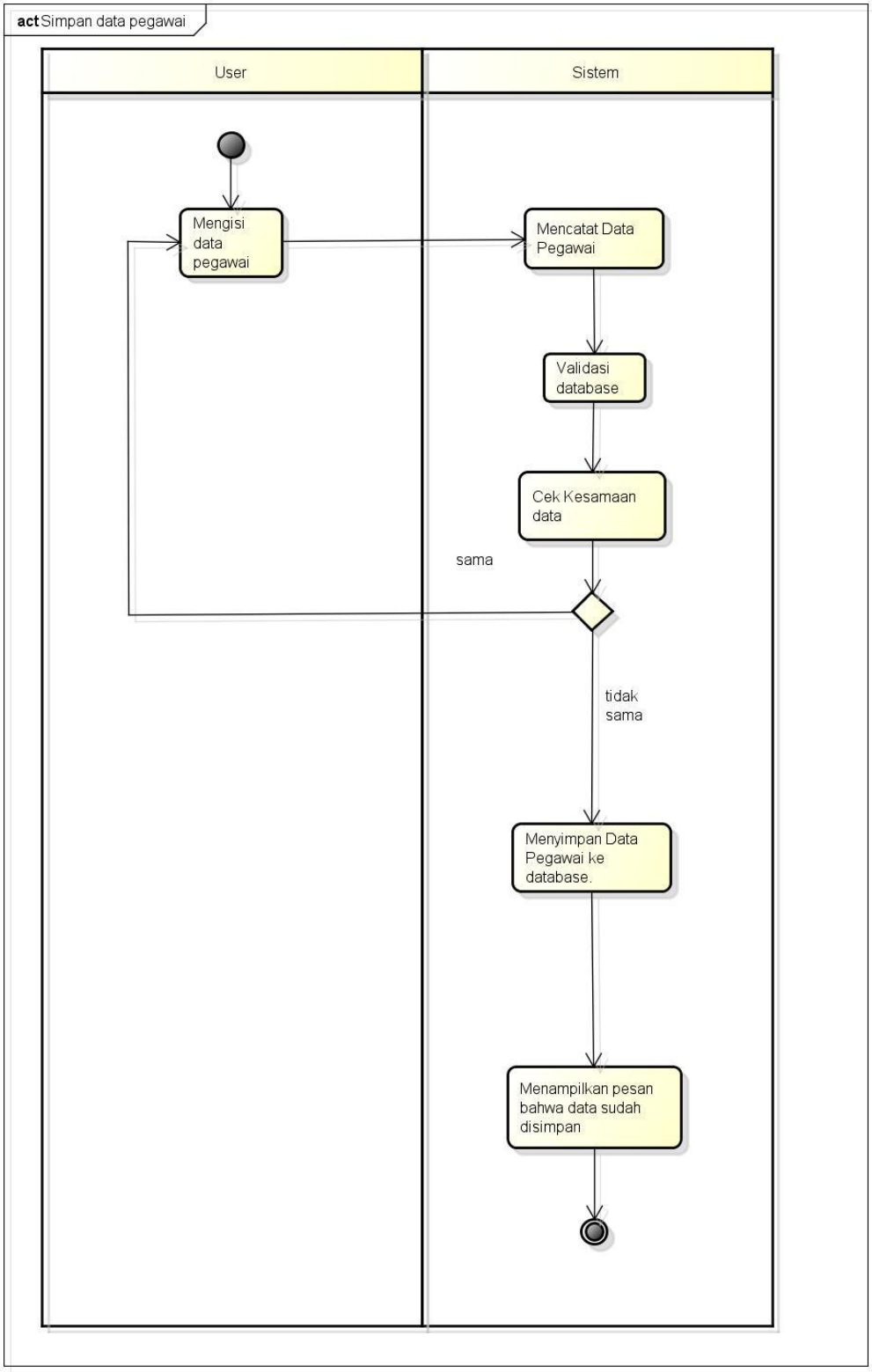
3.1.7.5 Activity Diagram

3.1.7.5.1 *Login*



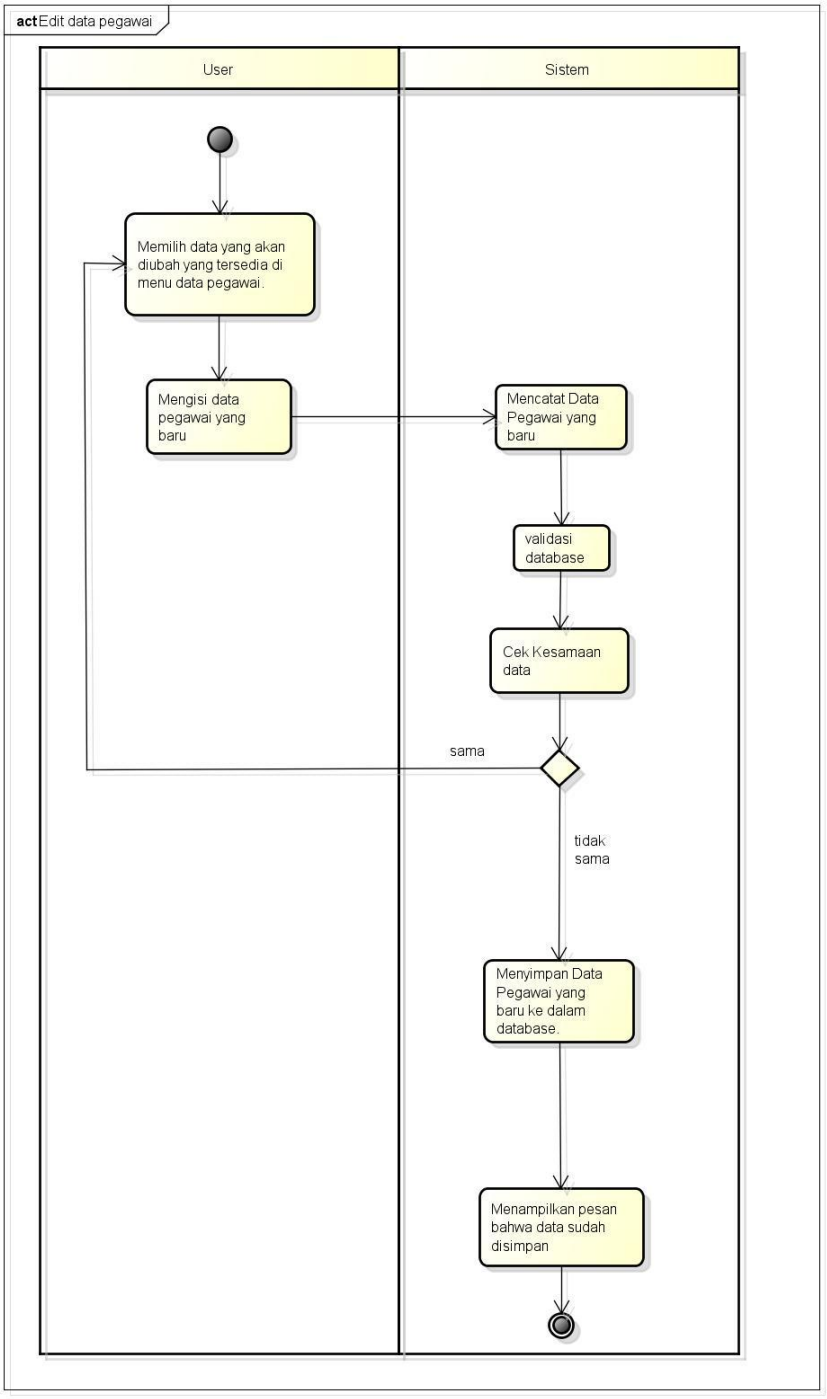
Gambar 3.5 *Activity Diagram login*

3.1.7.5.2 Menambah data pegawai



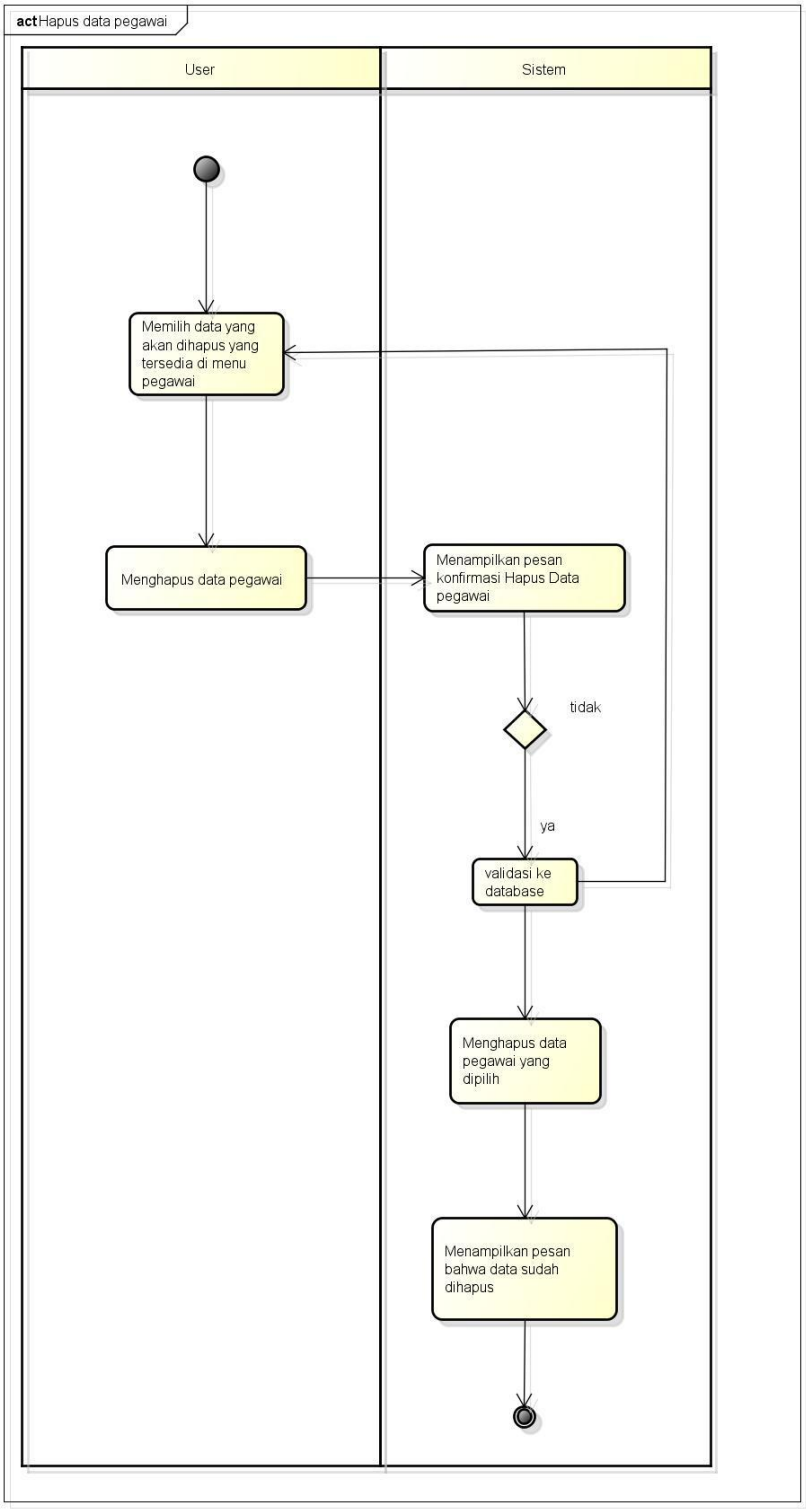
Gambar 3.6 *Activity Diagram* menambah data pegawai

3.1.7.5.3 *Edit* Data Pegawai



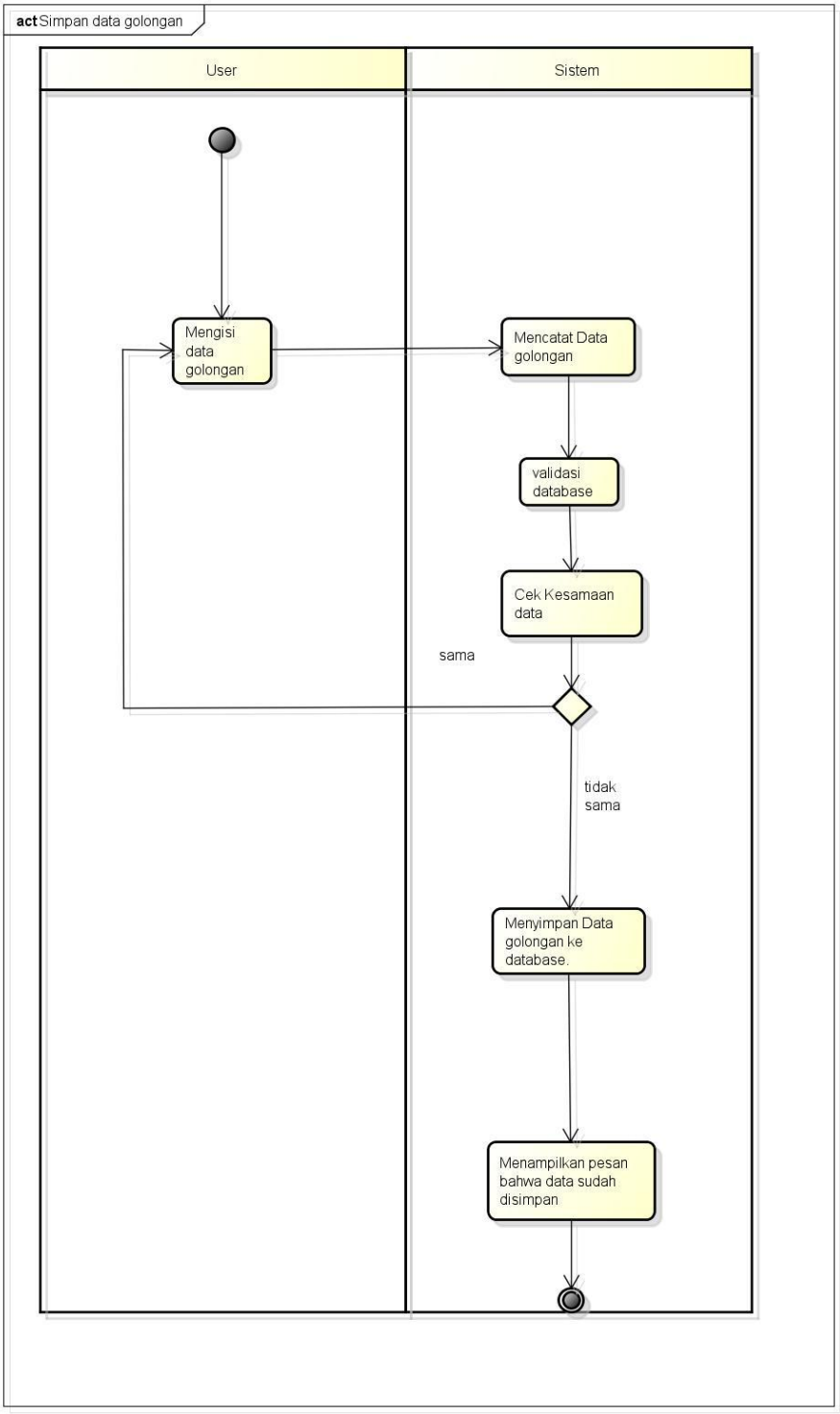
Gambar 3.7 *Activity Diagram Edit* data pegawai

3.1.7.5.4 Hapus data pegawai



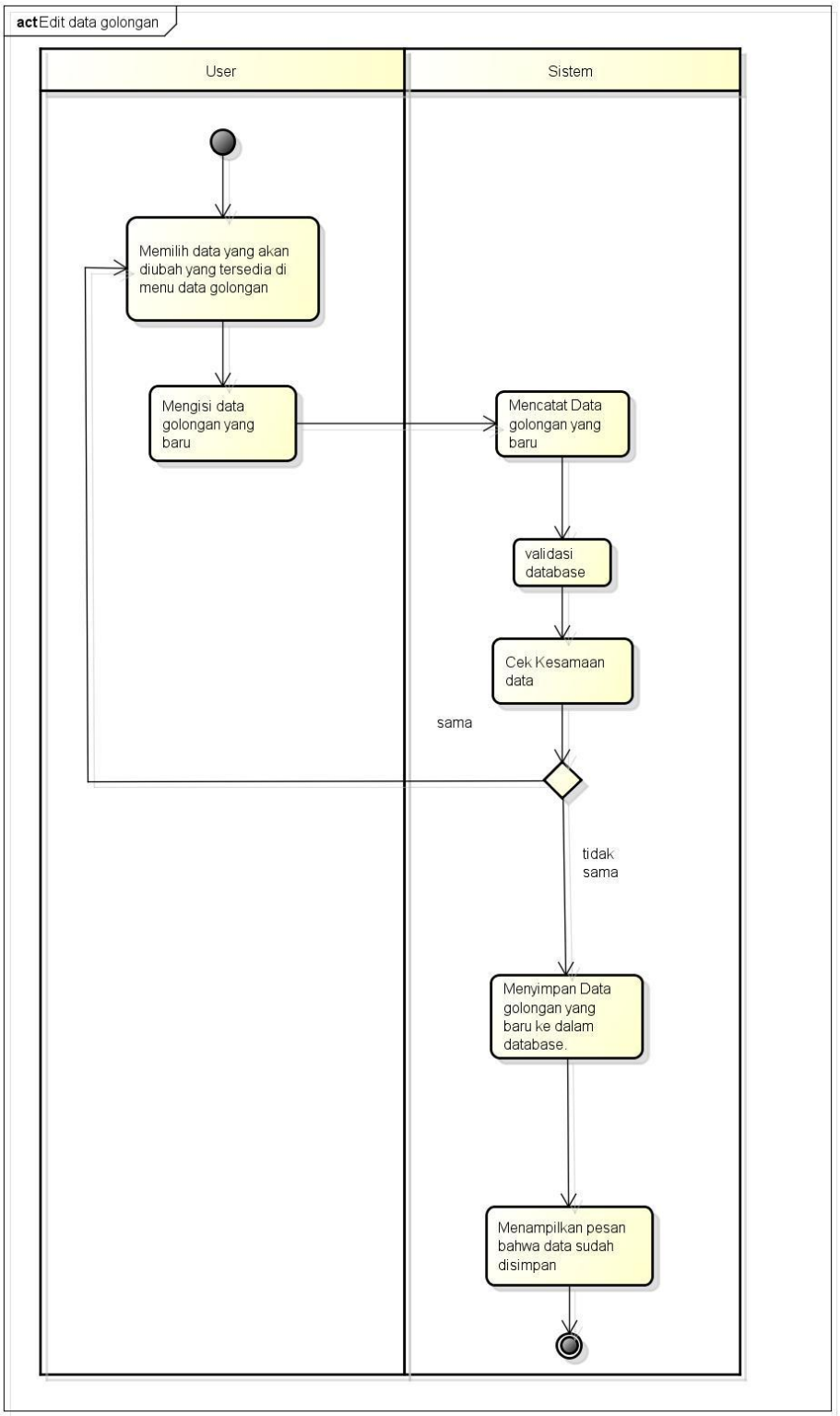
Gambar 3.8 *Activity Diagram* Hapus data pegawai

3.1.7.5.5 Menambah data golongan



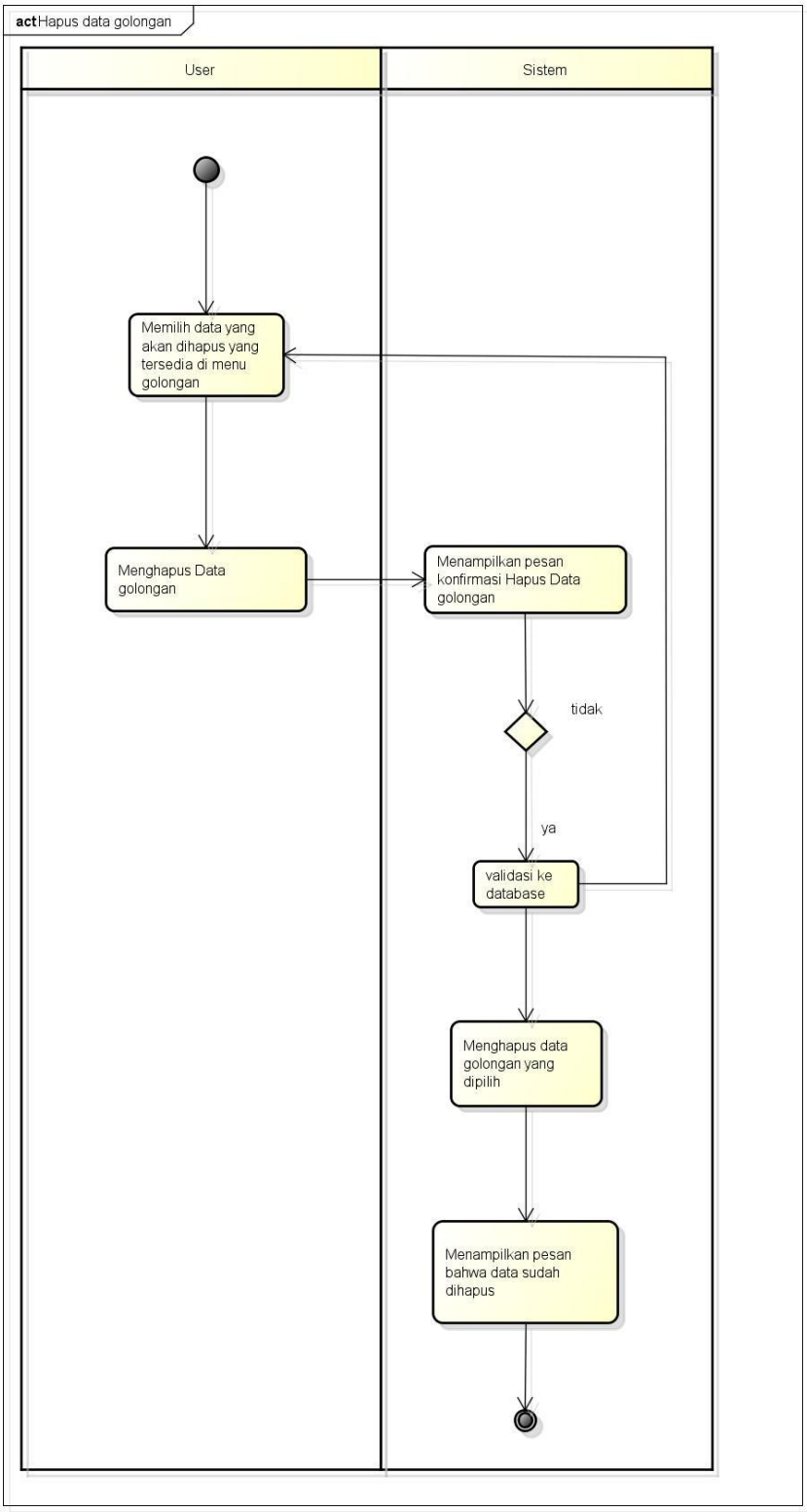
Gambar 3.9 *Activity Diagram* Menambah data golongan

3.1.7.5.6 *Edit* data Golongan



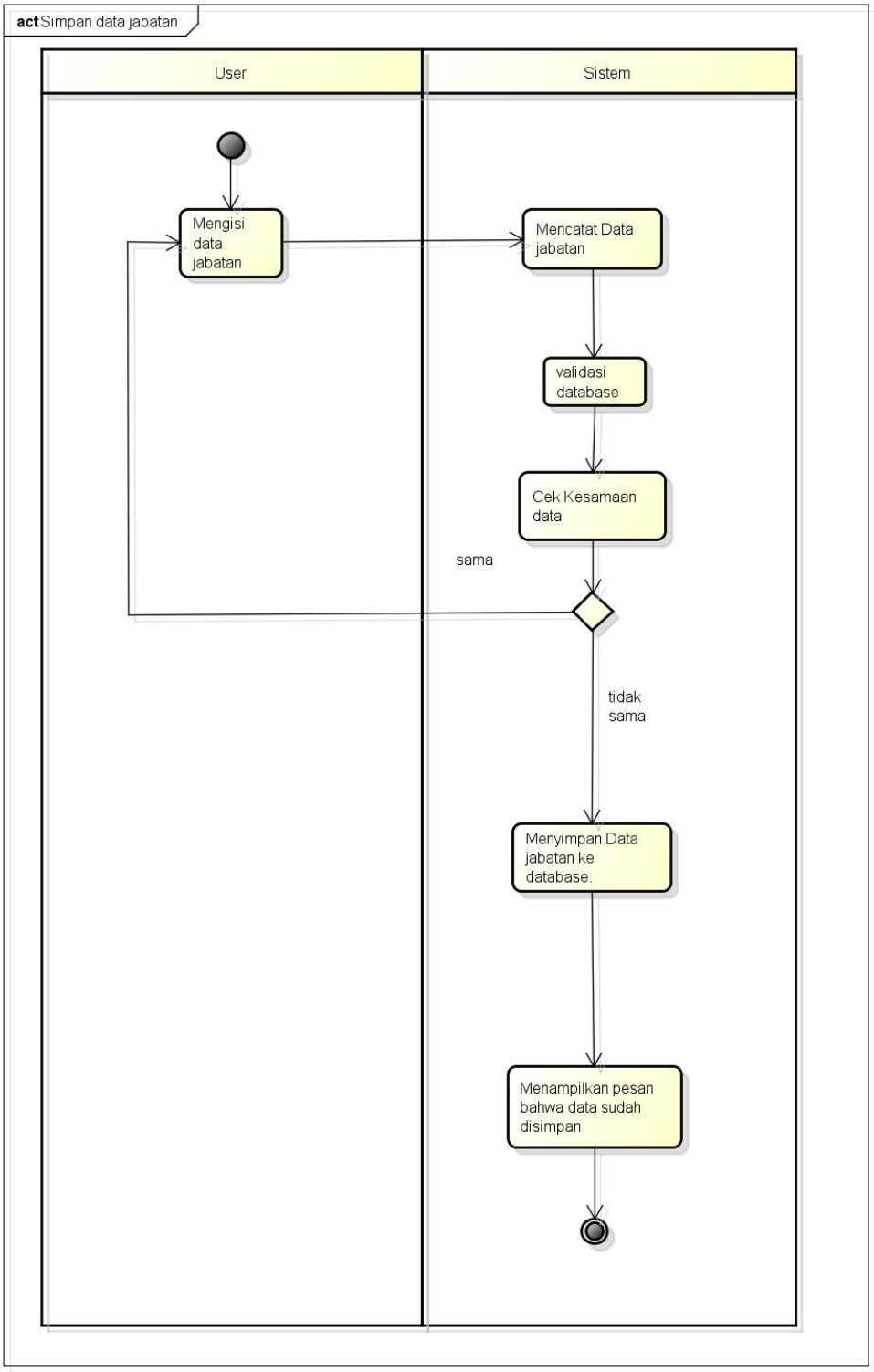
Gambar 3.10 *Activity Diagram Edit* data golongan

3.1.7.5.7 Hapus data golongan



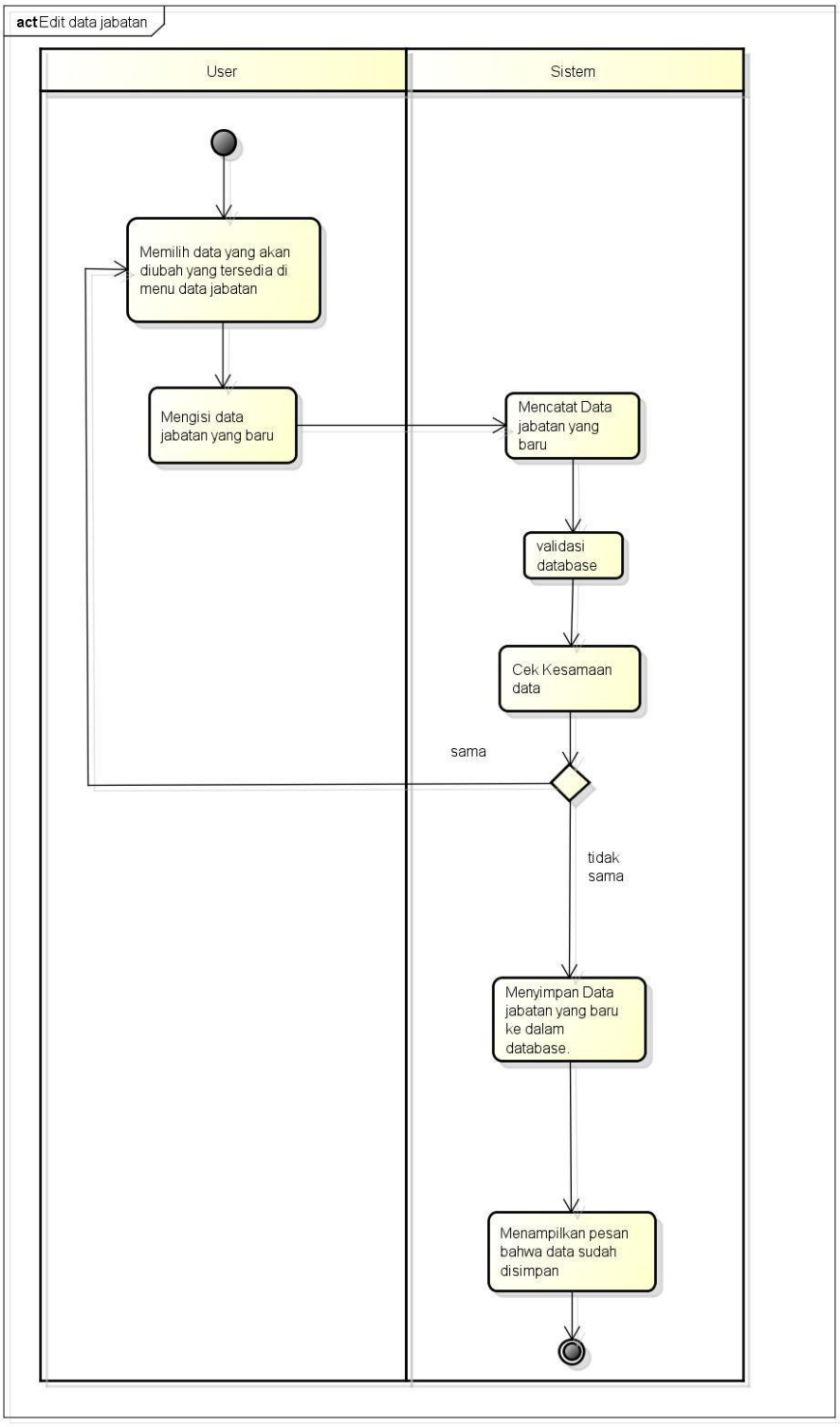
Gambar 3.11 *Activity Diagram* Hapus data golongan

