BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Melakukan analisis terhadap sistem yang sedang berjalan bertujuan sebagai dasar perancangan atau perbaikan sistem lama. Dari hasil analisis tersebut dapat diketahui kelemahan atau kekurangan sistem yang lama dan dapat dirancang atau diperbaiki menjadi sebuah sistem yang lebih efektif dan efisien.

3.1.1 Analisis Sistem Berjalan

Analisis sistem berjalan adalah analisis yang dilakukan pada sistem yang sedang berjalan di PT. PLN (Persero) pada bagian kepegawaian. Berdasarkan hasil observasi yang dilakukan dan wawancara dengan pihak bagian kepegawaian didapatkan prosedur yang sedang berjalan di bagian kepegawaian adalah sebagai berikut :

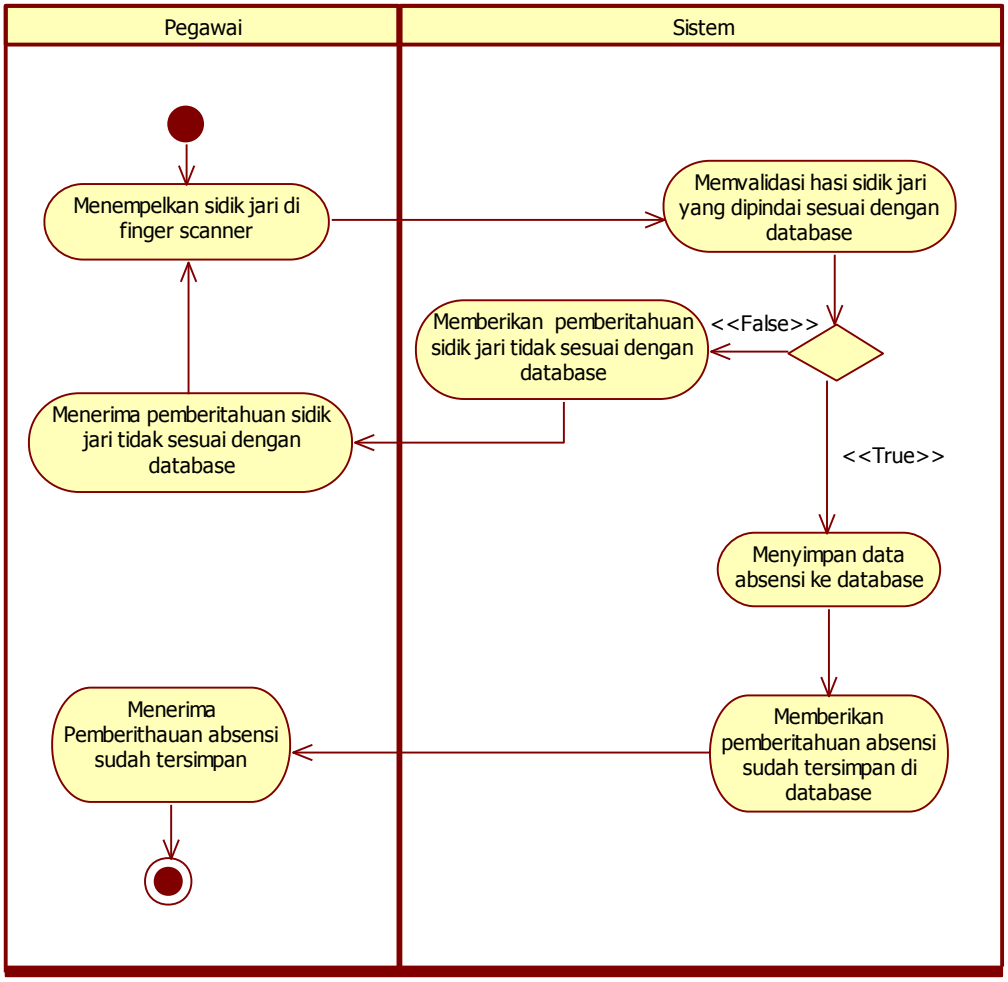
1. Proses absensi
2. Proses evaluasi absensi
3. Proses rekrutmen pegawai baru
4. Proses mutasi pegawai

Proses-proses yang berjalan tersebut dilakukan antara pegawai yang melakukan absensi, sistem yang menjadi jembatan antara pegawai dan admin, dan terakhir petugas yang melihat dan mengolah informasi yang dihasilkan oleh sistem. Penjelasan dari setiap prosedur adalah sebagai berikut :

3.1.1.1 Proses Absensi

Proses absensi adalah prosedur dimana pegawai PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung melakukan rutinitas absensi dengan menggunakan alat berupa pemindai sidik jari. Proses absensi di PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung sebenarnya sudah berjala secara terkomputerisasi, tetapi belum memiliki fitur yang dapat mengevaluasi absensi dari pegawai. Proses absensi digambarkan dengan *Activity Diagram* pada gambar 3.1. Proses-proses yang dilakukan pada prosedur absensi adalah sebagai berikut :

1. Pegawai menempelkan sidik jarinya dari ibu jarinya ke alat pemindai sidik jari.
2. Sistem akan memvalidasi hasil pemindaian sidik jari.
3. Apabila hasil pemindaian sidik jari tidak sesuai dengan *database* maka sistem akan memberikan pemberitahuan bahwa sidik jari tidak sesuai dengan *database*. Pegawai akan mengulangi proses pemindaian.
4. apabila hasil pemindaian sesuai dengan *database* maka data absensi akan disimpan ke *database.*
5. Setelah data absensi disimpan di *database*, sistem akan memberikan pemeberitahuan bahwa data absensi telah tersimpan di dalam *database.*

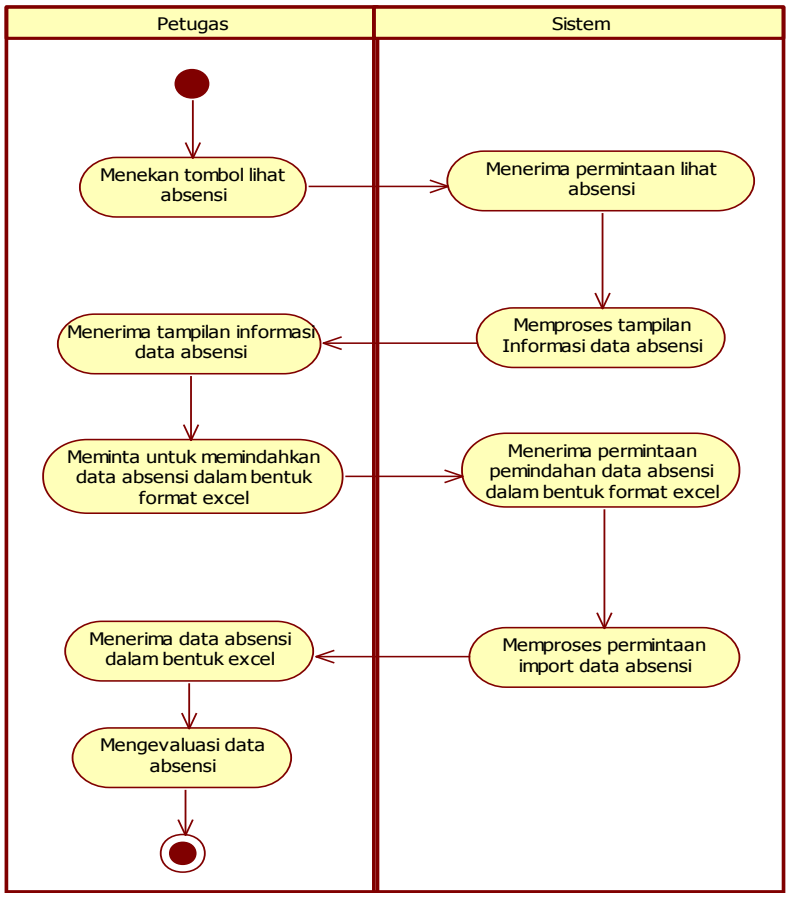


Gambar 3.1 *Activity Diagram* Proses Absensi

3.1.1.2 Proses Evaluasi Absensi

Proses evaluasi absensi dilakukan oleh petugas dari bagian kepegawaian untuk melihat informasi absensi yang dihasilkan oleh sistem dari proses absensi yang dilakukan pegawai sebelumnya dan juga untuk mengevaluasinya. Dengan kata lain petugas bagian kepegawaian memproses evaluasi sudah di luar sistem yang ada tetapi dengan menggunakan *Microsoft excel*. Proses absensi digambarkan dengan *Activity Diagram* pada gambar 3.2. Proses-proses yang dilakukan pada prosedur lihat absensi adalah sebagai berikut :

1. Petugas melakukan permintaan lihat absensi pada sistem.
2. Sistem memproses permintaan petugas kemudian menampilkan informasi data absensi dari *database.*
3. Petugas kemudian akan memindahkan informasi tersebut ke dalam format *Microsoft excel* untuk mengevaluasi absensi dari pegawai yang telah melakukan absen.
4. Sistem memproses permintaan untuk pemindahan ke format *Microsoft excel*.
5. Petugas menerima informasi data absensi dalam bentuk format **.xls.*
6. Petugas mengevaluasi informasi data absensi tersebut.

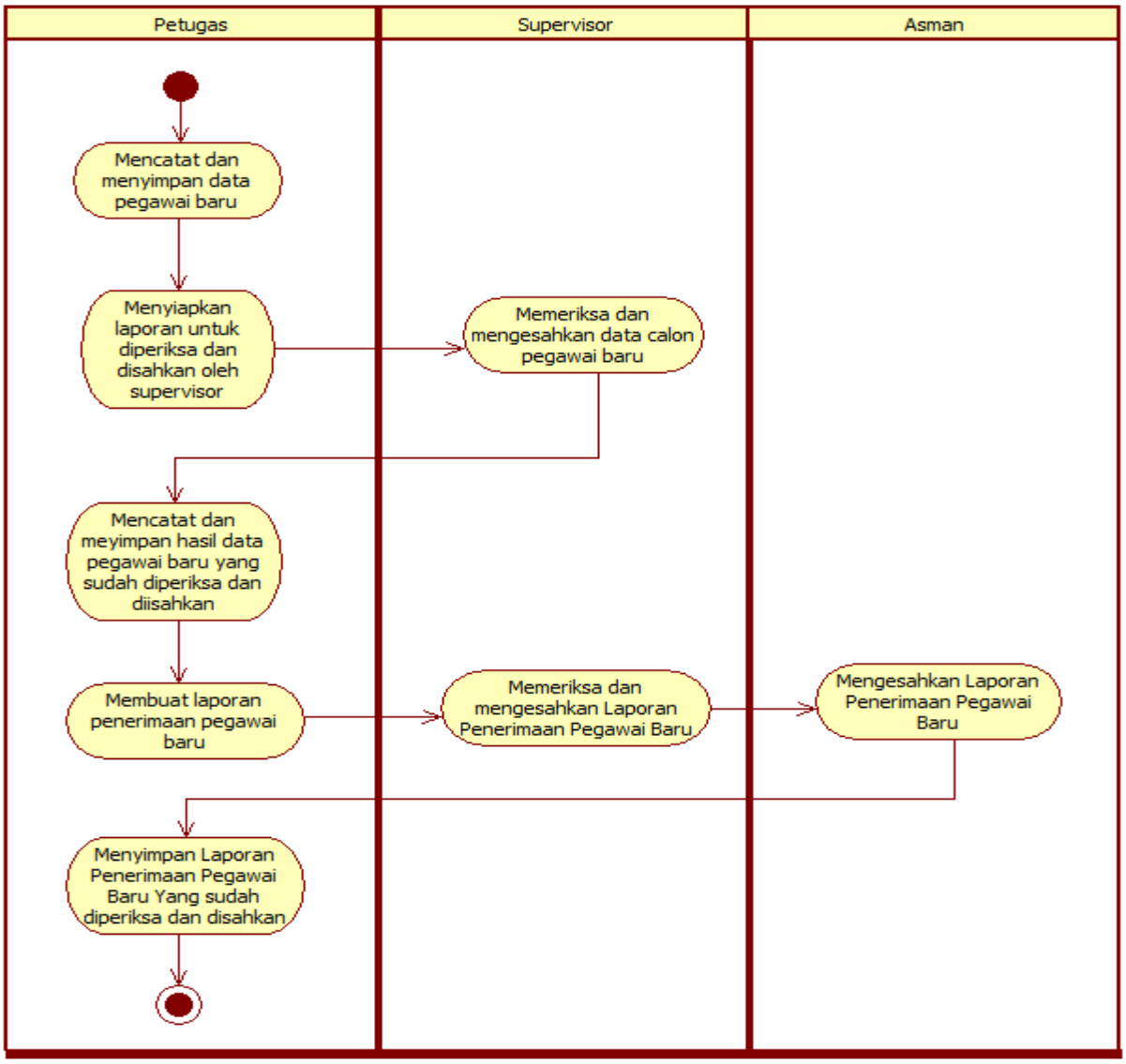


Gambar 3.2 *Activity Diagram* Proses Lihat Absensi

3.1.1.3 Proses Rekrutmen Pegawai Baru

PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung untuk meningkatkan kinerja dan meningkatkan daya saing tentunya selalu merekrut pegawai baru. Bagian kepegawaian menangani administrasi pencatatan dalam proses penerimaan pegawai baru, dan bertanggung jawab untuk melaporkan kepada atasan yaitu asman dan manajer bagian kepegawaian terhadap setiap penerimaan yang dilakukan. Sedangkan untuk proses penyeleksian calon pegawai baru dilakukan oleh tim penyeleksi yang berada di luar bagian kepegawaian, kemudian tim penyeleksi akan memberikan laporan kepada bagian kepegawaian untuk dilakukan pencatatan dalam hal administrasi. Pada gambar 3.3 akan dijelaskan mengenai *Activity Diagram* proses rekrutmen pegawai baru. Proses-proses yang dilakukan pada prosedur rekrutmen pegawai baru adalah sebagai berikut :

1. Petugas bagian kepegawaian akan mencatat dan menyimpan data-data yang terkait dengan penerimaan pegawai baru.
2. Petugas membuat laporan hasil seleksi dan penerimaan pegawai baru.
3. Supervisor SDM dan asman SDM akan memeriksa dan mengesahkan hasil penerimaan pegawai baru.
4. Petugas akan menyimpan data dari laporan pegawai baru.



Gambar 3.3 *Activity Diagram* Proses Rekrutmen Pegawai Baru

3.1.1.4 Proses Mutasi Pegawai

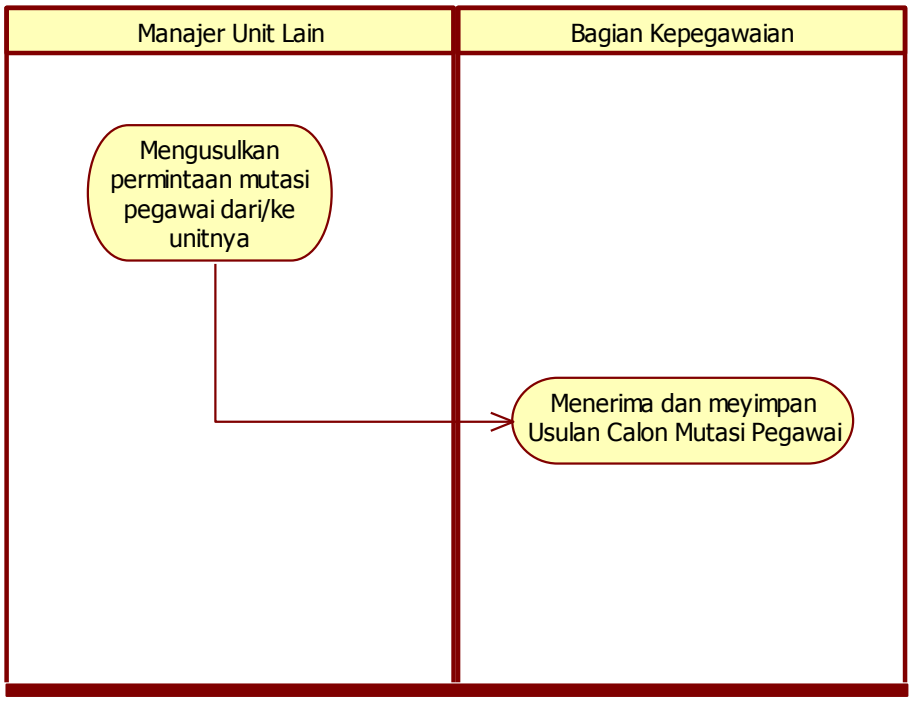
Proses mutasi pegawai adalah proses dimana seorang pegawai dipindah bagian yang masih dalam lingkup unit intern yaitu berada dalam lingkungan PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung ataupun berada di luar unit (ekstern) APJ tersebut. Bagian kepegawaian bertanggung jawab untuk mencatat semua permintaan mutasi pegawai dan melaporkan kepada atasan dari

unit yang terkait ketika proses mutasi telah berhasil dilaksanakan. Permintaan mutasi dilakukan oleh manajer dari unit yang bersangkutan untuk permintaan mutasi, dapat dilihat pada *Activity Diagram* gambar 3.4. Pada gambar 3.5 akan dijelaskan mengenai *Activity Diagram* proses mutasi pegawai. Proses-proses yang dilakukan pada prosedur mutasi pegawai adalah sebagai berikut :

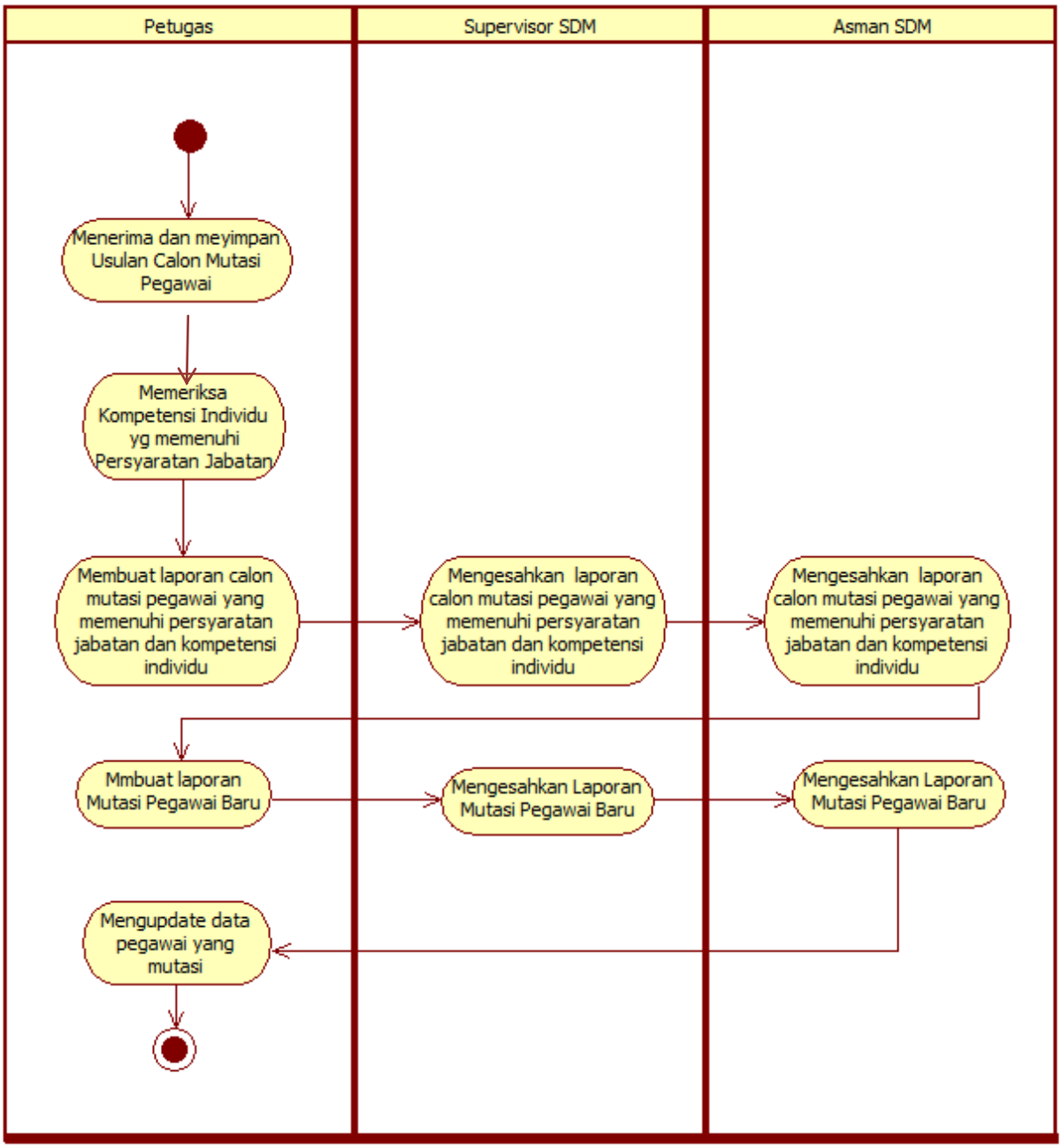
1. Manajer unit yang bersangkutan meminta permintaan mutasi kepada bagian kepegawaian.
2. Petugas menerima dan menyimpan usulan calon mutasi pegawai.
3. Petugas kemudian akan memeriksa lama berkerja calon mutasi pegawai, apakah pegawai tersebut sudah lama berkerja selama 1 tahun 6 bulan. Apabila estimasi waktu berkerja belum memenuhi kriteria yang ditentukan maka permintaan pengajuan mutasi akan ditolak.
4. Membuat laporan calon mutasi pegawai yang memenuhi kompetensi individu dan persyaratan jabatan kemudian diserahkan kepada supervisor dan asman SDM untuk diperiksa dan disahkan. Setelah disah
5. Supervisor dan asman SDM memeriksa dan mengesahkan laporan calon mutasi pegawai yang memenuhi kompetensi individu dan persyaratan jabatan.
6. Petugas kemudian dari laporan calon mutasi pegawai yang sudah disahkan, akan membuat laporan mutasi pegawai baru dan

diserahkan kembali kepada supervisor dan asman SDM untuk diperiksa dan disahkan.

7. Setelah laporan mutasi pegawai baru disahkan, petugas bagian kepegawaian akan meyimpan data mutasi pegawai baru tersebut dan mengupdate data pegawai.



Gambar 3.4 Proses Permintaan Usulan Mutasi Pegawai



Gambar 3.5 *Activity Diagram* Mutasi Pegawai Baru

Setelah melakukan analisis sistem yang berjalan pada bagian kepegawaian di PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung, dapat disimpulkan bahwa proses evaluasi absensi di bagian kepegawaian memakan waktu lama karena proses evaluasinya tidak disertakan di dalam sistem, tetapi

terpisah pada *Microsoft excel* sehingga petugas harus memproses proses sendiri evaluasi tersebut tanpa bantuan sistem yang otomatisasi. Selain itu belum adanya pemutakhiran teknologi pada pencatatan setiap perkrekrutan pegawai baru dan mutasi pegawai berpotensi untuk menimbulkan masalah dalam ketersajian data dan juga kesinkronisasian dalam penyimpanan data dengan data pegawai.

3.1.2 Analisis Proses Bisnis

Analisis Proses 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 proses bisnis yang terdapat di dalam bagian kepegawaian PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung:

3.1.2.1 Proses Bisnis Mutasi

Pada proses mutasi terdapat proses-proses bisnis, diantaranya adalah sebagai berikut :

1. Mutasi terdapat dua jenis yaitu mutasi intern dan mutasi ekstern. Mutasi intern adalah mutasi yang dilakukan masih dilakukan dalam satu lingkungan PLN yang sama, sedangkan mutasi ekstern adalah mutasi yang terjadi di lingkungan unit PLN yang satu ke unit yang lain.
2. Masa kerja pada jabatan terakhir paling sedikit 1 tahun 6 bulan.

3. Keputusan mutasi pegawai disahkan oleh pejabat yang berwenang yaitu supervisor dan asman (asisten manajer) bagian kepegawaian.

3.1.2.2 Proses Bisnis Rekrutmen Pegawai

Pada proses rekrutmen pegawai terdapat proses-proses bisnis, diantaranya adalah sebagai berikut :

1. Rekrutmen adalah proses penerimaan pegawai baru
2. Data calon rekrutmen akan diberikan kepada tim penyeleksi yang berada di luar bagian kepegawaian, kemudian setelah data calon rekrutmen telah diseleksi dan didapat informasi mengenai rekrutmen baru, data rekrutmen baru tersebut akan diberikan ke bagian kepegawaian untuk proses pencatatan.
3. Bagian kepegawaian akan melakukan pencatatan data rekrutmen baru kemudian akan diperiksa kembali oleh supervisor bagian kepegawaian dan asman kepegawaian.

3.1.2.3 Proses Bisnis Evaluasi Absensi Pegawai

Pada proses evaluasi absensi pegawai terdapat proses-proses bisnis, diantaranya adalah sebagai berikut :

1. Proses evaluasi dilakukan dengan mengimport file absensi ke dalam sistem kemudian evaluasi dilakukan oleh sistem
2. Evaluasi absensi adalah mengevaluasi siapa saja pegawai yang mengalami keterlambatan dan pulang sebelum waktunya juga alfa, izin, dan cuti

3.1.3 Analisis *Key Performance Indicators* (KPI)

Analisis KPI bertujuan untuk menganalisis bagaimana KPI diterapkan sesuai dengan tujuan mengevaluasi dari penunjuk kinerja pegawai. KPI berkaitan dengan sasaran strategis perusahaan sehingga harus dianalisis pembobotannya, penentuan objektif, serta cara penilaian akhir dari kinerja pegawai.

3.1.3.1 Pembobotan Index KPI

Setiap objektif *Key Performance Indicators* (KPI) pasti memiliki index untuk dijadikan acuan penilaian. Untuk PLN APJ Jabar dan Banten memiliki 4 objektif, dimana masing-masing objektif memiliki jangkauan nilai bobot antara 1-5 sesuai dengan ketentuan bagian kepegawaian di PLN. Berikut adalah pembobotan index KPI yang dapat dilihat pada tabel 3-1

Tabel 3-1 Index Objektif KPI

No	Objektif	Index	Bobot
1.	Terbuka terhadap Kritik Konstruktif	Tidak memuaskan	1
		Kurang	2
		Cukup memuaskan	3
		Memuaskan	4
		Sangat memuaskan	5
2.	Menunjukkan keterampilan dan pengetahuan yang diperlukan	Tidak memuaskan	1
		Kurang	2
		Cukup memuaskan	3
		Memuaskan	4
		Sangat memuaskan	5
3.	Menyelesaikan Seluruh Pekerjaan yang Telah Ditugaskan	Tidak memuaskan	1
		Kurang	2
		Cukup memuaskan	3
		Memuaskan	4
		Sangat memuaskan	5
4.	Memenuhi Persyaratan Kehadiran	Tidak memuaskan	1
		Kurang	2
		Cukup memuaskan	3

No	Objektif	Index	Bobot
		Memuaskan	4
		Sangat memuaskan	5

Objektif sendiri sifatnya dinamis, yaitu dapat ditambahkan sesuai dengan kebutuhan indikator untuk penilaian pencapaian KPI sendiri sesuai dengan penujuk ketercapaian target bagian kepegawaian yang dievaluasi.

3.1.3.2 Penghitungan Hasil Evaluasi KPI

Setelah index setiap objektifnya dipilih oleh penilai yang memiliki otoritas untuk menentukan index yang diperoleh pegawai yang dievaluasi. Berikut adalah daftar penilai dan otoritas yang dimiliki pada tabel 3-2 :

Tabel 3-2 Daftar Pemeriksa KPI

No.	Jabatan Pemeriksa	Otoritas Pemeriksa
1	Supervisor	Mengevaluasi pegawai dengan jabatan dibawah manajer, asman, supervisor.
2	Asman	Mengevaluasi pegawai dengan jabatan supervisor.