3.1.7.5.8 Menambah data jabatan



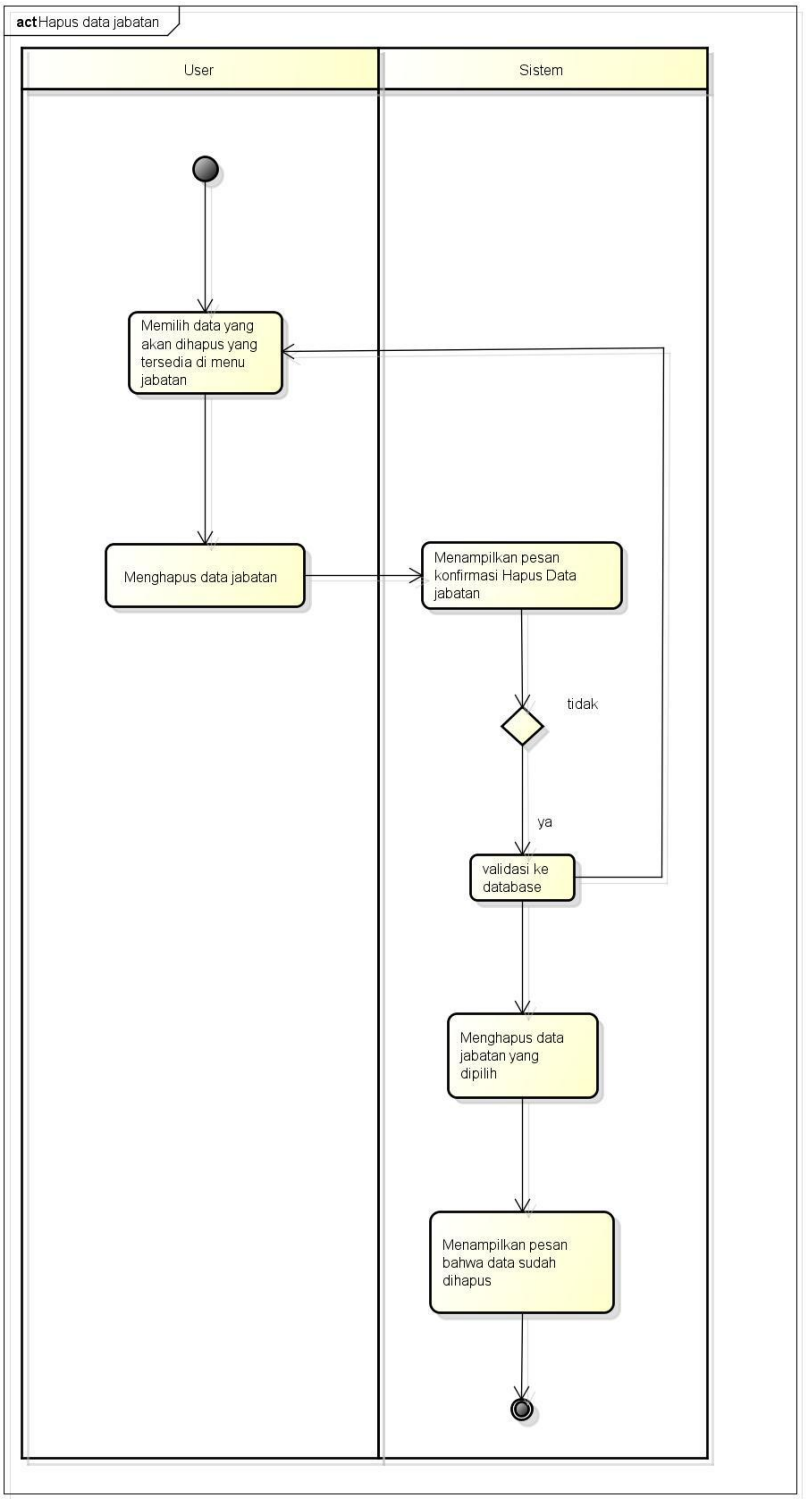
Gambar 3.12 *Activity Diagram* Menambah data jabatan

3.1.7.5.9 *Edit* data jabatan



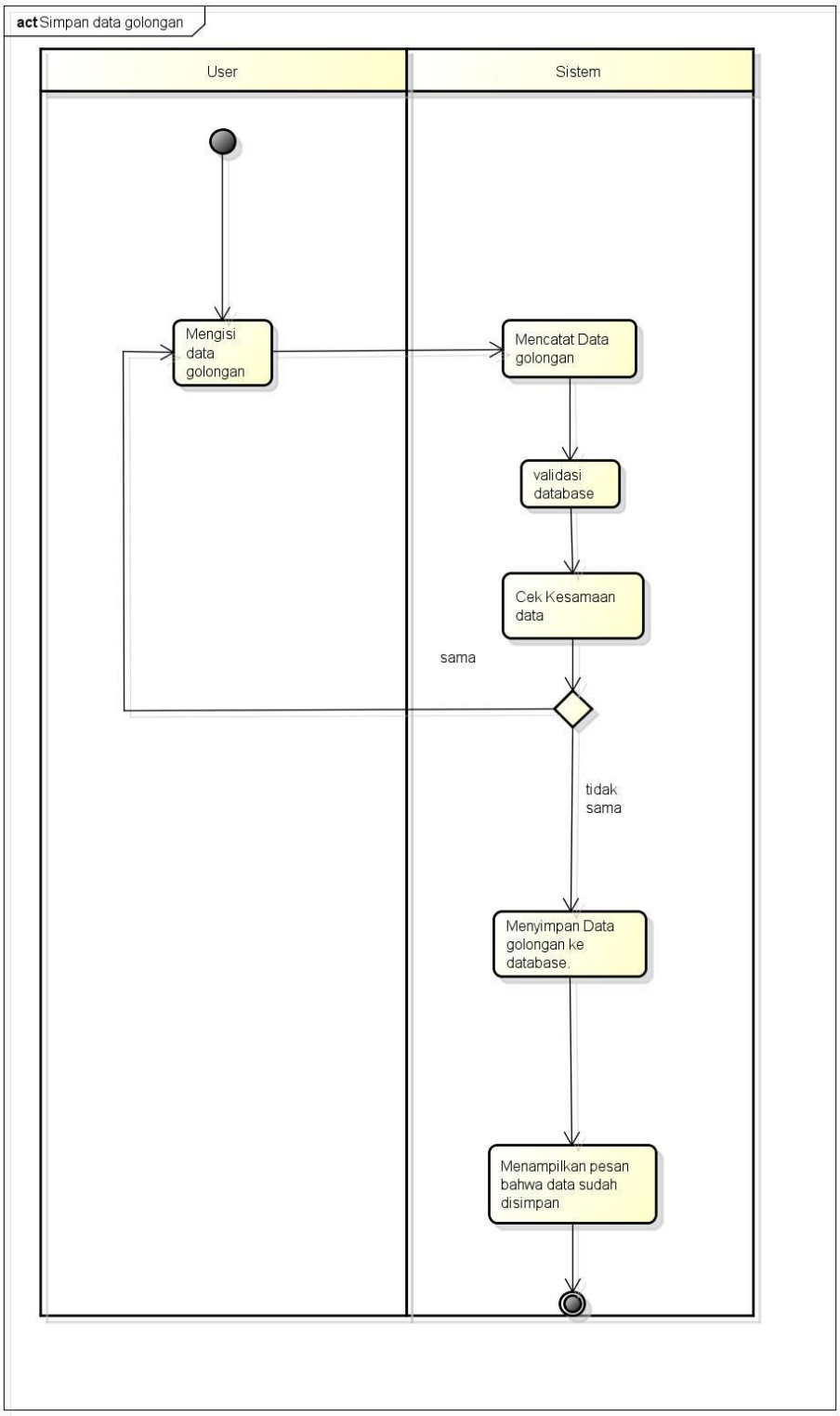
Gambar 3.13 *Activity Diagram Edit* data jabatan

3.1.7.5.10 Hapus Data jabatan



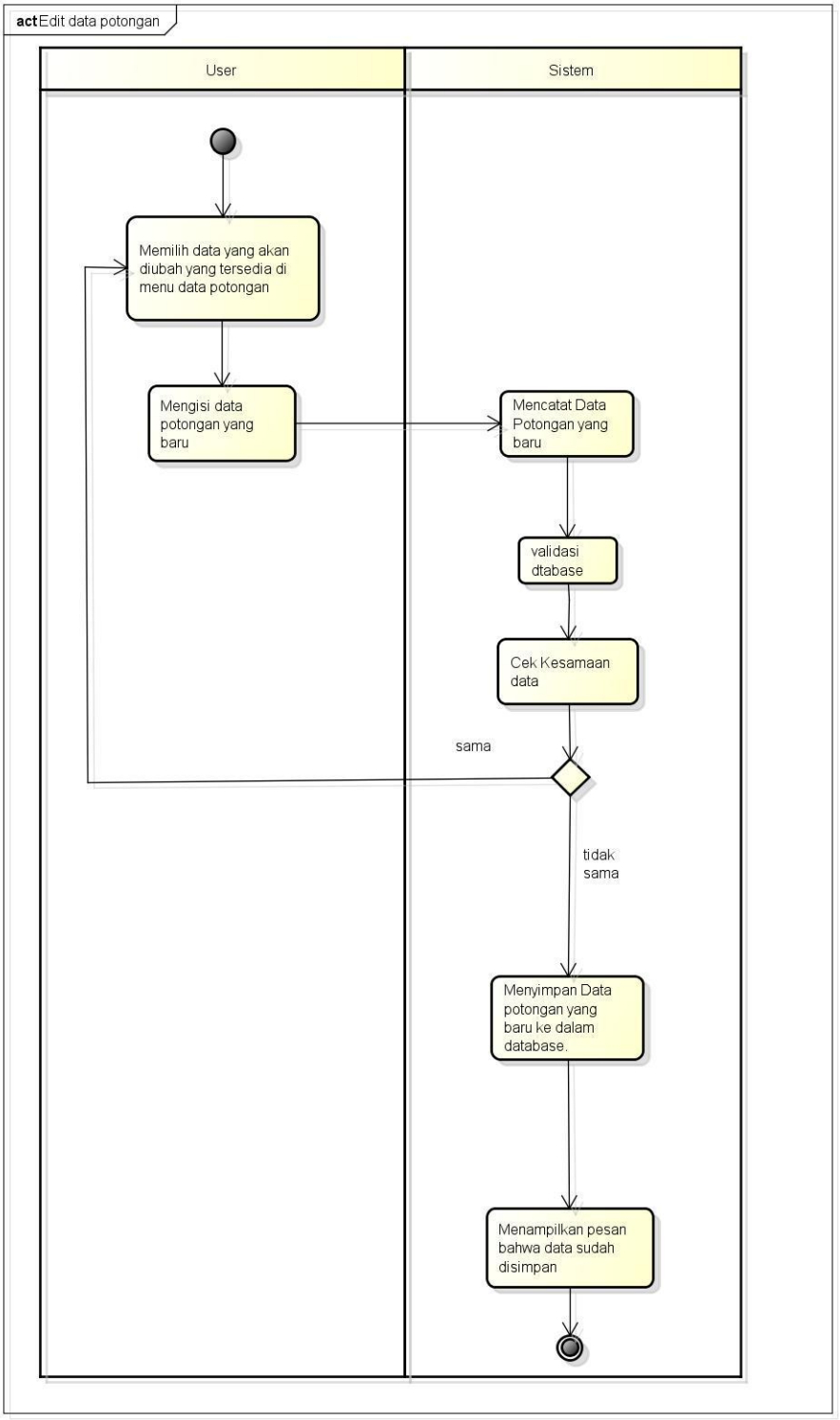
Gambar 3.14 *Activity Diagram* Hapus data jabatan

3.1.7.5.11 Menambah Data Potongan



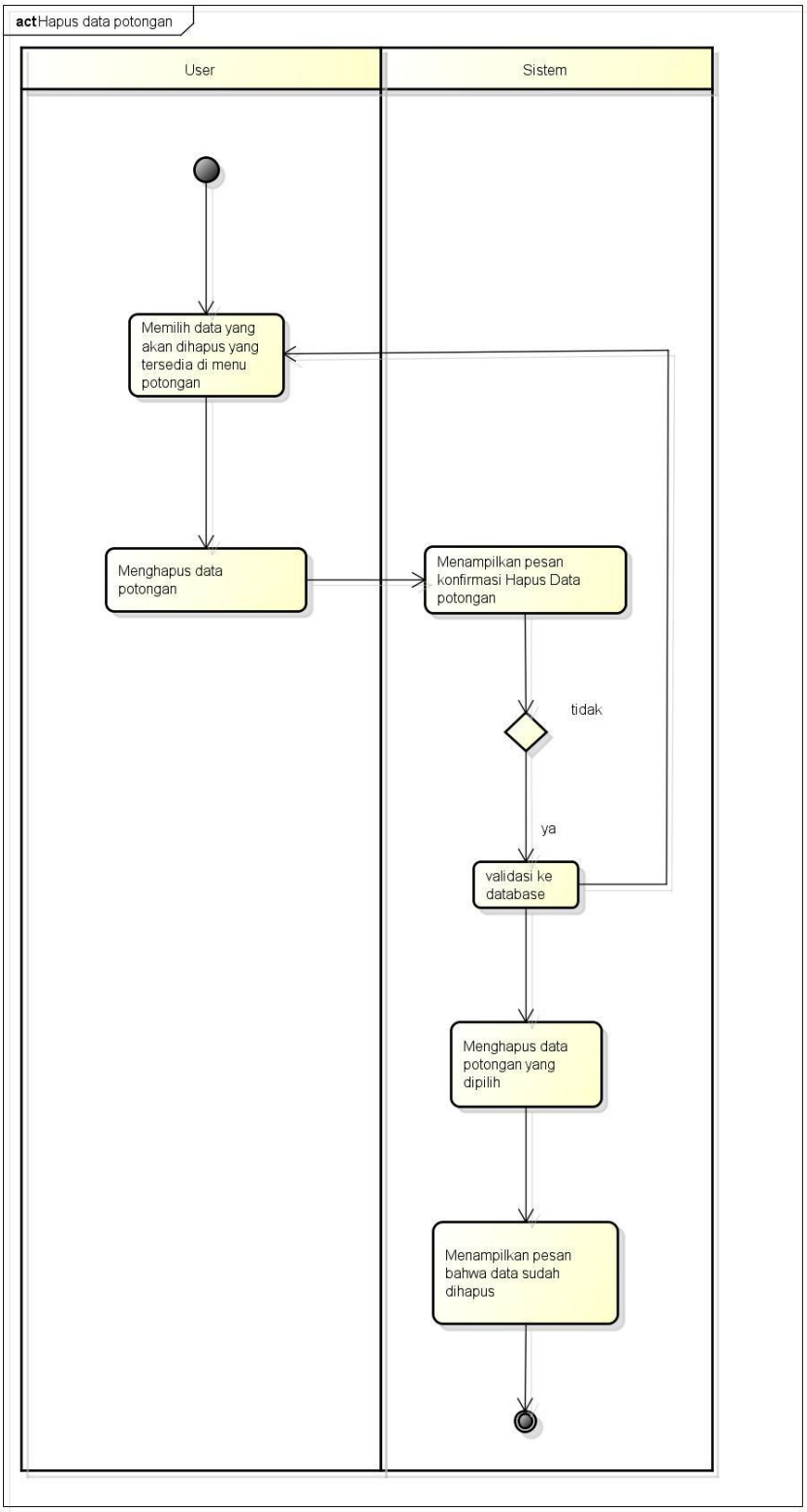
Gambar 3.15 *Activity Diagram* Menambah data potongan

3.1.7.5.12 *Edit* Data Potongan



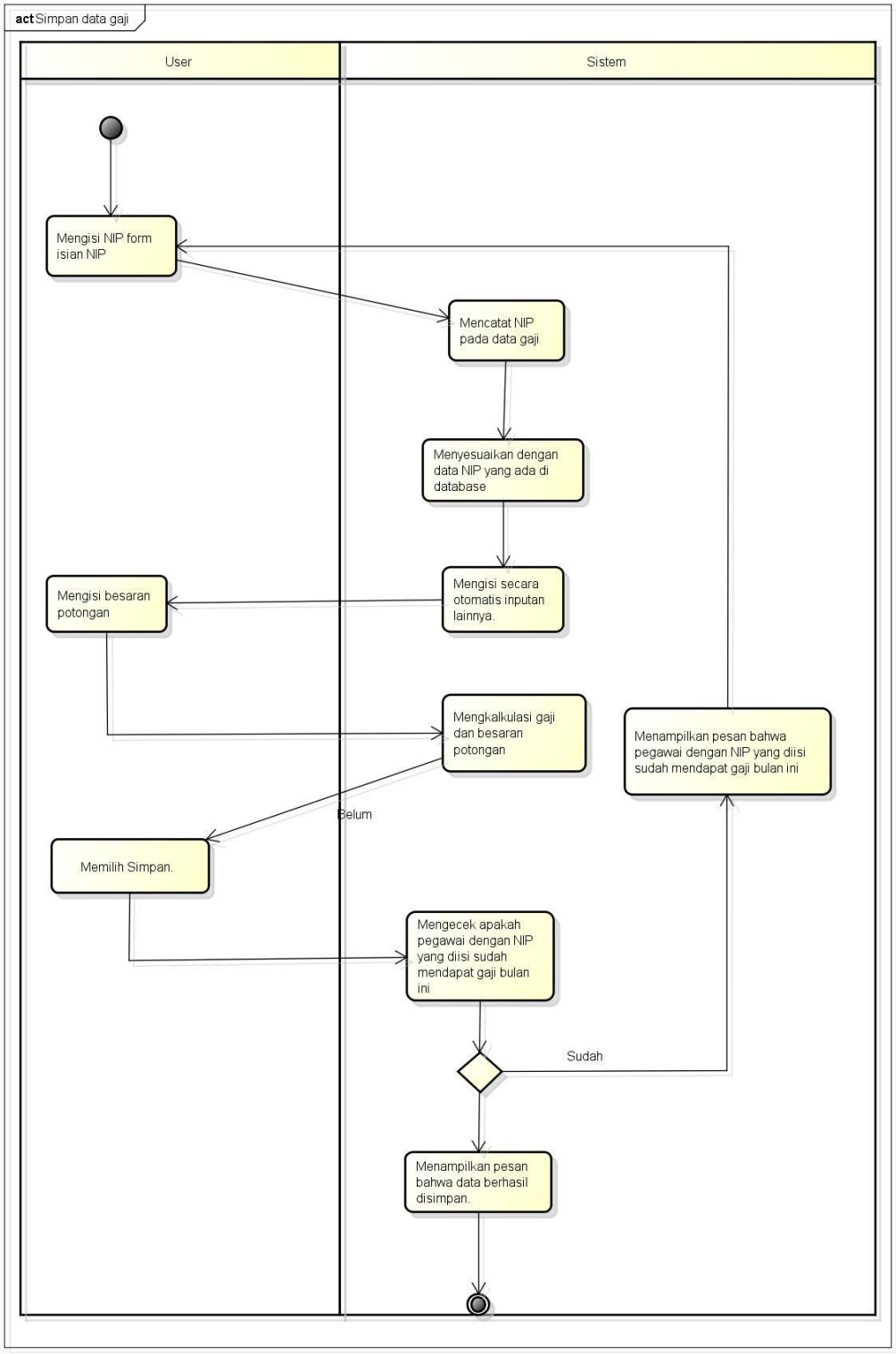
Gambar 3.15 *Activity Diagram edit* data potongan

3.1.7.5.13 Hapus data potongan



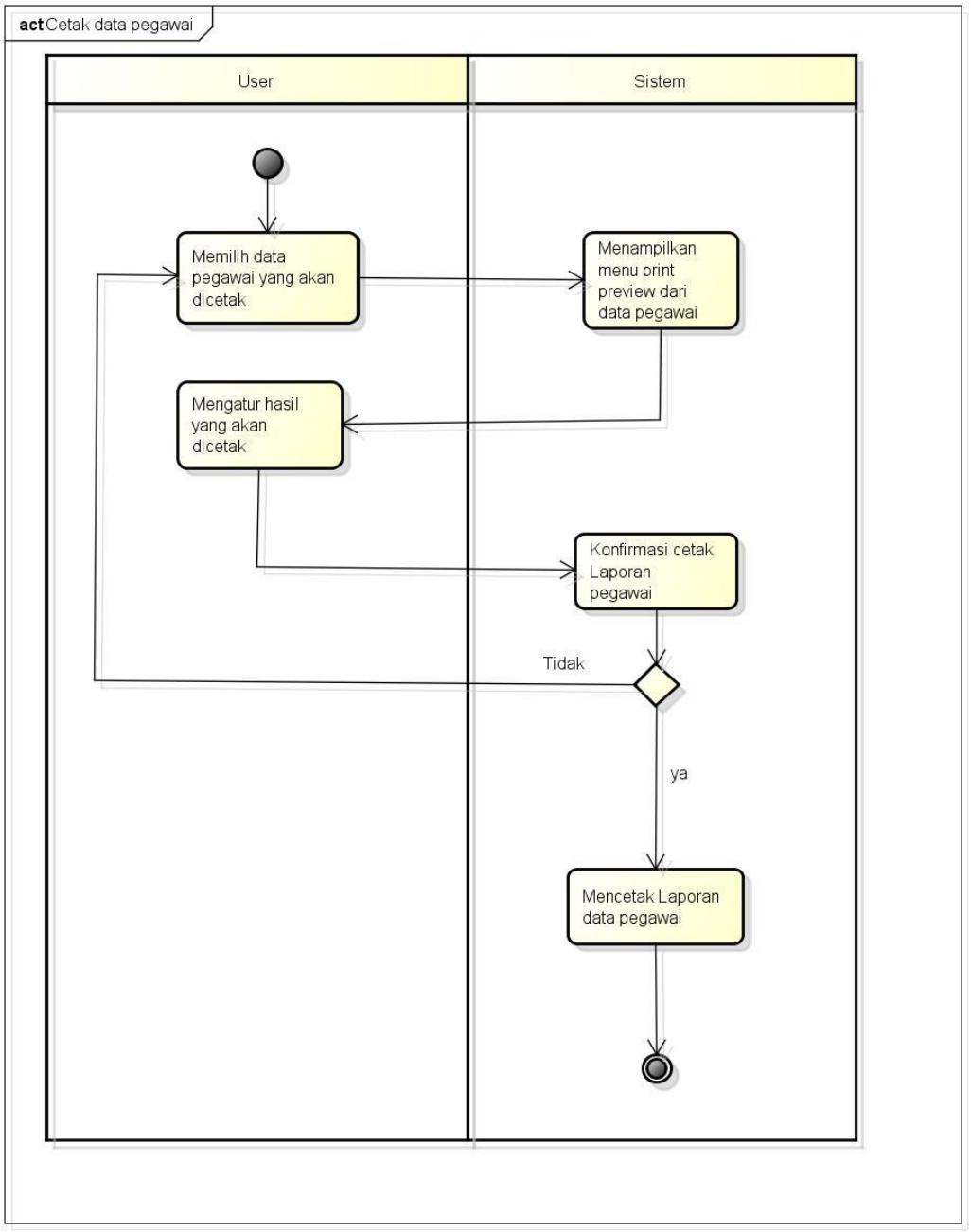
Gambar 3.16 *Activity Diagram* Hapus data potongan

3.1.7.5.14 Menyimpan data gaji



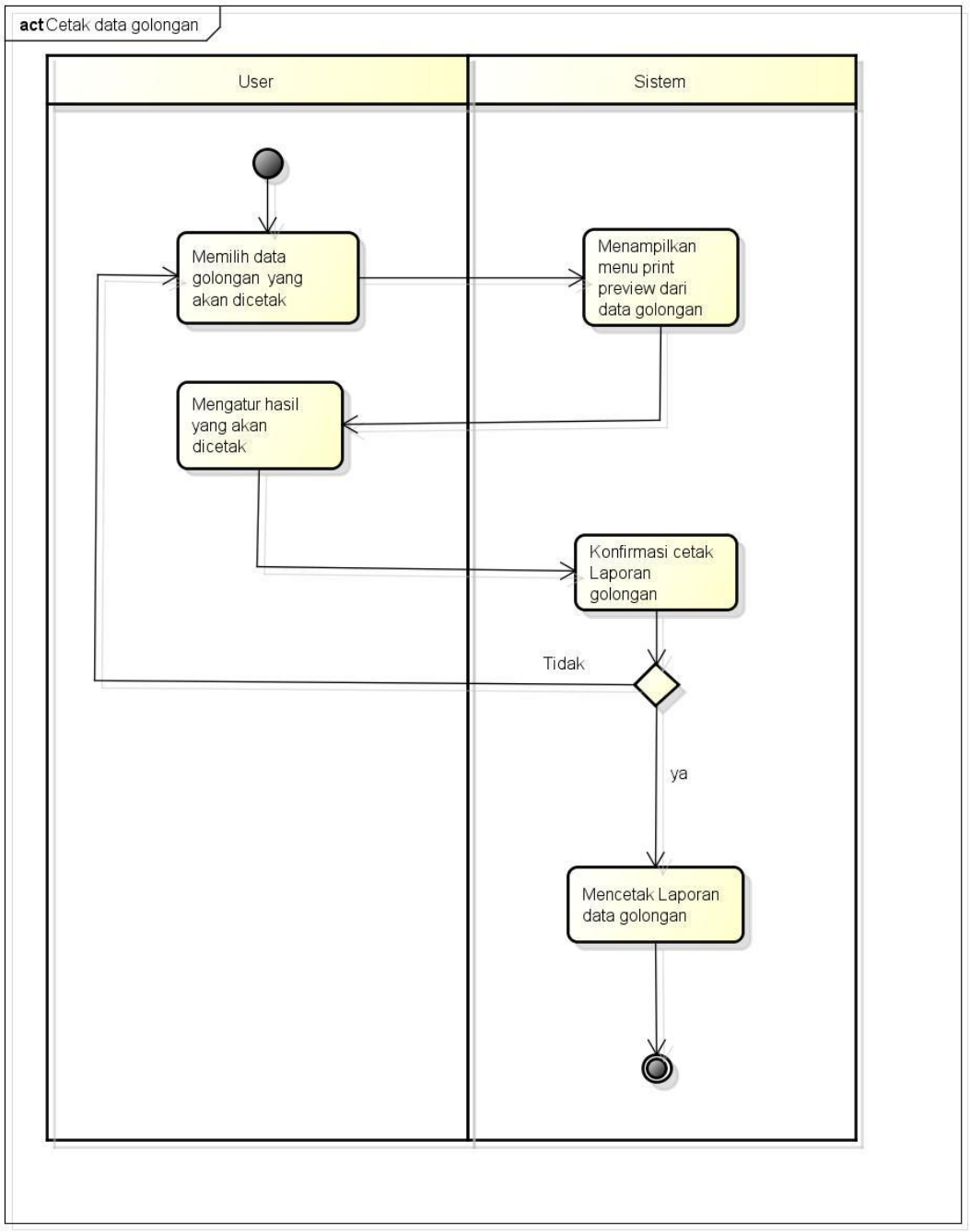
Gambar 3.17 *Activity Diagram* Menyimpan data gaji

3.1.7.5.15 Cetak laporan pegawai



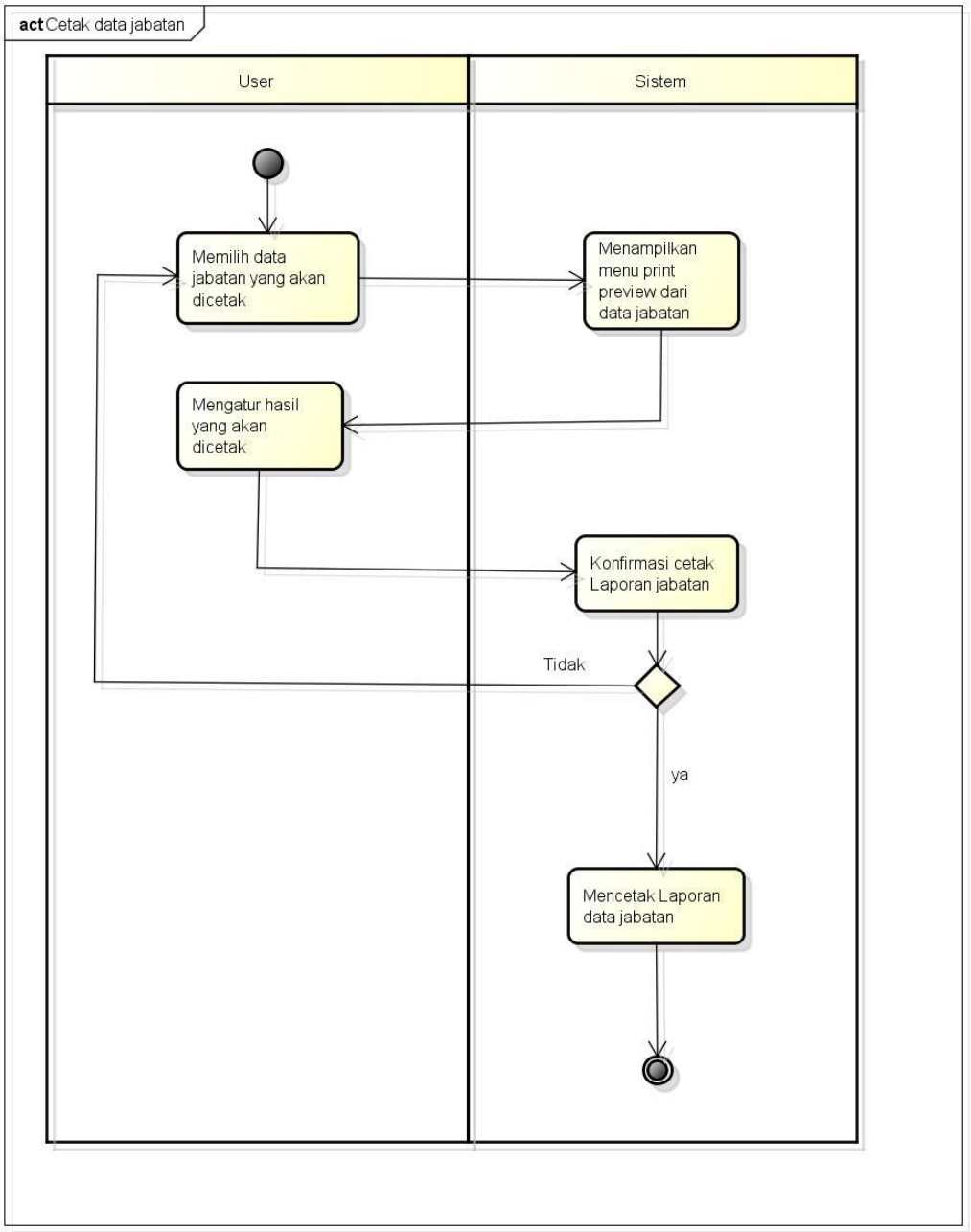
Gambar 3.18 *Activity Diagram* Cetak Laporan Pegawai

3.1.7.5.16 Cetak laporan golongan



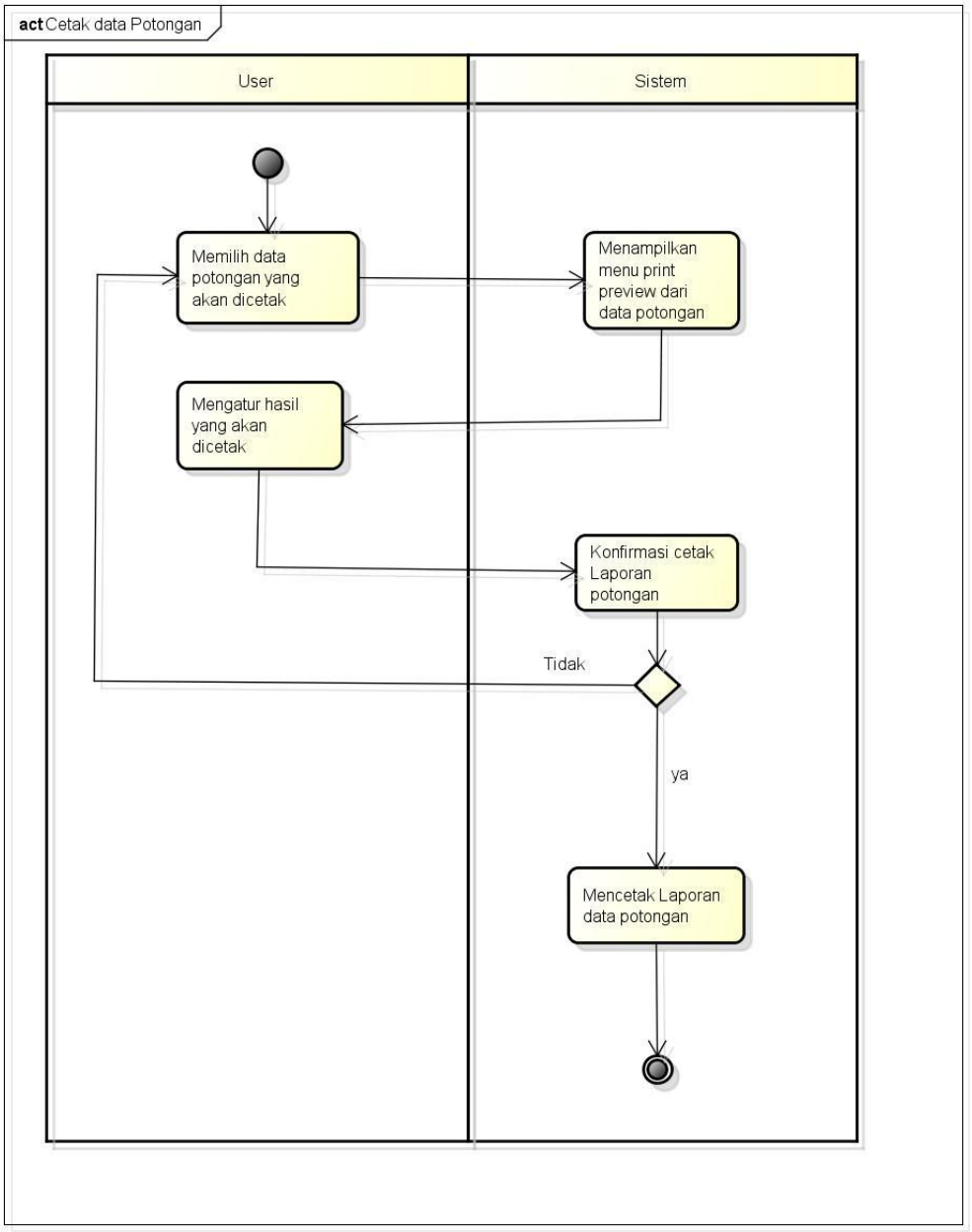
Gambar 3.19 *Activity Diagram* Cetak Laporan golongan

3.1.7.5.17 Cetak laporan jabatan



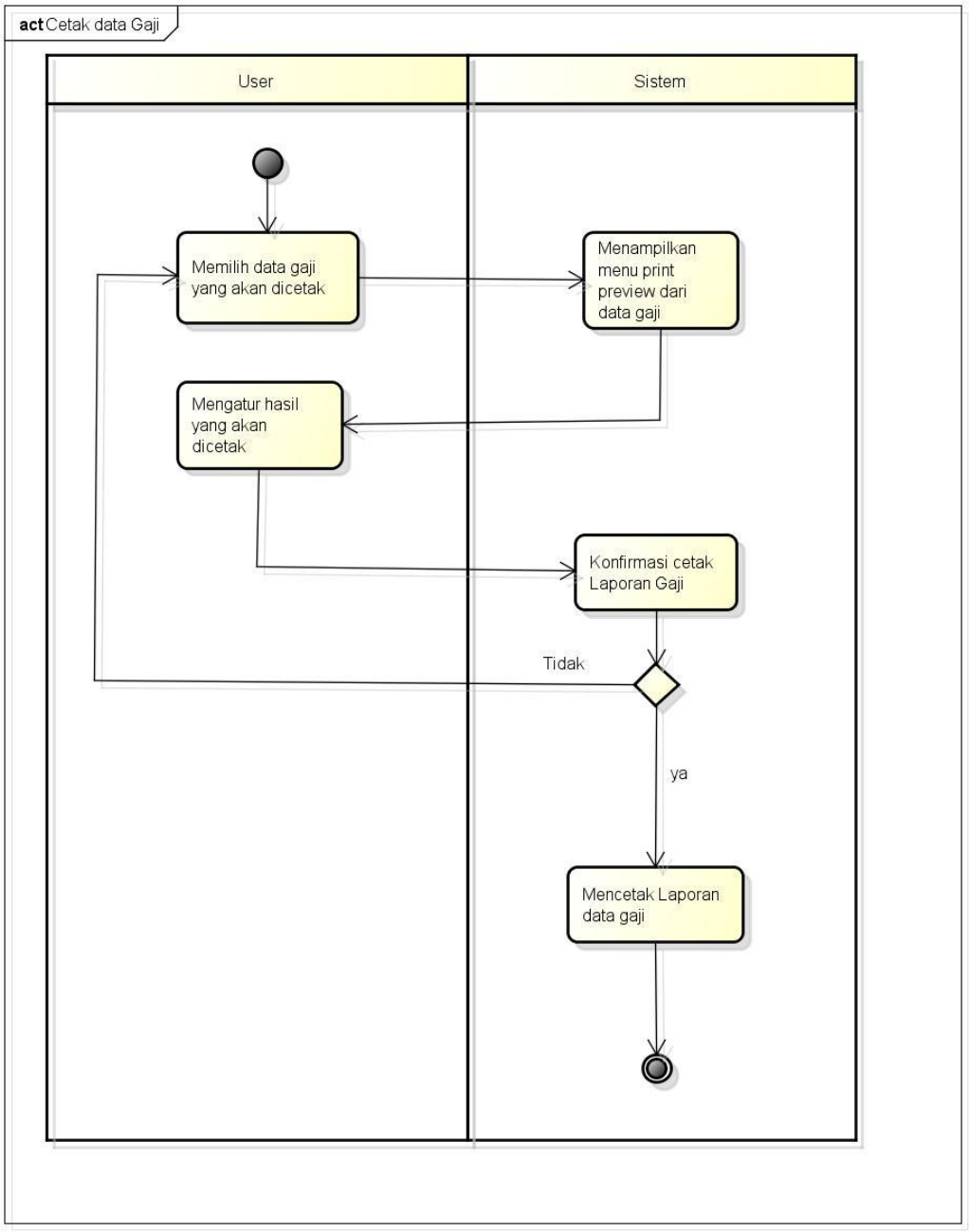
Gambar 3.20 *Activity Diagram* Cetak Laporan jabatan

3.1.7.5.18 Cetak laporan potongan



Gambar 3.21 *Activity Diagram* Cetak Laporan Potongan

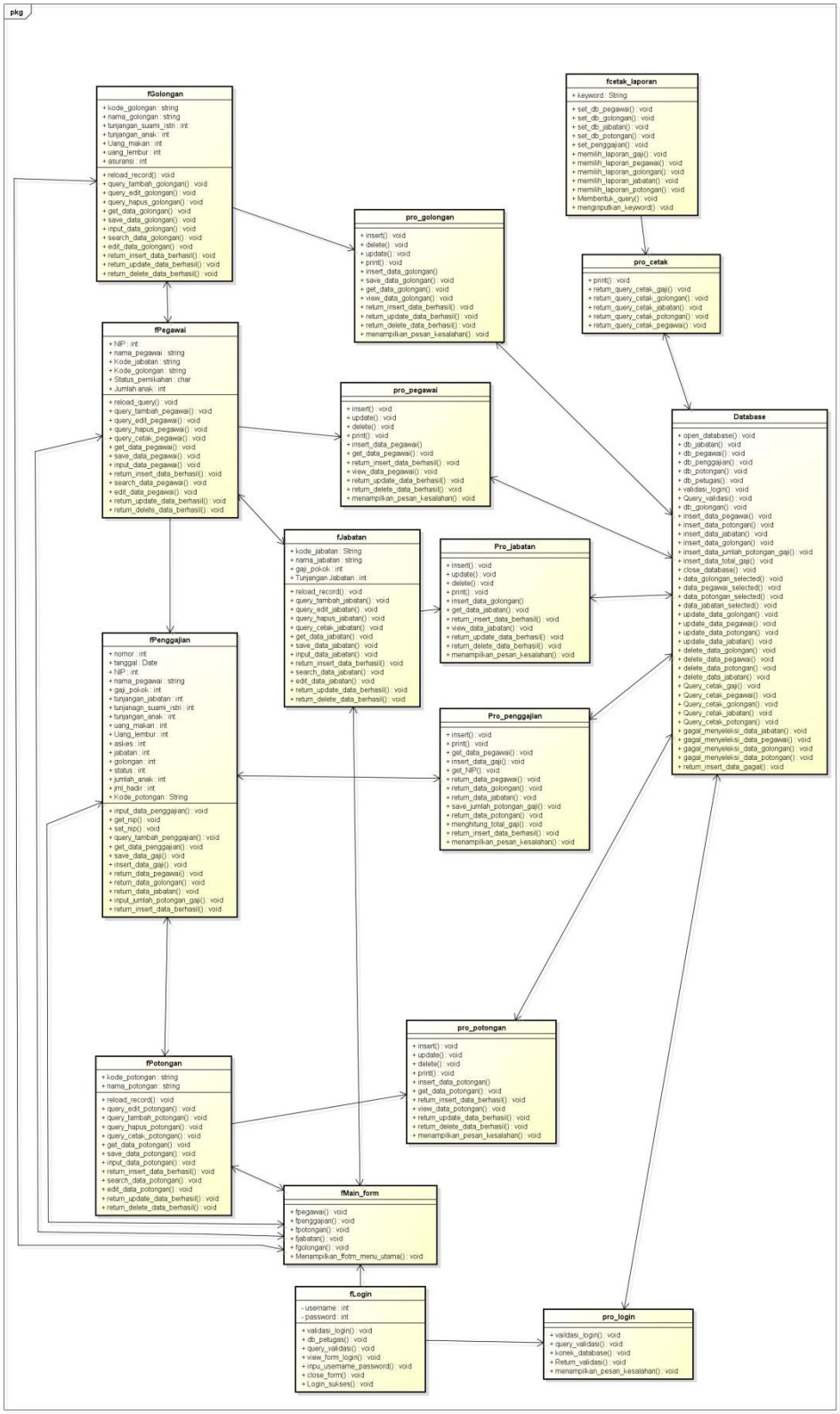
3.1.7.5.19 Cetak laporan Gaji



Gambar 3.22 *Activity Diagram* Cetak Laporan Gaji

3.1.7.6 Class Diagram

Menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. Dengan melihat karakteristik sistem informasi kepegawaian mulai dari data pegawai sampai dengan penggajian beserta proses-proses yang terjadi, maka dapat dibuat C*lass Diagram.* Berikut *Class Diagram* Aplikasi Kepegawaian di kantor Balai Besar Pengembangan Budidaya Air Tawar Sukabumi.



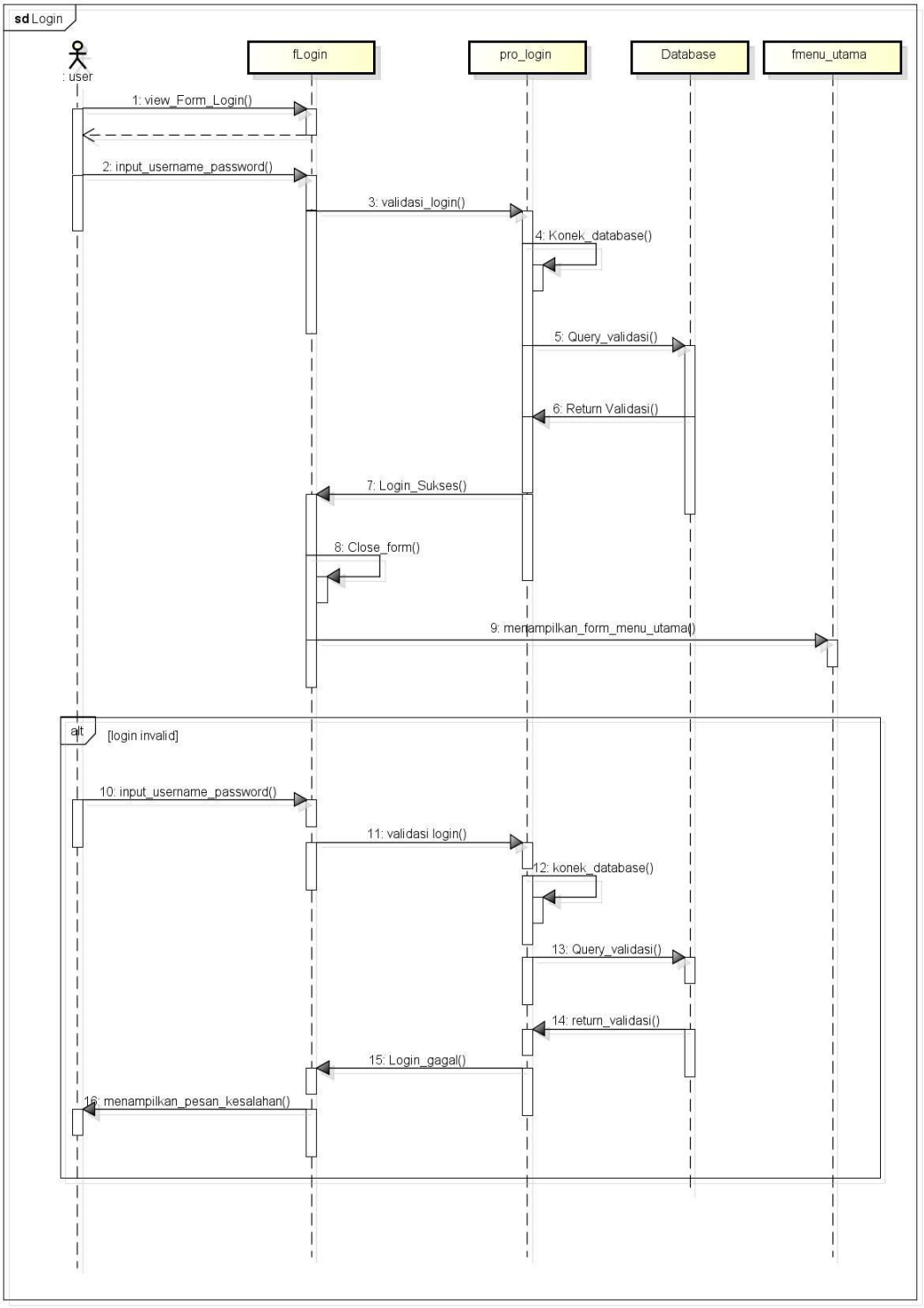
Gambar 3.23 *Class Diagram* Aplikasi Kepegawaian di BBPBAT

3.1.7.7 *Sequence diagram*

Sequence Diagram digunakan untuk menggambarkan interaksi antar objek dalam waktu yang berurutan. Tetapi pada dasarnya sequence Diagram selain digunakan dalam lapisan abstraksi model objek. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Komponen utama *sequence* diagram terdiri atas objek yang dituliskan dengan kotak segiempat bernama pesan diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan proses vertikal.

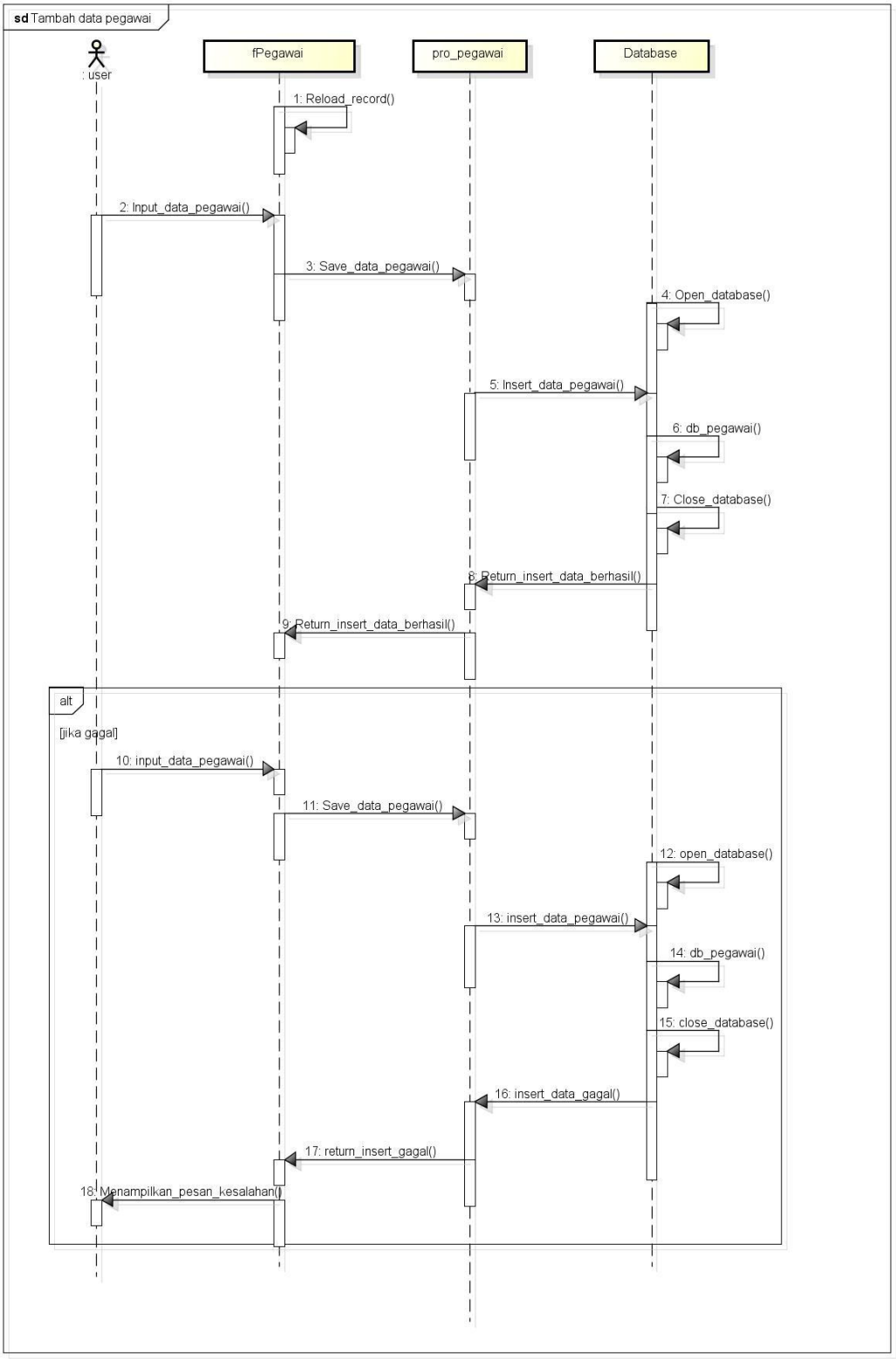
Berikut adalah *sequence diagram* yang ada pada kantor Balai Besar Pengembangan Budidaya Air Tawar Sukabumi, yaitu:

3.1.7.7.1 Login



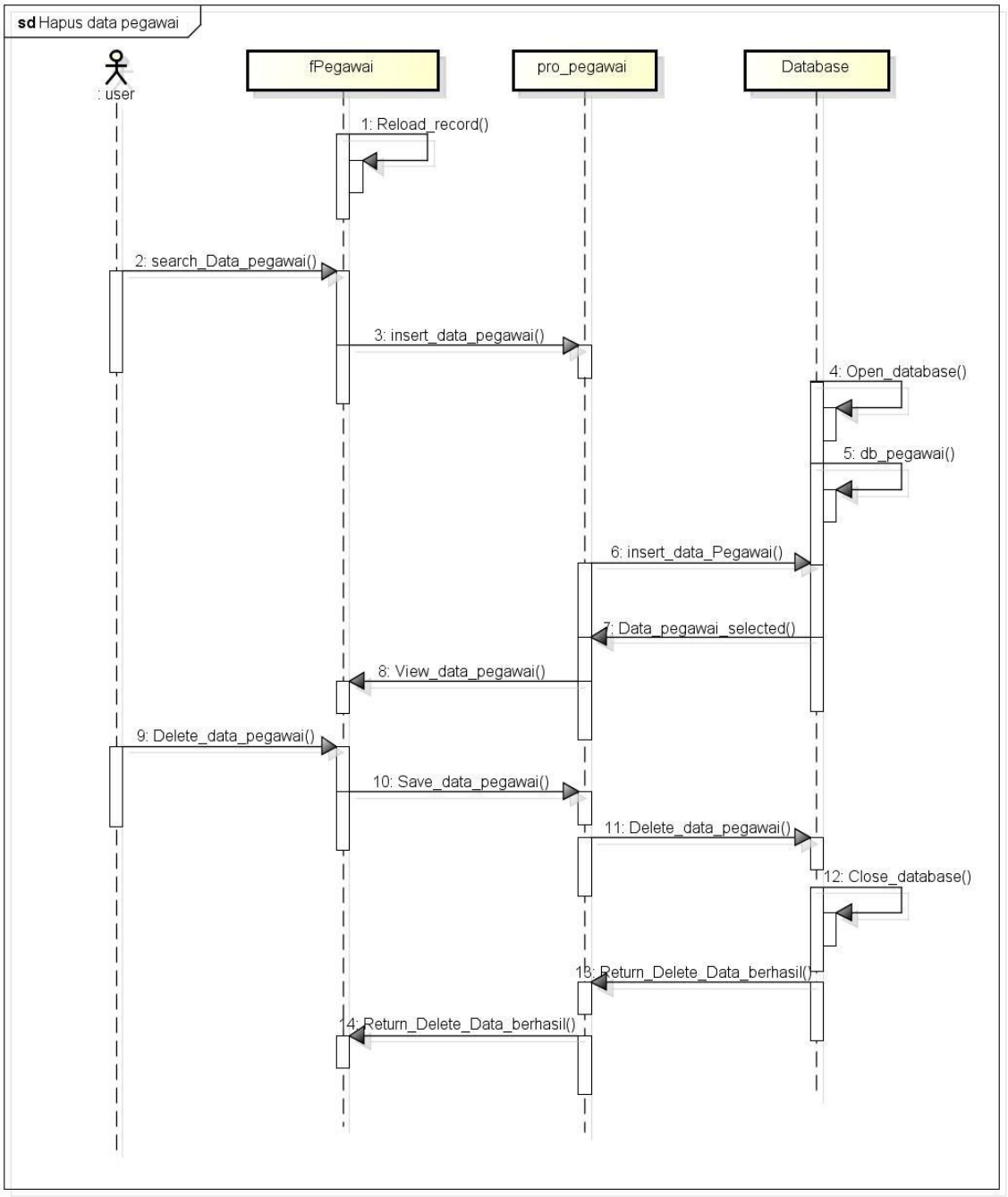
Gambar 3.24 *Sequence Diagram* Logi

3.1.7.7.2 Menambah data pegawai



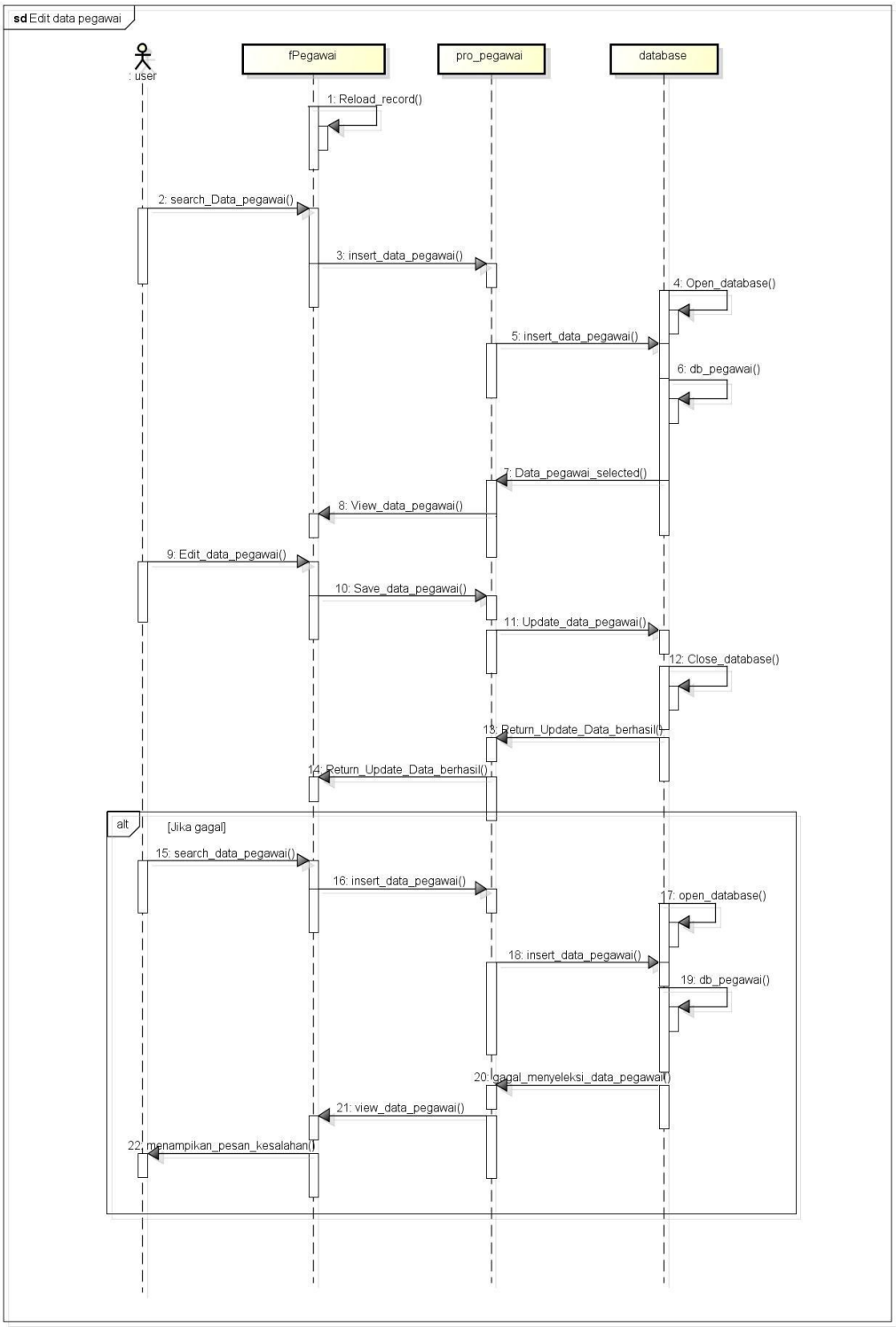
Gambar 3.25 *Sequence Diagram* Menambah data pegawai