Setelah pemeriksa menentukan index dari tiap-tiap objektif pegawai yang dievaluasi kemudian dapat dihitung berapa hasil kinerja seorang pegawai tersebut, kemudian diperoleh hasil penilaian berdasarkan objektif dan index yang telah ditentukan sebelumnya. Berikut penghitungan hasil evaluasi KPI yang dapat dilihat pada tabel 3-3.

Pegawai yang dievaluasi :

Nama Pegawai : Suherman

NIP : 6285108L

Jabatan : Staff

Tabel 3-3 Penghitungan Hasil Evaluasi KPI

No	Objektif	Index	Bobot
1	Terbuka terhadap Kritik Konstruktif	Cukup	3
2	Menunjukkan Keterampilan dan Pengetahuan yang Diperlukan	Memuaskan	4
3	Menyelesaikan Seluruh Pekerjaan yang Telah Ditugaskan	Memuaskan,	4
4	Memenuhi Persyaratan Kehadiran	Sangat memuaskan	5
Jumlah Bobot			**16**

Setelah diperoleh jumlah bobot dari objektif-objektif yang ditentukan indexnya untuk seorang pegawai, barulah dapat dihitung skor yang didaptkan seorang pegawai dan index pencapaian yang didapatkan oleh pegawai tersebut. Berikut adalah rumus penghitungan skor KPI.

$$\text{Skor KPI} = \frac{\text{Jumlah Bobot}}{\text{Jumlah Soal}} * 2$$

Penghitungan untuk hasil skor KPI pegawai atas nama Suherman :

$$\text{Skor KPI} = \frac{16}{4} * 2$$

$$\text{Skor KPI} = 8$$

Apabila skor KPI dalam bilangan desimal, maka perhitungan KPI tersebut akan dibulatkan untuk penyesuaian dengan level kinerja yang diraih pegawai.

Angka skor KPI tersebut akan dikonversikan menjadi suatu level kinerja sesuai dengan skor yang diraihnya. Berikut adalah tabel level kinerja sesuai dengan tetapan bagian kepegawaian untuk mengkonversi skor KPI seperti pada tabel 3-4:

Tabel 3-4 Level Kinerja Pegawai

No	Jumlah Bobot Didapat	Level Kinerja
1	10 (sepuluh)	Sangat Luar Biasa
2	9 (sembilan)	Luar Biasa
3	8 (delapan)	Sangat Optimal
4	7 (tujuh)	Sangat Potensial
5	6 (enam)	Optimal
6	5 (lima)	Potensial
7	4 (empat)	Kandidat Potensial
8	3 (tiga)	Perlu Penyesuaian
9	2 (dua)	Perlu Perhatian
10	1 (satu)	Sangat Perlu Perhatian

Skor KPI pegawai atas nama Suherman adalah 8 (delapan), maka level kinerja yang diperoleh adalah “sangat optimal”.

3.1.3.3 Proses Bisnis *Key Performance Indicators* (KPI)

Pada proses KPI terdapat proses-proses bisnis, diantaranya adalah sebagai berikut :

1. Pemeriksa yang memiliki otoritas sah adalah supervisor bagian kepegawaian, asman kepegawaian, dan manajer bagian kepegawaian
2. Supervisor kepegawaian memiliki otoritas memeriksa pegawai dengan jabatan dibawahnya danasman kepegawaian memiliki otoritas memeriksa pegawai dengan jabatan supervisor.
3. Selain pegawai yang tidak memiliki otoritas sah tidak dapat melakukan evaluasi KPI

4. Supervisor juga bertugas untuk melakukan pengolahan objektif, seandainya ada perubahan pada objektif.

Berdasarkan hasil analisis *Key Performance Indicators* (KPI) maka dapat disimpulkan evaluasi KPI memerlukan beberapa tahapan yaitu penentuan objektif yang ditentukan oleh supervisor, mengevaluasi pegawai, dan mendapatkan hasil evaluasi dari tiap pegawai.

3.1.4 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.

3.1.4.1 Analisis Perangkat Keras

Pembangunan sistem kepegawaian di bagian kepegawaian PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung 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 3-5.

Tabel 3-5 Spesifikasi Perangkat Lunak Kebutuhan Sistem

No	Perangkat Keras	Spesifikasi
1	Prosessor	Intel Pentium 4. 2,2 GHz
2	Monitor	Monitor VGA 17 inch
3	VGA	VGA Card On-Board 128 Mb

4	Memori	Memori DDR2 1 Gb
5	Hard disk	Kapasitas bebas minimal 1 Gb
6	*Optical Drive*	CD-R
7	*Keyboard*	*Keyboard*
8	*Mouse*	*Mouse*
9	Printer	Printer hitam putih
10	Jaringan Internet	-

Sedangkan spesifikasi perangkat keras yang sudah tersedia dan digunakan di bagian kepegawaian PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung ditunjukan di tabel 3-6.

Tabel 3-6 Spesifikasi Perangkat Keras di Bagian Kepegawaian

No	Perangkat Keras	Spesifikasi
1	Prosessor	Intel Pentium 4. 2,2 GHz
2	Monitor	Monitor LCD 19 inch
3	VGA	VGA Card On-Board 128 Mb
4	Memori	Memori DDR2 2 Gb (dual slot)
5	*Hard disk*	80 Gb (kapasitas bebas 40,73 Gb)
6	*Optical Drive*	DVD-R 52x
7	*Keyboard*	*Keyboard*
8	*Mouse*	*Mouse*
9	*Printer*	*Printer* multifungsi warna
10	Jaringan Internet	-

Setelah evaluasi mengenai spesifikasi perangkat keras di bagian kepegawaian PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung dengan membandingkan kebutuhan spesifikasi minimal untuk pembuatan sistem, dapat disimpulkan spesifikasi perangkat keras di bagian kepegawaian sudah memenuhi kebutuhan perangkat keras dari sistem kepegawaian.

3.1.4.2 Analisis Perangkat Lunak

Pembangunan sistem kepegawaian di bagian kepegawaian PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung, 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 3-7.

Tabel 3-7 Spesifikasi Kebutuhan Perangkat Lunak Sistem Kepegawaian

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows XP
2	Web Server Pembangun	XAMPP
3	Basis Data	MYSQL
4	Bahasa Pemrograman	PHP

Sedangkan spesifikasi perangkat lunak yang sudah tersedia dan digunakan di bagian kepegawaian PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung ditunjukan di tabel 3-8.

Tabel 3-8 Spesifikasi Perangkat Lunak di Bagian Kepegawaian

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows XP
2	Web Server Pembangun	WAMP
3	Basis Data	MYSQL
4	Bahasa Pemrograman	PHP

Setelah evaluasi mengenai spesifikasi perangkat lunak di bagian umum PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung dengan membandingkan kebutuhan spesifikasi minimal untuk pembuatan sistem, dapat disimpulkan spesifikasi perangkat lunak di bagian kepegawaian sudah memenuhi kebutuhan perangkat lunak dari sistem kepegawaian.

3.1.5 Analisis Kebutuhan Perangkat Pikir

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

Adapun perangkat pikir (pengguna) yang dimiliki saat ini hanya satu petugas di bagian kepegawaian yaitu :

1. **Petugas Bagian Kepegawaian**

Umur	:	35 tahun
Pendidikan Terakhir	:	Starata 1 (S1)
Kemampuan yang dimiliki	:	Mampu menggunakan perangkat lunak *Office Suite* dalam menjalankan tugasnya dengan bantuan sistem yang diberikan.

Untuk menjalankan sistem informasi yang dibangun perusahaan memerlukan tiga macam pengguna yang terlibat yaitu operator, kasir dan admin. Detail penjelasan pengguna tersebut bisa dilihat dalam rincian sebagai berikut :

2. **Admin**

Umur	:	23-40
Pendidikan Minimal	:	Diploma 3 (D3)
Kemampuan yang dimiliki	:	Mampu menggunakan aplikasi dari sistem informasi yang dibangun untuk pengolahan kepegawaian.

Melihat perbandingan perangkat pikir 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 admin calon pengguna dari sistem informasi kepegawaian. Pelatihan perangkat pikir yang ada bertujuan agar pengguna sistem dapat sebaik mungkin menggunakan aplikasi yang dibangun secara optimal.

3.1.6 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 kepegawain.

Analisis yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahapan pemodelan dalam analisis tersebut antara lain mengidentifikasi aktor, pembuatan *Use Case Diagram*, *Use Case Scenario, Activity Diagram, Sequence Diagram, Class Diagram, dan state diagram*.

3.1.6.1 Identifikasi Aktor

Aktor yang telibat pada sistem kepegawaian ada 3 aktor yaitu admin, asman dan supervisor. Deskripsi untuk setiap aktornya akan dijelaskan pada tabel 3-9.

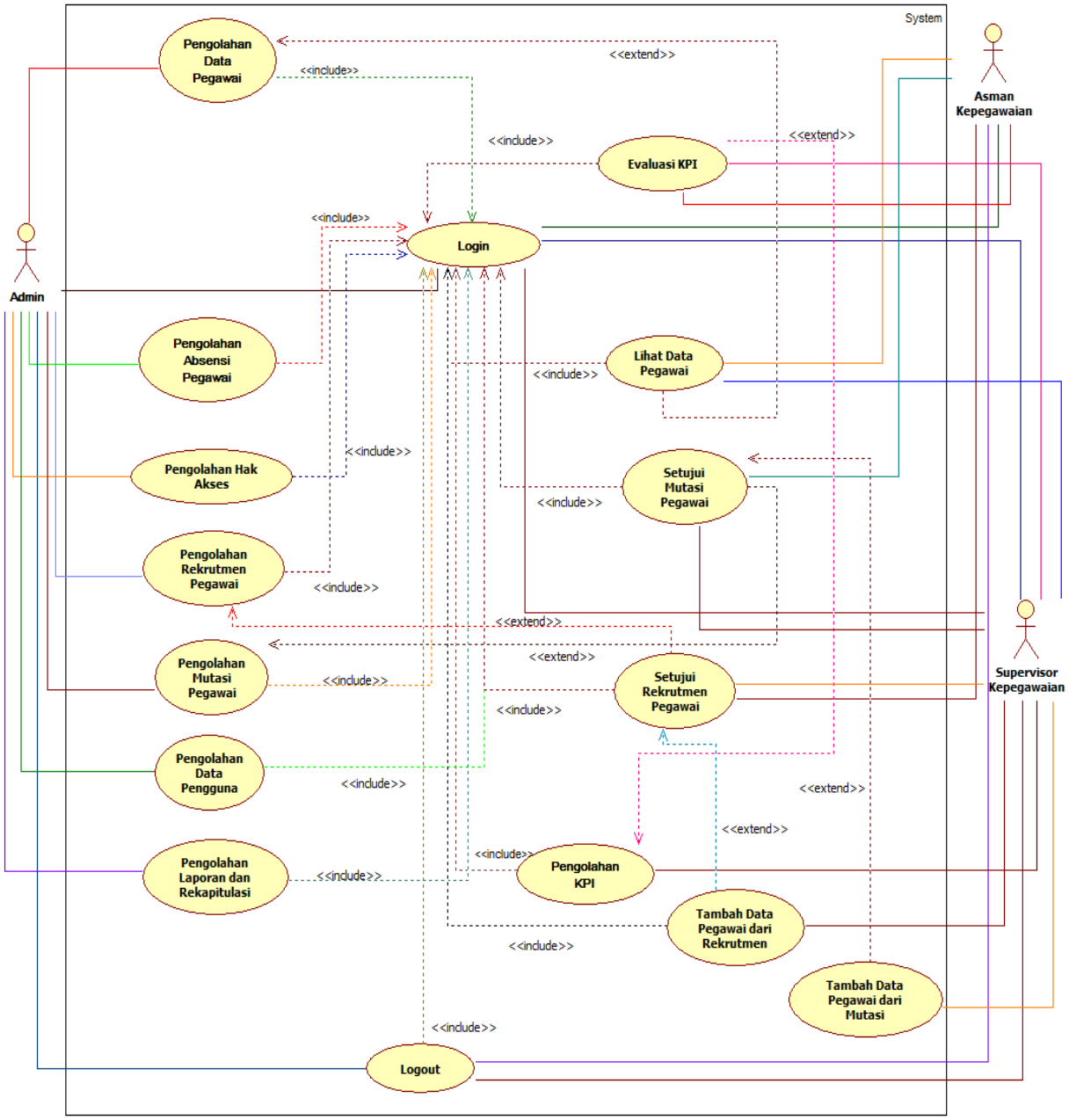
Tabel 3-9 Identifikasi Aktor

No	Aktor	Deskripsi
1	Admin	Sebagai pengatur atau pegawai yang mempunyai hak akses mengelola data petugas, akses masuk ke dalam sistem. Melakukan evaluasi pegawai
2	Asman	Melakukan pemeriksaan dan pengesahan pada laporan-laporan yang dibuat. Melakukan evaluasi pegawai
3	Supervisor	Mengawasi, mengatur pengolahan kpi dan Melakukan evaluasi pegawai

3.1.6.2 *Use Case Diagram*

Use Case Diagram adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsional yang terlihat pengguna. Adalah konstruksi untuk mendeskripsikan bagaimana sistem terlihat dimata pengguna.

Dari identifikasi aktor yang terlibat di atas maka *Use Case Diagram* dapat dilihat pada gambar 3.5.



Gambar 3.6 *Use Case Diagram*

3.1.6.3 *Use Case Scenario*

Use Case Scenario bertujuan untuk mendeskripsikan urutan langkah-langkah dalam proses bisnis baik yang dilakukan aktor terhadap sistem maupun yang dilikakukan oleh sistem terhadap aktor. Berdasarkan *Use Case Diagram* pada Gambar 3.6 maka *Use Case Scenario* untuk sistem yang dibangun terdiri dari *scenario* mengolah data pegawai, *scenario* mengolah absensi pegawai, *scenario* mengolah *Key Performance Indicators* (KPI) , *scenario* absensi pegawai, *scenario* mengolah mutasi, *scenario* mengolah rekruitmen dan *scenario* mengolah laporan dan rekapitulasi. Berikut ini penjelasan dari masing-masing *scenario* tersebut :

3.1.6.3.1 *Use Case Scenario* Login Pengguna

Interaksi antara aktor pengguna, yaitu admin dengan *use case* login dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3-10 *Use Case Scenario* Login Admin

Identifikasi Masalah	
Nama	Login
Tujuan	Masuk ke dalam sistem sebagai pengguna
Deskripsi	Proses login ini sebagai autentifikasi kewenangan sebagai pengguna dalam sistem
Aktor	admin, supervisor, dan asman
Usecase Yang Berkaitan	-
Skenario Utama	
Kondisi Awal	*Form* Login di tampilkan
Aksi Aktor	**Reaksi Sistem**
1) Mengisi semua inputan pada *form* Login kemudian menekan tombol login.	2) Mengautentifikasi data login dengan data pengguna pada basis data
	3) Menampilkan halaman utama untuk pengguna
Skenario Alternatif (Jika Gagal)	
Aksi Aktor	**Reaksi Sistem**
	1) Menampilkan pesan tidak terdaftar di database

Identifikasi Masalah	
	2) Menampilkan *form* Login
3) Mengisi semua inputan pada *form* Login kemudian menekan tombol login.	4) Menampilkan pesan bahwa data login salah
	5) Mengautentifikasi data login dengan data pengguna pada basis data
	6) Menampilkan halaman utama untuk pengguna
Kondisi Akhir	Pengguna dapat melakukan kegiatan pada sistem sesuai kewenangan dan hak akses masing-masing

3.1.6.3.2 *Use Case Scenario* Logout

Interaksi antara aktor pengguna dengan *use case* logout dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3-11 *Use Case Scenario* Logout

Identifikasi Masalah	
Nama	Logout
Tujuan	Keluar dari sistem
Deskripsi	Proses logout ini digunakan untuk keluar dari sistem
Aktor	Admin, supervisor, dan asman
Usecase Yang Berkaitan	-
Skenario Utama	
Kondisi Awal	Pengguna berada pada sistem dengan kondisi login
Aksi Aktor	**Reaksi Sistem**
1) Pengguna menekan tombol logout	2) Menampilkan pesan konfirmasi logout
3) Pengguna akan memilih tombol ya	4) Sistem merespon dengan merubah status log pengguna dan mengarahkan pengguna ke halaman login
Kondisi Akhir	Pengguna berhasil melakukan proses logout pada sistem

3.1.6.3.3 *Use Case Scenario* Pengolahan Data Pegawai

Interaksi antara aktor pengguna, yaitu admin dengan *use case* pengolahan data pegawai dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3-12 *Use Scenario* Pengolahan Data Pegawai

Identifikasi Masalah	
Nama	Pengolahan Data Pegawai
Tujuan	Untuk mengolah data pegawai
Deskripsi	Proses pengolahan data pegawai meliputi menambah, mengubah, menghapus dan mencari data pegawai
Aktor	Admin
Usecase Yang Berkaitan	-
Skenario Utama	
Kondisi Awal	Tampilan *Form* Pengolahan Data Pegawai
Aksi Aktor	**Reaksi Sistem**
1. Admin memilih tombol tambah data pegawai	2. Sistem akan menampilkan *form* isi tambah data kepada *admin* untuk mengisi data pegawai
3. setelah data diisi dengan valid dan benar admin memilih tombol simpan data pegawai	4. Sistem memeriksa data yang dimasukkan oleh *admin*, jika data yang dimasukkan itu *valid* dan benar maka data yang dimasukkan akan disimpan ke *database* dan mengarahkan admin ke halaman menu pegawai dan menampilkan notifikasi data baru berhasil tersimpan, tetapi jika data yang dimasukkan tidak valid,maka sistem akan menampilkan ulang *form* tambah data dan menampilkan pesan kesalahan.
5. Admin memilih tombol ubah pada baris data pegawai yang diinginkan	6. Sistem akan menampilkan *form* ubah data
7. Admin mengisi data pegawaiyang ingin dirubah dengan data yang benar dan valid	8. Sistem akan memeriksa data masukkan *admin*, jika data yang dimasukkan valid dan benar, maka data yang diubah akan disimpan dan menampilkan pesan notifikasi data berhasil diubah, jika data yang dimasukkan tidak *valid* maka sistem akanmenampilkan pesan kesalahan dan menampilkanulang *form* isian ubah data pegawai
9. Admin akan menghapus data pegawai yang diinginkan.	10. sistem akan menampilkan pesan kepada admin, apakah data pegawai ingin dihapus atau tidak
11. Admin akan menekan tombol Ok	12. Sistem akan menghapus data yang ingin dihapus dan menampilkan notifikasi pesan data berhasil dihapus
13. Admin menekan *link* pada NIP pegawai untuk melihat histori pegawai	14. Sistem menampilkan tampilan histori pegawai yang dipilih.
15. Admin memlih upload data pegawai	16. Sistem akan menampilkan form upload data pegawai
17. Admin akan memilih data yang akan diupload dengan ekstensi	18. Sistem akan memriksa apakah data yang akan sudah diplih, ekstensi data sesuai dengan

Identifikasi Masalah	
yang sesuai dan menekan tombol upload	ketentuan, dan tidak melebihi batas maksimum ukuran file. Apabila data yang diupload akan menampilkan notifikasi kesalahan dan mengarahkan kembali ke *form* upload data pegawai
Skenario Alternatif (Jika Gagal)	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan data pegawai gagal ditambah karena NIP yang ditambahkan/diubah sudah ada di data pegawai.
2. Mengisi kembali NIP data pegawai di *form* tambah/ubah data pegawai	
Kondisi Akhir	Data di sistem yang sudah mengalami proses pengolahan data pegawai

3.1.6.3.4 *Use Case Scenario* Pengolahan Absensi Pegawai

Interaksi antara aktor pengguna, yaitu admin dengan *use case* pengolahan absensi pegawai dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3-13 *Use Case Scenario* Pengolahan Absensi Pegawai

Identifikasi Masalah	
Nama	Pengolahan Absensi Pegawai
Tujuan	Untuk mengolah absensi pegawai
Deskripsi	Proses pengolahan data pegawai meliputi mengupload data absensi dan evaluasi absensi
Aktor	Admin
Usecase Yang Berkaitan	-
Skenario Utama	
Kondisi Awal	Tampilan *form* Pengolahan Absensi Pegawai
Aksi Aktor	**Reaksi Sistem**
1. Admin memilih tombol upload data *absensi*	2. Sistem akan menampilkan *form* upload data absensi
3. Admin memasukkan data .xls yang akan diupload	4. Sistem akan menerima inputan dan kemudian memasukkan ke dalam basisdata, apabila data berhasil di inputkan maka akan tampil notifikasi dan kembali ke menu absensi pegawai.
5. Admin memilih evaluasi Absensi berdasarkan kategori yang diinginkan.	6. Sistem akan menampilkan evaluasi yang dipilih oleh sistem dan menampilkan hasil evaluasi berdasarkan data pegawai

Identifikasi Masalah	
7. Admin memilih Rekap Absensi	8. Sistem akan menampilkan data absensi pegawai beserta rekapitulasi kehadirannya dan detail data absensi.
Kondisi Akhir	Admin dapat mengelola absensi pegawai sesuai dengan kebutuhan
	Data di sistem yang sudah mengalami proses pengolahan absensi pegawai

3.1.6.3.5 *Use Case Scenario* Pengolahan *Key Performance Indicators* (KPI)

Interaksi antara aktor pengguna, yaitu admin dengan *use case* pengolahan KPI dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3-14 *Use Case Scenario* Pengolahan KPI

Identifikasi Masalah	
Nama	Pengolahan KPI
Tujuan	Untuk mengolah *Key Performance Indicators* KPI
Deskripsi	Proses pengolahan KPI meliputi menambah objektif, mengubah objektif, menghapus objektif
Aktor	Supervisor
Usecase Yang Berkaitan	-
Skenario Utama	
Kondisi Awal	Tampilan *form* Pengolahan KPI
Aksi Aktor	**Reaksi Sistem**
1. *Supervisor* memilih tombol tambah objektif	2. Sistem akan menampilkan *form* isi tambah objektif *supervisor* untuk mengisi data objektif KPI
3. setelah data diisi dengan valid dan benar *supervisor* memilih tombol simpan data petugas	4. Sistem memeriksa data yang dimasukkan oleh *supervisor*, jika data yang dimasukkan itu *valid* dan benar maka data yang dimasukkan akan disimpan ke *database* dan mengarahkan *supervisor* ke halaman menu petugas, tetapi jika data yang dimasukkan tidak valid,maka sistem akan menampilkan ulang *form* tambah data dan menampilkan pesan kesalahan
5. *Supervisor* memilih tombol ubah objektif yang diinginkan	6. Sistem akan memeriksa apakah data yang dipilih oleh *supervisor*valid, jika valid maka sistem akan menampilkan form isian lengkap dengan data yang ingin diubah, jika data yang dipilih *supervisor*tidak valid maka sistem akan menampilkan pesan tidak valid dan mengarahkan *supervisor* ke halaman menu

Identifikasi Masalah	
	petugas
7. *Supervisor* mengisi objektif yang ingin dirubah dengan data yang benar dan valid	8. Sistem akan memeriksa data masukkan *supervisor*, jika data yang dimasukkan valid dan benar, maka data yang baru akan disimpan, jika data yang dimasukkan tidak *valid* maka sistem akanmenampilkan pesan kesalahan dan menampilkanulang *form* isian ubah data petugas
9. *Supervisor* akan menghapus objektif yang diinginkan.	10. sistem akan menampilkan pesan kepada *supervisor*apakah objektif ingin dihapus atau tidak
Skenario Alternatif (Jika Gagal)	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan pesan inputan pada *form* tambah atau ubah objektif KPI belum terisi.
2. Supervisor akan mengisikan kembali pada *form*	3. Menampilkan notifikasi bahwa data berhasil tersimpan.
Kondisi Akhir	Data di sistem yang sudah mengalami proses pengolahan KPI

3.1.6.3.6 *Use Case Scenario* Laporan dan Rekapitulasi

Interaksi antara aktor pengguna, yaitu admin dengan *use case* laporan dan rekaptulasi dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3-15 *Use Case Scenario* Laporan dan Rekapitulasi

Identifikasi Masalah	
Nama	Laporan dan Rekapitulasi
Tujuan	Untuk mengolah Laporan dan Rekapitulasi
Deskripsi	Proses yang akan menampilkan dan mencetak segala tentang laporan yang terdapat di dalam sistem informasi kepegawaian
Aktor	Admin
Usecase Yang Berkaitan	-
Skenario Utama	
Kondisi Awal	Tampilan *Form* laporan dan rekapitulasi
Aksi Aktor	**Reaksi Sistem**
	1 Menampilkan *form* laporan dan rekapitulasi
2 Memilih laporan atau rekapitulasi yang akan dilihat atau dicetak.	3. Menampilkan *form* laporan yang dipilih oleh admin

Identifikasi Masalah	
4. Menekan Tombol Cetak	5. Sistem akan menampilkan laporan dalam format **.pdf* kemudian laporan sudah siap dicetak.
Kondisi Akhir	Admin dapat mengolah laporan dan rekapitulasi sesuai dengan kebutuhan

3.1.6.3.7 *Use Case Scenario* Pengolahan Mutasi Pegawai

Interaksi antara aktor pengguna, yaitu admin dengan *use case* pengolahan mutasi pegawai dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3-16 *Use Case Scenario* Pengolahan Mutasi Pegawai

Identifikasi Masalah	
Nama	Pengolahan Mutasi Pegawai
Tujuan	Untuk mengolah Mutasi Pegawai
Deskripsi	Proses pengolahan Mutasi Pegawai seperti menambah permintaan mutasi, mengubah, menghapus
Aktor	Admin, Supervisor, Asman
Usecase Yang Berkaitan	Setujui Mutasi Pegawai dan tambah data pegawai dari mutasi.
Skenario Utama	
Kondisi Awal	Tampilan *form* Pengolahan Mutasi Pegawai
Aksi Aktor	**Reaksi Sistem**
1. Admin memilih tombol tambah data mutasi	2. Sistem akan menampilkan *form* isi tambah data mutasi
3. setelah data diisi dengan valid dan benar admin memilih tombol simpan data mutasi	4. Sistem memeriksa data yang dimasukkan oleh *admin*, jika data yang dimasukkan itu *valid* dan benar maka data yang dimasukkan akan disimpan ke *database* dan mengarahkan admin ke halaman menu mutasi, tetapi jika data yang dimasukkan tidak valid,maka sistem akan menampilkan ulang *form* tambah data dan menampilkan pesan kesalahan
5. Admin memilih tombol ubah data mutasi yang diinginkan	6. Sistem akan memeriksa apakah data yang dipilih oleh admin valid, jika valid maka sistem akan menampilkan form isian lengkap dengan data yang ingin diubah, jika data yang dipilih admintidak valid maka sistem akan menampilkan pesan tidak valid dan mengarahkan admin ke halaman menu mutasi
7. Admin mengisi data	8. Sistem akan memeriksa data masukkan *admin*,