3.1.7.7.3 Menghapus data Pegawai



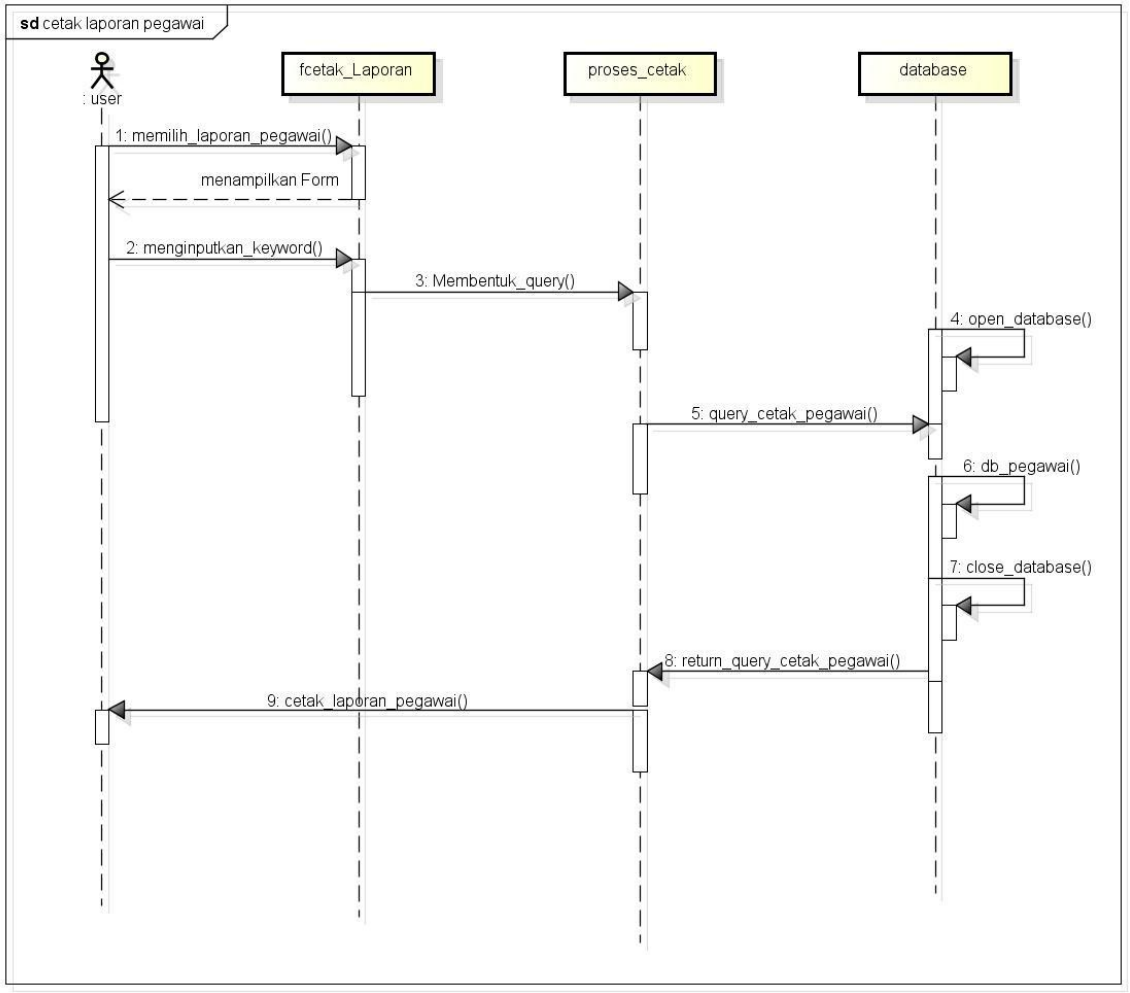
Gambar 3.26 *Sequence Diagram* Menghapus data pegawai

3.1.7.7.4 *Edit* data Pegawai



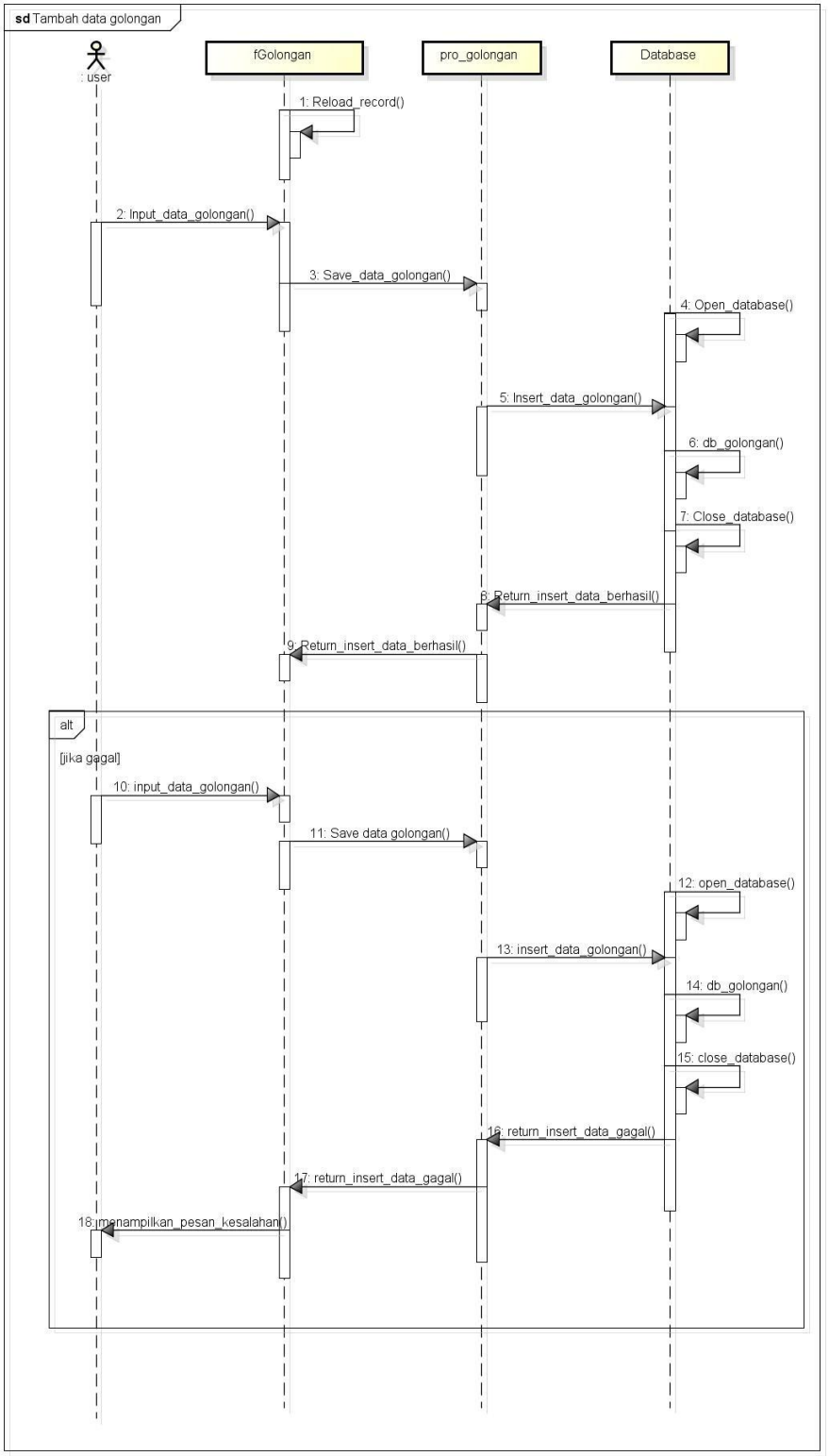
Gambar 3.27 *Sequence Diagram Edit* data pegawai

3.1.7.7.5 Cetak Laporan Pegawai



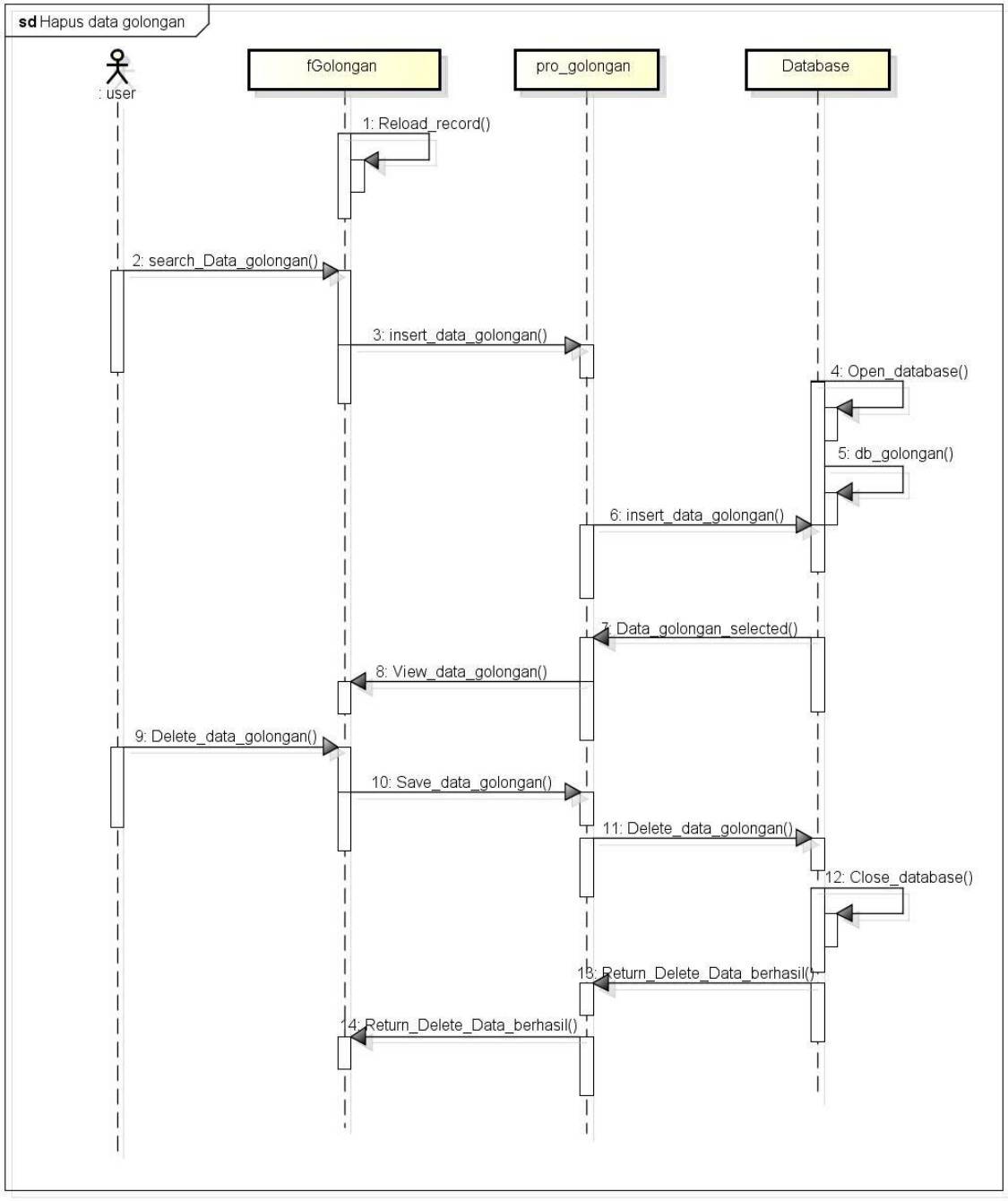
Gambar 3.28 *Sequence Diagram* Cetak data pegawai

3.1.7.7.6 Menambah data Golongan



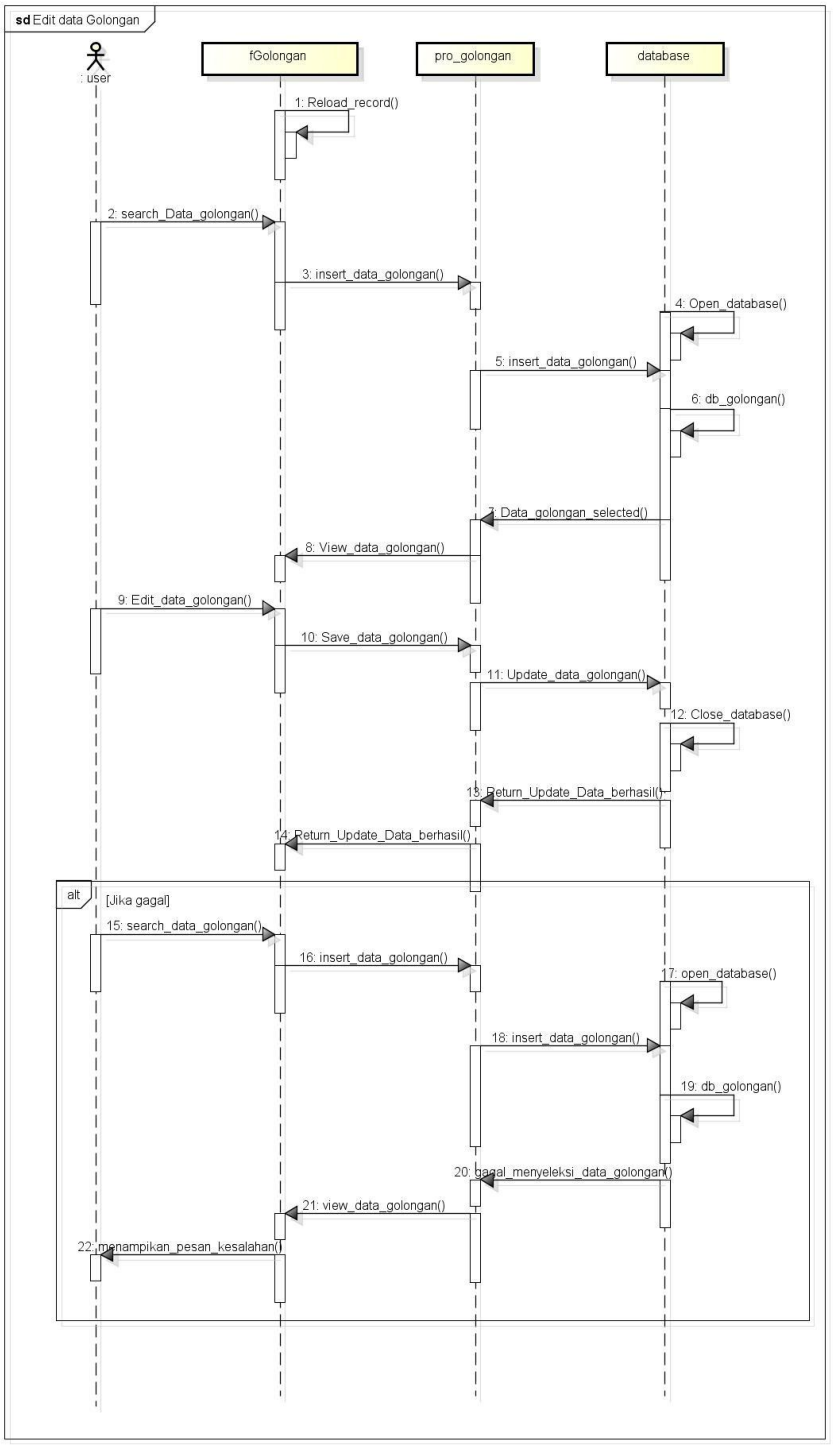
Gambar 3.29 *Sequence Diagram* Menambah data golongan

3.1.7.7.7 Menghapus data golongan



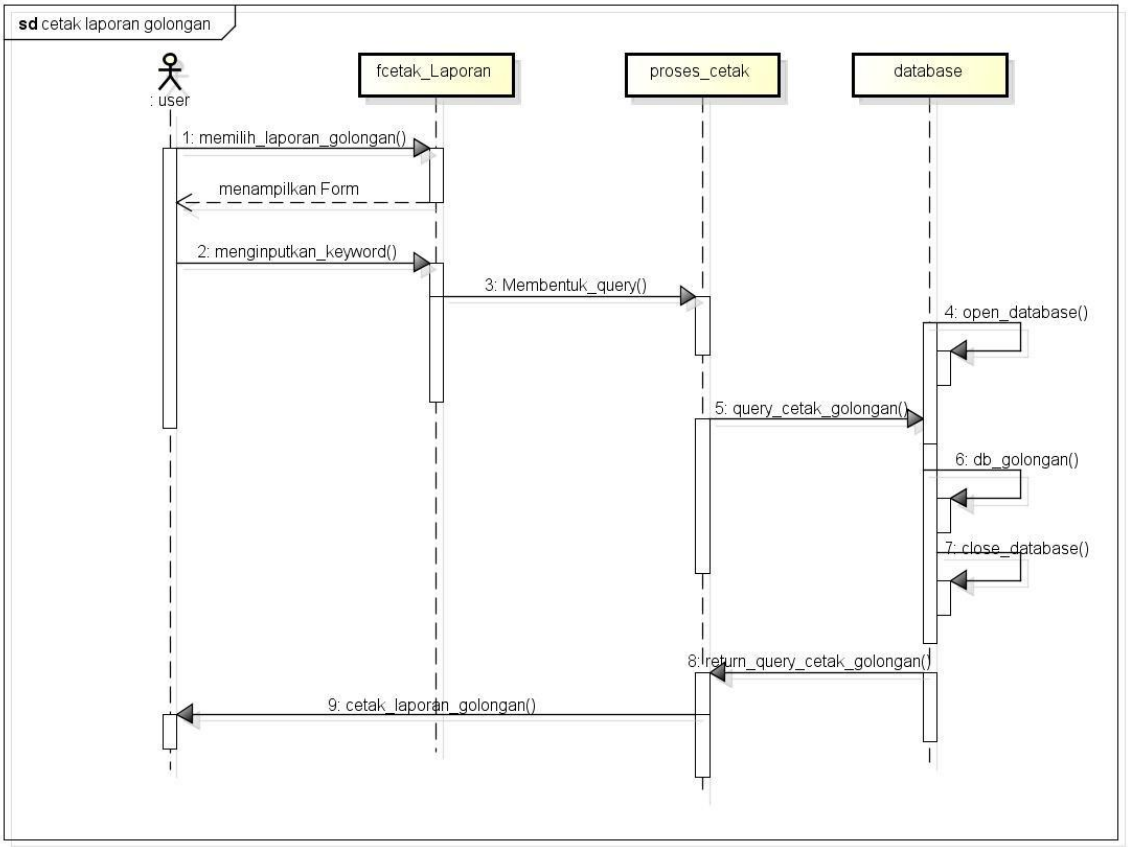
Gambar 3.30 *Sequence Diagram* Menghapus data golongan

3.1.7.7.8 *Edit* Data Golongan



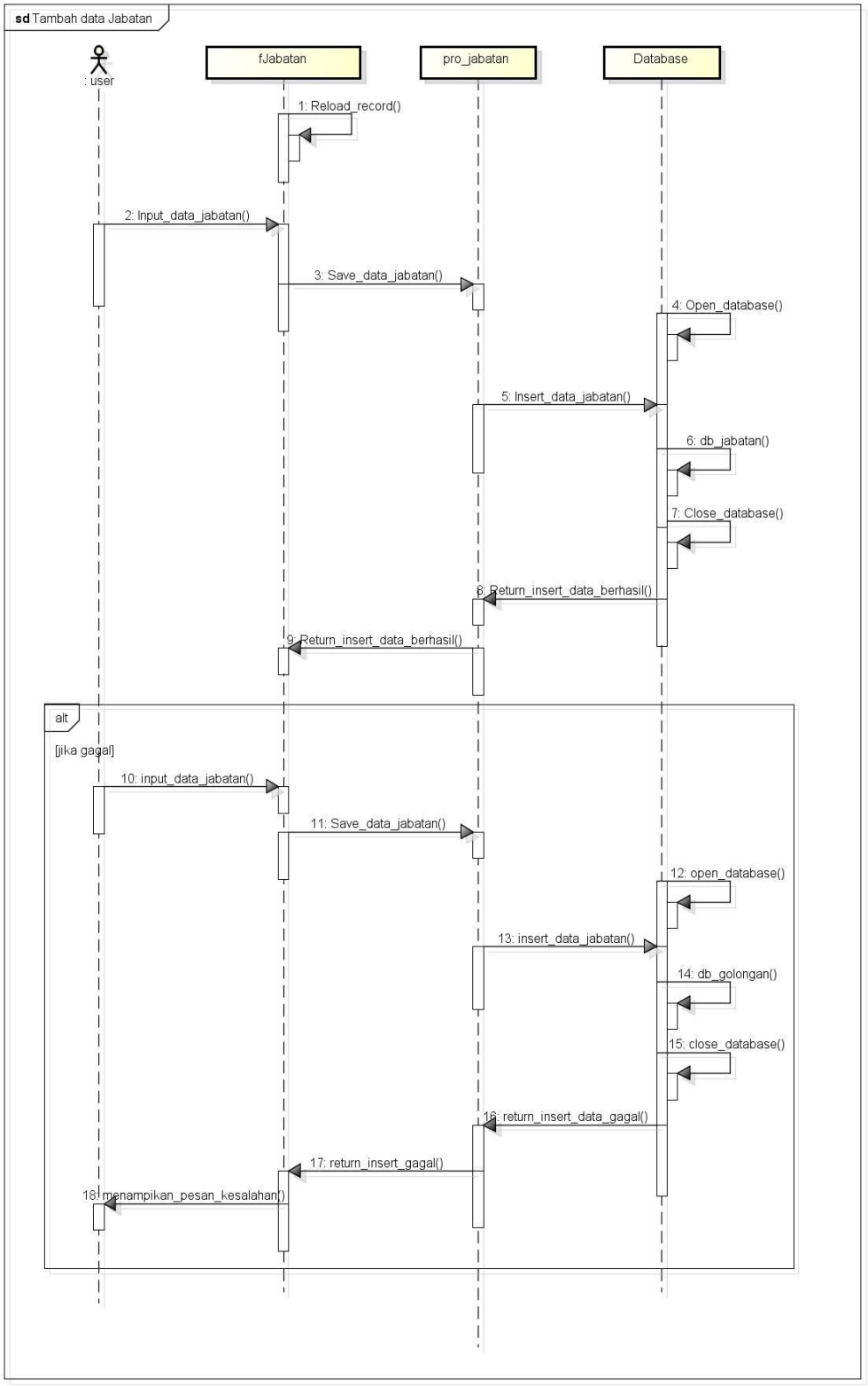
Gambar 3.31 *Sequence Diagram edit* data golongan

3.1.7.7.9 Cetak data golongan



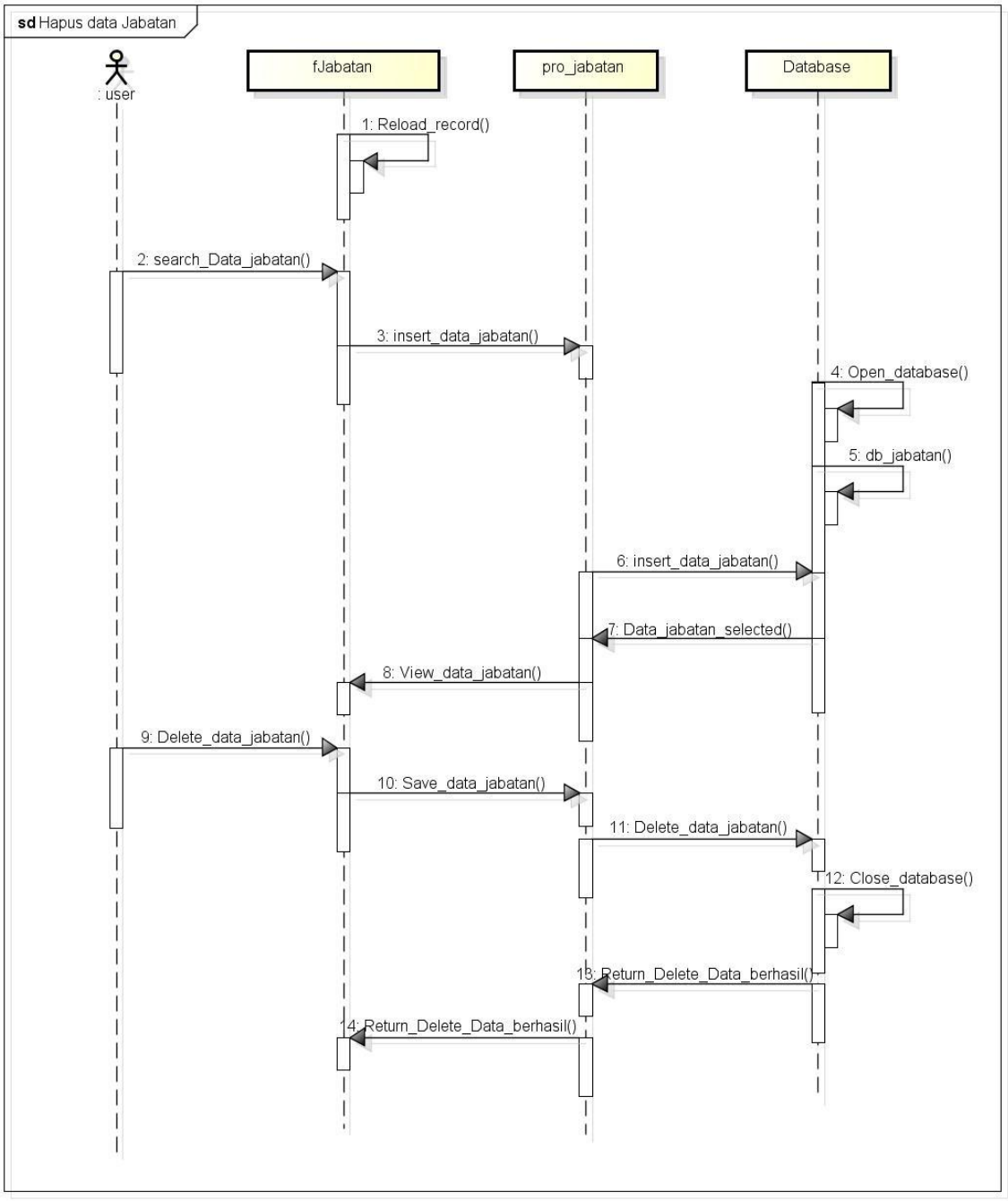
Gambar 3.32 *Sequence Diagram* Cetak data golongan

3.1.7.7.10 Menambah Data jabatan



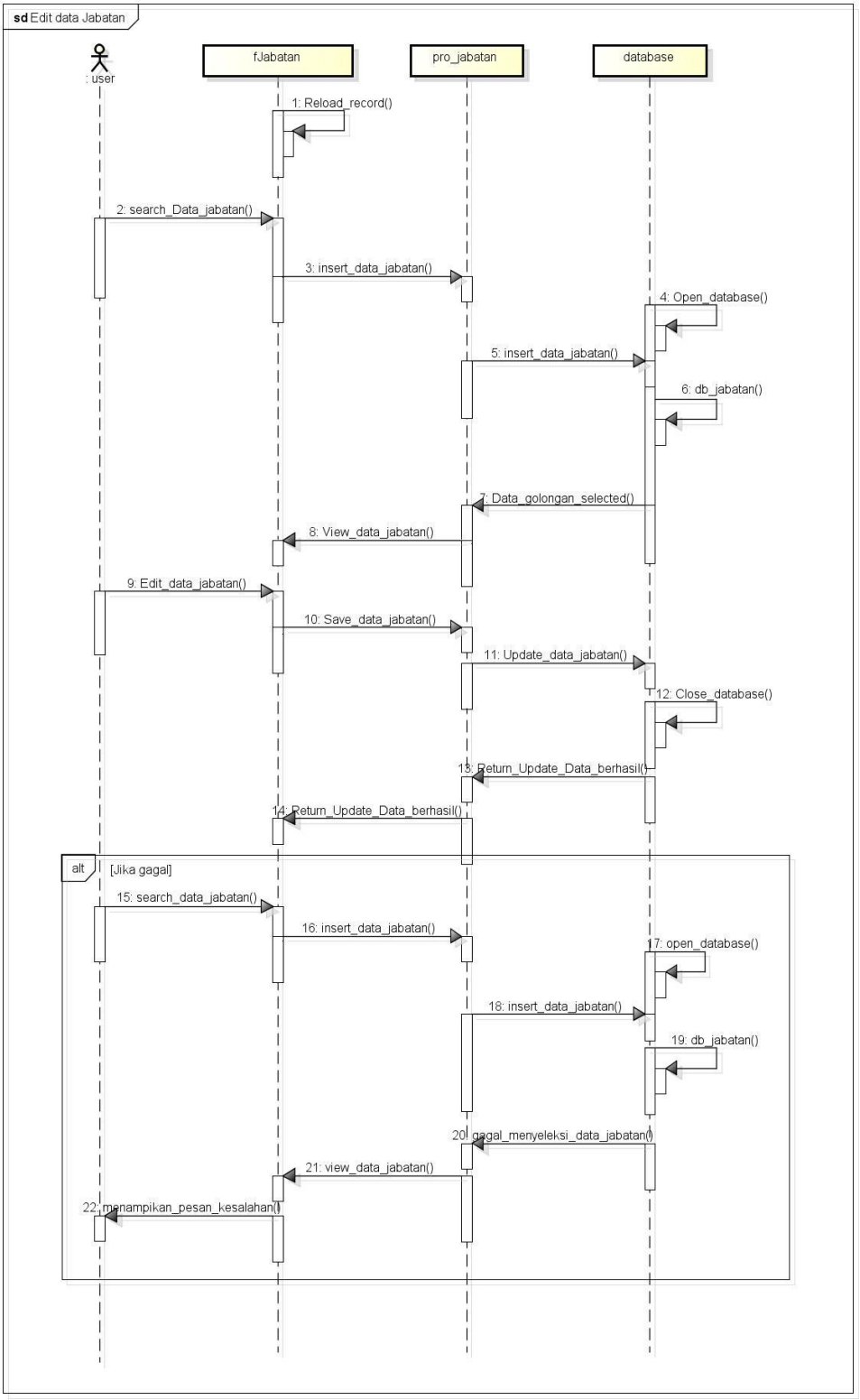
Gambar 3.33 *Sequence Diagram* Menambah data jabatan

3.1.7.7.11 Hapus data jabatan



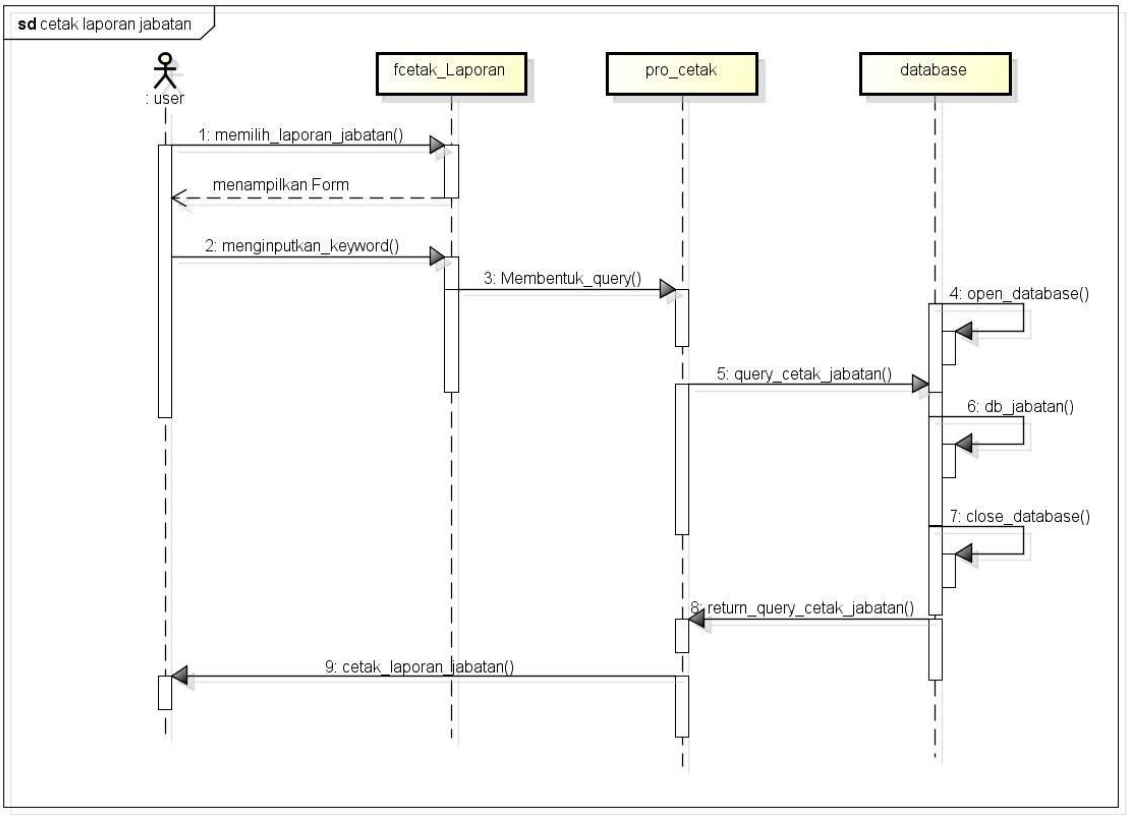
Gambar 3.34 *Sequence Diagram* Hapus data jabatan

3.1.7.7.12 *Edit* data jabatan



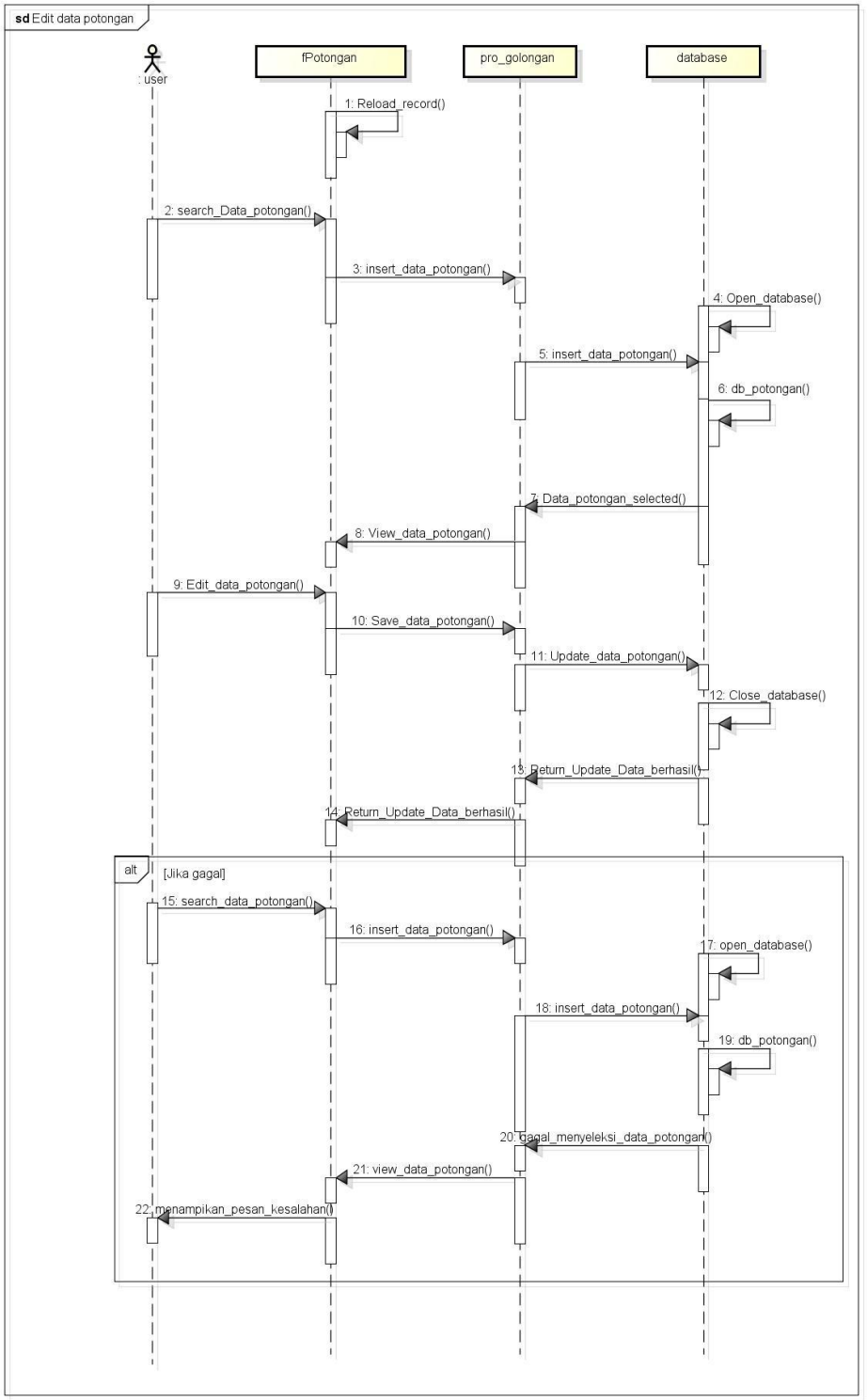
Gambar 3.35 *Sequence Diagram Edit* data jabatan

3.1.7.7.13 Cetak Data Jabatan



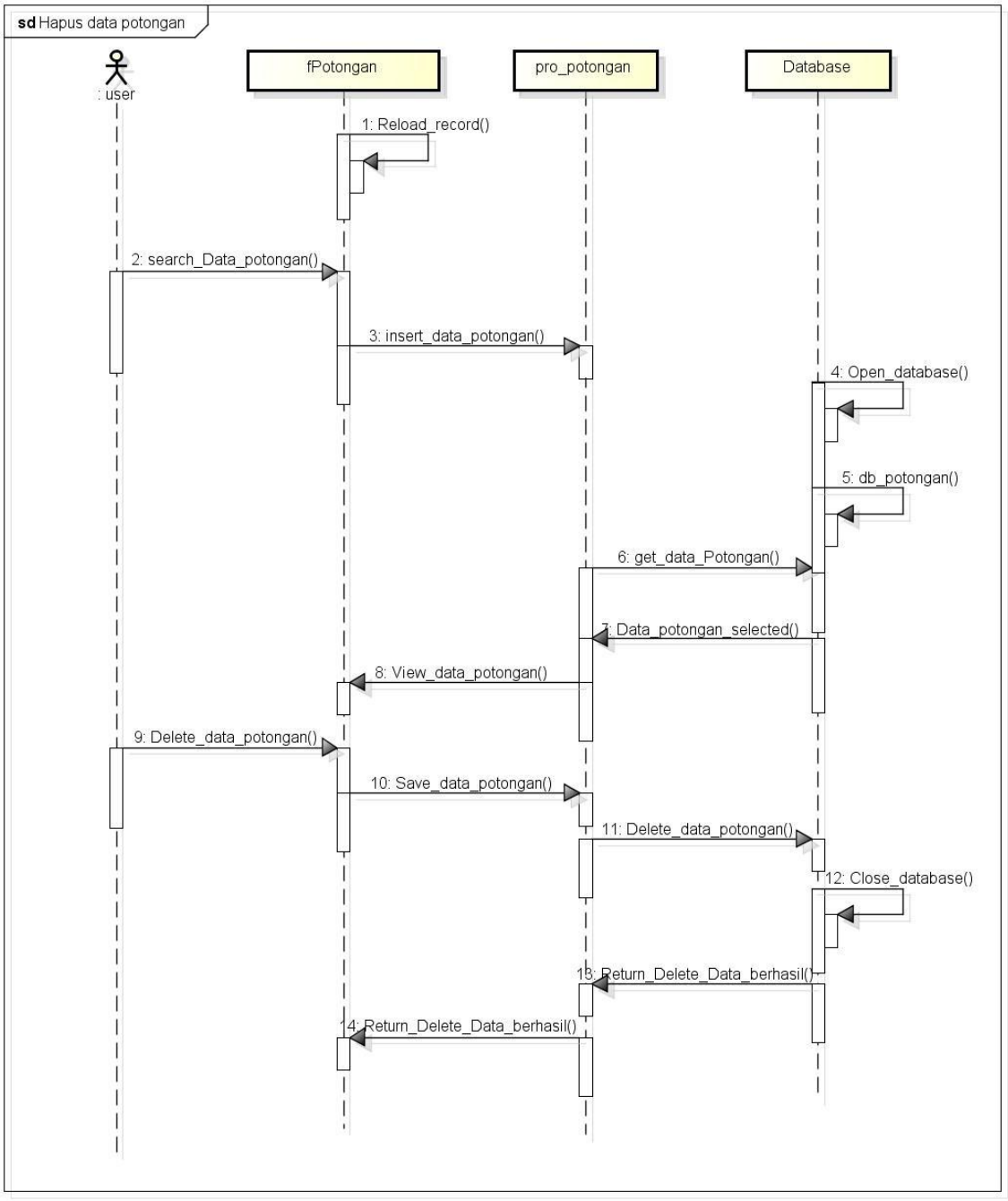
Gambar 3.36 *Sequence Diagram* cetak data jabatan

3.1.7.7.14 Menambah data potongan



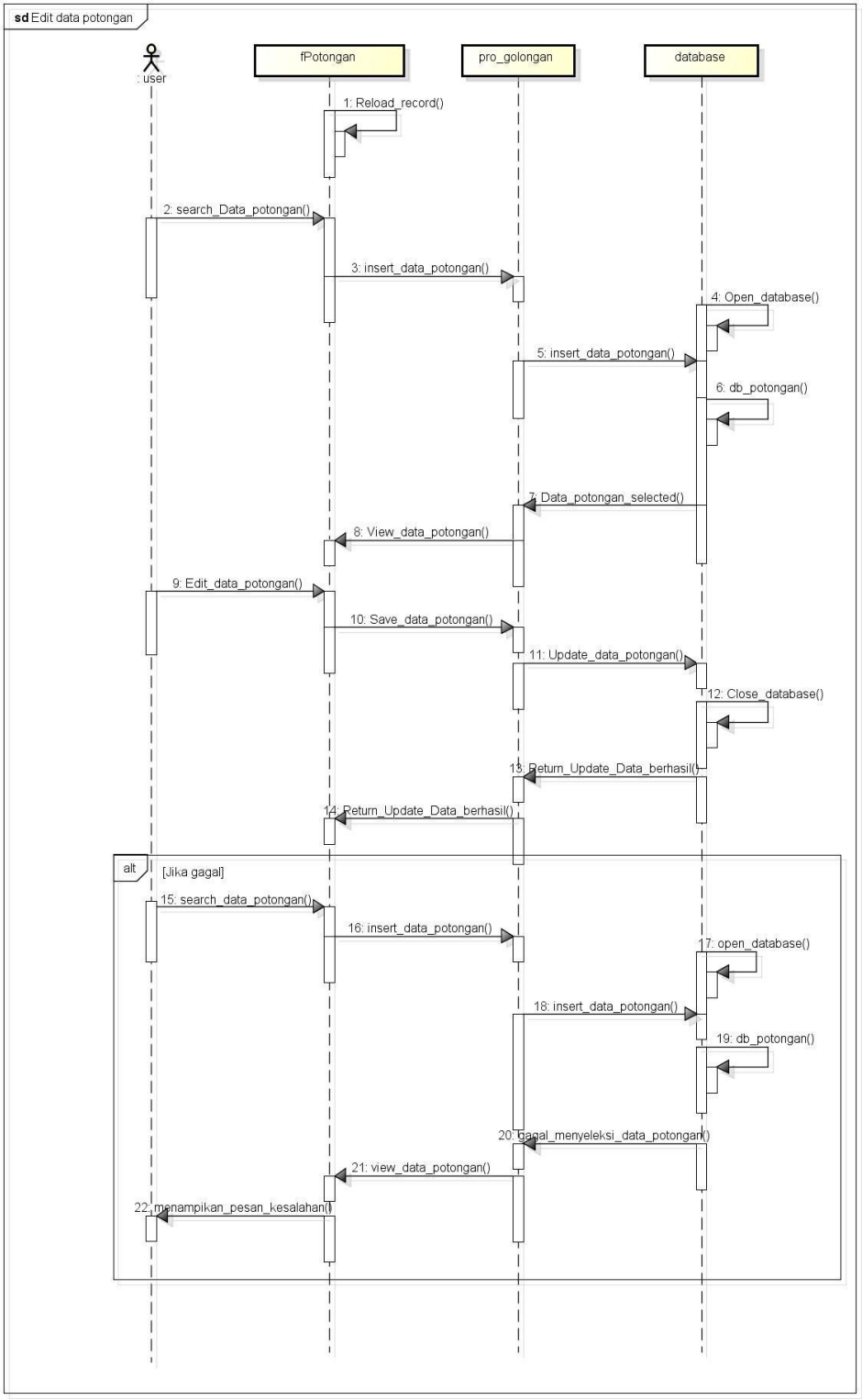
Gambar 3.37 *Sequence Diagram* Menambah data potongan

3.1.7.7.15 Hapus data potongan



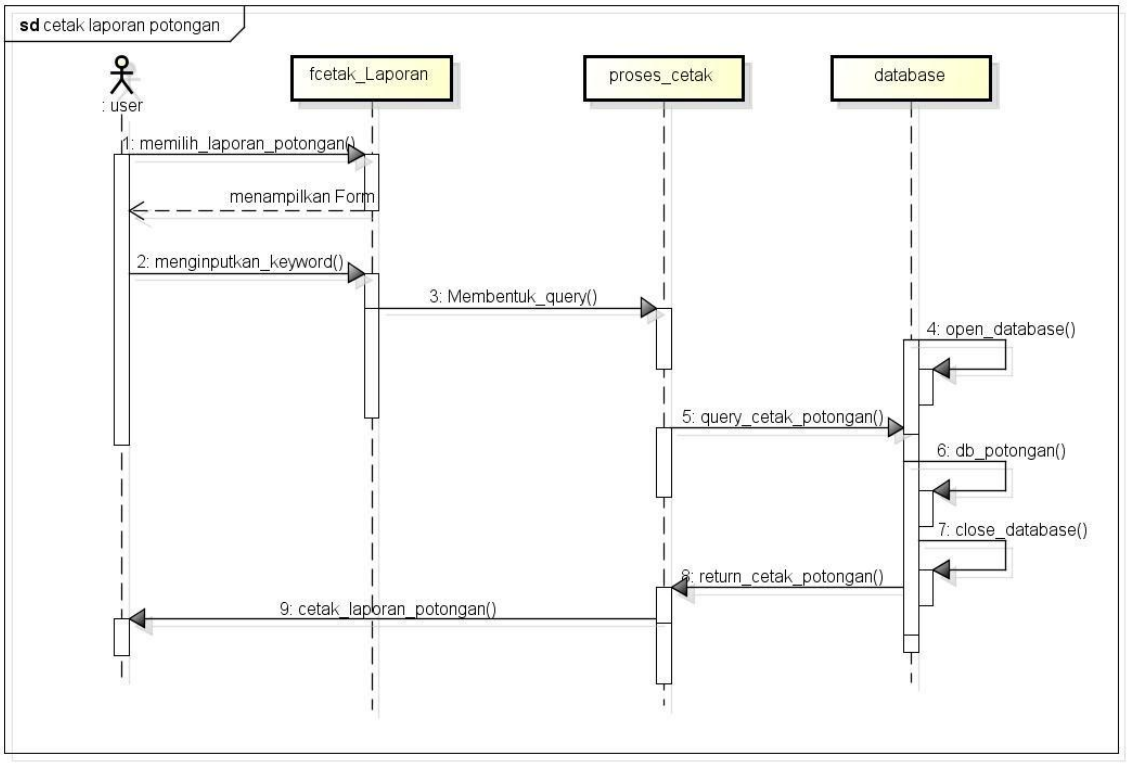
Gambar 3.38 *Sequence Diagram* Hapus data potongan

3.1.7.7.16 *Edit* data potongan



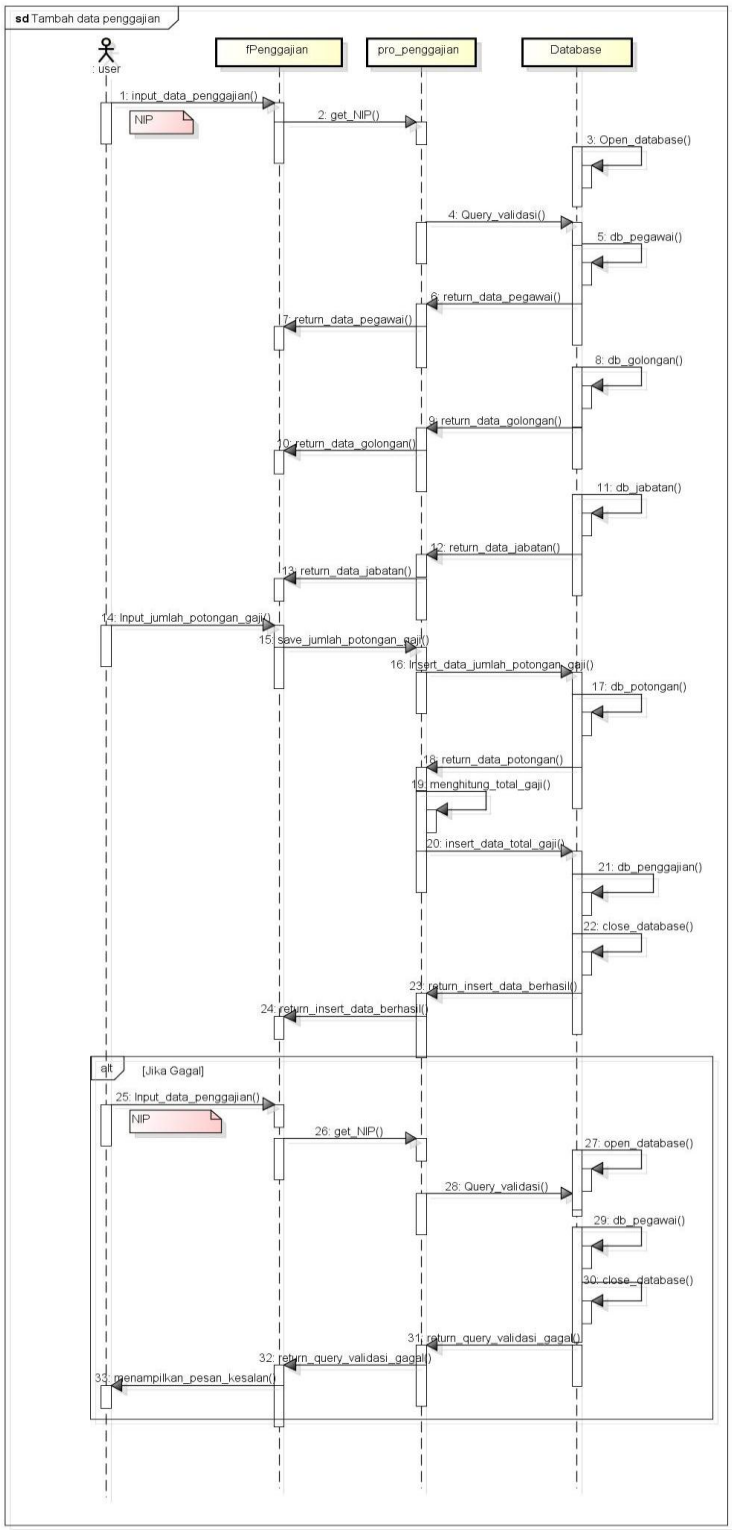
Gambar 3.39 *Sequence Diagram Edit* data potongan

3.1.7.7.17 Cetak data potongan



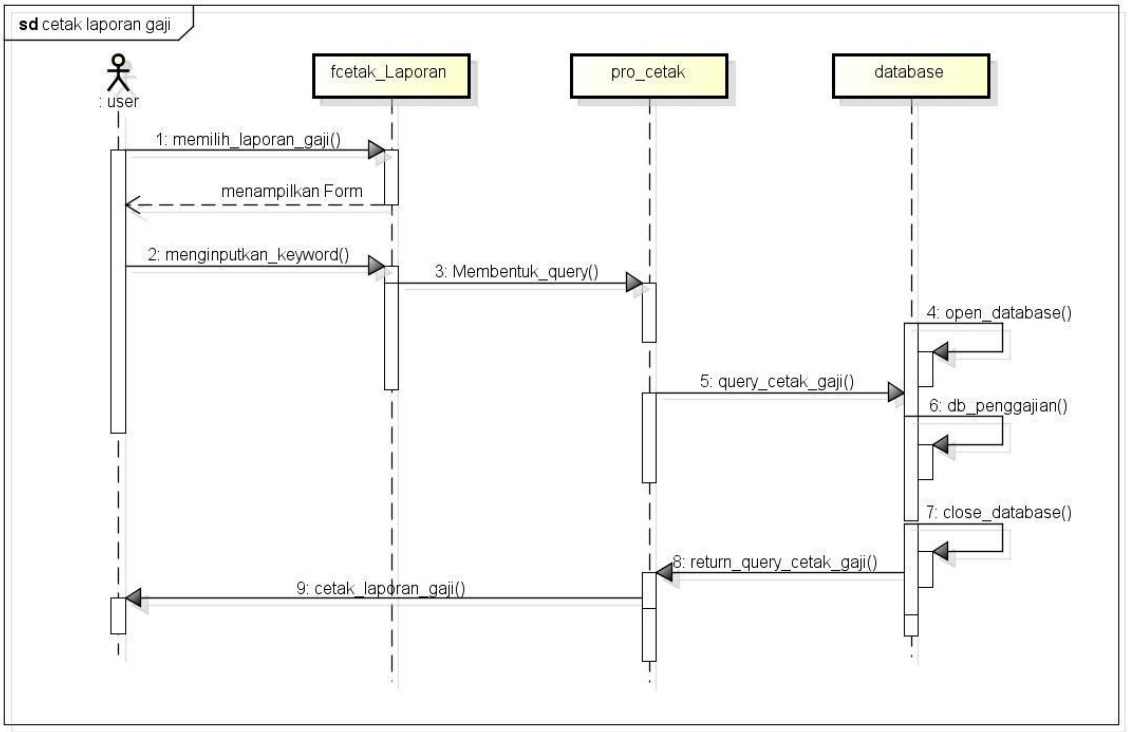
Gambar 3.40 *Sequence Diagram* Cetak data potongan

3.1.7.7.18 Menyimpan data gaji



Gambar 3.41 *Sequence Diagram* Menyimpan data gaji

3.1.7.7.19 Cetak data gaji



Gambar 3.42 *Sequence Diagram* cetak data gaji

3.2 Perancangan Sistem

Perancangan sistem adalah merancang atau mendesain suatu sistem yang baik, yang isinya adalah langkah-langkah operasi dalam proses pengolahan data dan prosedur untuk mendukung operasi sistem.

3.2.1 Perancangan Basis data

Perancangan basis data merupakan proses menciptakan perancangan untuk basis data yang akan mendukung operasi dan tujuan perusahaan[1]. Dalam merancang suatu basis data, digunakan metodologi-metodologi yang membantu dalam tahap perancangan basis data. Metodologi perancangan adalah pendekatan struktur dengan menggunakan prosedur, teknik, alat, serta bantuan dokumen untuk membantu dan memudahkan dalam proses perancangan. Dengan menggunakan teknik metode desain ini dapat

membantu dalam merencanakan, mengatur, mengontrol, dan mengevaluasi *database development project* [1].

Perancangan basis data terbagi menjadi tiga yaitu skema relasi, diagram relasi dan perancangan struktur tabel.

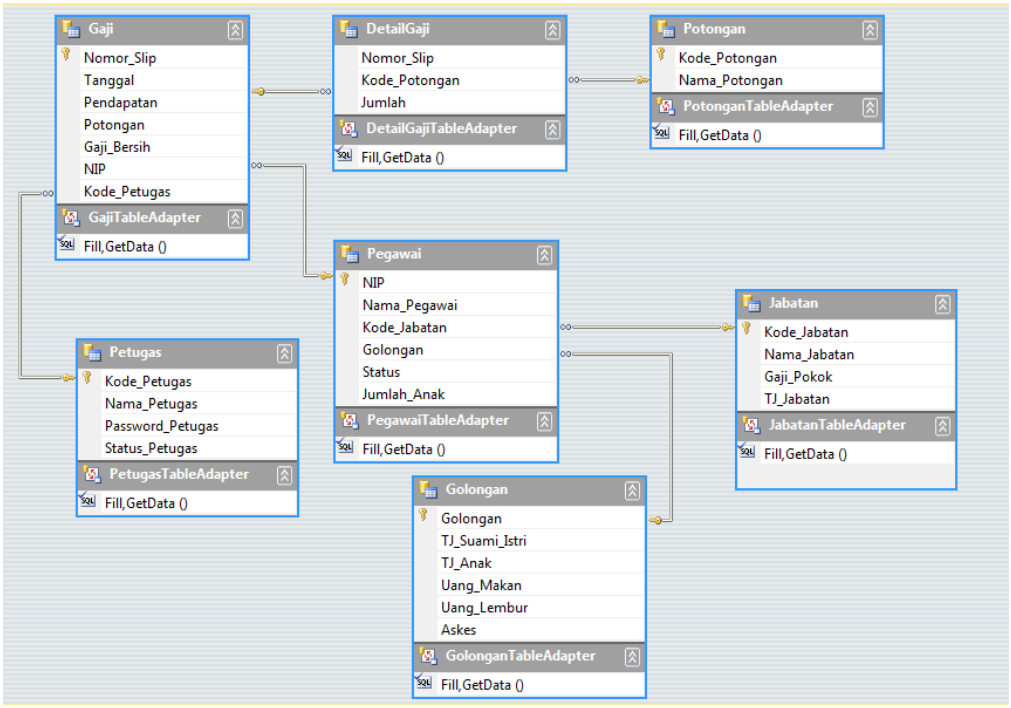
3.2.1.1 Skema Relasi

Berikut ini skema relasi dari basis data pada Aplikasi kepegawaian Balai Besar Pengembangan Budidaya Air Tawar.