Identifikasi Masalah	
mutasi yang ingin dirubah dengan data yang benar dan valid	jika data yang dimasukkan valid dan benar, maka data yang baru akan disimpan, jika data yang dimasukkan tidak *valid* maka sistem akan menampilkan pesan kesalahan dan menampilkanulang *form* isian ubah data mutasi
9. Admin akan menghapus data mutasi yang diinginkan.	10. sistem akan menampilkan pesan kepada admin apakah objektif ingin dihapus atau tidak.
11. Supervisor Menekan tombol setujui mutasi	12. Sistem akan mengkonfirmasi terlebih dahulu apakah yakin untuk menyetujui data mutasi tersebut. Apabila jenis mutasi ekstern maka sistem akan menampilkan form tambah pegawai. Apabila data mutasi intern, maka hasil validasi setujui data mutasi akan tersimpan di basis data.
13. Asman menekan tombol setujui mutasi	14. Sistem akan mengkonfirmasi terlebih dahulu apakah yakin untuk menyetujui data mutasi tersebut. Setelah dikonfirmasi maka data mutasi berhasil disetujui dan data mutasi akan tersimpan di basis data, dan data telah disahkan.
Skenario Alternatif (Jika Gagal)	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan notifikasi bahwa NIP pegawai yang dimasukan masih dalam proses mutasi (belum disetujui).
2. Admin Memasukan NIP pegawai yang tidak dalam proses mutasi.	3. Menampilkan notifikasi bahwa data mutasi yang dimasukan belum memenuhi persyaratan mutasi
4. Admin Memasukan data mutasi yang sudah memenuhi persyaratan mutasi.	
Kondisi Akhir	Pengguna dapat mengelola mutasi pegawai sesuai dengan kebutuhan
	Data di sistem yang sudah mengalami proses pengolahan mutasi pegawai

3.1.6.3.8 *Use Case Scenario* Pengolahan Rekrutmen Pegawai

Interaksi antara aktor pengguna, yaitu admin dengan *use case* Pengolahan Rekrutmen dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3-17 *Use Case Scenario* Pengolahan Rekrutmen Pegawai

Identifikasi Masalah	
Nama	Pengolahan Rekrutmen pegawai
Tujuan	Untuk mengolah Rekrutmen pegawai
Deskripsi	Proses pengolahan Rekrutmen pegawai seperti menambah data rekrutmen, mengubah, menghapus, menyetujui, dan menambah data pegawai
Aktor	Admin, supervisor, asman
Usecase Yang Berkaitan	Setujui rekrutmen pegawai dan tambah data pegawai dari rekrutmen
Skenario Utama	
Kondisi Awal	Tampilan *form* Pengolahan Rekrutmen pegawai
Aksi Aktor	**Reaksi Sistem**
1. Admin memilih tombol tambah data rekrutmen	2. Sistem akan menampilkan *form* isi tambah data rekrutmen
3. setelah data diisi dengan valid dan benar admin memilih tombol simpan data rekrutmen	4. Sistem memeriksa data yang dimasukkan oleh *admin*, jika data yang dimasukkan itu *valid* dan benar maka data yang dimasukkan akan disimpan ke *database* dan mengarahkan admin ke halaman menu mutasi, tetapi jika data yang dimasukkan tidak valid,maka sistem akan menampilkan ulang *form* tambah data dan menampilkan pesan kesalahan
5. Admin memilih tombol ubah data rekrutmen yang diinginkan	6. Sistem akan memeriksa apakah data yang dipilih oleh admin valid, jika valid maka sistem akan menampilkan form isian lengkap dengan data yang ingin diubah, jika data yang dipilih admintidak valid maka sistem akan menampilkan pesan tidak valid dan mengarahkan admin ke halaman menu mutasi
7. Admin mengisi data rekrutmen yang ingin dirubah dengan data yang benar dan valid	8. Sistem akan memeriksa data masukkan *admin*, jika data yang dimasukkan valid dan benar, maka data yang baru akan disimpan, jika data yang dimasukkan tidak *valid* maka sistem akan menampilkan pesan kesalahan dan menampilkanulang *form* isian ubah data rekrutmen

Identifikasi Masalah	
9. Admin akan menghapus rekrutmen yang diinginkan.	10. sistem akan menampilkan pesan kepada adminapakah rekrutemen ingin dihapus atau tidak
10. Supervisor Menekan tombol setujui rekrutmen	11. Sistem akan mengkonfirmasi terlebih dahulu apakah yakin untuk menyetujui data rekrutmen tersebut. Setelah dikonfirmasi maka supervisor sebagai actor akan menabahkan data rekrutmen tersebut ke dalam data pegawai dan data tesebut akan tersimpan di dalam basis data.
Kondisi Akhir	Admin dapat mengelola rekrutmen pegawai sesuai dengan kebutuhan
	Data di sistem yang sudah mengalami proses pengolahan rekrutmen pegawai

3.1.6.3.9 *Use Case Scenario* Evaluasi KPI

Interaksi antara aktor pengguna, yaitu pegawai dengan *use case* evaluasi KPI dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3-18 *Use Case Scenario* Evaluasi KPI

Identifikasi Masalah	
Nama	Evaluasi KPI
Tujuan	Untuk Melihat dan mengisi Evaluasi KPI
Deskripsi	Pegawai dapat melihat kategori KPI, mengisi evaluasi dan melihat hasil evaluasi
Aktor	Supervisor,Asman,Manajer
Usecase Yang Berkaitan	-
Skenario Utama	
Kondisi Awal	Tampilan Evaluasi KPI
Aksi Aktor	**Reaksi Sistem**
1. Pengguna menginputkan data pribadinya sebagai pemeriksa	2. Sistem akan menampilkan *form* evaluasi KPI apabila pengguna menginputkan data pribadinya sesuai dengan otoritas sebagai pemerksa,apabila pengguna tidak memiliki otoritas maka akan tampil pesan peringatan dan kembali menampilkan *form* inputan.
3. setelah data diisi dengan valid dan benar pengguna memilih	4. sistem akan memeriksa apakah isisan sudah valid, apabila isian belum valid maka sistem akan memberikan notifikasi. Apabila valid

Identifikasi Masalah	
tombol evaluasi pada data yang akan dievaluasi	maka sistem akan mengarahkan pengguna ke halaman evaluasi KPI.
5. Pengguna memilih tombol index objektif sesuai komprehensif pegawai tersebut	6. Sistem akan menghitung pilihan index yang dipilih oleh pengguna tersebut, dan kemudian akan menghitung index performansi dari pegawai yang diperiksa oleh pengguna.Sistem kemudian akan menampilkan form hasil
Kondisi Akhir	Pengguna berhasil mengevaluasi pegawai

3.1.6.3.10 *Use Case Scenario* Pengolahan Hak Akses

Interaksi antara aktor pengguna, yaitu admin dengan *use case* Pengolahan Hak akses dijelaskan dalam *use case* skenario sebagai berikut :

Tabel 3-19 *Use Case Scenario* Pengolahan Hak Akses

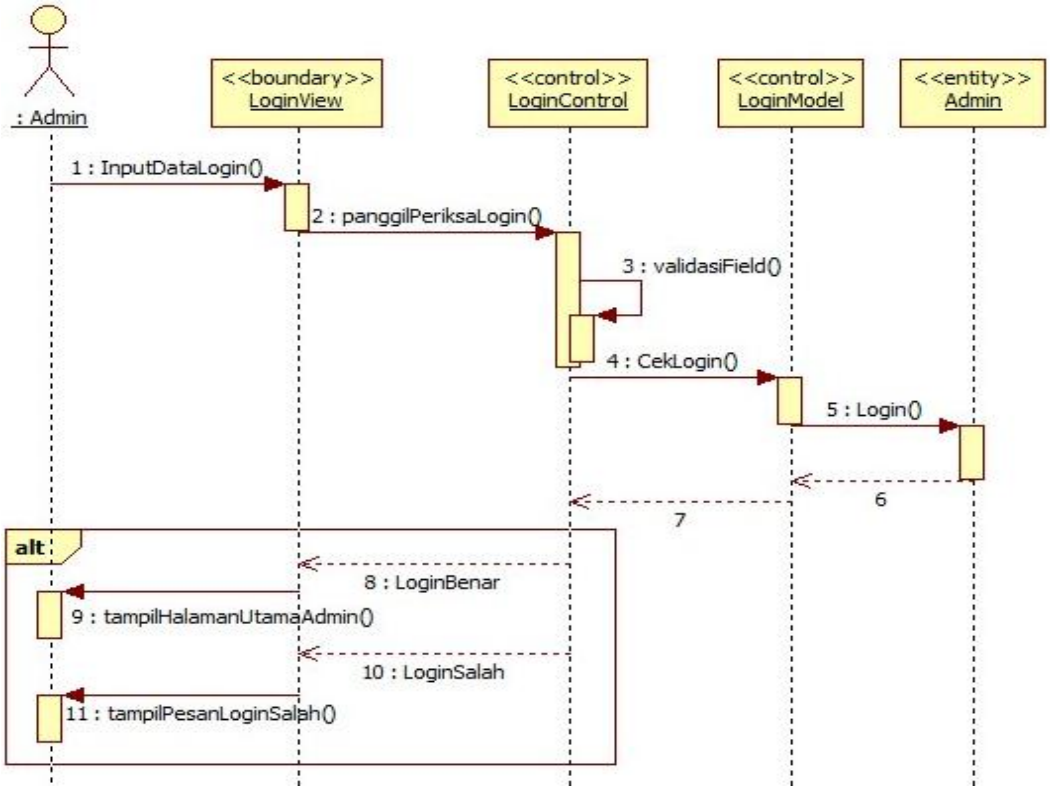
Identifikasi Masalah	
Nama	Pengolahan Hak akses pegawai
Tujuan	Untuk mengolah Hak akses pegawai
Deskripsi	Proses pengolahan Hak akses pegawai seperti mengubah data hak akses
Aktor	Admin
Usecase Yang Berkaitan	Ubah Hak Akses
Skenario Utama	
Kondisi Awal	Tampilan *form* Pengolahan Hak akses pegawai
Aksi Aktor	**Reaksi Sistem**
1. Admin memilih tombol ubah hak akses dari pengguna yang dipilih	2. Sistem akan menampilkan *form* ubah hak akses
3. setelah data diisi dengan valid dan benar admin memilih tombol simpan hak akses	4. Sistem memeriksa data yang dimasukkan oleh *admin*, jika data yang dimasukkan itu *valid* dan benar maka data yang dimasukkan akan disimpan ke *database* dan mengarahkan admin ke halaman menu rekrutmen, tetapi jika data yang dimasukkan tidak valid,maka sistem akan menampilkan ulang *form* ubah data dan menampilkan pesan kesalahan
Kondisi Akhir	Admin dapat mengelola hak akses pegawai sesuai dengan kebutuhan

3.1.6.4 *Sequence Diagram*

Use Case menggambarkan interaksi antar masing-masig objek pada setiap *Use Case* dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi. Berikut ini penjelasan dari masing-masing *Sequence Diagram* :

3.1.6.4.1 *Sequence Diagram* Login

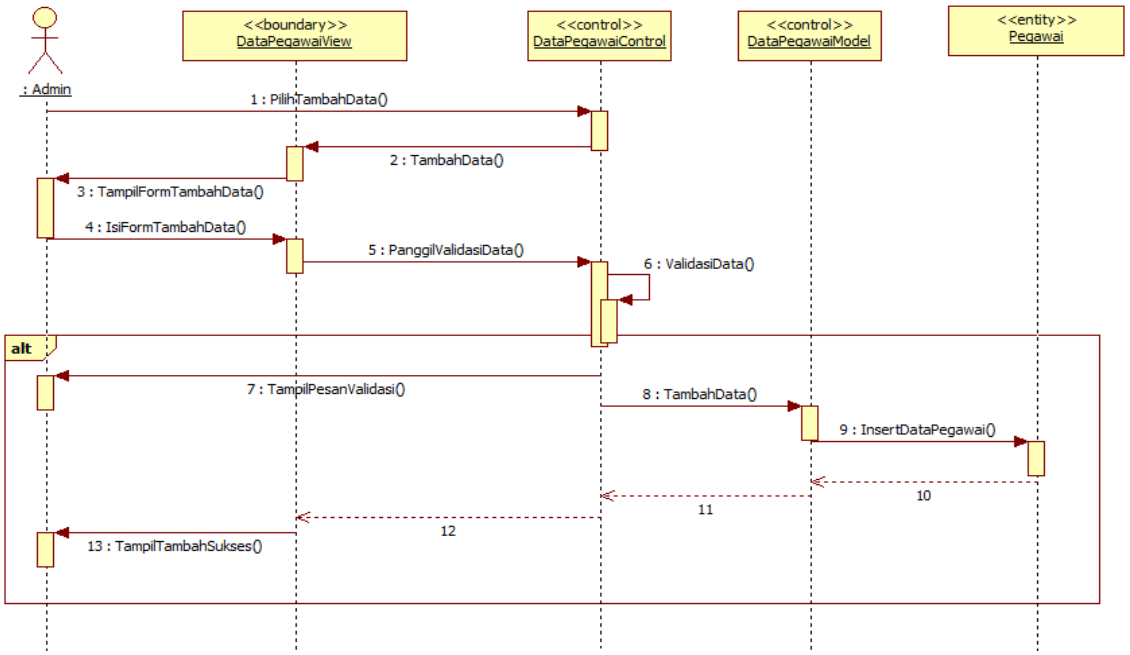
Interaksi antara aktor pengguna dengan *use case* login dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.7 *Sequence Diagram* Login

3.1.6.4.2 *Sequence Diagram* Tambah Data Pegawai

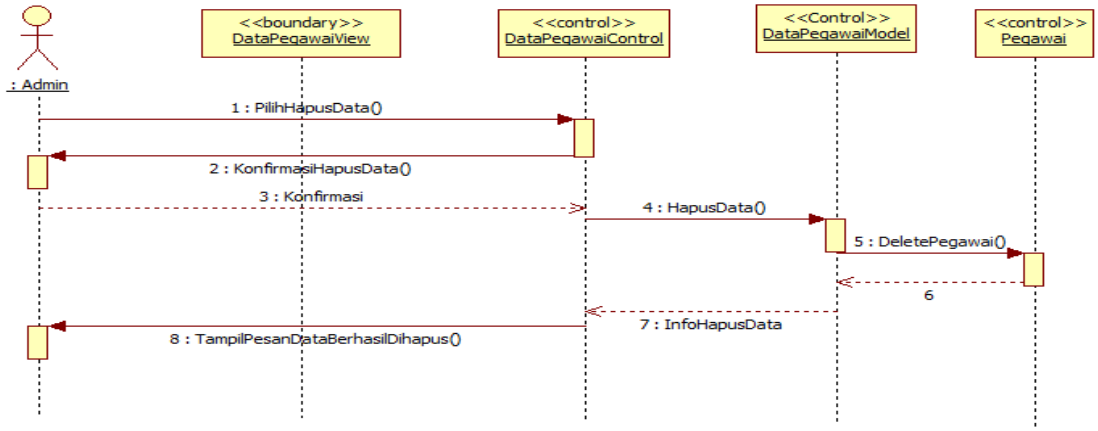
Interaksi antara aktor pengguna dengan *use case* tambah data pegawai dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.8 *Sequence Diagram* Tambah Data Pegawai

3.1.6.4.3 *Sequence Diagram* Ubah Data Pegawai

Interaksi antara aktor pengguna dengan *use case* ubah data pegawai dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.9 *Sequence Diagram* Ubah Data Pegawai

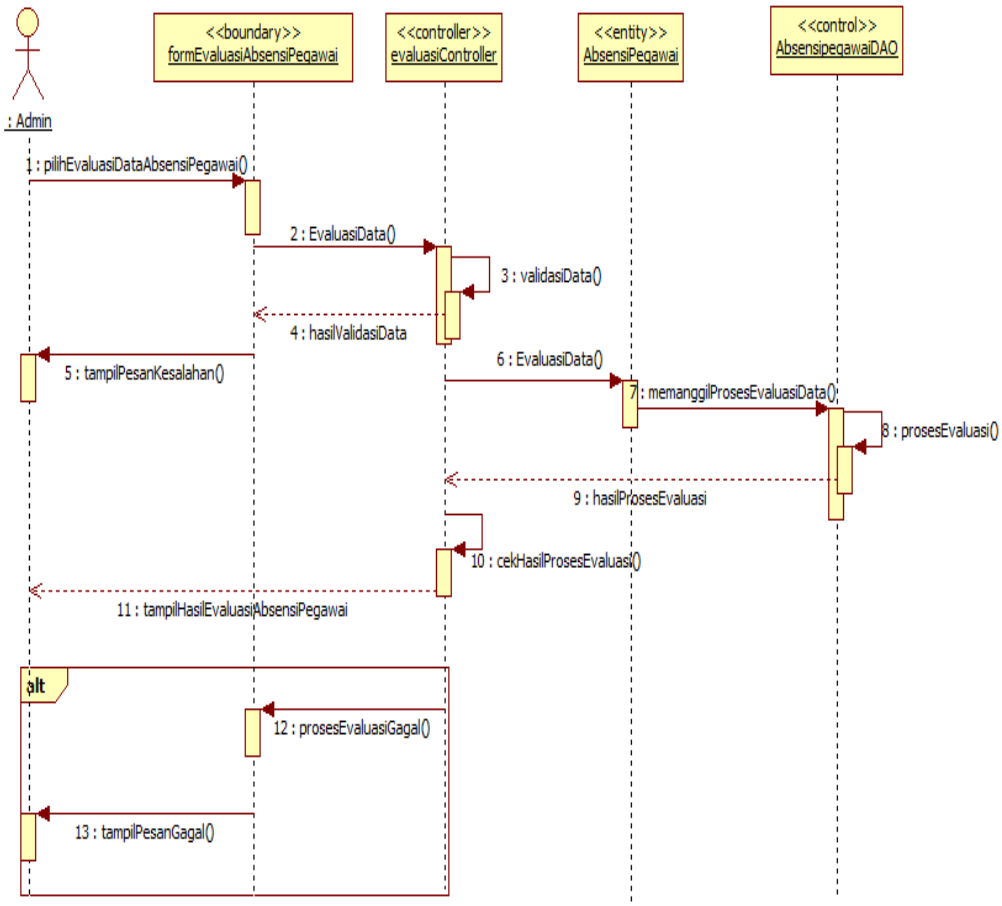
3.1.6.4.4 *Sequence Diagram* Pengolahan Absensi Pegawai

Interaksi antara aktor pengguna dengan *use case* pengolahan absensi pegawai dijelaskan dalam *Sequence Diagram* Rekap absensi pegawai, dan

evaluasi absensi pegawai. Berikut ini penjelasan dari masing-masing *Sequence Diagram* tersebut :

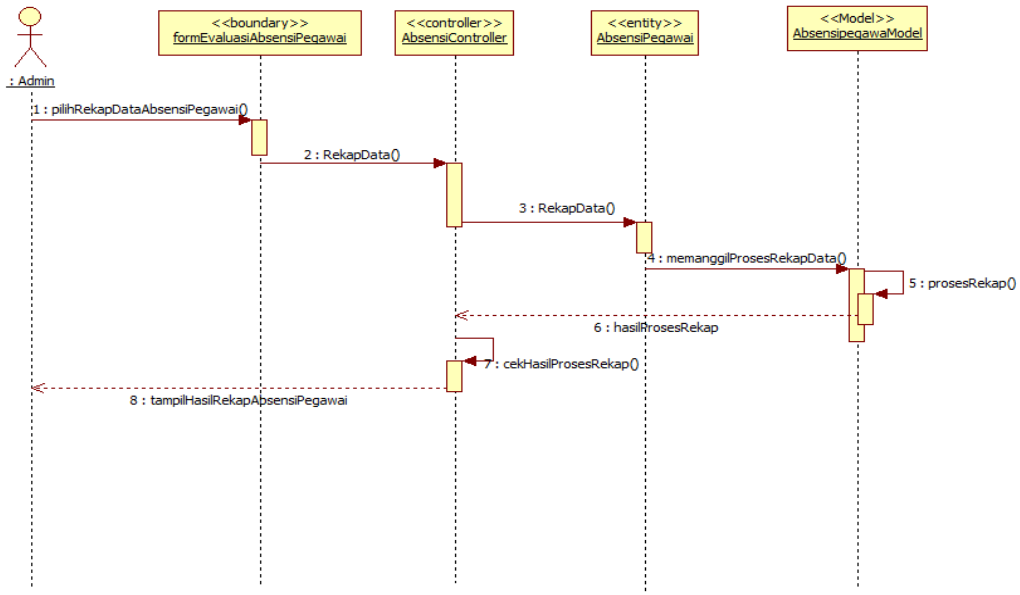
3.1.6.4.5 *Sequence Diagram* Evaluasi Absensi Pegawai

Interaksi antara aktor pengguna dengan *use case* evaluasi keterangan absensi pegawai dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.10 *Sequence Diagram* Evaluasi Absensi Pegawai

3.1.6.4.6 *Sequence Diagram* Rekap Absensi Pegawai



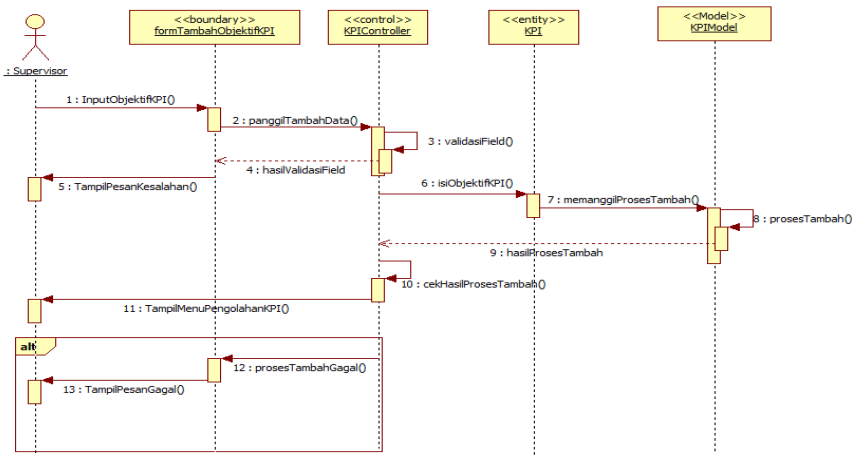
Gambar 3.11 *Sequence Diagram* Rekap Absensi Pegawai

3.1.6.4.7 *Sequence Diagram* Pengolahan KPI

Interaksi antara aktor pengguna dengan *use case* pengolahan (*Key Performance Indicators*) KPI dijelaskan dalam *Sequence Diagram* tambah *objektif* KPI, ubah *objektif* KPI, hapus *objektif* KPI, dan evaluasi *objektif* KPI. Berikut ini penjelasan dari masing-masing *Sequence Diagram* tersebut :

3.1.6.4.8 *Sequence Diagram* Tambah Objektif KPI

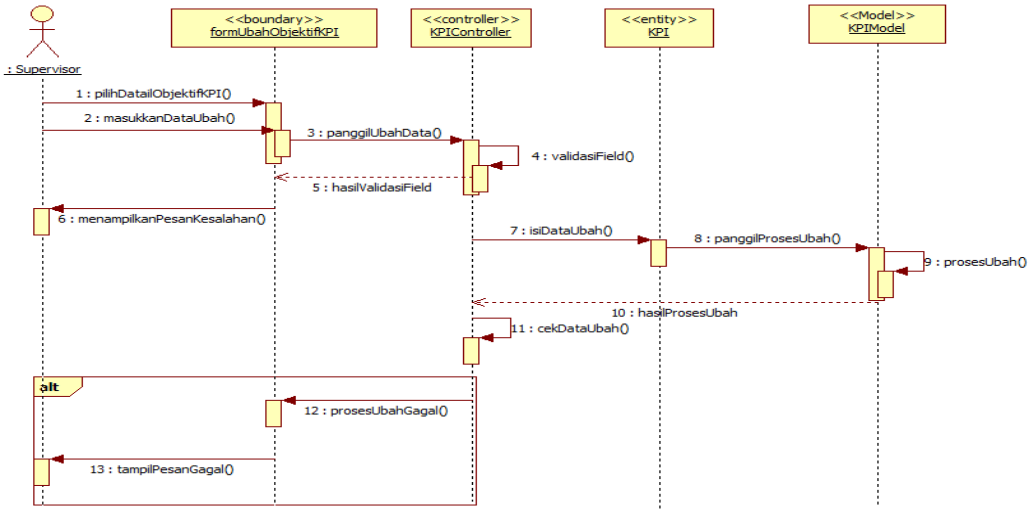
Interaksi antara aktor pengguna dengan *use case* tambah objektif KPI dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.12 *Sequence Diagram* Tambah Objektif KPI

3.1.6.4.9 *Sequence Diagram* Ubah Objektif KPI

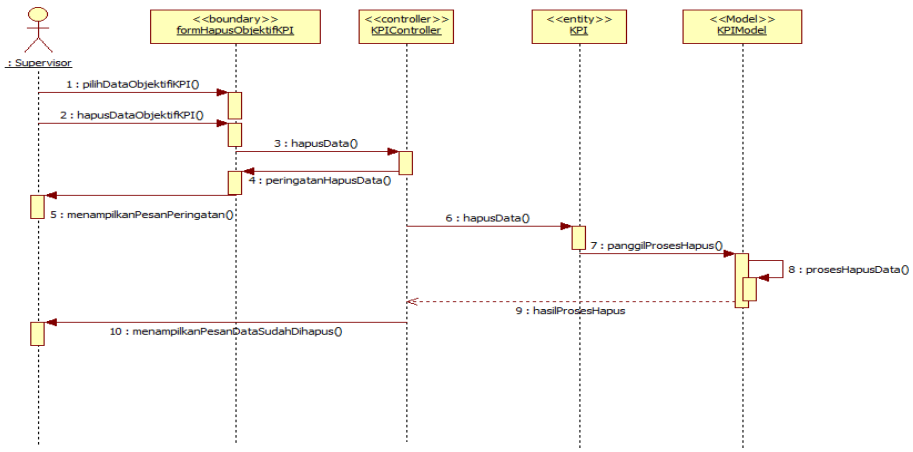
Interaksi antara aktor pengguna dengan *use case* ubah objektif KPI dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.13 *Sequence Diagram* Ubah Objektif KPI

3.1.6.4.10 *Sequence Diagram* Hapus Objektif KPI

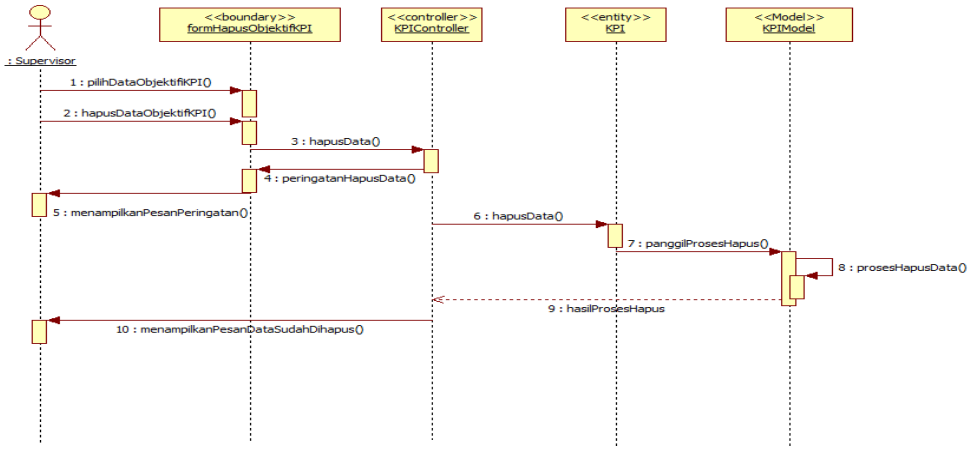
Interaksi antara aktor pengguna dengan *use case* hapus objektif KPI dijelaskan dalam *Sequence Diagram* sebagai berikut :