Pegawai = {NIP, Nama_Pegawai, Kode_Jabatan, Golongan, Status, Jumlah_Anak}
Gaji = {Nomor_Slip, Tanggal, Pendapatan, Potongan, Gaji_Bersih, NIP, Kode_Petugas}
DetailGaji = {Nomor_Slip, Kode_Potongan, Jumlah}
Potongan = {Kode_Potongan, Nama_Potongan}
Petugas = {Kode_Petugas, Nama_Petugas, Password_Petugas, Status_Petugas}
Jabatan = {Kode_Jabatan, Nama_Jabatan, Gaji_Pokok, Tj_Jabatan}
Golongan = {Golongan, Tj_Suami_Istri, Tj_Anak, Uang_Makan, Uang_Lembur, Askes}

3.2.1.2 Skema Relasi

Diagram relasi merupakan rangkaian hubungan antara dua tabel atau lebih pada sistem basis data. Pada gambar 3.42 akan diperlihatkan penjelasan rangkaian basis data pada Aplikasi kepegawaian Balai Besar Pengembangan Budidaya Air Tawar.



Gambar 3.43 Skema relasi *database* Aplikasi kepegawaian BBPBAT

3.2.1.3 Struktur Tabel

Dalam perancangannya sistem ini terdiri dari tujuh tabel *Database*, yaitu :

3.2.1.3.1 Struktur Tabel Pegawai

Tabel Pegawai digunakan untuk menampung data pegawai. Struktur tabel pegawai dapat dilihat pada tabel 3.28.

Tabel 3.28 Struktur tabel pegawai

Nama Field	Tipe Data	Keterangan
NIP	Varchar (20)	*Primary Key*
Nama_Pegawai	Varchar (30)	
Kode_Jabatan	Varchar (4)	*Foreign Key*
Golongan	Varchar (2)	*Foreign key*
Status	Varchar (15)	
Jumlah_Anak	Int	

3.2.1.3.2 Struktur Tabel DetailGaji

Tabel DetailGaji digunakan untuk menampung data detail gaji. Struktur tabel DetailGaji dapat dilihat pada tabel 3.29.

Tabel 3.29 Struktur tabel DetailGaji

Nama Field	Tipe Data	Keterangan
Nomor_Slip	Varchar (10)	*Foreign Key*
Kode_Potongan	Varchar (3)	*Foreign key*
Jumlah	Int	

3.2.1.3.3 Struktur Tabel Gaji

Tabel gaji digunakan untuk menampung data gaji. Struktur tabel gaji dapat dilihat pada tabel 3.30.

Tabel 3.30 Struktur tabel gaji

Nama Field	Tipe Data	Keterangan
Nomor_Slip	Varchar (10)	*Primary key*
Tanggal	Date/time	
Pendapatan	Int	
Potongan	Int	
Gaji_Bersih	Int	
NIP	Varchar (20)	*Foreign Key*
Kode_Petugas	Varchar (5)	*Foreign Key*

3.2.1.3.4 Struktur Tabel Golongan

Tabel golongan digunakan untuk menampung data golongan. Struktur tabel golongan dapat dilihat pada tabel 3.31.

Tabel 3.31 Struktur tabel golongan

Nama Field	Tipe Data	Keterangan
Golongan	Varchar (4)	*Primary Key*
Tj_Suami_Istri	Int	
Tj_Anak	Int	
Uang_Makan	Int	
Uang_Lembur	Int	

3.2.1.3.5 Struktur Tabel Jabatan

Tabel jabatan digunakan untuk menampung data jabatan. Struktur tabel jabatan dapat dilihat pada tabel 3.32.

Tabel 3.32 Struktur tabel jabatan

Nama Field	Tipe Data	Keterangan
Kode_Jabatan	Varchar (4)	*Primary key*
Nama_Jabatan	Varchar (20)	
Gaji_Pokok	Int	
Tj_Jabatan	Int	

3.2.1.3.6 Struktur Tabel Petugas

Tabel Petugas digunakan untuk menampung data petugas. Struktur tabel petugas dapat dilihat pada tabel 3.33

Tabel 3.33 Struktur tabel petugas

Nama Field	Tipe Data	Keterangan
Kode_Petugas	Varchar (5)	*Primary Key*
Nama_Petugas	Varchar (30)	
Password_Petugas	Varchar (10)	
Status_Petugas	Varchar (15)	

3.2.1.3.7 Struktur Tabel Potongan

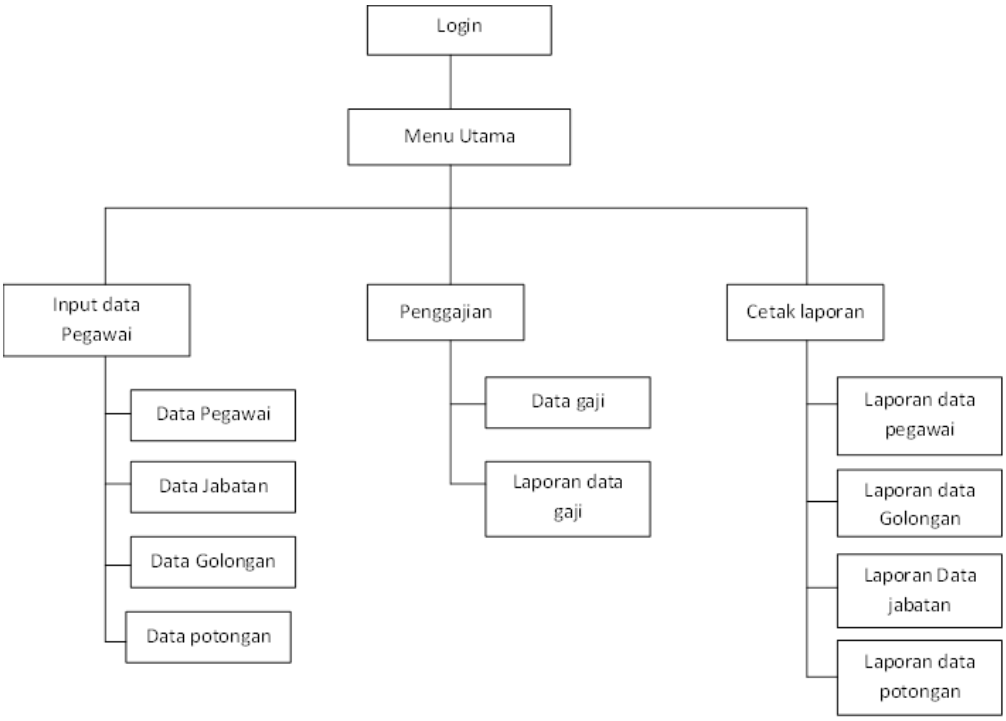
Tabel Potongan digunakan untuk menampung data potongan. Struktur tabel potongan dapat dilihat pada tabel 3.34.

Tabel 3.34 Struktur tabel potongan

Nama Field	Tipe Data	Keterangan
Kode_Potongan	Varchar (3)	*Primary Key*
Nama_Potongan	Varchar (30)	

3.2.2 Perancangan Struktur Menu

Pembangunan Aplikasi kepegawaian di kantor Balai Besar Pengembangan Budidaya Air Tawar sukabumi menggunakan struktur menu di bawah ini :



Gambar 3.44 Struktur menu

Ada tiga hal yang ditampilkan dalam menu utama aplikasi ini, yaitu halaman Input Data Pegawai, Penggajian, dan Cetak Laporan. Halaman Data Input pegawai berisi tentang data pegawai, data golongan, data jabatan dan data potongan. Halaman Penggajian menampilkan data gaji dan laporan data gaji. Halaman Cetak Laporan menampilkan laporan data pegawai, laporan data golongan, laporan dat jabatan dan laporan data potongan.

3.2.3 Perancangan antarmuka

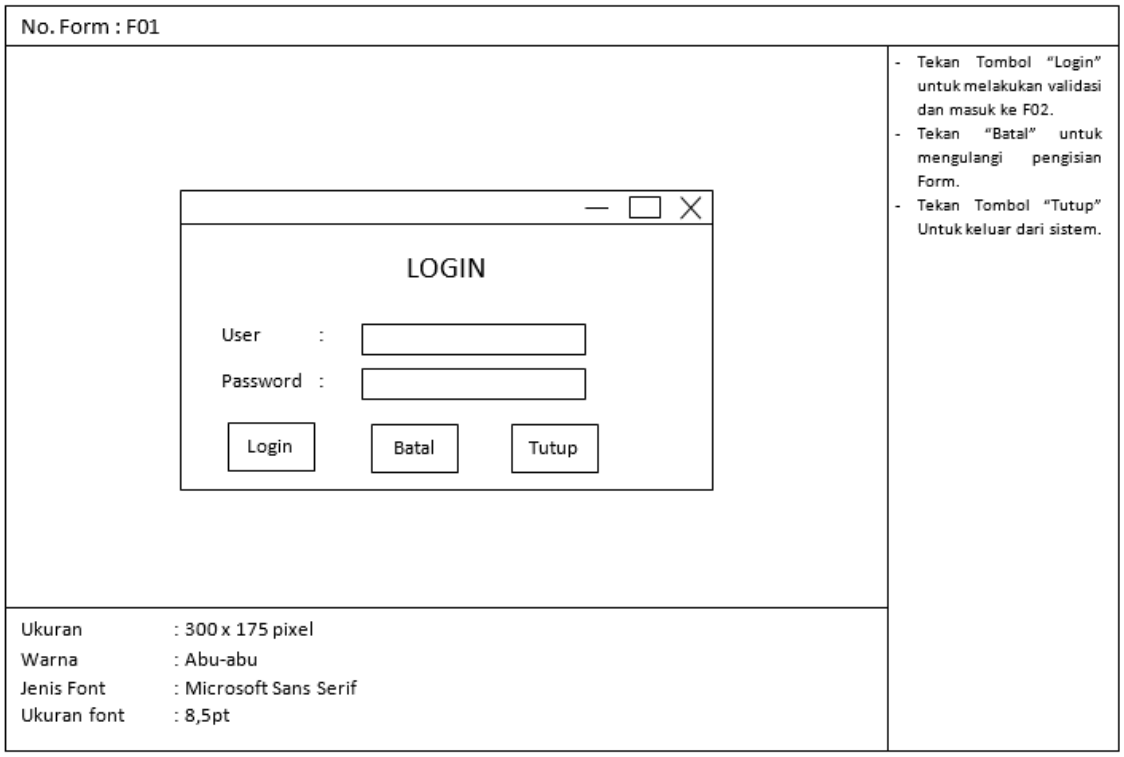
Sistem yang akan dibangun memiliki *user Interface* atau desain antar muka pengguna. Perancangan antarmuka mendeskripsikan rencana tampilan yang akan digunakan pada Aplikasi Kepegawaian di Balai Besar Pengembangan Budidaya Air Tawar Kota Sukabumi. Perancangan antarmuka pada Aplikasi Kepegawaian di Balai Besar Pengembangan Budidaya Air Tawar Kota Sukabumi terdiri dari *form* dan pesan.

3.2.3.1 Perancangan *Form*

Perancangan antarmuka *form* pada Aplikasi dapat dilihat pada gambar 3.45. sampai dengan gambar 3.56.

3.2.3.1.1 Form Login

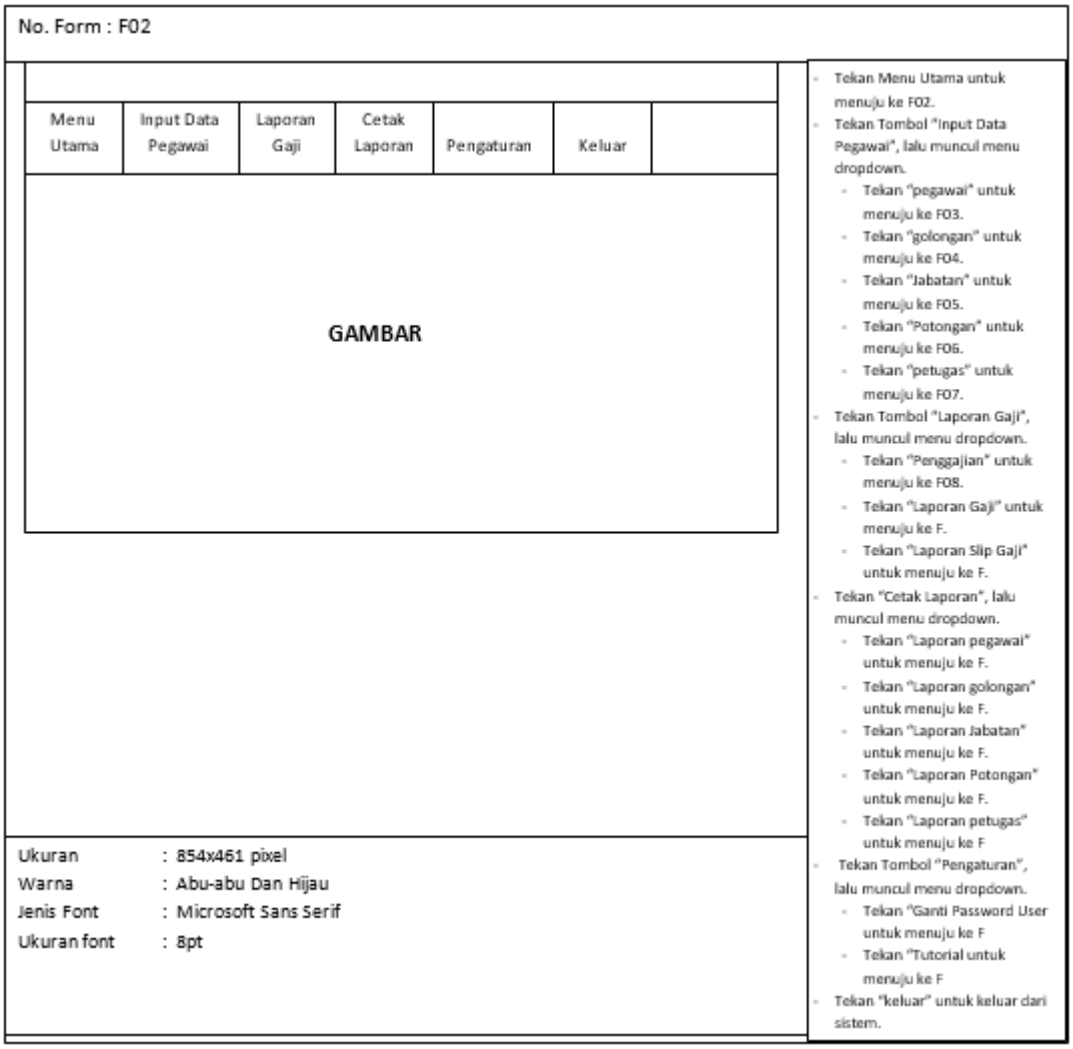
Ketika pertama kali mengakses program, petugas diwajibkan untuk login. Tampilan form login dapat dilihat pada gambar di bawah ini.



Gambar 3.45 Antarmuka *Form Login*

3.2.3.1.2 *Form* Menu Utama

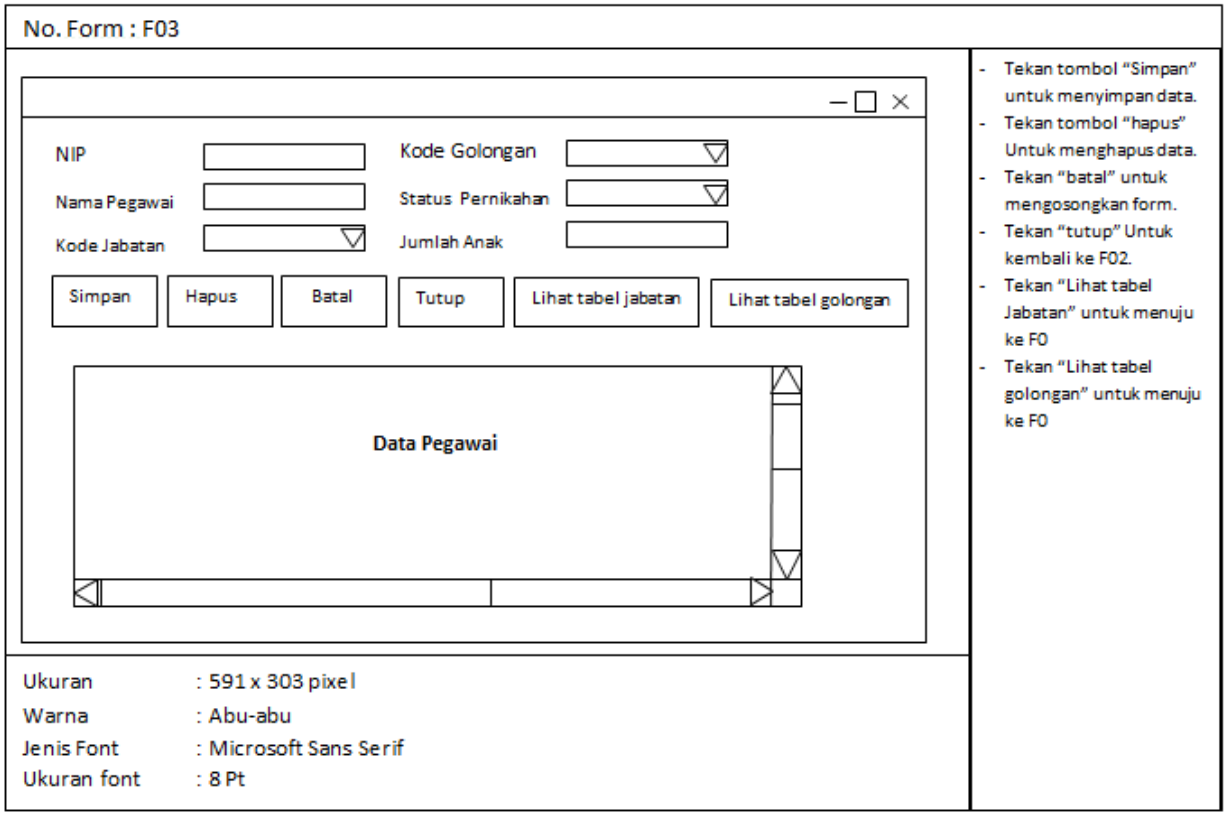
Jika *login* berhasil, selanjutnya akan otomatis masuk ke menu utama. Pada menu utama ini terdapat menu Input data pegawai, laporan gaji, cetak laporan, dan pengaturan. Tampilan *form* menu utama dapat dilihat pada gambar di bawah ini.



Gambar 3.46 Antarmuka *Form* Menu utama

3.2.3.1.3 *Form* Menu Pegawai

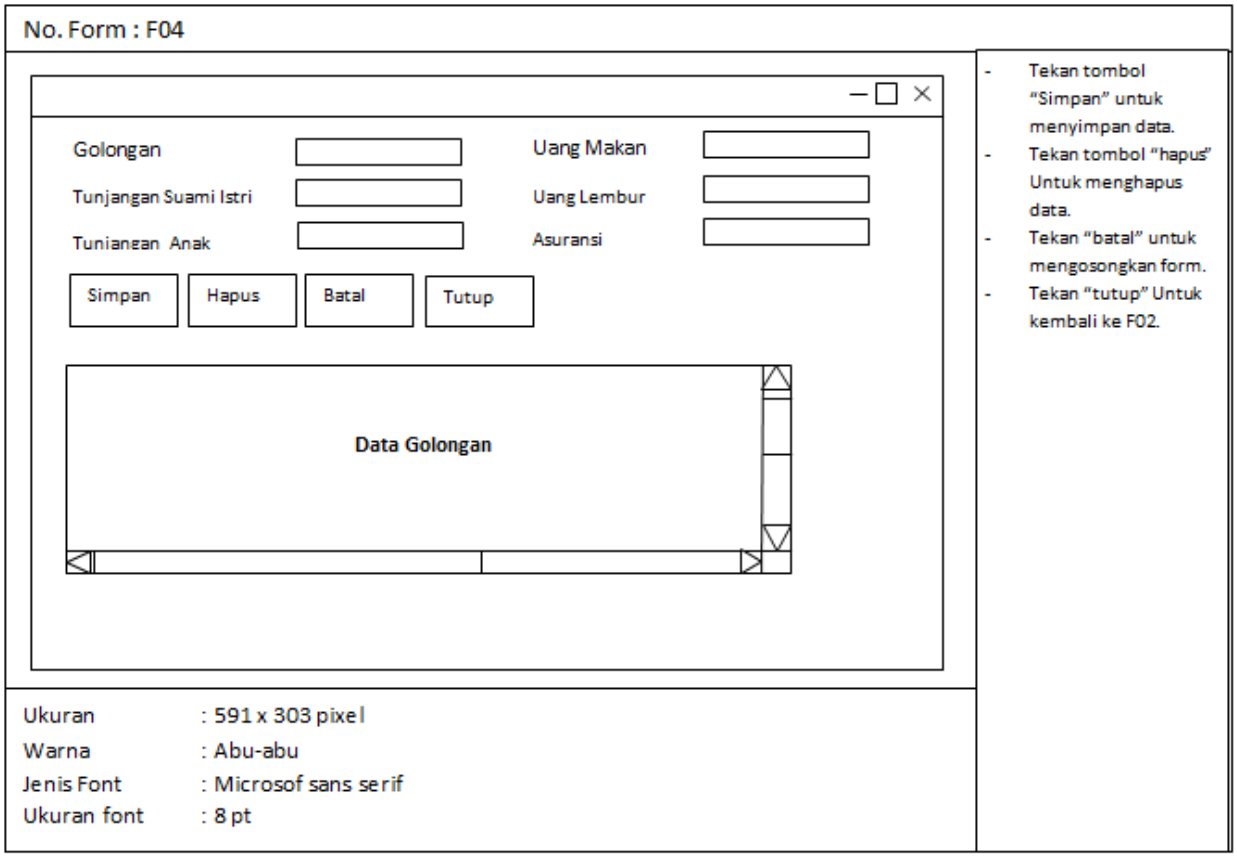
Pada *form* menu pegawai ini terdapat input NIP, Nama Pegawai, Kode Jabatan, Kode Golongan, status pernikahan dan jumlah anak. Tampilan *form* menu pegawai dapat dilihat pada gambar di bawah ini.



Gambar 3.47 Antarmuka *Form* menu Pegawai

3.2.3.1.4 *Form* Menu Golongan

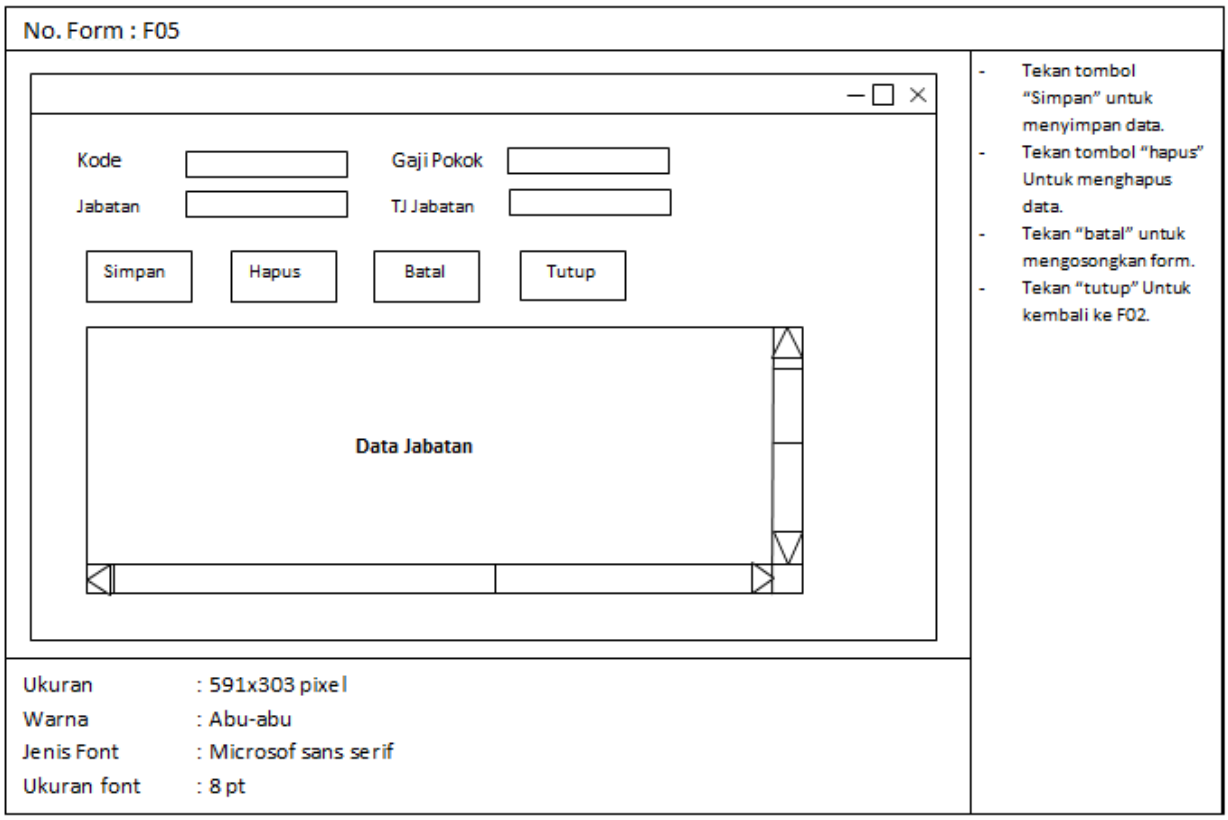
Pada *form* menu golongan ini terdapat menu input Golongan, Tunjangan Istri, Tunjangan Anak,Uang Makan, Uang Lembur, dan Asuransi. Tampilan *form* menu Golongan dapat dilihat pada gambar di bawah ini.



Gambar 3.48 Antarmuka *Form* menu Golongan

3.2.3.1.5 *Form* Menu Jabatan

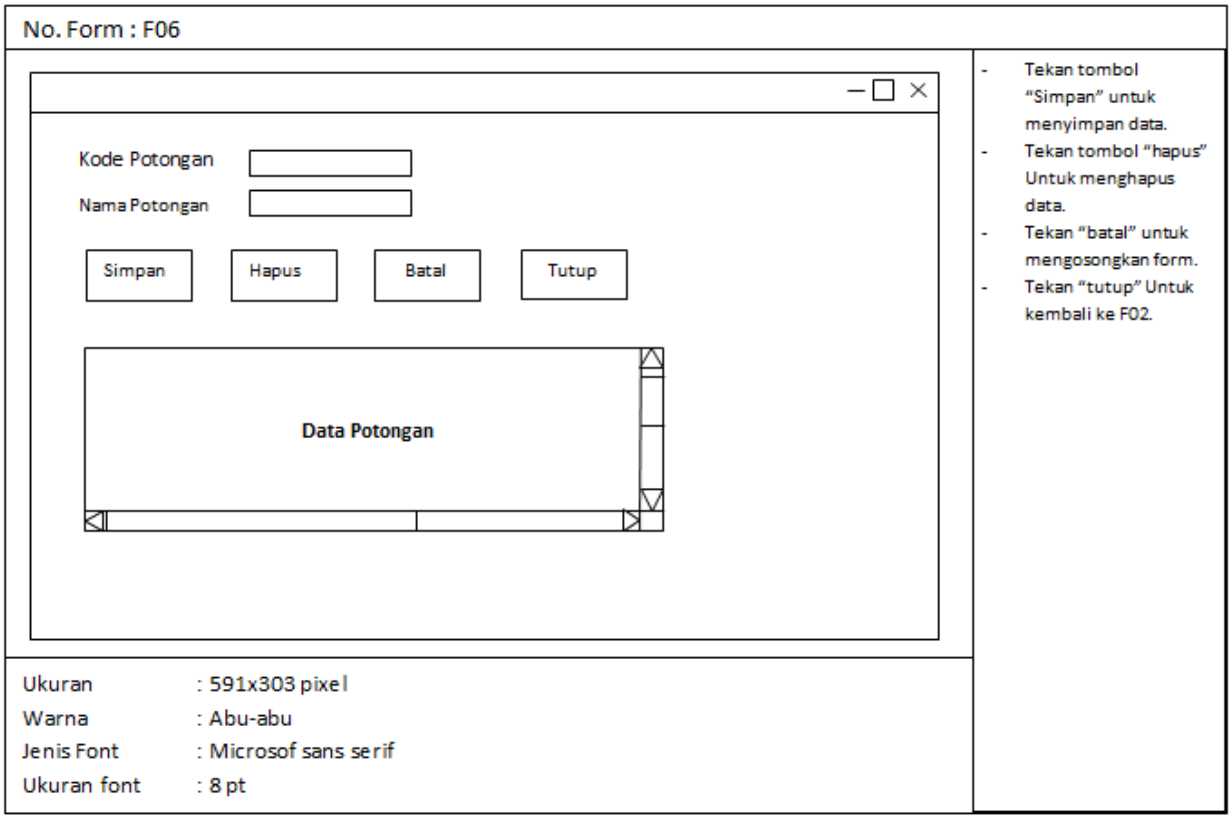
Pada *form* menu Jabatan ini terdapat input Kode, Jabatan, Gaji Pokok dan Tunjangan Jabatan. Tampilan *form* menu Jabatan dapat dilihat pada gambar di bawah ini.



Gambar 3.49 Antarmuka *Form* menu Jabatan

3.2.3.1.6 *Form* Menu Potongan

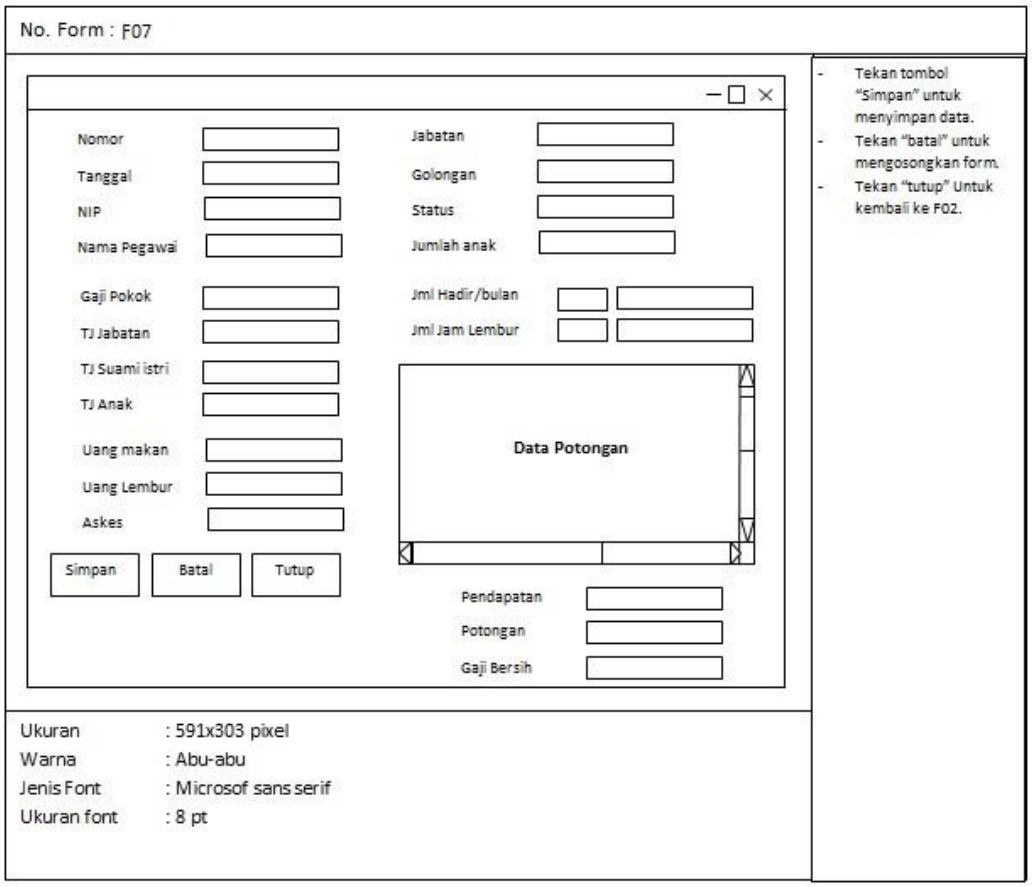
Pada *form* menu potongan ini terdapat input kode potongan dan nama potongan. Tampilan *form* menu Jabatan dapat dilihat pada gambar di bawah ini.



Gambar 3.50 Antarmuka *Form* menu potongan

3.2.3.1.7 *Form* Menu Gaji

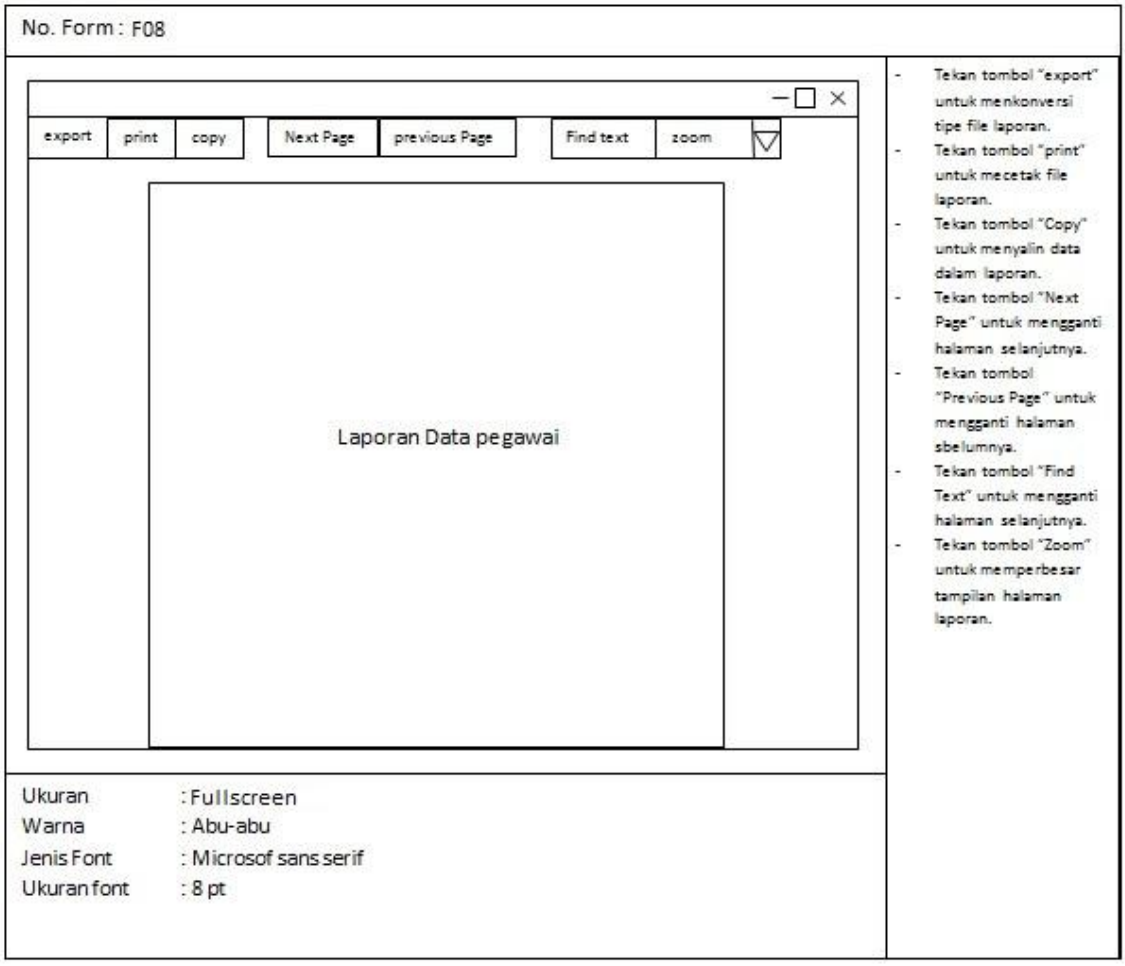
Pada *form* menu Gaji ini terdapat input NIP yang akan langsung secara otomatis keluar data tiap data pegawai. Tampilan *form* menu Gaji dapat dilihat pada gambar di bawah ini.



Gambar 3.51 Antarmuka *Form* Menu Gaji

3.2.3.1.8 *Form* Laporan Data Pegawai

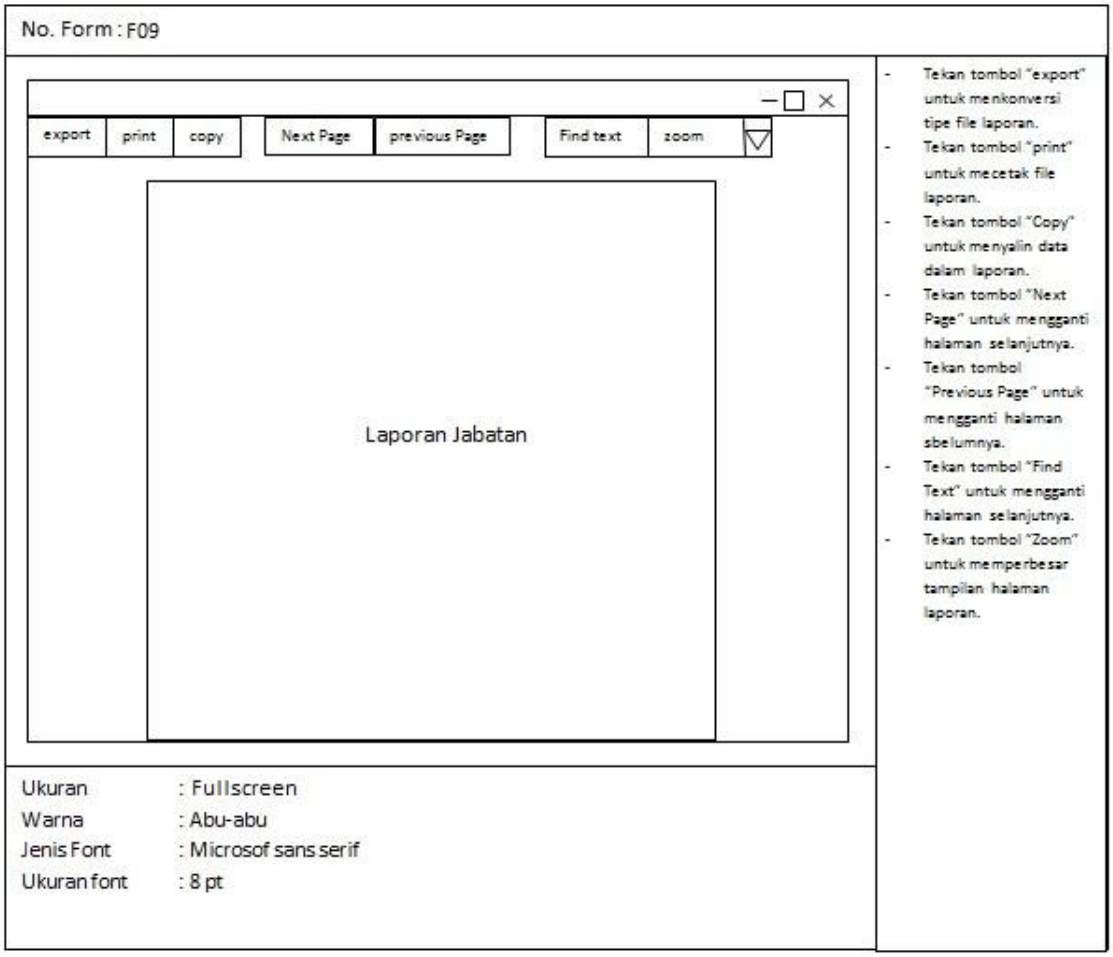
Pada *form* menu Laporan data pegawai ini menampilkan cetak data pegawai sebelum datanya dicetak (*print preview*). Tampilan *form* menu Laporan data pegawai dapat dilihat pada gambar di bawah ini.



Gambar 3.52 Antarmuka *Form* Laporan data pegawai

3.2.3.1.9 *Form* Laporan Data Jabatan

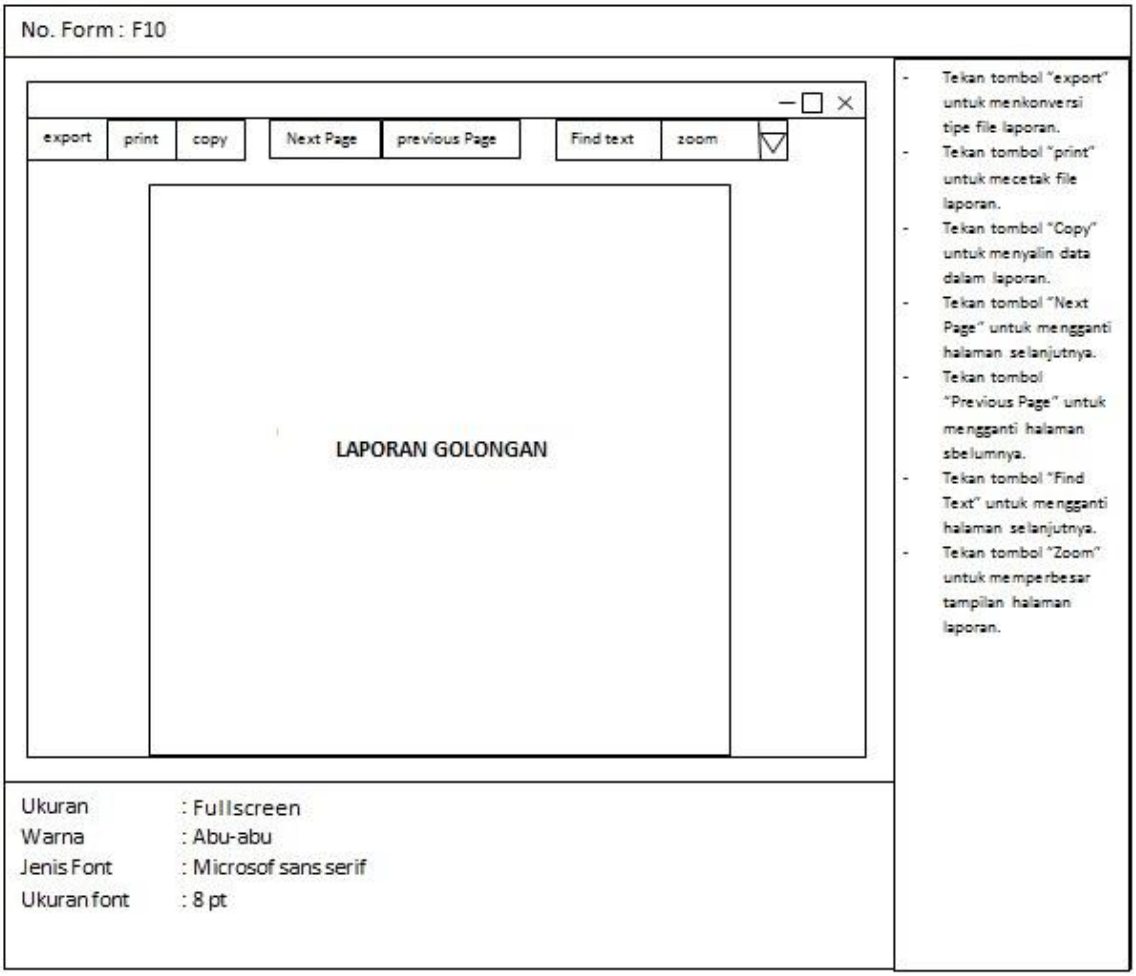
Pada *form* menu Laporan data jabatan ini menampilkan cetak data jabatan sebelum datanya dicetak (*print preview*). Tampilan *form* menu Laporan data jabatan dapat dilihat pada gambar di bawah ini.



Gambar 3.53 Antarmuka *Form* Laporan data jabatan

3.2.3.1.10 *Form* Laporan Data golongan

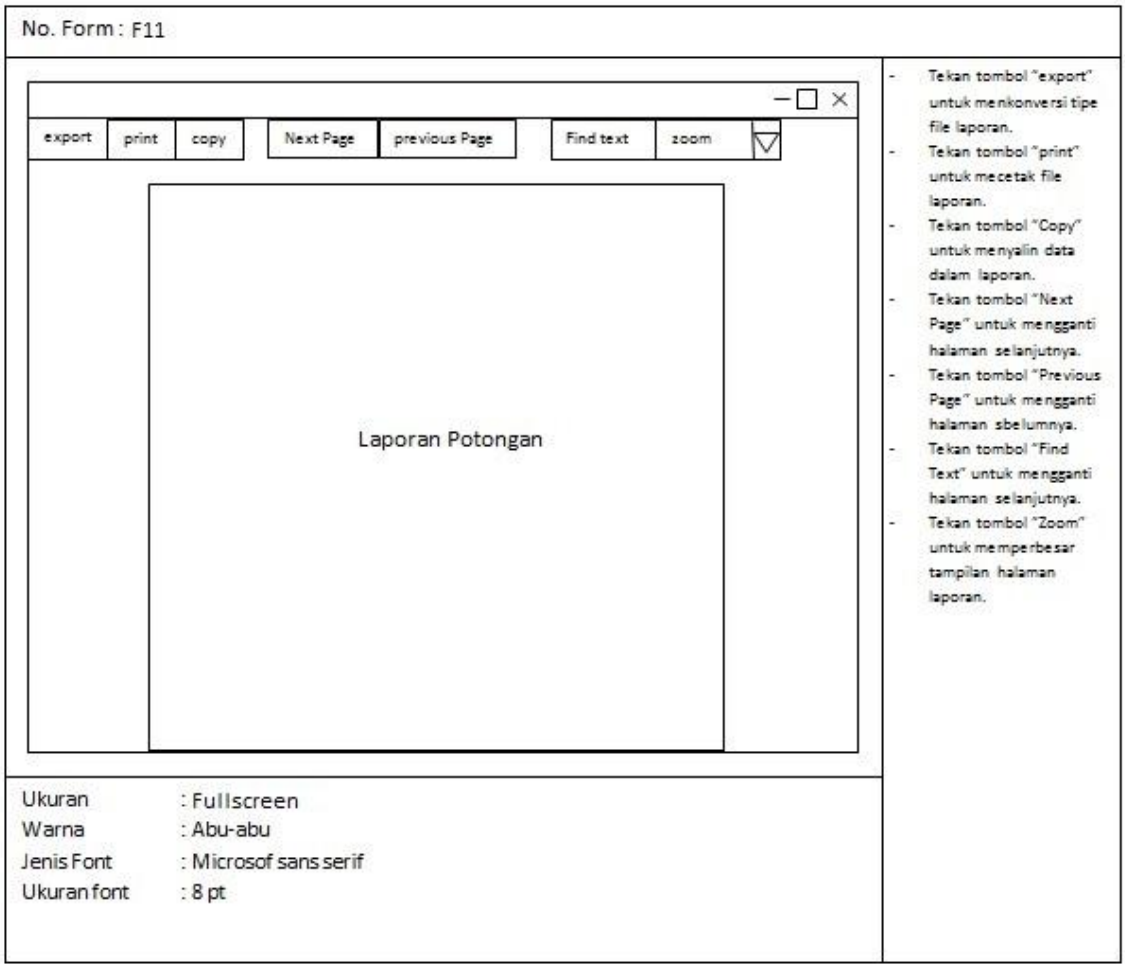
Pada *form* menu Laporan data golongan ini menampilkan cetak data golongan sebelum datanya dicetak (*print preview*). Tampilan *form* menu Laporan data golongan dapat dilihat pada gambar di bawah ini.



Gambar 3.54 Antarmuka *Form* Laporan data golongan

3.2.3.1.11 *Form* Laporan Data Potongan

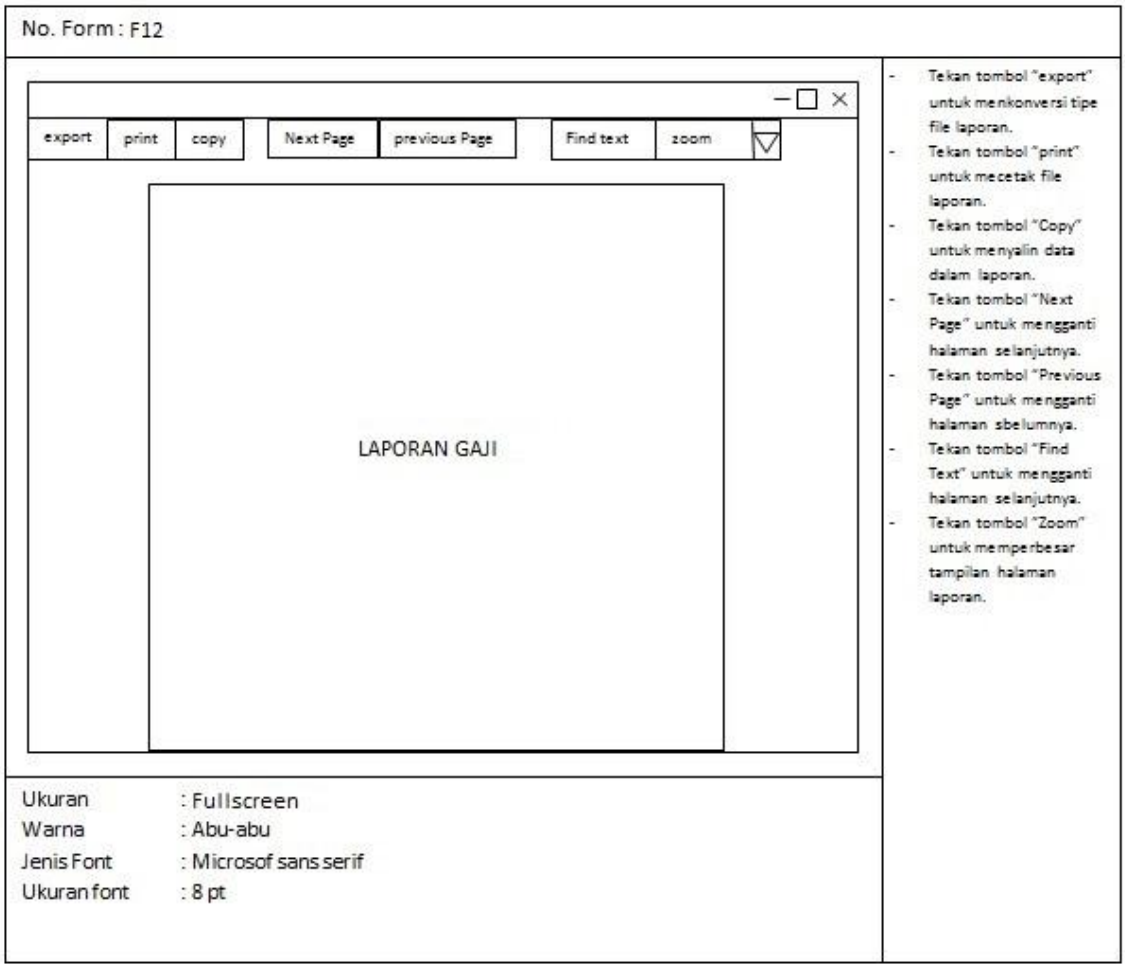
Pada *form* menu Laporan data potongan ini menampilkan cetak data potongan sebelum datanya dicetak (*print preview*). Tampilan *form* menu Laporan data potongan dapat dilihat pada gambar di bawah ini.



Gambar 3.55 Antarmuka *Form* Laporan data potongan

3.2.3.1.12 *Form* Laporan Data Gaji

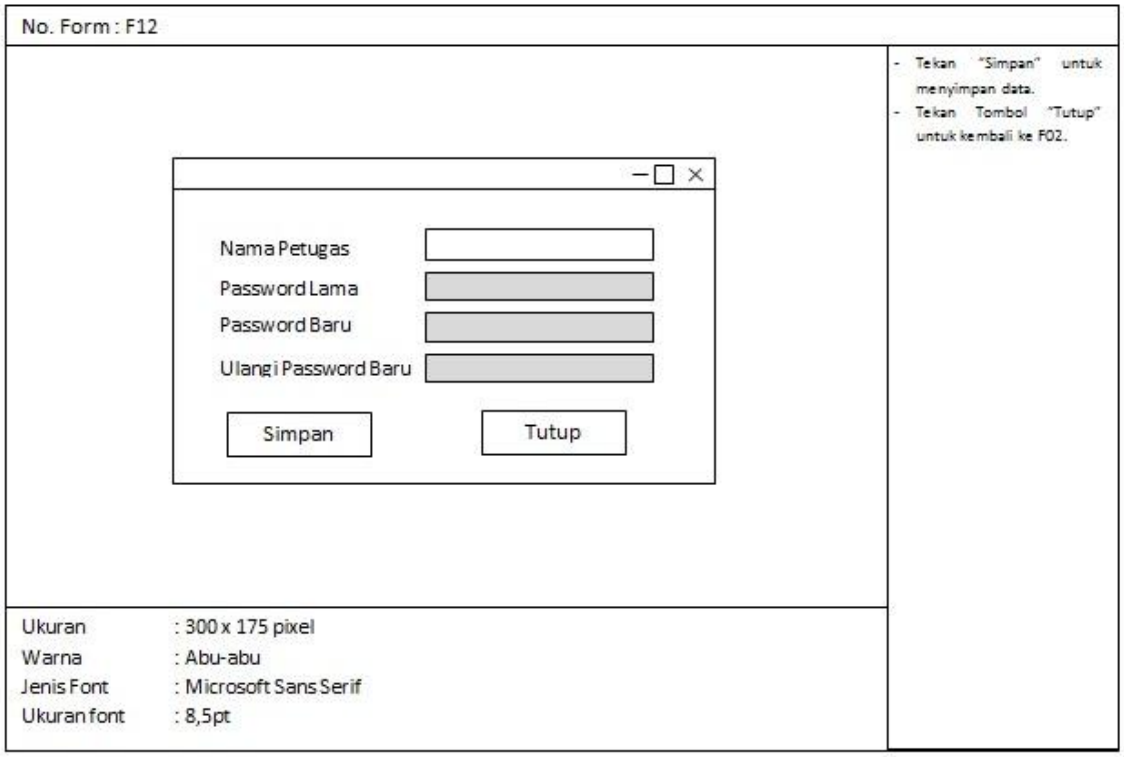
Pada *form* menu Laporan data gaji ini menampilkan cetak data gaji sebelum datanya dicetak (*print preview*). Tampilan *form* menu Laporan data gaji dapat dilihat pada gambar di bawah ini.



Gambar 3.56 Antarmuka *Form* Laporan data Gaji

3.2.3.1.13 *Form* Ubah Password

Pada *form* menu Ubah Password ini menampilkan Input Nama Petugas, Password Lama, Password Baru, Ulangi Password Baru. Tampilan *form* menu Ubah Password dapat dilihat pada gambar dibawah ini.



Gambar 3.57 Antarmuka *Form* ubah password

3.2.3.2 Perancangan Pesan

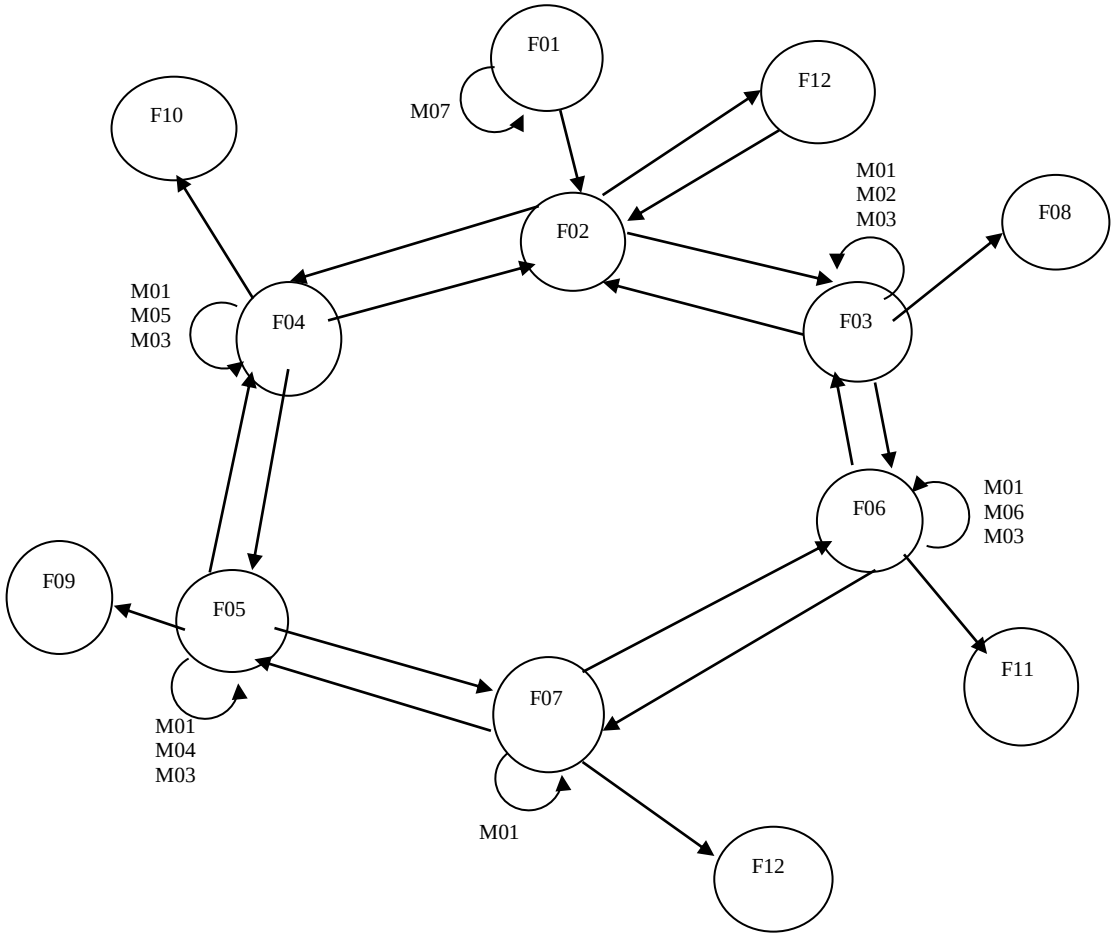
Perancangan pesan pada Aplikasi dapat dilihat pada gambar 3.57.



Gambar 3.58 Perancangan Pesan

3.2.4 Jaringan Semantik

Jaringan semantik pada aplikasi Kepegawaian Balai Besar Pengembangan Budidaya Air Tawar sukabumi sebagai berikut :



Gambar 3.59 Jaringan semantik

3.2.5 Perancangan prosedural

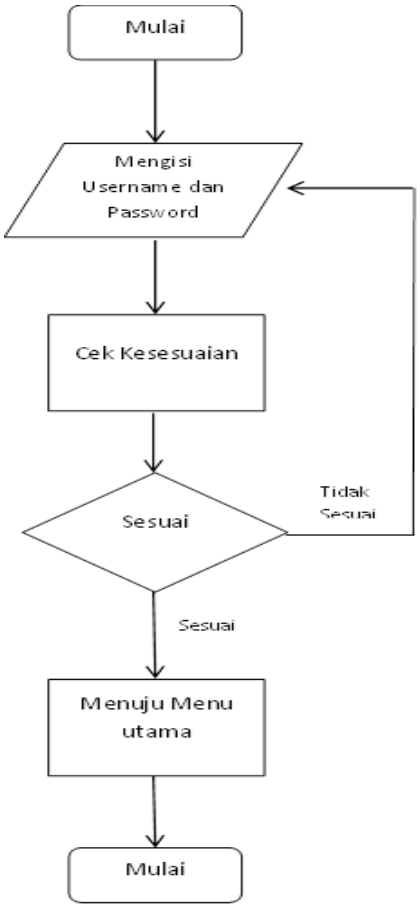
Perancangan prosedural dilakukan dengan membuat *flowchart* sistem untuk menunjukkan alir (*flow*) didalam program atau prosedur sistem secara logika.