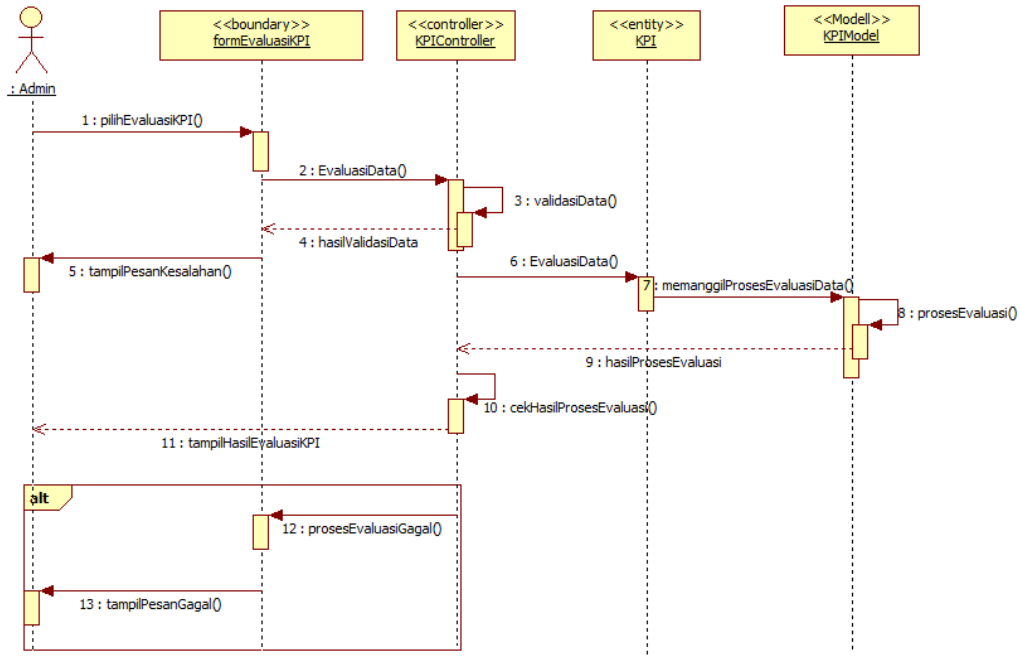
Gambar 3.14 *Sequence Diagram* Hapus Objektif KPI

3.1.6.4.11 *Sequence Diagram* Evaluasi KPI

Interaksi antara aktor pengguna dengan *use case* Evaluasi KPI dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.15 *Sequence Diagram* Hapus Objektif KPI



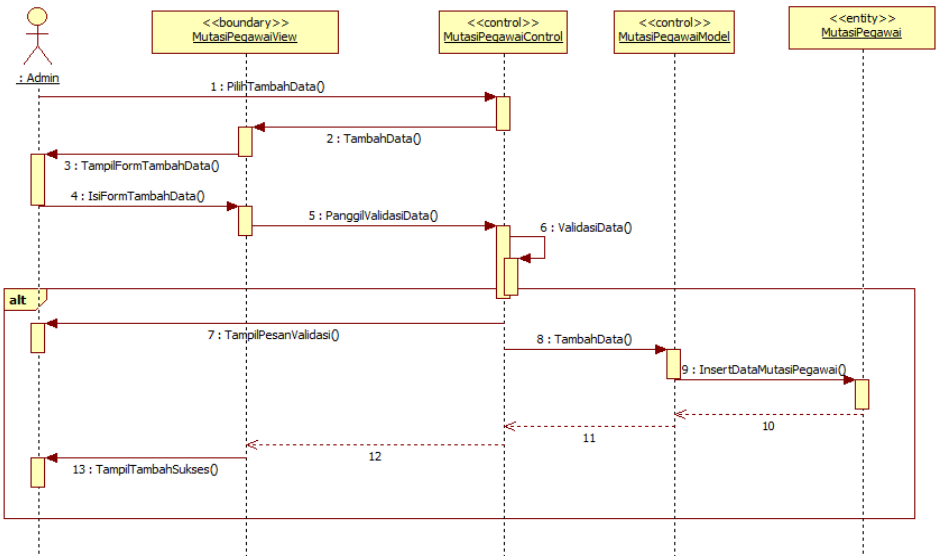
Gambar 3.16 *Sequence Diagram* Evaluasi kategori KPI

3.1.6.4.12 *Sequence Diagram* Pengolahan Mutasi Pegawai

Interaksi antara aktor pengguna dengan *use case* pengolahan mutasi pegawai dijelaskan dalam *Sequence Diagram* tambah mutasi pegawai, ubah mutasi pegawai, hapus mutasi pegawai, setujui mutasi pegawai, dan tambah data pegawai dari mutasi. Berikut ini penjelasan dari masing-masing *Sequence Diagram* tersebut :

3.1.6.4.13 *Sequence Diagram* Tambah Mutasi Pegawai

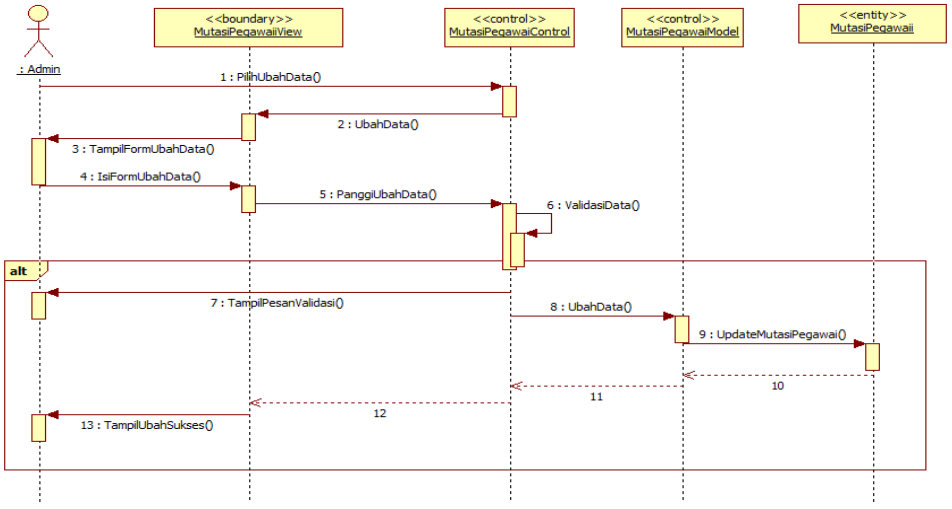
Interaksi antara aktor pengguna dengan *use case* tambah mutasi pegawai dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.17 *Sequence Diagram* Tambah Mutasi Pegawai

3.1.6.4.14 *Sequence Diagram* Ubah Mutasi Pegawai

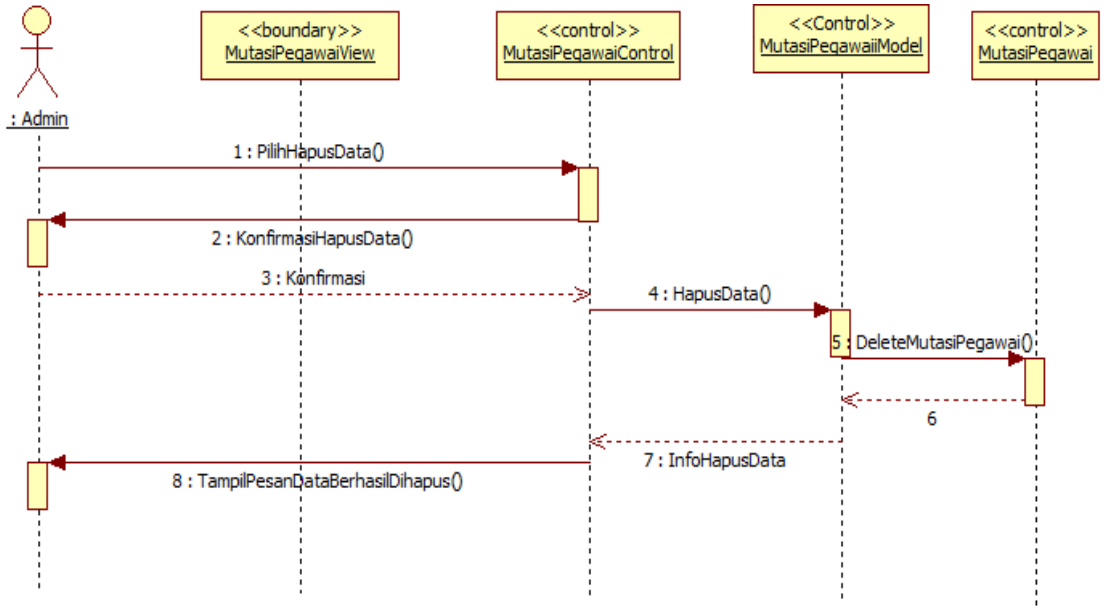
Interaksi antara aktor pengguna dengan *use case* ubah mutasi pegawai dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.18 *Sequence Diagram* Ubah Mutasi Pegawai

3.1.6.4.15 *Sequence Diagram* Hapus Mutasi Pegawai

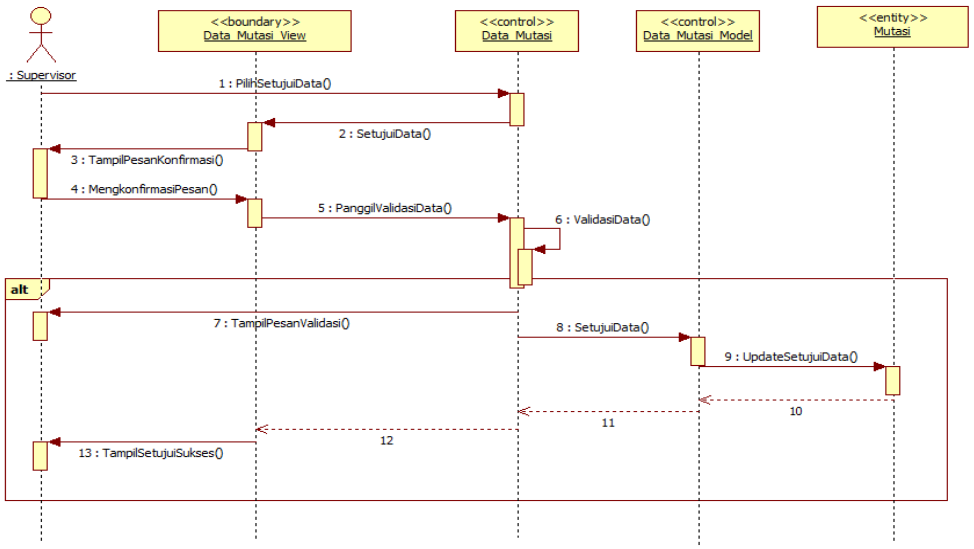
Interaksi antara aktor pengguna dengan *use case* hapus mutasi pegawai dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.19 *Sequence Diagram* Hapus Mutasi Pegawai

3.1.6.4.16 *Sequence Diagram* Setujui Mutasi Pegawai

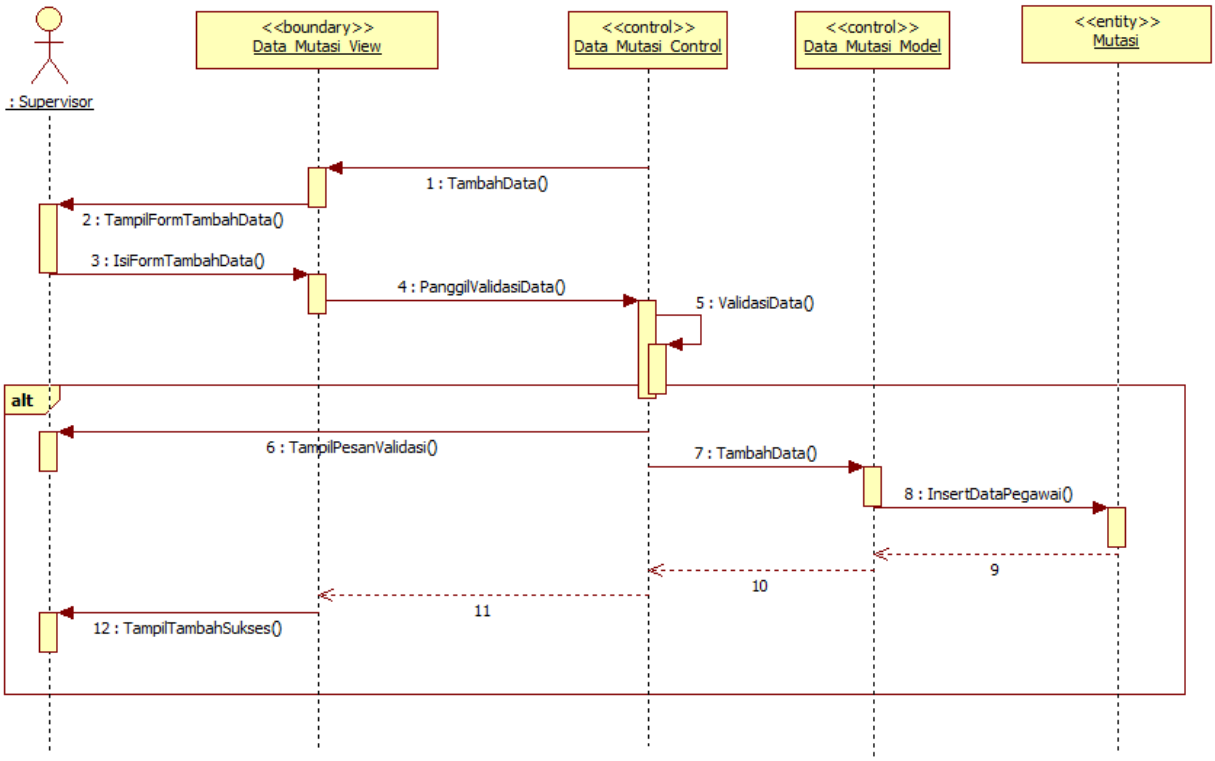
Interaksi antara aktor pengguna dengan *use case* setujui mutasi pegawai dijelaskan dalam *Sequence Diagram* setujui mutasi pegawai. Berikut ini penjelasan dari *Sequence Diagram* tersebut :



Gambar 3.20*Sequece Diagram* Setujui Mutasi Pegawai

3.1.6.4.17 *Sequence Diagram* Tambah Data Pegawai dari Mutasi

Interaksi antara aktor pengguna dengan *use case* tambah mutasi pegawai dijelaskan dalam *Sequence Diagram* sebagai berikut :



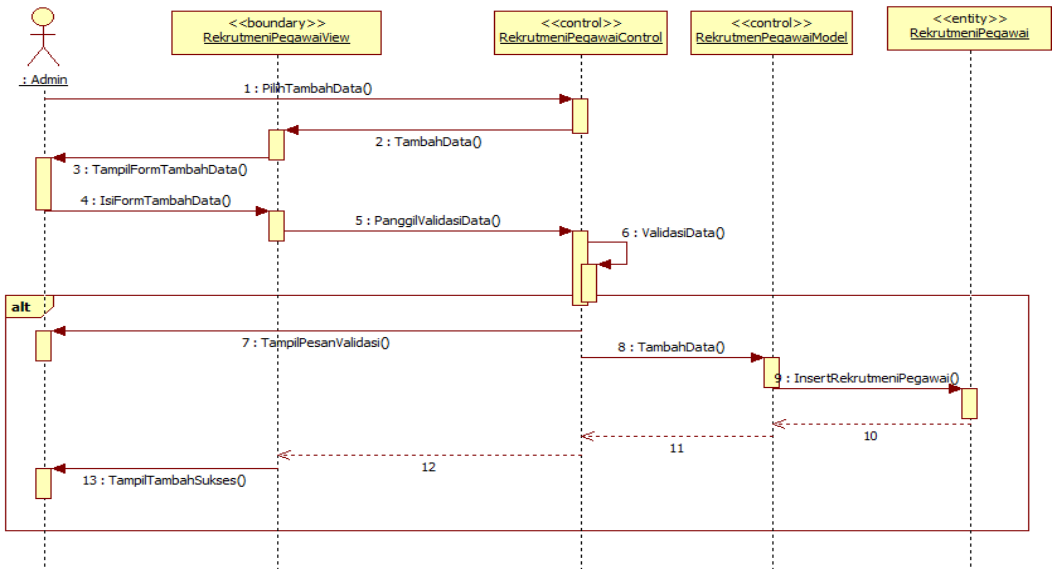
Gambar 3.21 *Sequence Diagram* Tambah Data Pegawai dari Mutasi

3.1.6.4.18 *Sequence Diagram* Pengolahan Rekrutmen Pegawai

Interaksi antara aktor pengguna dengan *use case* pengolahan rekrutmen pegawai dijelaskan dalam *Sequence Diagram* tambah rekrutmen pegawai, ubah rekrutmen pegawai, dan hapus rekrutmen pegawai. Berikut ini penjelasan dari masing-masing *Sequence Diagram* tersebut :

3.1.6.4.19 *Sequence Diagram* Tambah Rekrutmen Pegawai

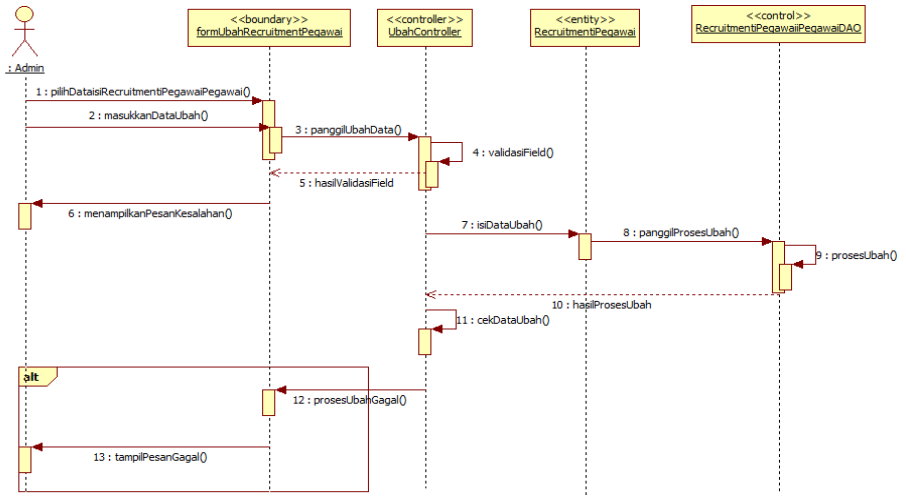
Interaksi antara aktor pengguna dengan *use case* tambah rekrutmen pegawai dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.22 *Sequence Diagram* Tambah Rekrutmen Pegawai

3.1.6.4.20 *Sequence Diagram* Ubah Rekrutmen Pegawai

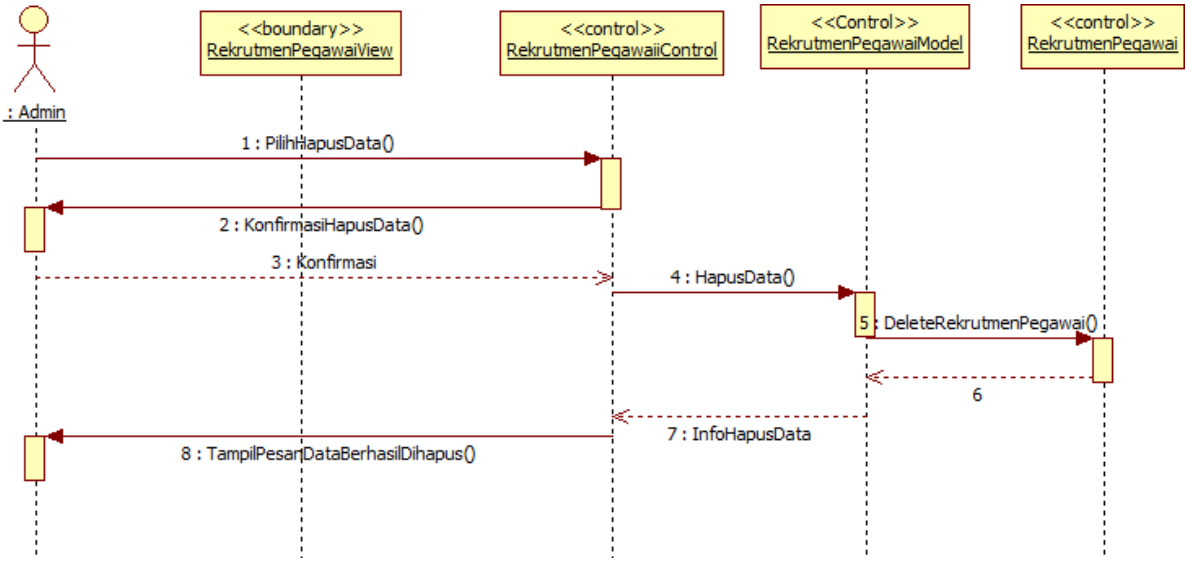
Interaksi antara aktor pengguna dengan *use case* ubah rekrutmen pegawai dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.23 Sequence Diagaram Ubah Rekrutmen Pegawai

3.1.6.4.21 *Sequence Diagram* Hapus Rekrutmen Pegawai

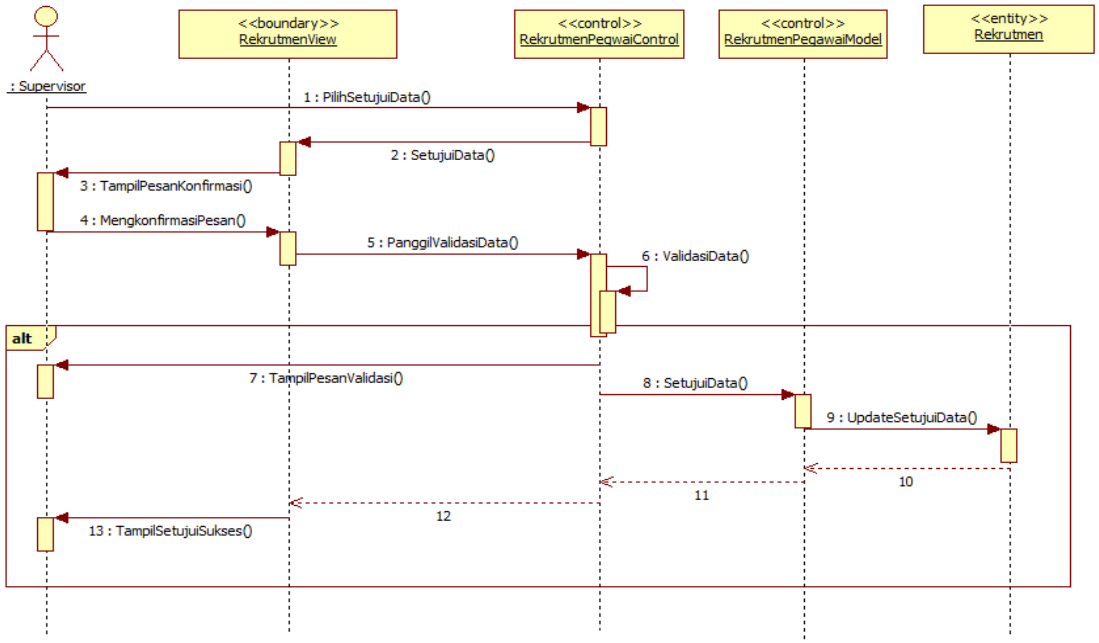
Interaksi antara aktor pengguna dengan *use case* hapus rekrutmen pegawai dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.24 *Sequence Diagram* Hapus Rekrutmen Pegawai

3.1.6.4.22 *Sequence Diagram* Setujui Rekrutmen Pegawai

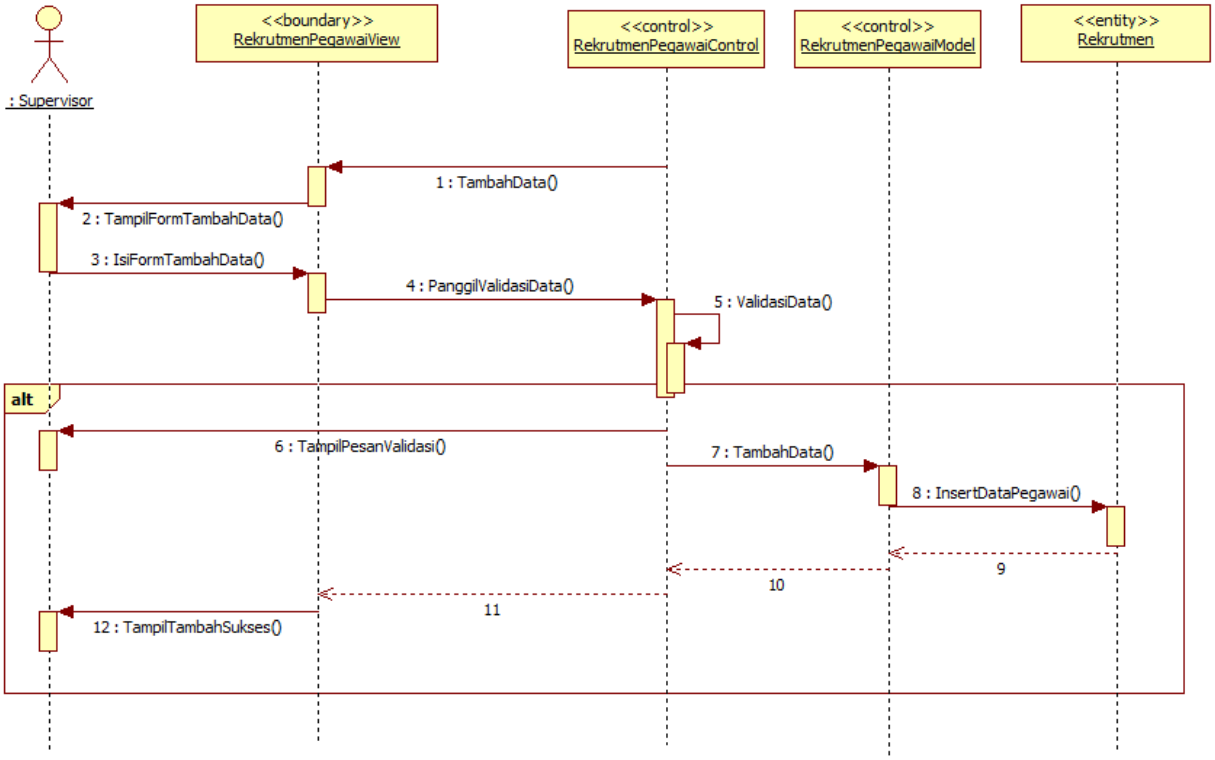
Interaksi antara aktor pengguna dengan *use case* setujui rekrutmen pegawai dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.25 *Sequence Diagram* Setujui Rekrutmen Pegawai

3.1.6.4.23 *Sequence Diagram* Tambah Data Pegawai dari Rekrutmen

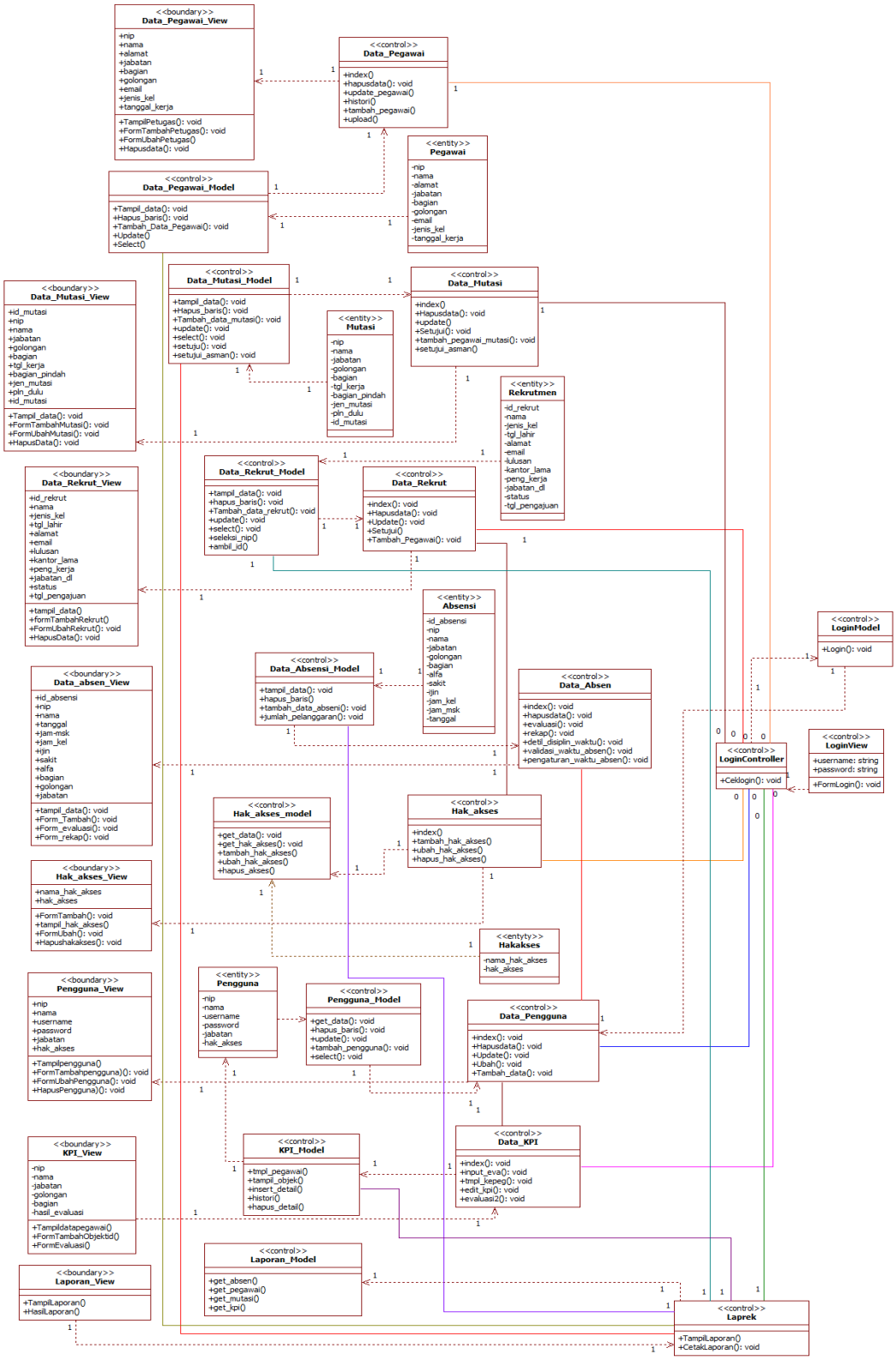
Interaksi antara aktor pengguna dengan *use case* tambah data pegawai dari rekrutmen dijelaskan dalam *Sequence Diagram* sebagai berikut :



Gambar 3.26 *Sequence Diagram* Tambah Data Pegawai dari Rekrutmen

3.1.6.5 *Class Diagram*

Class Diagram menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur itu meliputi atribut-atribut dan *method-method* yang ada pada masing-masing *Class*, sedangkan hubungnnya meliputi pewarisan asosiasi, generalilasi dll. *Class Diagram* dalam sistem informasi kepegawaian di PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung dapat dilihat di gambar 3.7. Setiap kelas jugas memiliki jenis-jenisnya sendiri dapat dilihat di tabel 3.19.



Gambar 3.27 *Class Diagram*

Setiap kelas pada *Class Diagram* gambar 3.26 memiliki jenis kelasnya masing-masing seperti yang terlihat pada tabel 3.20.

Tabel 3-20 Jenis-jenis Kelas

No.	Nama Kelas	Jenis Kelas
1.	Login_*View*	Boundary
2.	Login_Controller	Control
3.	Login_model	Control
4.	Data_Pegawai__*View*	Boundary
5.	Data_Pegawai	Control
6.	Data_Pegawai__model	Control
7.	Data_Rekrutmen__*View*	Boundary
8.	Data_Rekrutmen	Control
9.	Data_Rekrutmen__model	Control
10.	KPI_*View*	Boundary
11.	KPI	Control
12.	KPI_model	Control
13.	Absensi_Pegawai__*View*	Boundary
14.	Data_Absen	Control
15.	Data_Absen _model	Control
16.	Data_Absen __*View*	Boundary
17.	Data_Rekrut	Control
18.	Data_Rekrut_model	Control
19.	LaporanRekapitulasi_*View*	Boundary
20.	LaporanRekapitulasiController	Control
21.	LaporanRekapitulasi_model	Control
22.	HakAkses_*View*	Boundary
23.	HakAksesControl	Control
24.	HakAkses_model	Control
25.	Pengguna_*View*	Boundary
26.	Pengguna	Control
27.	Pengguna_model	Control

Deskripsi dari setiap *Class* yang ada dijelaskan sebagai berikut :

1. *Class* Data_Pegawai

Class Data_Pegawai merupakan *Class* yang digunakan untuk mengatur proses pengolahan pada data pegawai dan bertugas untuk memanggil *Class-Class* yang lainnya dan mengatur alur pada *Class model* maupun *Class View.*

Tabel 3-21 *Class* DataPegawaiController

Nama Operasi/*Method*	**Keterangan**
TampilPegawai	Prosedur yang dipanggil ketika pengguna meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *Class*

Nama Operasi/*Method*	Keterangan
	Data_Pegawai_Model untuk mengambil data pada database.
TambahPegawai	Meminta kelas Data_Pegawai_Model untuk menambah data pegawai dan menyimpannya ke *database* dan meminta kelas Data_Pegawai_*View* untuk menampilkan form Tambah pegawai
UbahPegawai	Meminta kelas Data_Data_Pegawai_Model untuk Mengubah data pegawai dan meminta kelas Data_Pegawai_*View* untuk menampilkan form ubah pegawai
HapusPegawai	Meminta kelas Data_Pegawai_Model untuk Menghapus data pegawai dan meminta kelas Data_Pegawai_*View* untuk menampilkan form hapus pegawai
ValidasiPegawai	Memvalidasi data pegawai yang dimasukan pada form tambah atau ubah, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
HistoriPegawai	Mengambil data pegawai yang diakses dari tabel pegawai dan diakses oleh *Class* data_pegawai_model

2. Class Data_Pegawai_Model

Class Data_Pegawai_Model merupakan *Class* yang berisi prosedur-prosedur untuk mengolah data pegawai. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3-22 *Class* Data_Pegawai_Model

Nama Operasi/*Method*	Keterangan
GetPegawai	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data pegawai.
TambahPegawai	Prosedur yang dipanggil ketika proses tambah data pegawai, melakukan perintah simpan data pada *database*.
UbahPegawai	Prosedur yang dipanggil ketika proses ubah data pegawai, melakukan perintah update data pada *database*.
HapusPegawai	Prosedur yang dipanggil ketika hapus data, menghapus data pegawai pada *database*.

GetHistori	Prosedur yang dipanggil ketika menampilakan histori dari seorang pegawai pada *database*.

3. *Class* Data_Pegawai_*View*

Class Data_Pegawai_*View* merupakan *Class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan pengguna.

Tabel 3-23 *Class* Data_Pegawai_*View*

Nama Operasi/*Method*	**Keterangan**	
TampilPegawai	Menampilkan data pegawai	
FormTambahPegawai	menampilkan form tambah pegawai	
FormUbahPegawai	menampilkan form ubah pegawai	
FormHapusPegawai	menampilkan form hapus pegawai	
Nama Atribut	**Tipe**	**Keterangan**
NIP	Varchar	Primary Key
nama	varchar	
tanggal_lahir	date	
jenis_kel	varchar	
email	varchar	
bagian	varchar	
golongan	varchar	
tanggal_kerja	date	

4. *Class* Data_Rekrutmen

Class Data_Rekrutmen merupakan *Class* yang digunakan untuk mengatur proses pengolahan pada data rekrutmen dan bertugas untuk memanggil *Class-Class* yang lainnya dan mengatur alur pada *Class model* maupun *Class View.*

Tabel 3-24 *Class* Data Rekrutmen

Nama Operasi/*Method*	**Keterangan**
TampilRekrutmen	Prosedur yang dipanggil ketika pengguna meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *Class* Data_Rekrutmen_Model untuk mengambil data pada database.
TambahRekrutmen	Meminta kelas Data_Rekrutmen_Model untuk menambah data rekrutmen dan menyimpannya ke *database* dan meminta kelas Data_Rekrutmen_*View* untuk menampilkan

Nama Operasi/*Method*	Keterangan
	form Tambah rekrutmen
UbahRekrutmen	Meminta kelas Data_Rekrutmen_Model untuk Mengubah data rekrutmen dan meminta kelas Data_Rekrutmen_*View* untuk menampilkan form ubah rekrutmen
HapusRekrutmen	Meminta kelas Data_Rekrutmen_Model untuk Menghapus data rekrutmen dan meminta kelas Data_Rekrutmen_*View* untuk menampilkan form hapus rekrutmen
ValidasiRekrutmen	Memvalidasi data rekrutmen yang dimasukan pada form tambah atau ubah juga setujui, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
SetujuiRekrutmen	Meminta *Class* Data_Rekrutmen_Model untuk merubah data rekrutmen sesuai perubahan yang dilakukan pada saat proses setuju rekrutmen.

5. *Class* Data_Rekrutmen_Model

Class Data_Rekrutmen_Model merupakan *Class* yang berisi prosedur-prosedur untuk mengolah data rekrutmen. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3-25 *Class* Data_Rekrutmen_Model

Nama Operasi/*Method*	Keterangan
GetRekrutmen	Prosedur mengambil data dari *database,* digunakan untuk menampilkan seluruh data rekrutmen.
TambahRekrutmen	Prosedur yang dipanggil ketika proses tambah data rekrutmen, melakukan perintah simpan data pada *database*.
UbahRekrutmen	Prosedur yang dipanggil ketika proses ubah data rekrutmen, melakukan perintah update data pada *database*.
HapusRekrutmen	Prosedur yang dipanggil ketika hapus data, menghapus data rekrutmen pada *database*.
SetujuiRekrutmen	Prosedur yang dipanggil ketika mengubah *database* pada tabel rekrutmen ketika proses setujui rekrutmen.
FormTambahPegawaiRekrutmen	menampilkan form tambah pegawai dari rekrutmen

6. *Class* Data_Rekrutmen_*View*

Class Data_Rekrutmen_*View* merupakan *Class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan pengguna.

Tabel 3-26 *Class* Data_Rekrutmen_*View*

Nama Operasi/*Method*	**Keterangan**	
TampilRekrutmen	Menampilkan data rekrutmen	
FormTambahRekrutmen	menampilkan form tambah rekrutmen	
FormUbahRekrutmen	menampilkan form ubah rekrutmen	
FormHapusRekrutmen	menampilkan form hapus rekrutmen	
Nama Atribut	**Tipe**	**Keterangan**
id_rekrutmen	varchar	primary key
NIP	Varchar	
nama	varchar	
jabatan	varchar	
bagian	varchar	
golongan	varchar	
tanggal_kerja	date	
bagin_pindah	varchar	
tgl_pengajuan	date	
tgl_setujui	date	

7. *Class* Data_Rekrutmen

Class Data_Rekrutmen merupakan *Class* yang digunakan untuk mengatur proses pengolahan pada data rekrutmen dan bertugas untuk memanggil *Class-Class* yang lainnya dan mengatur alur pada *Class model* maupun *Class View.*

Tabel 3-27 *Class* Data_Rekrutmen

Nama Operasi/*Method*	**Keterangan**
TampilRekrutmen	Prosedur yang dipanggil ketika pengguna meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *Class* Data_Rekrutmen_Model untuk mengambil data pada database.
TambahRekrutmen	Meminta kelas Data_Rekrutmen_Model untuk menambah data rekrutmen dan menyimpannya ke *database* dan meminta kelas Data_Rekrutmen_*View* untuk menampilkan form Tambah rekrutmen
UbahRekrutmen	Meminta kelas Data_Rekrutmen_Model untuk

Nama Operasi/*Method*	Keterangan
	Mengubah data rekrutmen dan meminta kelas Data_Rekrutmen_*View* untuk menampilkan form ubah rekrutmen
HapusRekrutmen	Meminta kelas Data_Rekrutmen_Model untuk Menghapus data rekrutmen dan meminta kelas Data_Rekrutmen_*View* untuk menampilkan form hapus rekrutmen
ValidasiRekrutmen	Memvalidasi data rekrutmen yang dimasukan pada form tambah atau ubah juga setujui, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
SetujuiRekrutmen	Meminta *Class* Data_Rekrutmen_Model untuk merubah data rekrutmen sesuai perubahan yang dilakukan pada saat proses setuju rekrutmen.

8. *Class* Data_Rekrutmen_Model

Class Data_Rekrutmen_Model merupakan *Class* yang berisi prosedur-prosedur untuk mengolah data rekrutmen. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3-28 *Class* Data_Rekrutmen_Model

Nama Operasi/*Method*	Keterangan
GetRekrutmen	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data rekrutmen.
TambahRekrutmen	Prosedur yang dipanggil ketika proses tambah data rekrutmen, melakukan perintah simpan data pada *database*.
UbahRekrutmen	Prosedur yang dipanggil ketika proses ubah data rekrutmen, melakukan perintah update data pada *database*.
HapusRekrutmen	Prosedur yang dipanggil ketika hapus data, menghapus data rekrutmen pada *database*.
SetujuiRekrutmen	Prosedur yang dipanggil ketika mengubah *database* pada tabel rekrutmen ketika proses setujui rekrutmen.

9. *Class* Data_Rekrutmen_*View*

Class Data_Rekrutmen_*View* merupakan *Class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan pengguna.

Tabel 3-29 *Class* Data_Rekrutmen_*View*

Nama Operasi/*Method*	**Keterangan**	
TampilRekrutmen	Menampilkan data rekrutmen	
FormTambahRekrutmen	menampilkan form tambah rekrutmen	
FormUbahRekrutmen	menampilkan form ubah rekrutmen	
FormHapusRekrutmen	menampilkan form hapus rekrutmen	
FormSetujuiRekrutmen	menampilkan form setujui pegawai	
FormTambahPegawaiRekrutmen	menampilkan form tambah pegawai dari rekrutmen	
Nama Atribut	**Tipe**	**Keterangan**
id_rekrutmen	varchar	primary key
nama	varchar	
tgl_lahir	date	
alamat	varchar	
email	varchar	
lulusan	varchar	
tempat_berkerja_dulu	varchar	
jabatan_dahulu	varchar	
pengalaman_berkerja	varchar	

10. *Class* Pengguna

Class Data_Pengguna merupakan *Class* yang digunakan untuk mengatur proses pengolahan pada data pengguna dan bertugas untuk memanggil *Class-Class* yang lainnya dan mengatur alur pada *Class model* maupun *Class View.*

Tabel 3-30 *Class* Data_Pengguna

Nama Operasi/*Method*	**Keterangan**
TampilPengguna	Prosedur yang dipanggil ketika pengguna meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *Class* Data_Pengguna_Model untuk mengambil data pada database.
TambahPengguna	Meminta kelas Data_Pengguna_Model untuk menambah data pengguna dan menyimpannya ke *database* dan meminta kelas Data_Pengguna_*View* untuk menampilkan form Tambah pengguna
UbahPengguna	Meminta kelas Data_Pengguna_Model untuk Mengubah data pengguna dan meminta kelas Data_Pengguna_*View* untuk menampilkan form ubah pengguna
HapusPengguna	Meminta kelas Data_Pengguna_Model untuk

Nama Operasi/*Method*	Keterangan
	Menghapus data pengguna dan meminta kelas Data_Pengguna_*View* untuk menampilkan form hapus pengguna
ValidasiPengguna	Memvalidasi data pengguna yang dimasukan pada form tambah atau ubah juga setujui, akan menampilkan kesalahan jika data yang dimasukan tidak valid.

11. *Class* Pengguna_Model

Class Data_Pengguna_Model merupakan *Class* yang berisi prosedur-prosedur untuk mengolah data pengguna. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3-31 *Class* Data_Pengguna_Model

Nama Operasi	Keterangan
GetPengguna	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data pengguna.
TambahPengguna	Prosedur yang dipanggil ketika proses tambah data pengguna, melakukan perintah simpan data pada *database*.
UbahPengguna	Prosedur yang dipanggil ketika proses ubah data pengguna, melakukan perintah update data pada *database*.
HapusPengguna	Prosedur yang dipanggil ketika hapus data, menghapus data pengguna pada *database*.

12. *Class* Pengguna_*View*

Class Data_pengguna_*View* merupakan *Class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan pengguna.

Tabel 3-32 *Class* Data_Pengguna_*View*

Nama Operasi/*Method*	Keterangan	
TampilPengguna	Menampilkan data pengguna	
FormTambahPengguna	menampilkan form tambah pengguna	
FormUbahPengguna	menampilkan form ubah pengguna	
FormHapusPengguna	menampilkan form hapus pengguna	
Nama Atribut	**Tipe**	**Keterangan**
NIP	Varchar	Primary Key
nama	varchar	
username	varchar	

Nama Operasi/*Method*	Keterangan	
password	varchar	
jabtan	varchar	
hak akses	varchar	

13. *Class* Kpi

Class Data_Kpi merupakan *Class* yang digunakan untuk mengatur proses pengolahan pada data kpi dan bertugas untuk memanggil *Class-Class* yang lainnya dan mengatur alur pada *Class model* maupun *Class View*.

Tabel 3-33 *Class* Data_Kpi

Nama Operasi/*Method*	Keterangan
TampilKpi	Prosedur yang dipanggil ketika pengguna meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *Class* Data_Kpi_Model untuk mengambil data pada database.
TambahKpi	Meminta kelas Data_Kpi_Model untuk menambah data kpi dan menyimpannya ke *database* dan meminta kelas Data_Kpi_*View* untuk menampilkan form Tambah kpi
UbahKpi	Meminta kelas Data_Kpi_Model untuk Mengubah data kpi dan meminta kelas Data_Kpi_*View* untuk menampilkan form ubah kpi
HapusKpi	Meminta kelas Data_Kpi_Model untuk Menghapus data kpi dan meminta kelas Data_Kpi_*View* untuk menampilkan form hapus kpi
ValidasiKpi	Memvalidasi data kpi yang dimasukan pada form tambah atau ubah juga setujui, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
EvluasiKpi	Meminta *Class* Data_Kpi_Model untuk merubah data kpi sesuai perubahan yang dilakukan pada saat proses setuju kpi.

14. *Class* Kpi_Model

Class Data_Kpi_Model merupakan *Class* yang berisi prosedur-prosedur untuk mengolah data kpi. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3-34 *Class* Data_Kpi_Model

Nama Operasi/*Method*	Keterangan
GetKpi	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data kpi.
TambahKpi	Prosedur yang dipanggil ketika proses tambah data kpi, melakukan perintah simpan data pada *database*.
UbahKpi	Prosedur yang dipanggil ketika proses ubah data kpi, melakukan perintah update data pada *database*.
HapusKpi	Prosedur yang dipanggil ketika hapus data, menghapus data kpi pada *database*.
evaluasiKpi	Prosedur yang dipanggil ketika mengubah *database* pada tabel kpi ketika proses evaluasi kpi.

15. *Class* Kpi_*View*

Class Data_Kpi_Model merupakan *Class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan pengguna.

Tabel 3-35 *Class* Data_Kpi_*View*

Nama Operasi	**Keterangan**	
TampilKpi	Menampilkan data kpi	
FormTambahKpi	menampilkan form tambah kpi	
FormUbahKpi	menampilkan form ubah kpi	
FormHapusKpi	menampilkan form hapus kpi	
FormEvaluasiKpi	menampilkan form evaluasi kpi	
Nama Atribut	**Tipe**	**Keterangan**
NIP	Varchar	Primary Key
nama	varchar	
jabatan	varchar	
bagian	varchar	
golongan	varchar	
hasil evaluasi	varchar	
jumlah objektif	varchar	
jumlah bobot	varchar	

16. *Class* Absen_Pegawai

Class Absen_pegawai_Model merupakan *Class* yang berisi prosedur-prosedur untuk mengolah data absen_pegawai. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3-36 *Class* Absen_pegawai

Nama Operasi/*Method*	Keterangan
TampilAbsen_pegawai	Prosedur yang dipanggil ketika pengguna meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *Class* Absen_pegawai_Model untuk mengambil data pada database.
TambahAbsen_pegawai	Meminta kelas Absen_pegawai_Model untuk menambah data absen_pegawai dan menyimpannya ke *database* dan meminta kelas Absen_pegawai_*View* untuk menampilkan form Tambah absen_pegawai
UbahAbsen_pegawai	Meminta kelas Absen_pegawai_Model untuk Mengubah data absen_pegawai dan meminta kelas Absen_pegawai_*View* untuk menampilkan form ubah absen_pegawai
HapusAbsen_pegawai	Meminta kelas Absen_pegawai_Model untuk Menghapus data absen_pegawai dan meminta kelas Absen_pegawai_*View* untuk menampilkan form hapus absen_pegawai
ValidasiAbsen_pegawai	Memvalidasi data absen_pegawai yang dimasukan pada form tambah atau ubah juga setujui, akan menampilkan kesalahan jika data yang dimasukan tidak valid.
evaluasiAbsen_pegawai	Meminta *Class* Absen_pegawai_Model untuk merubah data absen_pegawai sesuai perubahan yang dilakukan pada saat proses evaluasi absen_pegawai.

17. *Class* Absen_Pegawai_Model

Class Data_Absen_pegawai_Model merupakan *Class* yang berisi prosedur-prosedur untuk mengolah data absen_pegawai. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3-37 *Class* Data_Absen_pegawai_Model

Nama Operasi/*Method*	Keterangan
GetAbsen_pegawai	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data absen_pegawai.
TambahAbsen_pegawai	Prosedur yang dipanggil ketika proses tambah data absen_pegawai, melakukan perintah simpan data pada *database*.
UbahAbsen_pegawai	Prosedur yang dipanggil ketika proses ubah data absen_pegawai, melakukan perintah update data pada *database*.
HapusAbsen_pegawai	Prosedur yang dipanggil ketika hapus data, menghapus data absen_pegawai pada *database*.
EvaluasiAbsen_pegawai	Prosedur yang dipanggil ketika mengubah *database* pada tabel absen_pegawai ketika proses evaluasi absen_pegawai.

18. *Class* Absen_Pegawai_*View*

Class Data_absen_Pegawai *_View* merupakan *Class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan pengguna.

Tabel 3-38 *Class* Data_Rekrutmen_*View*

Nama Operasi/*Method*	Keterangan	
TampilAbsen	Menampilkan data absen	
FormTambahketerangan	menampilkan form tambah absen	
FormUbahketerangan	menampilkan form ubah absen	
FormHapusketerangan	menampilkan form hapus absen	
FormEvaluasiketerangan	menampilkan form evaluasi pegawai	
Nama Atribut	**Tipe**	**Keterangan**
id_absen	varchar	primary key
nama	varchar	
tgl_lahir	date	
alamat	varchar	
email	varchar	
lulusan	varchar	
tempat_berkerja_dulu	varchar	
jabatan_dahulu	varchar	
pengalaman_berkerja	varchar	

19. *Class* Laprek

Class Laprek_Model merupakan *Class* yang berisi prosedur-prosedur untuk mengolah data rekrutmen. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3-39 *Class* Laprek

Nama Operasi/*Method*	Keterangan
TampilLaporan	Prosedur untuk melakukan pembuatan laporan form yang sudah diisi sebelumnya dan kemudian menampilkannya.
CetakLaporan	Prosedur untuk mencetak laporan yang sudah dibuat.

20. *Class* Laprek_*View*

Class Laprek_*View* merupakan *Class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan pengguna.

Tabel 3-40 *Class* Laprek_*View*

Nama Operasi/*Method*	Keterangan
TampilLaporan	Prosedur untuk melakukan pembuatan laporan form yang sudah diisi sebelumnya dan kemudian menampilkannya.
CetakLaporan	Prosedur untuk mencetak laporan yang sudah dibuat.

21. *Class* Laprek_Model

Class Laprek_Model merupakan *Class* yang berisi prosedur-prosedur untuk mengolah data rekrutmen. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3-41 *Class* Laprek_Model

Nama Operasi	Keterangan
GetLaporan	Prosedur mengambil data dari database, digunakan untuk menampilkan seluruh data berdasarkan jangka waktu yang dimasukan.
CetakLaporan	Prosedur yang dipanggil ketika proses cetak laporan.

22. *Class* Hak_Akses

Class Hak_akses merupakan *Class* yang berisi prosedur-prosedur untuk mengolah data hak_akses. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3-42 *Class* Data_Hak_akses

Nama Operasi/*Method*	Keterangan
TampilHak_akses	Prosedur yang dipanggil ketika pengguna meminta untuk menampilkan data, prosedur ini akan memberi perintah kepada *Class* Data_Hak_akses_Model untuk mengambil data pada database.
TambahHak_akses	Meminta kelas Data_Hak_akses_Model untuk menambah data hak_akses dan menyimpannya ke *database* dan meminta kelas Data_Hak_akses_*View* untuk menampilkan form Tambah hak_akses
UbahHak_akses	Meminta kelas Data_Hak_akses_Model untuk Mengubah data hak_akses dan meminta kelas Data_Hak_akses_*View* untuk menampilkan form ubah hak_akses
HapusHak_akses	Meminta kelas Data_Hak_akses_Model untuk Menghapus data hak_akses dan meminta kelas Data_Hak_akses_*View* untuk menampilkan form hapus hak_akses
ValidasiHak_akses	Memvalidasi data hak_akses yang dimasukan pada form tambah atau ubah juga setujui, akan menampilkan kesalahan jika data yang dimasukan tidak valid.

23. *Class* Hak_Akses_*View*

Class Hak_Akses_*View* merupakan *Class* yang berisi prosedur-prosedur untuk mengatur tampilan yang berinteraksi langsung dengan pengguna.

Tabel 3-43 *Class* Hak_Akses_*View*

Nama Operasi/*Method*	Keterangan
TampilHak_akses	Menampilkan data hak_akses
FormTambahHak_akses	menampilkan form tambah hak_akses

FormUbahHak_akses	menampilkan form ubah hak_akses	
FormHapusHak_akses	menampilkan form hapus hak_akses	
FormSetujuiHak_akses	menampilkan form setujui pegawai	
FormTambahPegawaiHak_akses	menampilkan form tambah pegawai dari hak_akses	
Nama Atribut	**Tipe**	**Keterangan**
id_hak_akses	varchar	primary key
nama_hak_akses	varchar	
hak_tambah_data	date	
hak_ubah_data	varchar	
hak_hapus_data	varchar	
hak_setujui_data	varchar	

24. *Class* Hak_Akses_Model

Class Hak_akses_Model merupakan *Class* yang berisi prosedur-prosedur untuk mengolah data rekrutmen. *Class* ini berhubungan langsung dengan *database* untuk proses mengambil, menyimpan dan menghapus data.

Tabel 3-44 *Class* Hak_akses_Model

Nama Operasi/*Method*	Keterangan
GetHak_akses	Prosedur mengambil data dari *database*, digunakan untuk menampilkan seluruh data hak_akses.
TambahHak_akses	Prosedur yang dipanggil ketika proses tambah data hak_akses, melakukan perintah simpan data pada *database*.
UbahHak_akses	Prosedur yang dipanggil ketika proses ubah data hak_akses, melakukan perintah update data pada *database*.
HapusHak_akses	Prosedur yang dipanggil ketika hapus data, menghapus data hak_akses pada *database*.