3.2.5.1 Flowchart

Flowchart merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Flowchart menjelaskan tentang urutan-urutan dari prosedur yang ada di dalam sistem dengan menggunakan simbol-simbol.

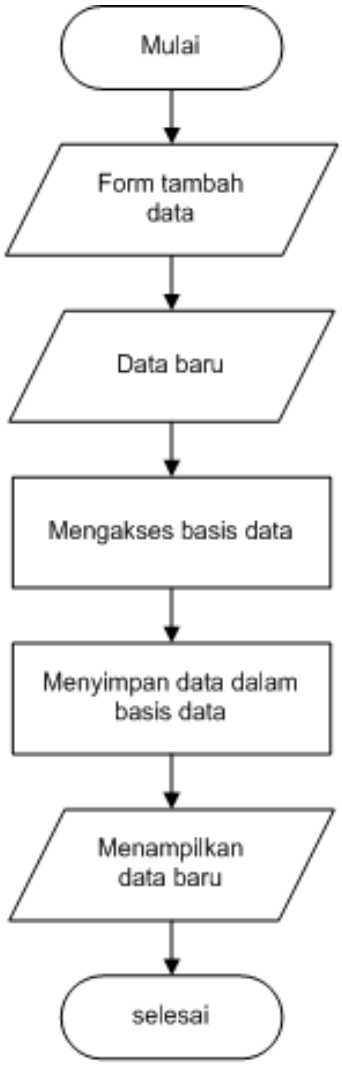
Flowchart pembuatan Aplikasi kepegawaian Balai Besar Pengembangan Budidaya Air Tawar Sukabumi adalah sebagai berikut:

3.2.5.1.1 Proses *Login*



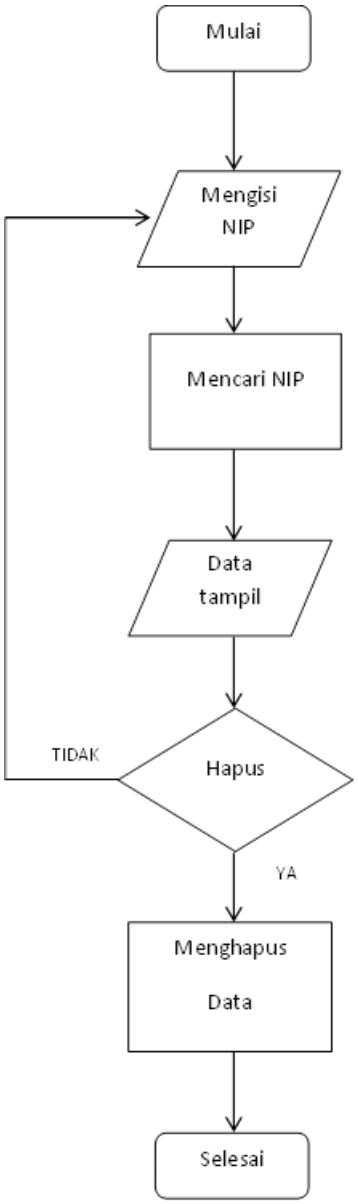
Gambar 3.60 Flowchart Proses *Login*

3.2.5.1.2 Proses Tambah data



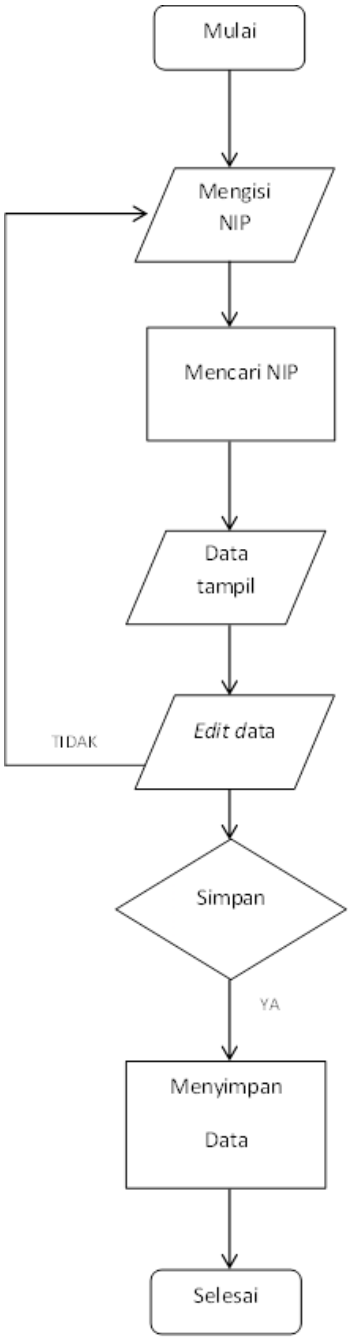
Gambar 3.61 Flowchart Proses tambah Data

3.2.5.1.3 Proses Hapus Data



Gambar 3.62 Flowchart Proses hapus data

3.2.5.1.4 Proses *edit* data



Gambar 3.63 Flowchart Proses *edit* data

3.3 Implementasi

Setelah sistem dianalisis dan didesain secara rinci, maka akan menuju tahap implementasi. Implementasi merupakan tahap meletakkan sistem sehingga siap untuk dioperasikan. Implementasi bertujuan untuk mengkonfirmasi modul-modul perancangan, sehinga pengguna dapat memberikan masukan kepada pembangun sistem.

3.3.1 Implementasi sistem

Implementasi sistem merupakan tahap yang dilakukan setelah melakukan tahap perancangan sistem yaitu menjelaskan mengenai pembuatan sistem serta tampilan dari aplikasi yang telah dibuat, yang sesuai dengan analisis dan perancangan sebelumnya. Setelah tahap implementasi dilakukan maka dibutuhkan sebuah pengujian sistem untuk membuktikan bahwa aplikasi dapat berjalan sesuai dengan yang diharapkan.

3.3.1.1 Perangkat keras yang digunakan

Perangkat keras yang digunakan untuk mengimplementasikan sistem adalah sebagai berikut :

Tabel 3.35 Perangkat keras yang digunakan

No	Perangkat Keras	Keterangan
1	*Processor*	Intel Core i5 2.13 GHz (4 CPUs)
2	memori	2 GB
3	*Harddisk*	500 GB
4	Monitor	LED 15”
5	Keyboard	Genius
6	Mouse	Genius
7	Printer	Canon MP-2000

3.3.1.2 Perangkat lunak yang digunakan

Perngkat lunak yang digunakan untuk mengimplementasikan sistem adalah sebagai berikut :

Tabel 3.36 Perangkat Lunak yang digunakan

No	Perangkat Keras	Keterangan
1	Sistem Operasi	Windows 7 Ultimate 32-bit
2	Bahasa Pemrograman	Microsoft Visual Basic 2010
3	Database	Microsoft Access 2010

3.3.2 Implementasi Basis Data

Implementasi *database* dilakukan dengan menggunakan aplikasi *database* Microsoft Access 2010 sesuai dengan kebutuhan *database* Pada Kantor Balai Besar Pengembangan Budidaya Air Tawar.

Microsoft Access 2010 dipilih sebagai *database*, dikarenakan aplikasi ini berhubungan dengan *database* sistem Aplikasi yang akan di gunakan di bagian karyawan dan penggajian

Berikut ini Query dalam *database* karyawan dan penggajian yang digunakan dalam pembuatan aplikasi kepegawaian Kantor Balai Besar Pengembangan Budidaya Air Tawar.

1. Query Tabel Gaji

```
CREATE TABLE IF NOT EXISTS `gaji` (
  `Nomor_Slip` varchar(10) NOT NULL,
  `Tanggal` datetime DEFAULT NULL,
  `Pendapatan` int(11) DEFAULT '0',
  `Potongan` int(11) DEFAULT '0',
  `Gaji_Bersih` int(11) DEFAULT '0',
  `NIP` varchar(9) DEFAULT NULL,
  `Kode_Petugas` varchar(5) DEFAULT NULL,
  PRIMARY KEY (`Nomor_Slip`),
    FOREIGN KEY (Kode_Petugas) REFERENCES petugas
  (Kode_petugas),
  FOREIGN KEY (NIP) REFERENCES pegawai  (NIP)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

2. Query DetailGaji

```
CREATE TABLE IF NOT EXISTS `detailgaji` (
  `Nomor_Slip` varchar(10) DEFAULT NULL,
  `Kode_Potongan` varchar(3) DEFAULT NULL,
  `Jumlah` int(11) DEFAULT '0',
 PRIMARY KEY `Nomor_Slip` (`Nomor_Slip`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

3. Query Tabel golongan

```
CREATE TABLE IF NOT EXISTS `golongan` (
  `Golongan` varchar(4) NOT NULL,
  `TJ_Suami_Istri` int(11) DEFAULT '0',
  `TJ_Anak` int(11) DEFAULT '0',
  `Uang_Makan` int(11) DEFAULT '0',
  `Uang_Lembur` int(11) DEFAULT '0',
  `Askes` int(11) DEFAULT '0',
  PRIMARY KEY (`Golongan`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

4. Query Tabel jabatan

```
CREATE TABLE IF NOT EXISTS `jabatan` (
  `Kode_Jabatan` varchar(4) NOT NULL,
  `Nama_Jabatan` varchar(20) DEFAULT NULL,
  `Gaji_Pokok` int(11) DEFAULT '0',
  `TJ_Jabatan` int(11) DEFAULT '0',
  PRIMARY KEY (`Kode_Jabatan`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

5. Query tabel Pegawai

```
CREATE TABLE IF NOT EXISTS `pegawai` (
  `NIP` varchar(8) NOT NULL,
  `Nama_Pegawai` varchar(30) DEFAULT NULL,
  `Kode_Jabatan` varchar(4) DEFAULT NULL,
  `Golongan` varchar(2) DEFAULT NULL,
  `Status` varchar(15) DEFAULT NULL,
  `Jumlah_Anak` tinyint(4) DEFAULT '0',
  PRIMARY KEY (`NIP`),
  FOREIGN KEY (Golongan) REFERENCES golongan
  (Golongan),
    FOREIGN KEY (Kode_Jabatan) REFERENCES jabatan
  (Kode_Jabatan)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

6. Query Tabel Petugas

```
CREATE TABLE IF NOT EXISTS `petugas` (
  `Kode_Petugas` varchar(5) NOT NULL,
  `Nama_Petugas` varchar(30) DEFAULT NULL,
  `Password_Petugas` varchar(10) DEFAULT NULL,
  `Status_Petugas` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`Kode_Petugas`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

7. Query Tabel Potongan

```
CREATE TABLE IF NOT EXISTS `potongan` (
  `Kode_Potongan` varchar(3) NOT NULL,
  `Nama_Potongan` varchar(30) DEFAULT
NULL,
  PRIMARY KEY (`Kode_Potongan`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

3.3.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap halaman aplikasi yang dibuat dan pengkodeannya dalam bentuk file program. Berikut ini adalah implementasi antarmuka yang dibuat.

Tabel 3.37 Implementasi antarmuka Aplikasi kepegawaian BBPBAT Sukabumi

No	Menu	Deskripsi	Nama *File*
1.	Login	Digunakan oleh petugas sebagai salah satu syarat untuk masuk ke dalam sistem.	Login.vb
2	Menu utama	Digunakan oleh petugas untuk masuk ke dalam sub menu pilihan	MenuUtama.vb
3	Pegawai	Digunakan oleh petugas untuk menyimpan, menambah, mengubah dan menghapus data pegawai	Pegawai.vb
4	Golongan	Digunakan oleh petugas untuk menyimpan, menambah, mengubah dan menghapus data golongan.	Golongan.vb
5	Jabatan	Digunakan oleh petugas untuk menyimpan, menambah, mengubah dan menghapus data golongan.	Jabatan.vb
6	potongan	Digunakan oleh petugas untuk menyimpan, menambah, mengubah dan menghapus data golongan.	Potongan.vb
7	Gaji	Digunakan oleh petugas untuk menyimpan data gaji.	Penggajian.vb
8	Laporan pegawai	Digunakan oleh petugas untuk mencetak laporan pegawai.	Laporan_pg.vb
9	Laporan golongan	Digunakan oleh petugas untuk mencetak laporan golongan.	Laporan.vb

No	Menu	Deskripsi	Nama *File*
10	Laporan jabatan	Digunakan oleh petugas untuk mencetak laporan jabatan.	Laporan_j.vb
11	Laporan potongan	Digunakan oleh petugas untuk mencetak laporan potongan.	Laporan_pt.vb
12	Laporan Gaji	Digunakan oleh petugas untuk mencetak laporan gaji.	Laporan_GJ.vb
13	Ubah password	Digunakan oleh petugas untuk mengganti password..	Ganti password.vb

3.3.4 Pengujian perangkat lunak

Pengujian yang digunakan untuk menguji sistem yang baru adalah metode pengujian *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.

3.3.4.1 Rencana Pengujian

Rencana pengujian yang akan dilakukan dengan menguji sistem secara *alpha* dan *beta*. Pengujian Aplikasai Kepegaiawaian pada Balai Besar Pengembangan Budidaya Air Tawar Sukabumi ini, menggunakan data uji berdasarkan data yang telah didapat dari Bagian kepegawaian pada Balai Besar Pengembangan Budidaya Air Tawar Sukabumi. Rencana pengujian selengkapnya terlihat pada tabel dibawah ini:

Tabel 3.38 Rencana Pengujian Aplikasi Kepegawaian BBPBAT Sukabumi

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Verifikasi *Username*	*Black box*
	Verifikasi *Password*	*Black box*
Pengolahan Data Pegawai	Tambah Data Pegawai	*Black Box*
	Simpan Data pegawai	*Black Box*

	Edit data Pegawai	*Black Box*
	Hapus data pegawai	*Black Box*
Pengolahan Data Golongan	Tambah Data Golongan	*Black Box*
	Simpan Data Golongan	*Black Box*
	Edit data Golongan	*Black Box*
	Hapus data Golongan	*Black Box*
Pengolahan Data Jabatan	Tambah Data Jabatan	*Black Box*
	Simpan Data Jabatan	*Black Box*
	Edit data Jabatan	*Black Box*
	Hapus data Jabatan	*Black Box*
Pengolahan Data Potongan	Tambah Data Potongan	*Black Box*
	Simpan Data Potongan	*Black Box*
	Edit data Potongan	*Black Box*
	Hapus data Potongan	*Black Box*
Pengolahan Data Potongan	Tambah Data Gaji	*Black Box*
	Simpan Data Gaji	*Black Box*

3.3.4.2 *Equivalance Partitioning Testing*

Berdasarakan rencana pengujian yang telah disusun, maka dapat dilakukan pengujian sebagai berikut :

3.3.4.2.1 Pengujian halaman *Login*

Tabel 3.39 Rencana pengujian halaman *login*

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Username dan *Password* yang benar	Data login dimasukkan dan klik login maka dilakukan proses pengecekan data login. Apabila data login benar maka user tersebut mendapatkan haknya sebagai pengolah data kepegawaian. Klik batal untuk membatalkan	Dapat mengisi data login sebagai user dan tombol login dapat berfungsi. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak
Kasus dan Hasil Uji (Data login tidak seseuai dengan *database*)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Username dan *password* kosong.	Dapat menampilkan pesan kesalahan "Login Salah, silahkan periksa kembali username dan password anda !"	Pesan kesalahan "Login Salah, silahkan periksa kembali username dan password anda !" ditampilkan, user tidak dapat melakukan login. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak

3.3.4.2.2 Pengujian tambah data pegawai

Tabel 3.40 Rencana pengujian tambah data pegawai

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field NIP : 0000001 Nama Pegawai : Bambang Kode Jabatan : 02 Kode Golongan : 3 Status Pernikahan : Menikah jumlah anak : 2	Proses pemasukkan selesai dilakukan. Klik simpan, data yang baru akan masuk ke *database*, maka akan langsung ditampilkan pada datagrid data pegawai.	Data pegawai sukses dimasukkan setelah klik tombol simpan. Sesuai dengan yang diharapkan	☑ Diterima ☐ Ditolak
Kasus dan Hasil Uji (Data yang sudah ada dalam *database*)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field NIP: 0000001.	Dapat menampilkan pesan kesalahan "Data sudah ada!"	Data pegawai tidak masuk ke *database* Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak

3.3.4.2.3 Pengujian Tambah data Golongan

Tabel 3.41 Rencana pengujian tambah data golongan

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field golongan: 1, Tunjangan suami istri: 200000, Tunjangan anak: 400000, uang makan: 150000, Asuransi: 100000.	Proses pemasukkan selesai dilakukan. Klik simpan, data yang baru akan masuk ke database, maka akan langsung ditampilkan pada datagrid data golongan.	Data golongan sukses dimasukkan setelah klik tombol simpan. Sesuai dengan yang diharapkan	☑ Diterima ☐ Ditolak
Kasus dan Hasil Uji (Data data yang sudah ada dalam *database*)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field Golongan: 1.	Dapat menampilkan pesan kesalahan "Data sudah ada!"	Data golongan tidak masuk ke *database* Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak

3.3.4.2.4 Pengujian tambah data Jabatan

Tabel 3.42 Rencana pengujian tambah data jabatan

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field kode: 02, Jabatan: kepala bag. Keuangan, Gaji pokok: 3200000, TJ Jabatan: 200000	Proses pemasukkan selesai dilakukan. Klik simpan, data yang baru akan masuk ke database, maka akan langsung ditampilkan pada datagrid data jabatan.	Data jabatan sukses dimasukkan setelah klik tombol simpan. Sesuai dengan yang diharapkan	☑ Diterima ☐ Ditolak
Kasus dan Hasil Uji (Data data yang sudah ada dalam *database*)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field kode: 02.	Dapat menampilkan pesan kesalahan "Data sydah ada!"	Data jabatan tidak masuk ke *database* Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak

3.3.4.2.5 Pengujian tambah data potongan

Tabel 3.43 Rencana pengujian tambah data potongan

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field kode potongan: 01K Nama potongan: Koperasi	Proses pemasukkan selesai dilakukan. Klik simpan, data yang baru akan masuk ke database, maka akan langsung ditampilkan pada datagrid data potongan.	Data potongan sukses dimasukkan setelah klik tombol simpan. Sesuai dengan yang diharapkan	☑ Diterima ☐ Ditolak
Kasus dan Hasil Uji (Data data yang sudah ada dalam *database*)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field Kode Potongan: 02	Dapat menampilkan pesan kesalahan "Data Sudah ada!"	Data potongan tidak masuk ke *database* Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak

3.3.4.2.6 Pengujian menyimpan data Gaji

Tabel 3.44 Rencana pengujian menyimpan data gaji

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field NIP pegawai pada data Gaji .	Proses pemasukkan akan secara otomatis dilakukan oleh sistem untuk field Lain. Klik simpan, data yang baru akan masuk ke *database*	Data Gaji sukses dimasukkan setelah klik tombol simpan. Sesuai dengan yang diharapkan	☑ Diterima ☐ Ditolak
Kasus dan Hasil Uji (Data tidak diisi)			
Data masukan	Hasil yang diharapkan	Hasil sesuai uji kasus	Kesimpulan
Field NIP Kososng.	Dapat menampilkan pesan kesalahan "Data Belum Lengkap!"	Data penggajian tidak dimasukkan secara otomatis. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak

3.3.4.2.7 Pengujian *Edit* data pegawai

Tabel 3.44 Rencana pengujian *Edit* data pegawai

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field NIP pegawai: 0000001 Atau memilih dat yang ada pada *datagrid.*	Melakukan pencarian data yang akan diubah berdasarkan NIP yang diisi.	Mencari data sesuai NIP yang diisi. Lalu ditampilkan pada *form*. Sesuai dengan yang diharapkan	☑ Diterima ☐ Ditolak

	Dapat melakukan pengubahan data terhadap field yang akan diubah.	Data pegawai yang telah diubah dapat ditampilkan pada datagrid data pegawai. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak
Kasus dan Hasil Uji (Data tidak ada dalam *database*)			
Data masukan	Hasil yang diharapkan	Hasil sesuai uji kasus	Kesimpulan
Mengisi Field NIP: 1234567 .	Dapat menampilkan pesan kesalahan "Data tidak ditemukan!"	Data tidak ditemukan. Dan menampilkan pesan kesalahan. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak

3.3.4.2.8 Pengujian *Edit* data golongan

Tabel 3.44 Rencana pengujian *Edit* data golongan

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field NIP kode golongan: 1 Atau memilih dat yang ada pada *datagrid.*	Melakukan pencarian data yang akan diubah berdasarkan kode golongan yang diisi.	Mencari data sesuai kode golongan yang diisi. Lalu ditampilkan pada *form*. Sesuai dengan yang diharapkan	☑ Diterima ☐ Ditolak

	Dapat melakukan pengubahan data terhadap field yang akan diubah.	Data golongan yang telah diubah dapat ditampilkan pada datagrid data golongan. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak
Kasus dan Hasil Uji (Data tidak ada dalam *database*)			
Data masukan	Hasil yang diharapkan	Hasil sesuai uji kasus	Kesimpulan
Mengisi Field kode golongan: 6	Dapat menampilkan pesan kesalahan "Data tidak ditemukan!"	Data tidak ditemukan. Dan menampilkan pesan kesalahan. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak

3.3.4.2.9 Pengujian *Edit* data jabatan

Tabel 3.44 Rencana pengujian *Edit* data jabatan

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field kode: 02 Atau memilih dat yang ada pada *datagrid.*	Melakukan pencarian data yang akan diubah berdasarkan kode yang diisi.	Mencari data sesuai kode yang diisi. Lalu ditampilkan pada *form*. Sesuai dengan yang diharapkan	☑ Diterima ☐ Ditolak
	Dapat melakukan pengubahan data terhadap field yang akan diubah.	Data jabatan yang telah diubah dapat ditampilkan pada datagrid data jabatan. Sesuai dengan yang	☑ Diterima ☐ Ditolak

		diharapkan.	
Kasus dan Hasil Uji (Data tidak ada dalam *database*)			
Data masukan	Hasil yang diharapkan	Hasil sesuai uji kasus	Kesimpulan
Mengisi Field kode: 0	Dapat menampilkan pesan kesalahan "Data tidak ditemukan!"	Data tidak ditemukan. Dan menampilkan pesan kesalahan. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak

3.3.4.2.10 Pengujian *Edit* data potongan

Tabel 3.44 Rencana pengujian *Edit* data potongan

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field kode potongan: 01K Atau memilih dat yang ada pada *datagrid.*	Melakukan pencarian data yang akan diubah berdasarkan kode potongan yang diisi.	Mencari data sesuai kode potongan yang diisi. Lalu ditampilkan pada *form*. Sesuai dengan yang diharapkan	☑ Diterima ☐ Ditolak
	Dapat melakukan pengubahan data terhadap field yang akan diubah.	Data potongan yang telah diubah dapat ditampilkan pada datagrid data potongan. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak
Kasus dan Hasil Uji (Data tidak ada dalam *database*)			
Data masukan	Hasil yang diharapkan	Hasil sesuai uji kasus	Kesimpulan

Mengisi Field NIP: 10K.	Dapat menampilkan pesan kesalahan “Data tidak ditemukan!”	Data tidak ditemukan. Dan menampilkan pesan kesalahan. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak

3.3.4.2.11 Pengujian Hapus data pegawai

Tabel 3.44 Rencana pengujian Hapus data pegawai

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field NIP pegawai: 0000001 Atau memilih dat yang ada pada *datagrid.*	Melakukan pencarian data yang akan diubah berdasarkan NIP yang diisi.	Mencari data sesuai NIP yang diisi. Lalu ditampilkan pada *form*. Sesuai dengan yang diharapkan	☑ Diterima ☐ Ditolak
	Dapat melakukan Hapus data terhadap field yang akan diubah.	Data pegawai yang telah dihapus sudah tidak terdapat pada datagrid data pegawai. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak
Kasus dan Hasil Uji (Data tidak ada dalam *database*)			
Data masukan	Hasil yang diharapkan	Hasil sesuai uji kasus	Kesimpulan
Mengisi Field NIP: 1234567 .	Dapat menampilkan pesan kesalahan “Data tidak ditemukan!”	Data tidak ditemukan. Dan menampilkan pesan kesalahan. Sesuai dengan yang	☑ Diterima ☐ Ditolak

		diharapkan.	

3.3.4.2.12 Pengujian Hapus data golongan

Tabel 3.44 Rencana pengujian Hapus data golongan

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field NIP kode golongan: 1 Atau memilih dat yang ada pada *datagrid.*	Melakukan pencarian data yang akan diubah berdasarkan kode golongan yang diisi.	Mencari data sesuai kode golongan yang diisi. Lalu ditampilkan pada *form*. Sesuai dengan yang diharapkan	☑ Diterima ☐ Ditolak
	Dapat melakukan Hapus data terhadap field yang akan diubah.	Data golongan yang telah dihapus sudah tidak terdapat pada datagrid data pegawai. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak
Kasus dan Hasil Uji (Data tidak ada dalam *database*)			
Data masukan	Hasil yang diharapkan	Hasil sesuai uji kasus	Kesimpulan
Mengisi Field kode golongan: 6	Dapat menampilkan pesan kesalahan “Data tidak ditemukan!”	Data tidak ditemukan. Dan menampilkan pesan kesalahan. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak

3.3.4.2.13 Pengujian *Hapus* data jabatan

Tabel 3.44 Rencana pengujian *Hapus* data jabatan

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field kode: 02 Atau memilih dat yang ada pada *datagrid.*	Melakukan pencarian data yang akan diubah berdasarkan kode yang diisi.	Mencari data sesuai kode yang diisi. Lalu ditampilkan pada *form*. Sesuai dengan yang diharapkan	☑ Diterima ☐ Ditolak
	Dapat melakukan Hapus data terhadap field yang akan diubah.	Data jabatan yang telah dihapus sudah tidak terdapat pada datagrid data jabatan. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak
Kasus dan Hasil Uji (Data tidak ada dalam *database*)			
Data masukan	Hasil yang diharapkan	Hasil sesuai uji kasus	Kesimpulan
Mengisi Field kode: 0	Dapat menampilkan pesan kesalahan “Data tidak ditemukan!”	Data tidak ditemukan. Dan menampilkan pesan kesalahan. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak

3.3.4.2.14 Pengujian Hapus data jabatan

Tabel 3.44 Rencana pengujian Hapus data jabatan

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Hasil yang diharapkan**	**Hasil sesuai uji kasus**	**Kesimpulan**
Mengisi field kode potongan: 01K Atau memilih dat yang ada pada *datagrid.*	Melakukan pencarian data yang akan diubah berdasarkan kode potongan yang diisi.	Mencari data sesuai kode potongan yang diisi. Lalu ditampilkan pada *form*. Sesuai dengan yang diharapkan	☑ Diterima ☐ Ditolak
	Dapat melakukan Hapus data terhadap field yang akan diubah.	Data potongan yang telah dihapus sudah tidak terdapat pada datagrid data potongan. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak
Kasus dan Hasil Uji (Data tidak ada dalam *database*)			
Data masukan	Hasil yang diharapkan	Hasil sesuai uji kasus	Kesimpulan
Mengisi Field NIP: 10K.	Dapat menampilkan pesan kesalahan "Data tidak ditemukan!"	Data tidak ditemukan. Dan menampilkan pesan kesalahan. Sesuai dengan yang diharapkan.	☑ Diterima ☐ Ditolak

Berdasarkan hasil pengujian dengan kasus *sample* uji yang telah dilakukan memberikan kesimpulan bahwa secara fungsional sistem sudah dapat menghasilkan *output* yang diharapkan.

3.3.4.3 Pengujian Beta

Pengujian beta merupakan pengujian langsung kepada pengguna untuk mencoba aplikasi yang baru. Pengujian yang dilakukan adalah dengan cara wawancara kepada pengguna aplikasi, dalam hal ini adalah seorang petugas di Bagian kepegawaian BBPBAT.

3.3.4.3.1 Hasil Wawancara

1. Pertanyaan pertama diberikan pada responden mengenai hasil pengelolaan data pegawai yang diperoleh oleh aplikasi kepegawaian BBPBAT.

   Pertanyaan dijawab oleh responden bahwa hasil pengelolaan data pegawai yang diperoleh oleh aplikasi kepegawaian BBPBAT tersebut cukup efisien dan memuaskan.
2. Pertanyaan kedua diberikan pada responden mengenai hasil pengelolaan gaji yang diperoleh oleh aplikasi.

   Pertanyaan dijawab oleh responden bahwa hasil pengelolaan gaji sangat cepat dan tepat, tidak merepotkan serta kesalahan perhitungan yang dibuatnya hampir tidak ada.
3. Pertanyaan ketiga diberikan pada responden mengenai hasil pembuatan laporan yang dihasilkan oleh aplikasi kepegawaian BBPBAT..

   Pertanyaan dijawab oleh responden bahwa hasil penmbuatan laporan sangat baik, rapi dan tidak memakan waktu yang lama.

3.3.4.4 Kesimpulan Hasil Pengujian Beta

Dari pengujian beta yang dilakukan yaitu wawancara, dapat diambil kesimpulan bahwa Aplikasi kepegawaian BBPBAT sukabumi ini dinilai memiliki hasil pengelolaan data pegawai yang efisien, pengelolan gaji sangat cepat dan tepat serta kesalahan perhitungannya sangat minim, dan dalam pembuatan laporannya Aplikasi kepegawaian BBPBAT sukabumi ini sangat baik, rapi dan tidak memakan waktu yang lama.