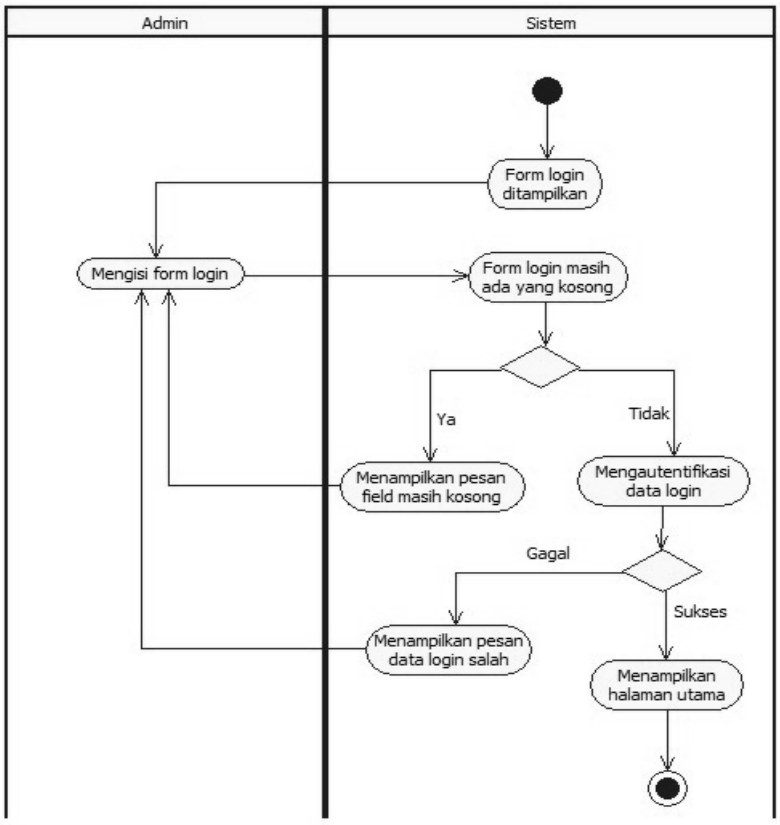
3.1.6.6 *Activity Diagram*

Activity Diagram memodelkan aliran kerja atau *workflow* dari urutan aktifitas dalam suatu proses yang mengacu pada *Use Case Diagram* yang ada. *Activity Diagram* pada aplikasi pengolahan data *kepegawaian* terdiri dari *Activity Diagram* mengolah data rekrutmen , *Activity Diagram* mengolah data pegawai,

Activity Diagram mengolah data rekrutmen, *Activity Diagram* mencetak laporan, data kpi . Berikut ini penjelasan dengan *Activity Diagram* untuk penjelasan dari masing-masing *Use Case Scenario*:

3.1.6.6.1 *Activity Diagram* Login Admin

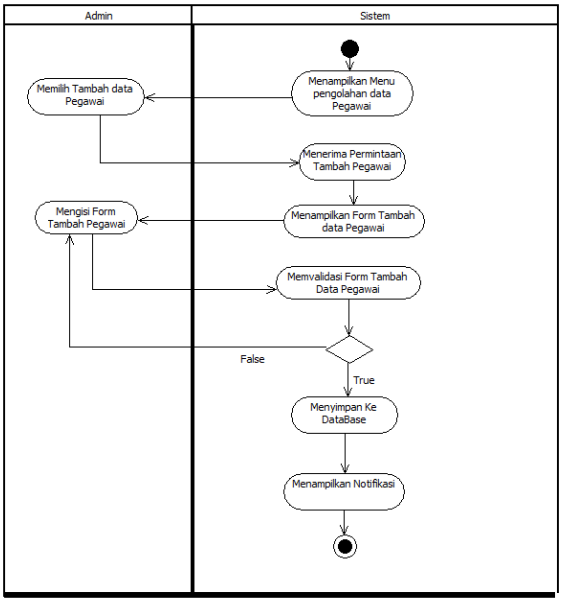
Interaksi antara aktor pengguna dengan *use case* login dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.28 Acticity Diagram Login Admin

3.1.6.6.2 *Activity Diagram* Tambah Data Pegawai

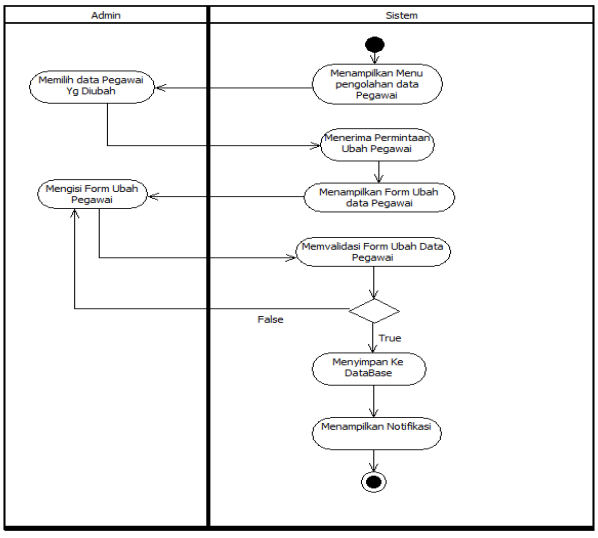
Interaksi antara aktor pengguna dengan *use case* tambah data pegawai dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.29 *Activity Diagram* Tambah Data Pegawai

3.1.6.6.3 *Activity Diagram* Ubah Data Pegawai

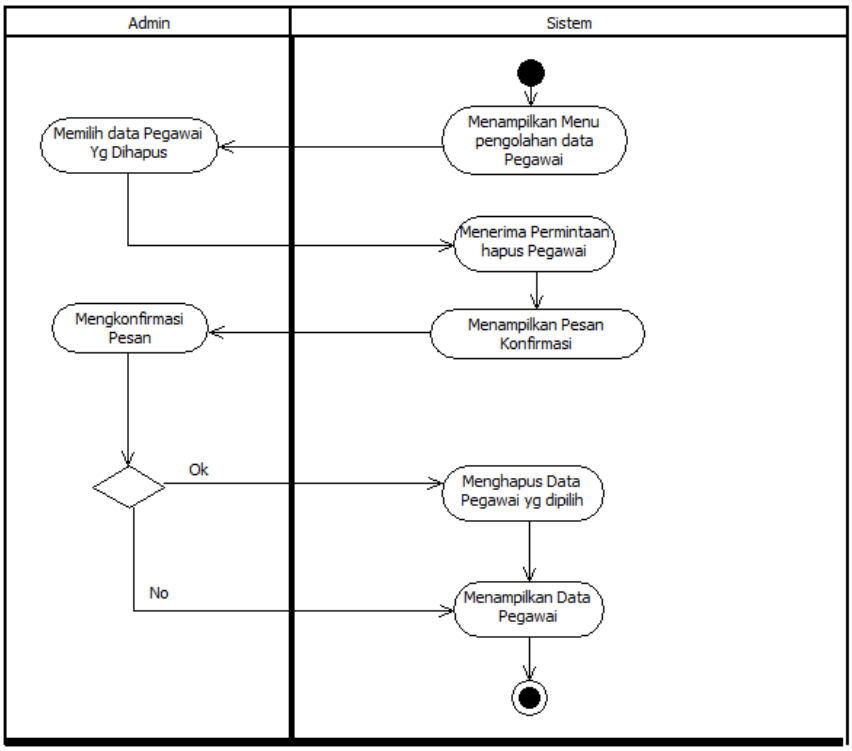
Interaksi antara aktor pengguna dengan *use case* ubah data pegawai dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.30 *Activity Diagram* Ubah Data Pegawai

3.1.6.6.4 *Activity Diagram* Hapus Data Pegawai

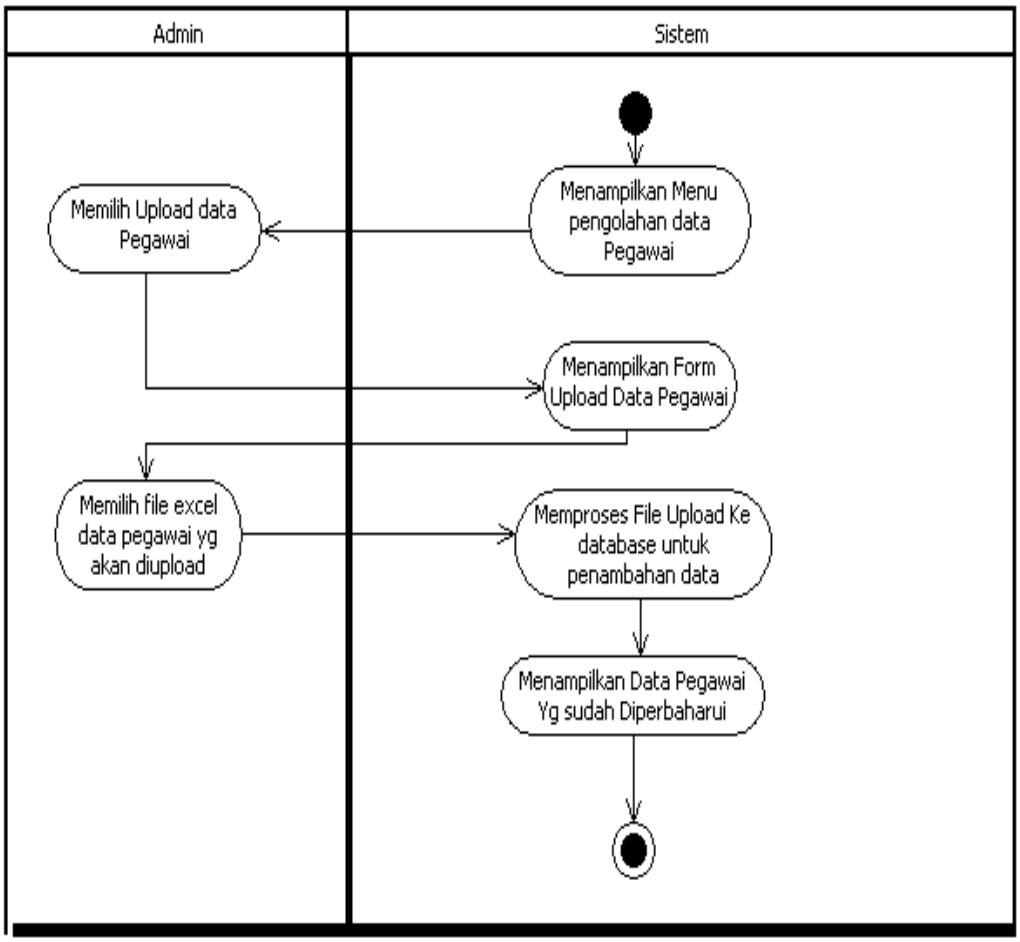
Interaksi antara aktor pengguna dengan *use case* hapus data pegawai dijelaskan dalam *Activity Diagram* sebagai berikut



Gambar 3.31 *Activity Diagram* Hapus Data Pegawai

3.1.6.6.5 *Activity Diagram* Upload Data Pegawai

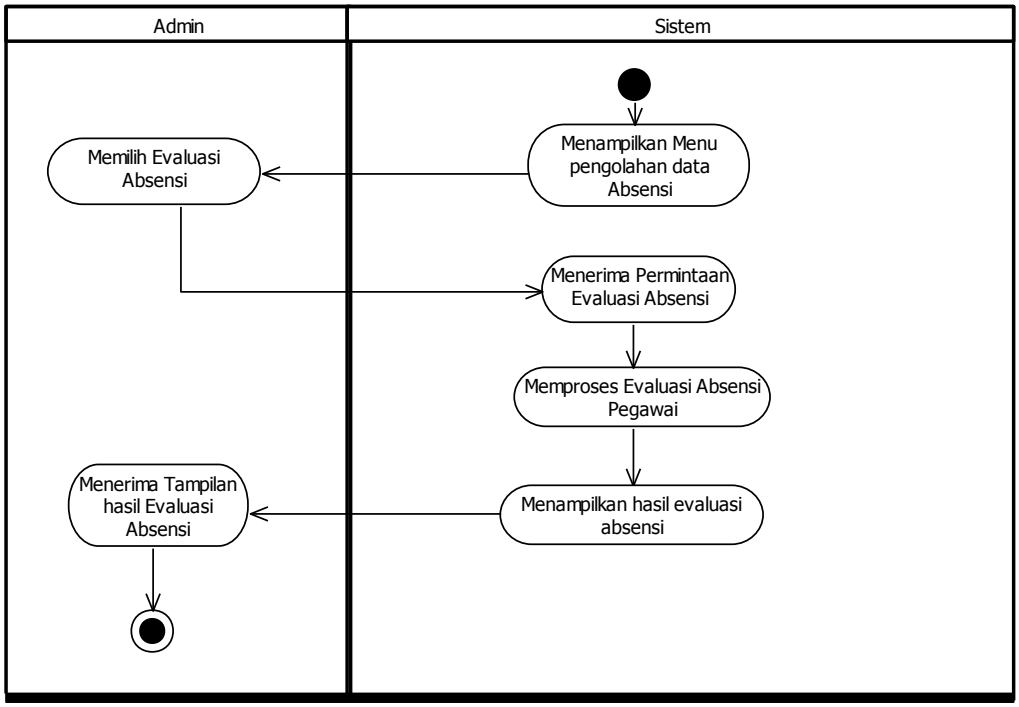
Interaksi antara aktor pengguna dengan *use case* upload data pegawai dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.32 *Activity Diagram* Upload Data Pegawai

3.1.6.6.6 *Activity Diagram* Evaluasi Absensi Pegawai

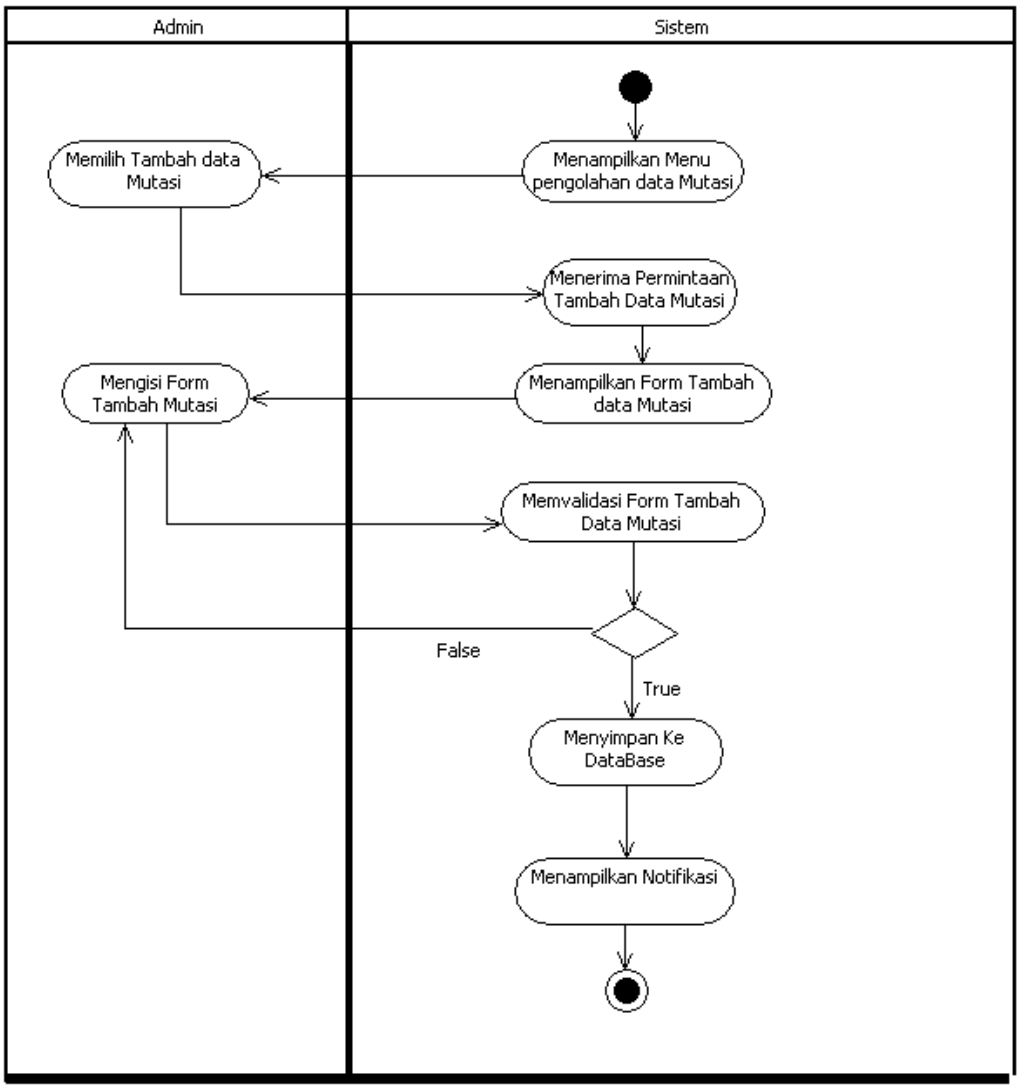
Interaksi antara aktor pengguna dengan *use case* evaluasi absensi pegawai dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.33 *Activity Diagram* Evaluasi Absensi

3.1.6.6.7 *Activity Diagram* Tambah Rekrutmen Pegawai

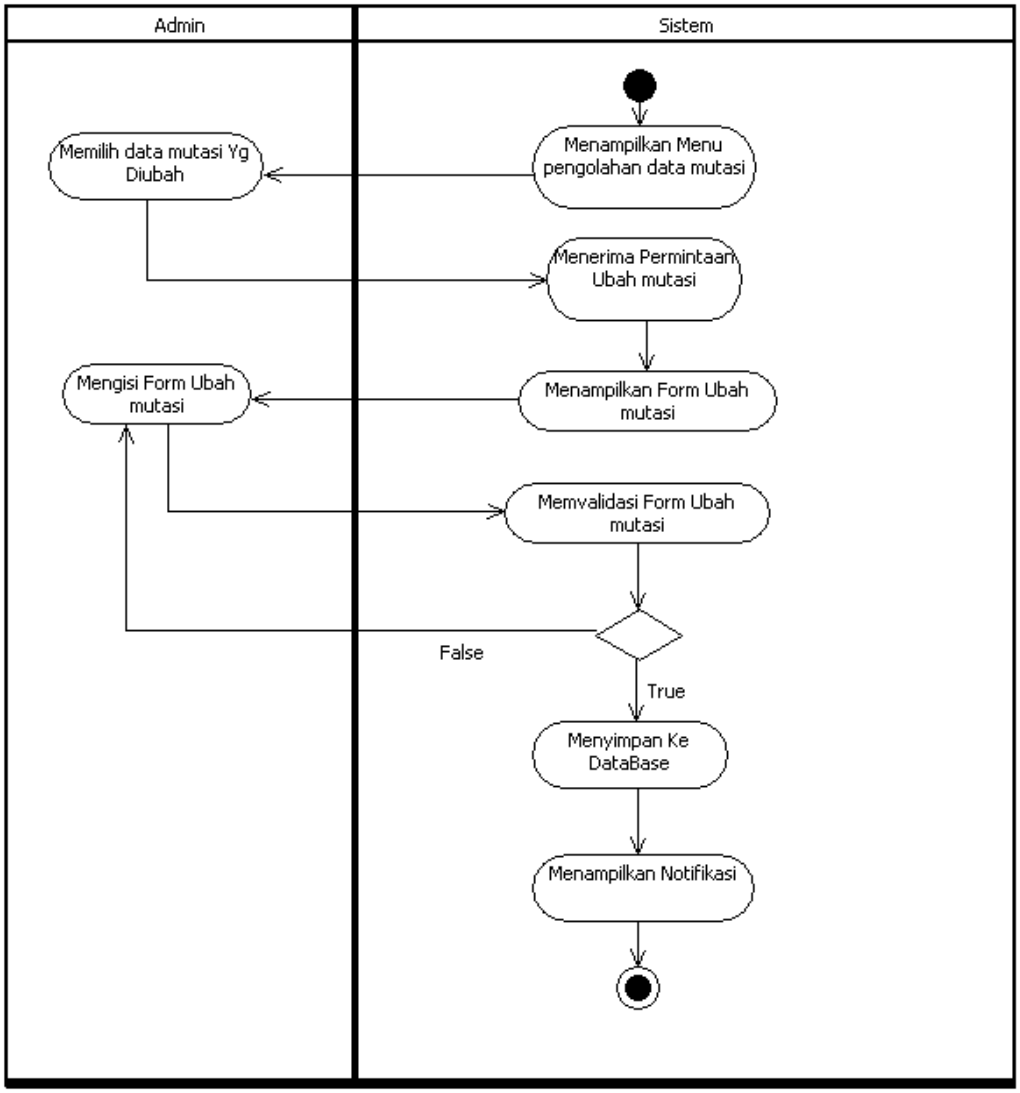
Interaksi antara aktor pengguna dengan *use case* tambah rekrutmen pegawai dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.34 *Activity Diagram* Tambah Rekrutmen Pegawai

3.1.6.6.8 *Activity Diagram* Ubah Rekrutmen Pegawai

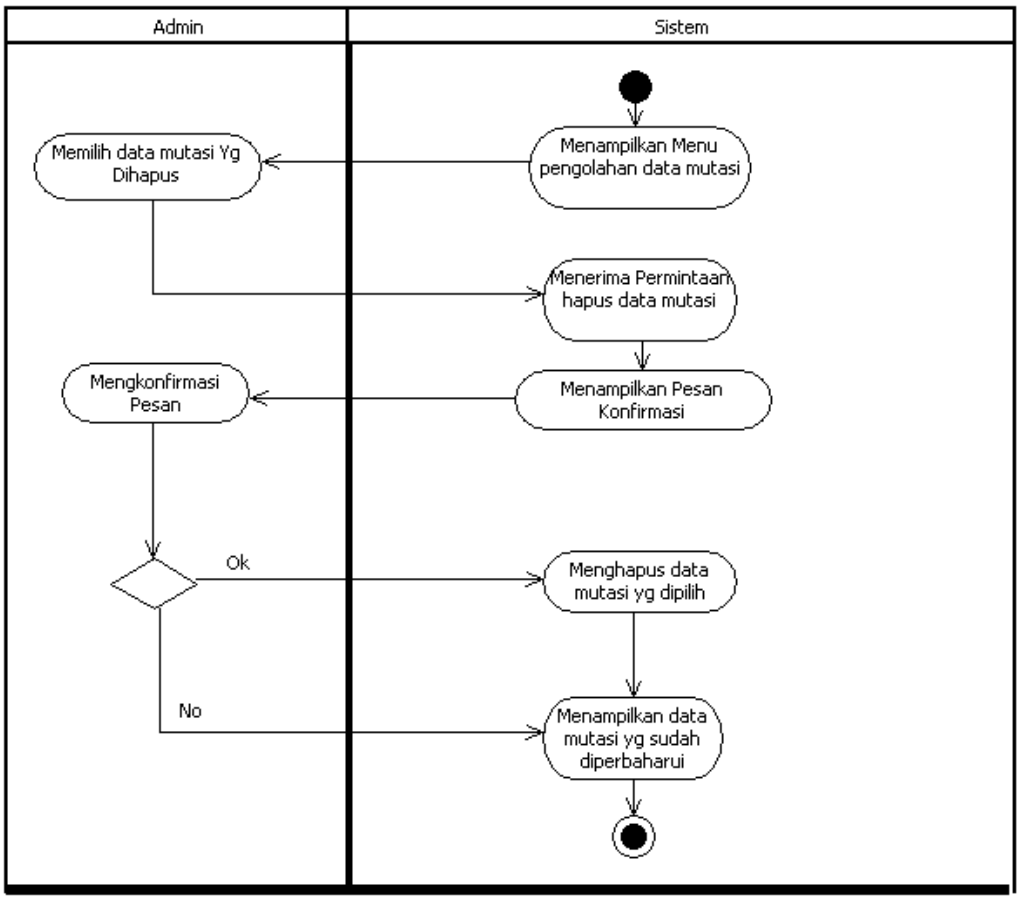
Interaksi antara aktor pengguna dengan *use case* ubah rekrutmen pegawai dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.35 ***Activity Diagram*** **Ubah Data Rekrutmen Pegawai**

3.1.6.6.9 *Activity Diagram* Hapus Rekrutmen Pegawai

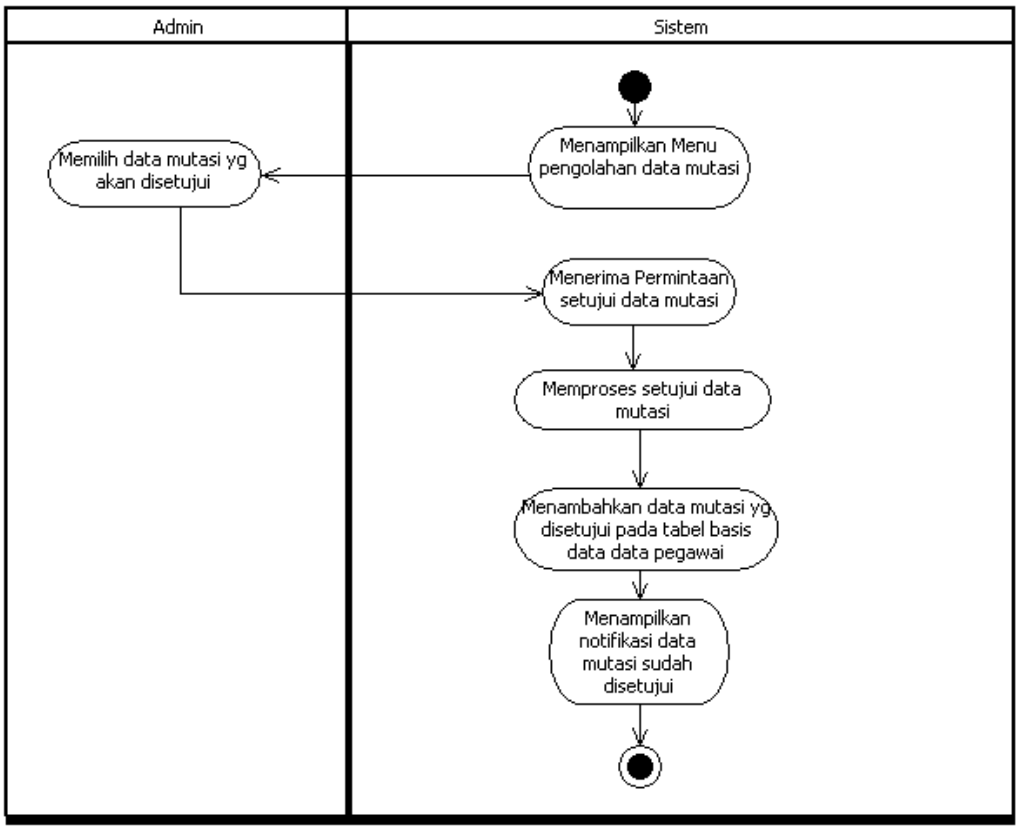
Interaksi antara aktor pengguna dengan *use case* hapus rekrutmen pegawai dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.36 *Activity Diagram* Hapus Rekrutmen Pegawai

3.1.6.6.10 *Activity Diagram* Setujui Rekrutmen Pegawai

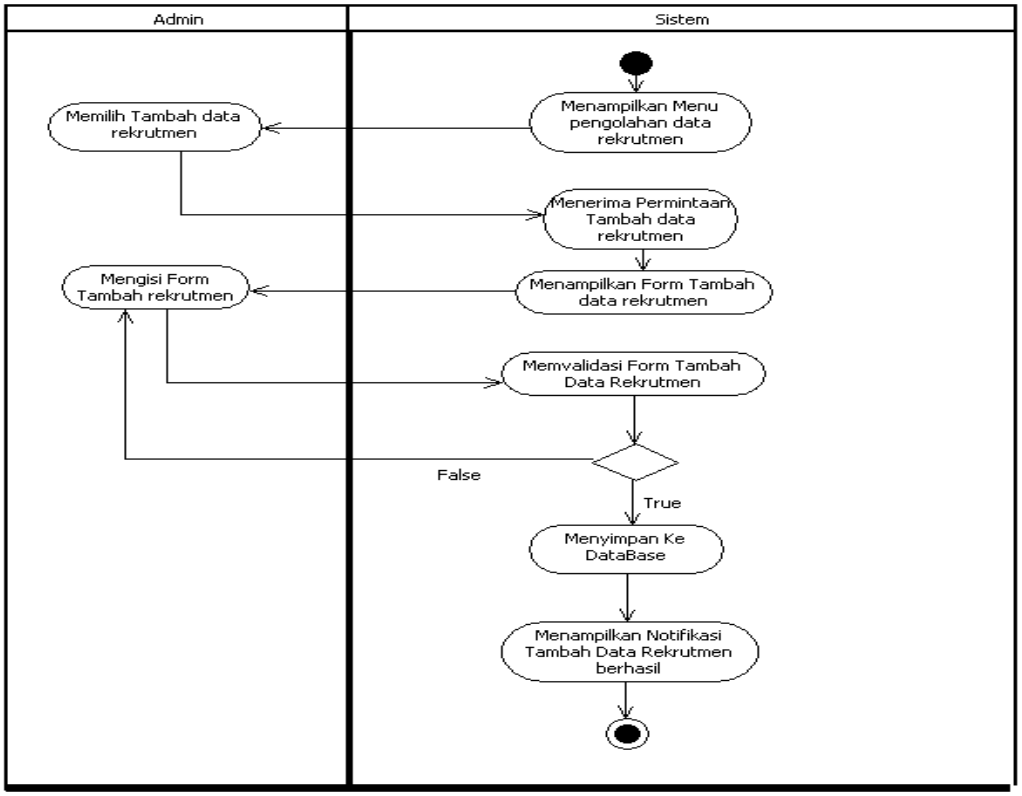
Interaksi antara aktor pengguna dengan *use case* Setujui Rekrutmen Pegawai dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.37 *Activity Diagram* Setujui Data Rekrutmen

3.1.6.6.11 *Activity Diagram* Tambah Rekrutmen Pegawai

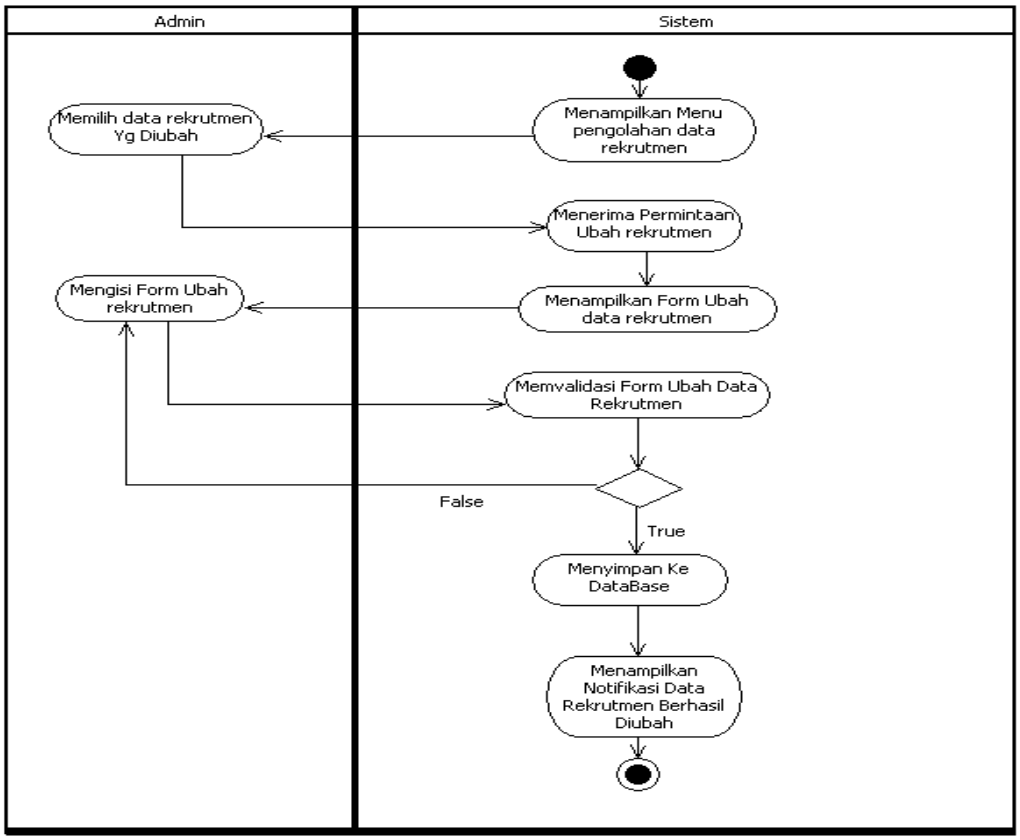
Interaksi antara aktor pengguna dengan *use case* tambah rekrutmen pegawai dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.38 ***Activity Diagram*** **Tambah Data Rekrutmen Pegawai**

3.1.6.6.12 *Activity Diagram* Ubah Rekrutmen Pegawai

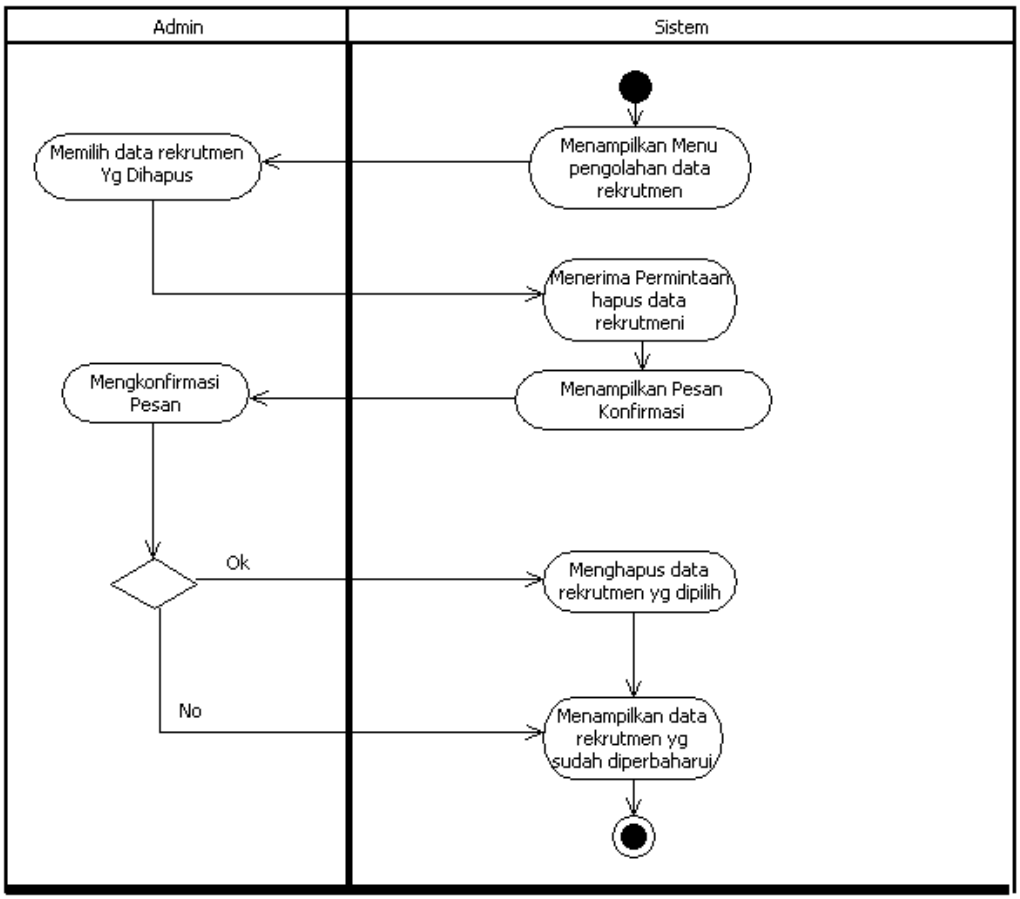
Interaksi antara aktor pengguna dengan *use case* ubah rekrutmen pegawai dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.39 *Activity Diagram* Ubah Data Rekrutmen

3.1.6.6.13 *Activity Diagram* Hapus Rekrutmen Pegawai

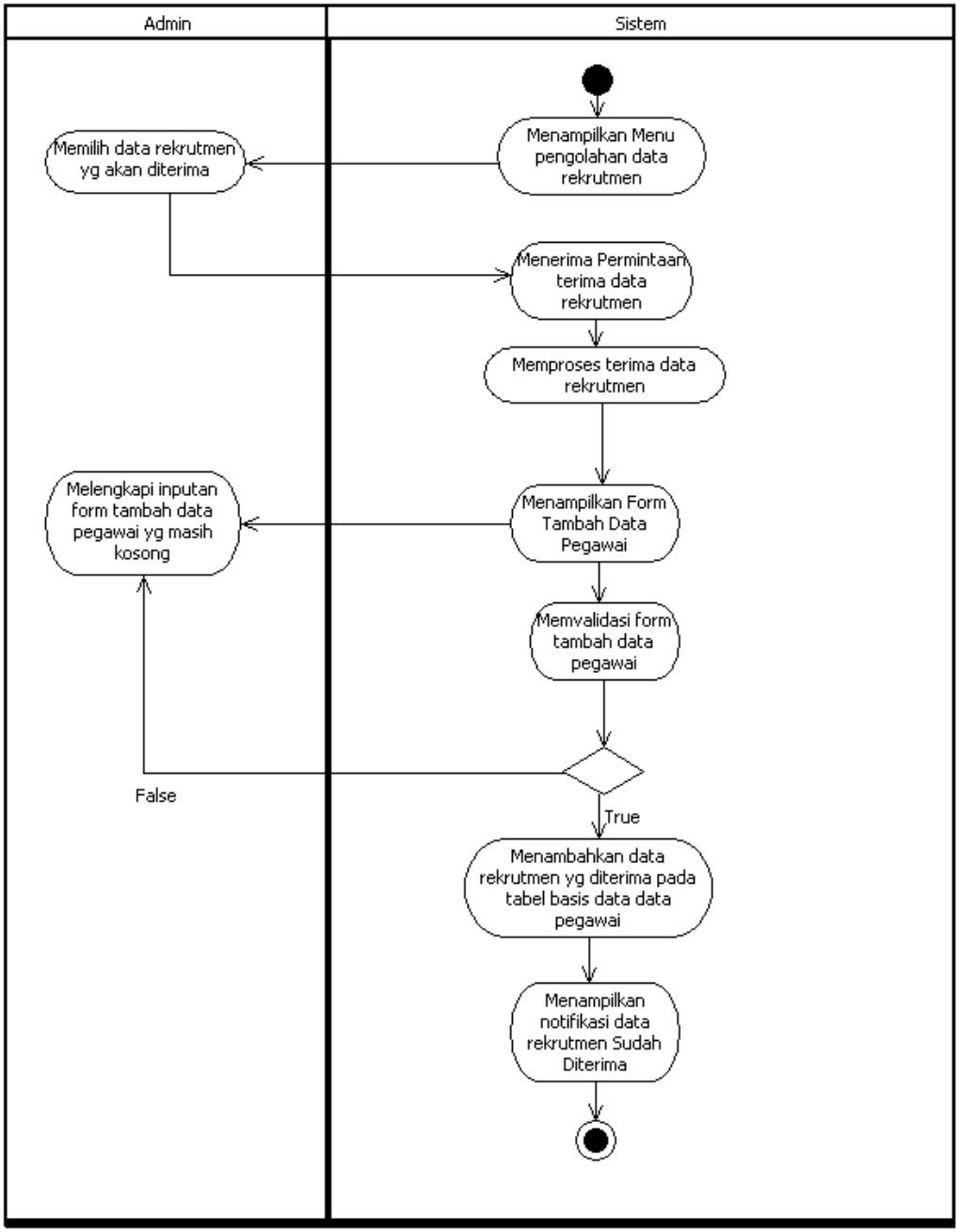
Interaksi antara aktor pengguna dengan *use case* hapus rekrutmen pegawai dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.40 *Activity Diagram* Hapus Data Rekrutmen Pegawai

3.1.6.6.14 *Activity Diagram* Setujui Rekrutmen Pegawai

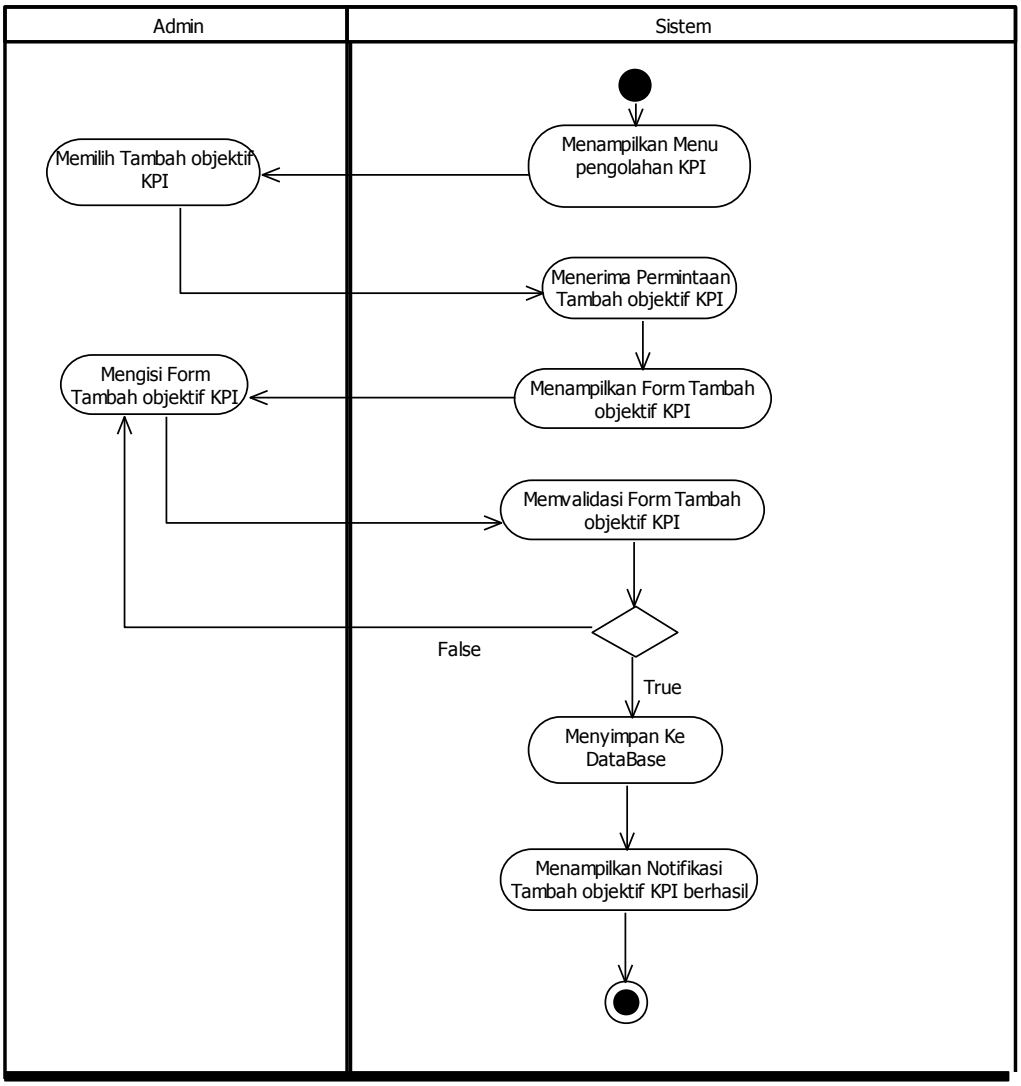
Interaksi antara aktor pengguna dengan *use case* terima rekrutmen pegawai dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.41 *Activity Diagram* Setujui Data Rekrutmen Pegawai

3.1.6.6.15 *Activity Diagram* Tambah Objektif KPI

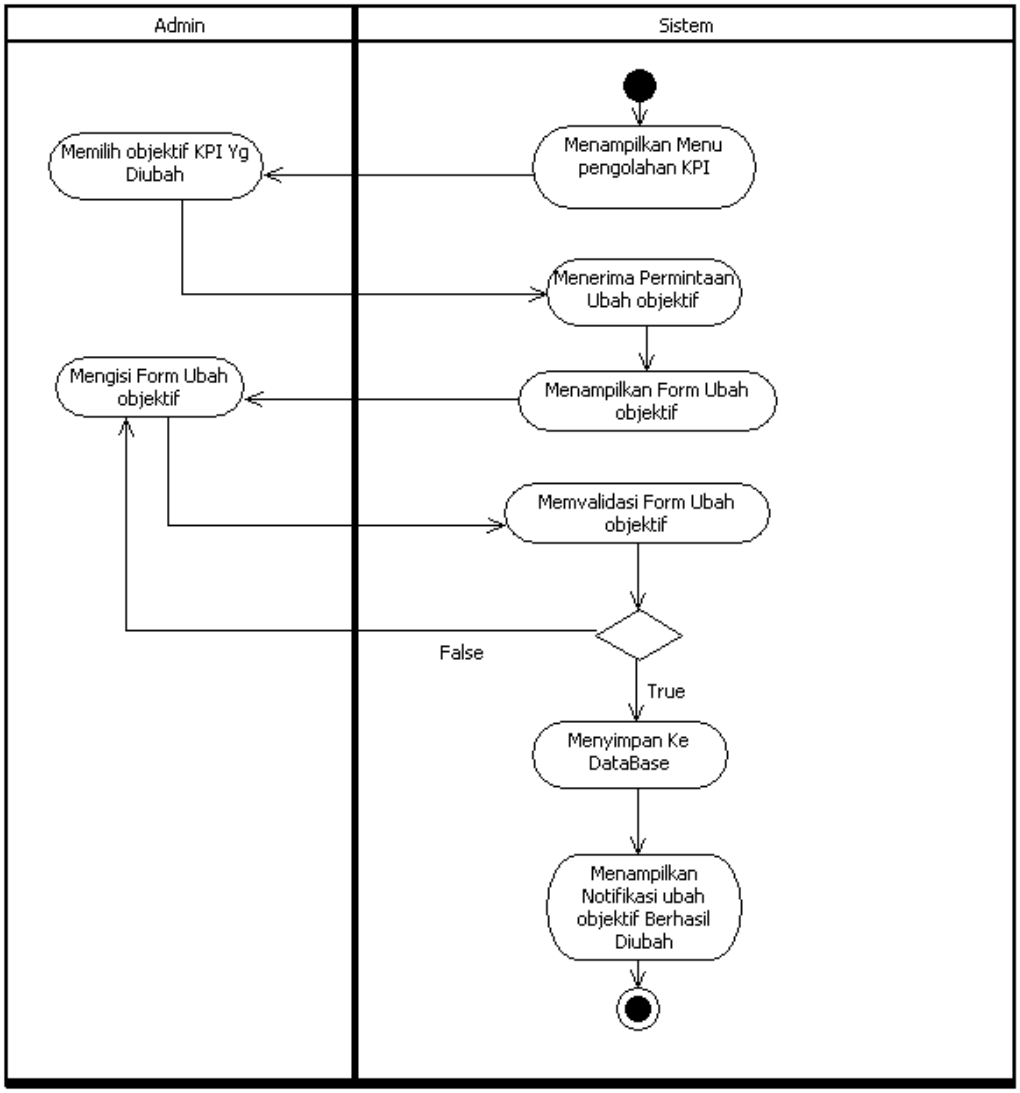
Interaksi antara aktor pengguna dengan *use case* tambah objektif KPI dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.42 *Activity Diagram* Tambah Objektif KPI

3.1.6.6.16 *Activity Diagram* Ubah Objektif KPI

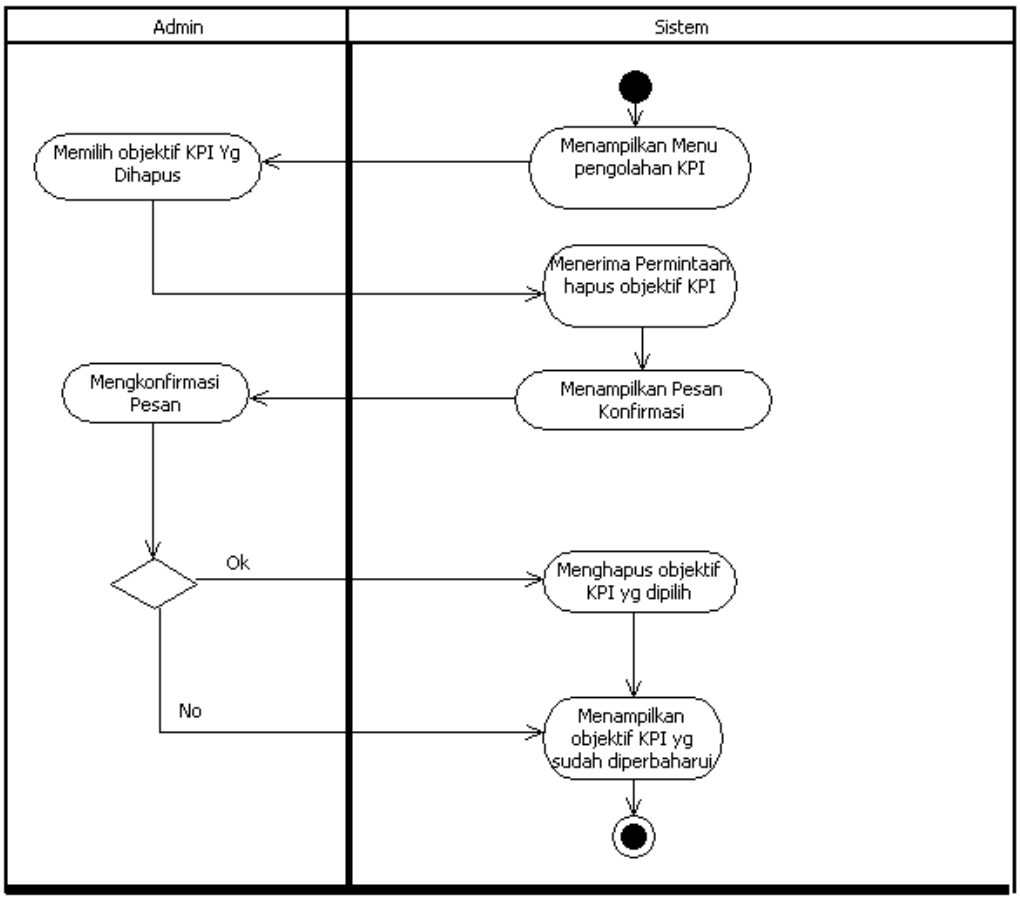
Interaksi antara aktor pengguna dengan *use case* Ubah objektif KPI dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.43 *Activity Diagram* Ubah Objektif KPI

3.1.6.6.17 *Activity Diagram* Hapus Objektif KPI

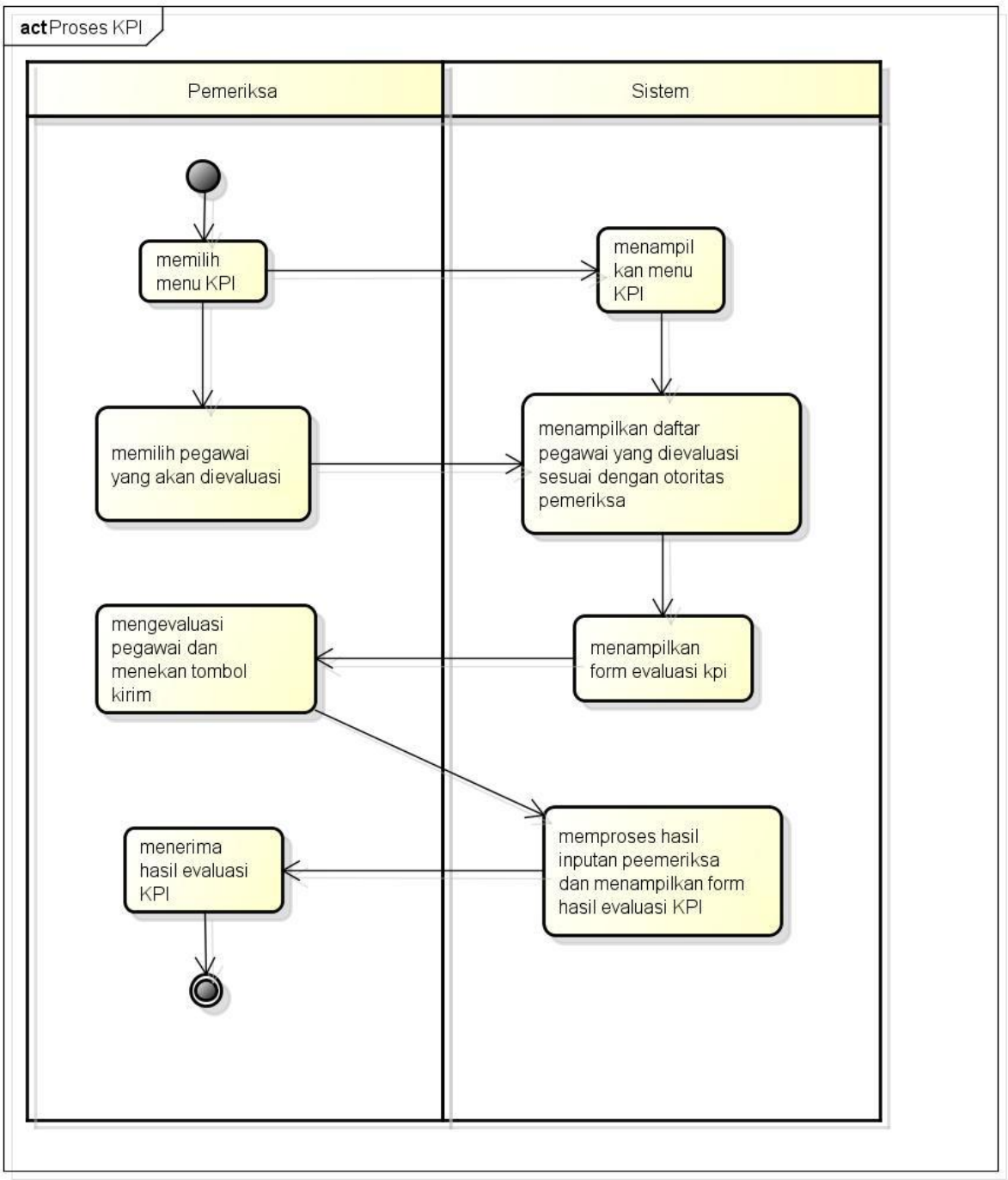
Interaksi antara aktor pengguna dengan *use case* hapus objektif KPI dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.44 *Activity Diagram* Hapus Objektif KPI

3.1.6.6.18 *Activity Diagram* Evaluasi KPI

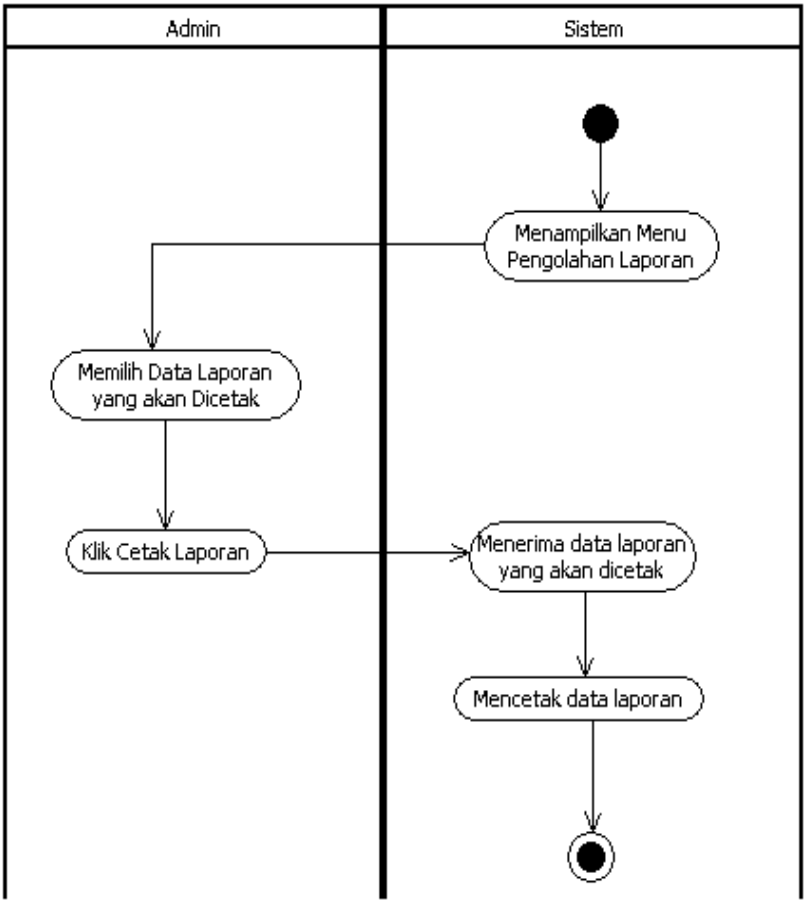
Interaksi antara aktor pengguna dengan *use case* evaluasi KPI dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.45 *Activity Diagram* Evaluasi KPI

3.1.6.6.19 *Activity Diagram* Cetak Laporan

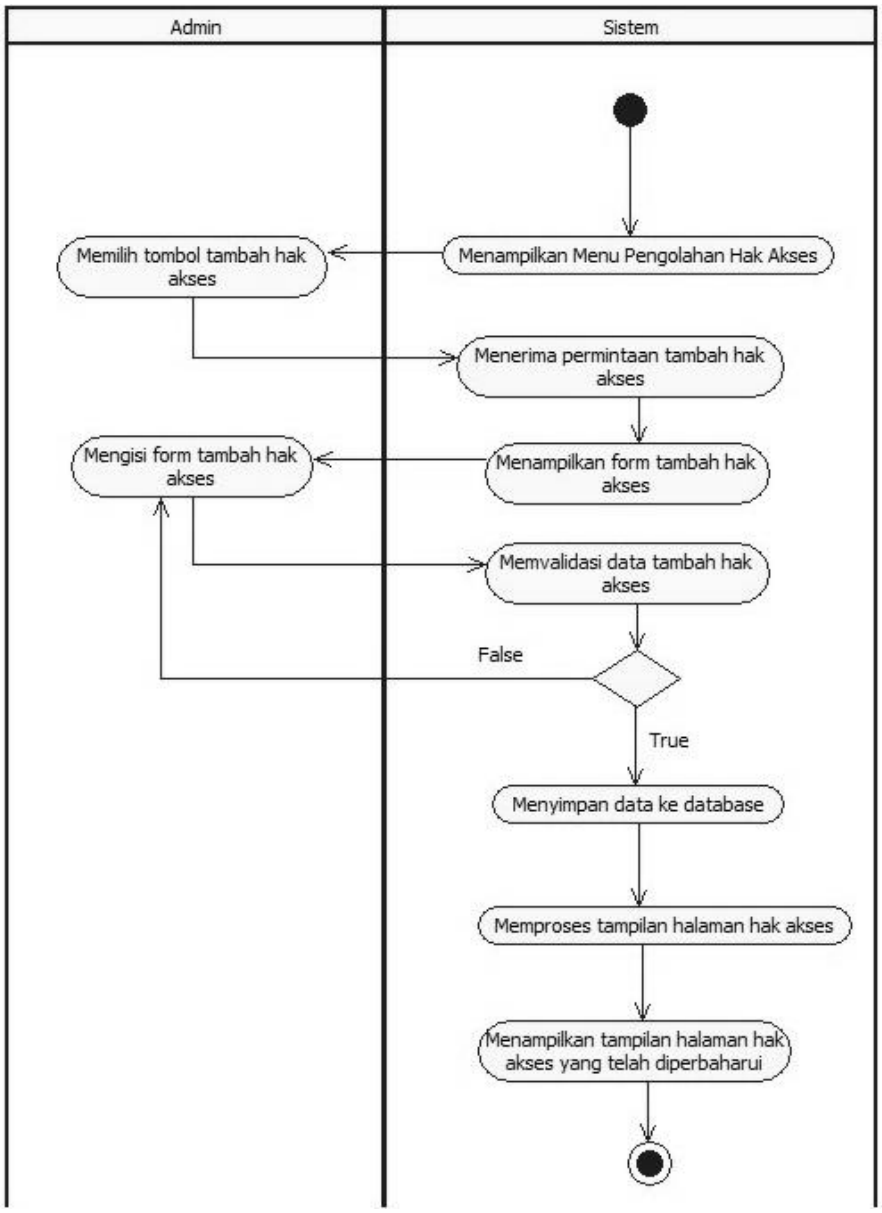
Interaksi antara aktor pengguna dengan *use case* cetak laporan dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.46 *Activity Diagram* Cetak Laporan

3.1.6.6.20 *Activity Diagram* Tambah Hak Akses

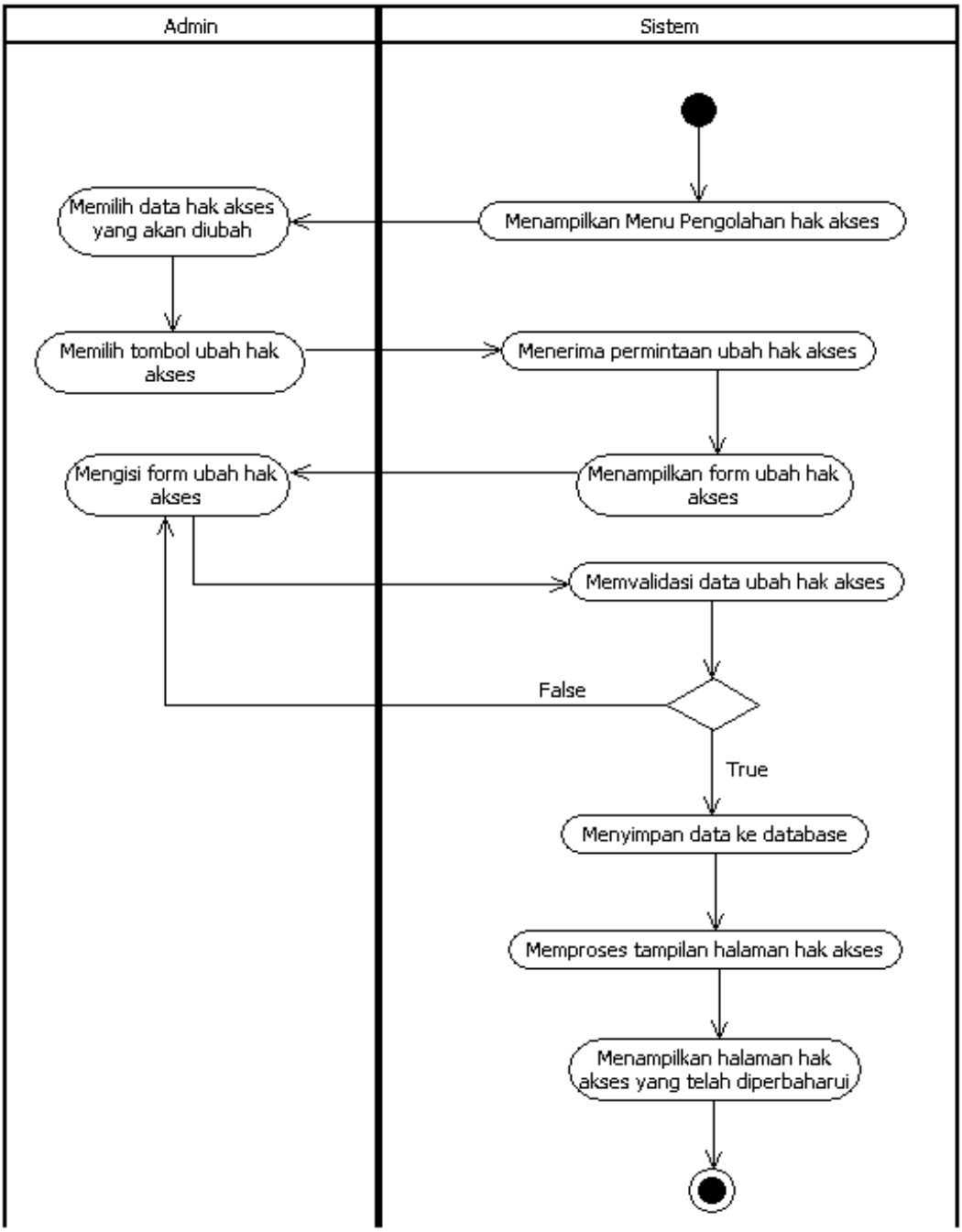
Interaksi antara aktor pengguna dengan *use case* tambah hak akses dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.47 *Activity Diagram* Tambah Hak Akses

3.1.6.6.21 *Activity Diagram* Ubah Hak Akses

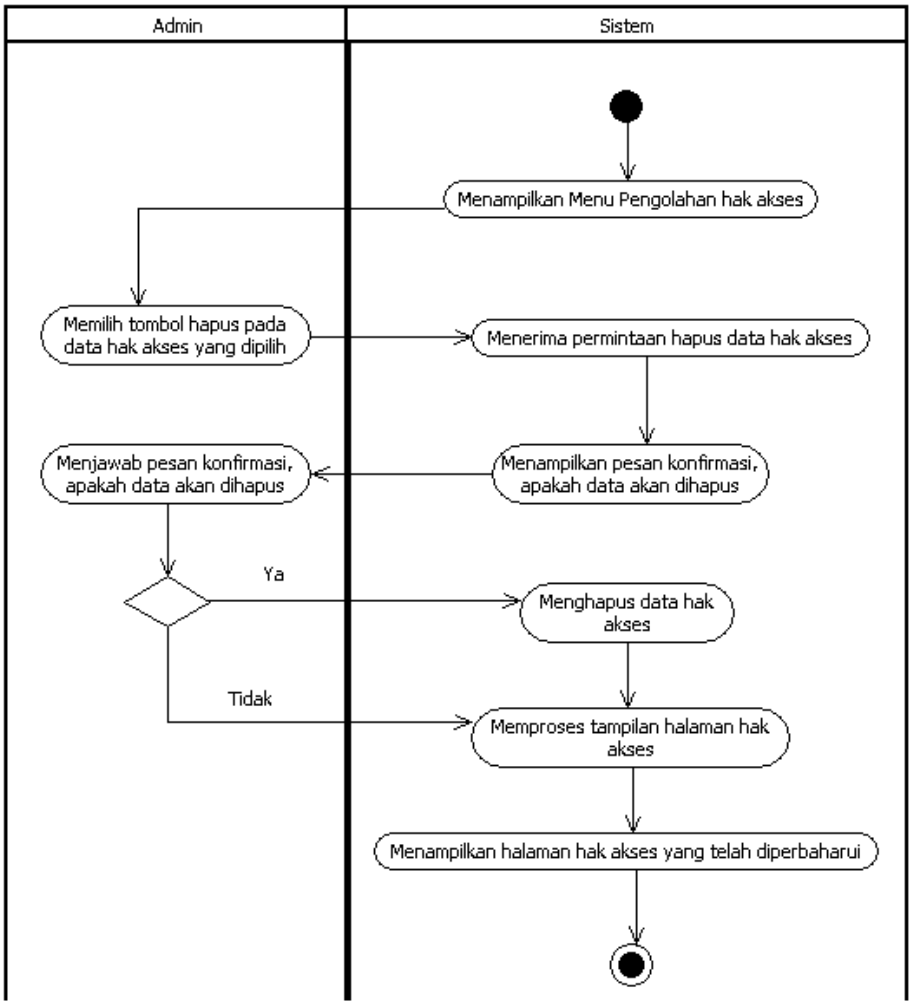
Interaksi antara aktor pengguna dengan *use case* ubah hak akses dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.48 *Activity Diagram* Ubah Hak Akses

3.1.6.6.22 *Activity Diagram* Hapus Hak Akses

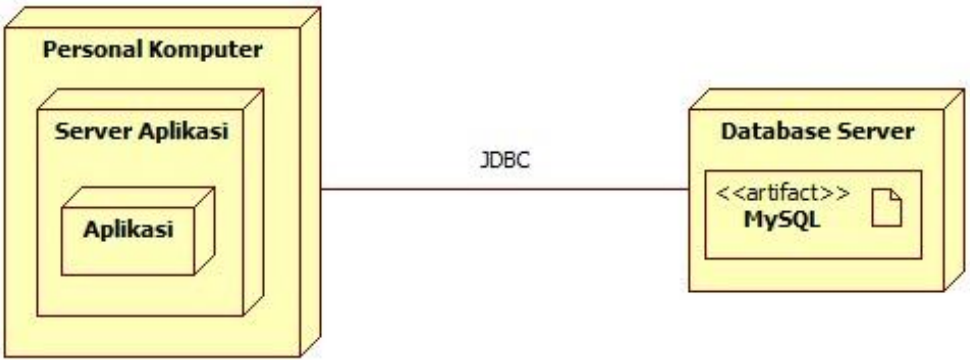
Interaksi antara aktor pengguna dengan *use case* hapus hak akses dijelaskan dalam *Activity Diagram* sebagai berikut :



Gambar 3.49 *Activity Diagram* Hapus Hak Akses

3.1.6.7 *Deployment Diagram*

Secara harifiah *Deployment Diagram* mengandung arti yaitu diagram pendisitribusian, jadi definisi *Deployment Diagram* adalah menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal. *Deployment Diagram* pada sistem informasi kepegawaian di PT. PLN (Persero) Distribusi Jabar dan Banten APJ Bandung dapat dilihat di gambar 3.48.



Gambar 3.50 ***Deployment Diagram***

3.2 Perancangan Sistem

Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh. Tahapan ini meliputi mengkonfigurasi komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem. Adapun perancangan sistem dari sistem informasi kepegawaian yang dibuat dijelaskan sebagai berikut.

3.2.1 Perancangan Data

Perancangan data merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai. Perancangan data terbagi menjadi skema relasi, diagram skema, dan perancangan struktur tabel. Berikut penjelasan detail perancangan data tersebut :

3.2.1.1 Skema Relasi

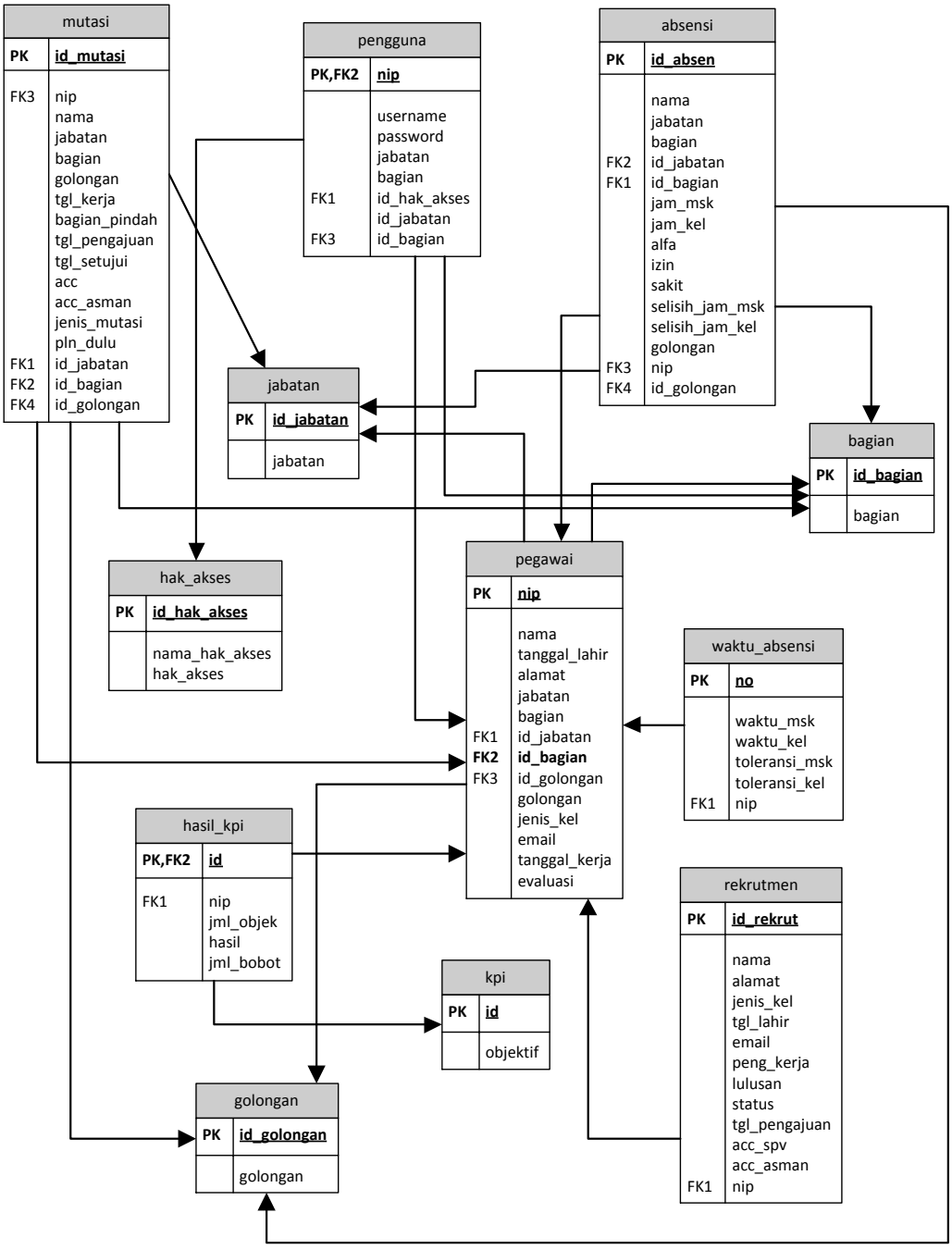
Skema relasi adalah dalah relasi yang memiliki nama, didefinisikan oleh himpunan pasangan atribut dan domainnya. Berikut adalah skema relasi dari basis data yang terdapat di dalam sistem informasi kepegawaian.

Pegawai	(nip, nama, tanggal_lahir, jenis_kel, alamat, jabatan, golongan, bagian, email, tanggal_kerja,id_jabatan, id_bagian, evaluasi),
Rekrutmen	(id_rekrutmen, nip, nama, jabatan, bagian, golongan, tgl_kerja, bagian_pindah, tgl_pengajuan, id_jabatan, id_bagian,tgl_setujui, acc, acc_asman, jenis_rekrutmen, pln_dulu),
Rekrutmen	(id_rekrut, nama, alamat, jenis_kel, tanggal_lahir, email, peng_kerja, kantor_lama, lulusan, status, tgl_pengajuan, acc_spv, acc_asman),
Absensi	(id_absen, nip, nama, jabatan, golongan, bagian, jam_msk,jam_kel,alfa,izin,sakit,selisih_jam_msk, id_jabatan, id_bagian, selisih_jam_kel),
Pengguna	(nip, password, jabatan, hak_akses, hak_ubah, hak_hapus, hak_setujui, id_jabatan, id_bagian,)
KPI	(id, objektif),
Hasil_KPI	(no, id, nip, jml_objek, hasil, jml_bobot)
Hak_Akses	(id_hak_akses, nama_hak_akses, hak_akses)
Waktu_Absensi	(no, waktu_msk, waktu_kel, toleransi_msk, toleransi_kel)
jababatan	(id_jabatan, jabatan)
bagian	(id_bagian, bagian)
golongan	(id_golongan, golongan)

3.2.1.2 Diagram Relasi

Diagram relasi adalah diagram yang menggambarkan hubungan antar data dalam suatu basis data berdasarkan objek-objek dasar data yang mempunyai

hubungan antara relasi. Berikut adalah gambar untuk diagram relasi pada basis data sistem informasi kepegawaian.



Gambar 3.51 Diagram Relasi

3.2.1.3 Struktur File

Struktur tabel menggambarkan detail tabel yang berisi field, tipe data, panjang data, dan keterangan lainnya. Adapun tabel-tabel yang digunakan dalam database sistem informasi kepegawaian ini adalah sebagai berikut:

1. Tabel Pegawai

Tabel pegawai digunakan untuk menyimpan data pegawai yang memudahkan dalam proses pengolahan data pegawai. Berikut struktur dari tabel pegawai :

Tabel 3-45 Tabel Pegawai

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
nip	Varchar	10	*Primary Key*	*NotNull*
nama	Varchar	45		*NotNull*
tanggal_lahir	Date			*NotNull*
jenis_kel	Varchar	6		*NotNull*
alamat	Text			*NotNull*
jabatan	varchar	30		*NotNull*
golongan	varchar	10		*NotNull*
bagian	Varchar	30		*NotNull*
email	varchar	50		*NotNull*
tanggal_kerja	date			*NotNull*
id_bagian	int		*Foreign Key references* bagian(id_bagian)	*NotNull*
id_jabatan	int		*Foreign Key references* jabatan(id_jabatan)	*NotNull*
evaluasi	Varchar	30		*NotNull*

2. Tabel Rekrutmen

Tabel rekrutmen digunakan untuk menyimpan data rekrutmen pegawai yang memudahkan dalam proses pengolahan data rekrutmen pegawai. Berikut struktur dari tabel rekrutmen pegawai :

Tabel 3-46 Tabel Rekrutmen

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id_rekrutmen	varchar	5	*Primary Key*	*NotNull*
nip	Varchar	10	*Foreign Key references* pegawai(nip)	*NotNull*
nama	Varchar	50		*NotNull*
jabatan	varchar	30		*NotNull*
golongan	varchar	10		*NotNull*
bagian	Varchar	20		*NotNull*
tanggal_kerja	date			*NotNull*
bagian_pindah	varchar	20		*NotNull*
tgl_pengajuan	date			*Null*
tgl_setujui	date			*Null*
acc	varchar	2		*Null*
acc_asman	varchar	2		*Null*
jenis_rekrutmen	varchar	10		*Null*
pln_dulu	varchar	10		*Null*
id_jabatan	int		*Foreign Key references* jabatan(id_jabatan)	*NotNull*
id_bagian	int		*Foreign Key references* bagian(id_bagian)	*NotNull*

3. Tabel Rekrutmen

Tabel rekrutmen digunakan untuk menyimpan data rekrutmen pegawai yang memudahkan dalam proses pengolahan data rekrutmen pegawai. Berikut struktur dari tabel rekrutmen pegawai :

Tabel 3-47 Tabel Rekrutmen

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id_rekrut	varchar	10	*Primary Key*	*NotNull*
nama	Varchar	50		*NotNull*
tanggal_lahir	Date			*NotNull*
jenis_kel	Varchar	6		*NotNull*
alamat	Text			*NotNull*
email	varchar	50		*NotNull*
peng_kerja	varchar	10		*Null*
kantor_lama	int	1		*Null*
lulusan	varchar	30		*Null*
tgl_pengajuan	date			*Null*

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
jabatan_dulu	varchar	40		*Null*
acc_spv	varchar	2		*Null*
acc_asman	varchar	2		*Null*
status	varchar	8		*Null*

4. Tabel Absensi

Tabel absensi digunakan untuk menyimpan data absensi pegawai yang memudahkan dalam proses pengolahan data absensi pegawai. Berikut struktur dari tabel absensi pegawai :

Tabel 3-48Tabel Absensi

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id_absensi	varchar	5	*Primary Key*	*NotNull*
nip	Varchar	10	*Foreign Key references* pegawai(nip)	*NotNull*
nama	Varchar	50		*NotNull*
jabatan	varchar	30		*NotNull*
golongan	varchar	10		*NotNull*
bagian	varchar	30		*NotNull*
jam_msk	time			*NotNull*
jam_kel	time			*NotNull*
id_jabatan	int		*Foreign Key references* jabatan (id_jabatan)	*NotNull*
id_bagian	int		*Foreign Key references* bagian (id_bagian)	*NotNull*
alfa	int			*Null*
izin	int			*Null*
sakit	int			*Null*

5. Tabel KPI

Tabel objektif digunakan untuk menyimpan data objektif KPI yang ada di dalam sistem informasi kepegawaian. Berikut struktur dari tabel objektif :

Tabel 3-49 Tabel Objektif

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id	int		*Primary Key*	*notnull,autoincrement*
objektif	Varchar	100		*NotNull*

6. Tabel Hasil KPI

Tabel objektif digunakan untuk menyimpan data dari hasil KPI yang ada di dalam sistem informasi kepegawaian. Berikut struktur dari tabel hasil KPI :

Tabel 3-50 Tabel Hasil KPI

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id	int		*Primary Key*	*notnull,autoincrement*
nip	Varchar	10	*Foreign Key references* pegawai(nip)	*NotNull*
jml_objek	int			*notnull*
hasil	varchar	50		*notnull*
jml_bobot	int			*notnull*

7. Tabel Pengguna

Tabel pengguna digunakan untuk menyimpan data pengguna pegawai yang memudahkan dalam proses pengolahan data pengguna pegawai. Berikut struktur dari tabel pengguna pegawai :

Tabel 3-51 Tabel Pengguna

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
nip	Varchar	10	*primary key, Foreign Key references* pegawai(nip)	*NotNull*
nama	Varchar	50		*NotNull*
username	varchar	30		*NotNull*
password	varchar	15		*NotNull*
jabatan	varchar	30		*NotNull*
id_hak_akses	varchar	10	*Foreign Key references* hak_akses(id_hak_akses)	*NotNull*

8. Tabel Bagian

Tabel bagian digunakan untuk menyimpan data bagian pegawai yang memudahkan dalam proses pengolahan data bagian pegawai. Berikut struktur dari tabel bagian pegawai :

Tabel 3-52 Tabel Bagian

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id_bagian	int		*primary key*	*NotNull,autoincrement*
bagian	Varchar	20		*NotNull*

9. Tabel Jabatan

Tabel jabatan digunakan untuk menyimpan data jabatan pegawai yang memudahkan dalam proses pengolahan data jabatan pegawai. Berikut struktur dari tabel jabatan pegawai :

Tabel 3-53 Tabel Jabatan

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id_jabatan	int		*primary key*	*NotNull,autoincrement*
jabatan	Varchar	20		*NotNull*

10. Tabel Waktu Absensi

Tabel waktu absensi digunakan untuk menyimpan data waktu absensi pegawai yang memudahkan dalam proses pengolahan data absensi pegawai sebagai acuan penentu waktu dari absensi. Berikut struktur dari tabel waktu absensi pegawai :

Tabel 3-54 Tabel Waktu Absensi

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
no	int		*primary key*	*NotNull,autoincrement*
waktu_msk	time			*NotNull*
waktu_kel	time			*NotNull*
toleransi_msk	time			*NotNull*
toleransi_kel	time			*NotNull*

11. Tabel Hak Akses

Tabel hak akses digunakan untuk menyimpan data hak akses dari pengguna sistem informasi kepegawaian yang memudahkan dalam proses pengolahan data pengguna itu sendiri. Berikut struktur dari tabel waktu hak akses :

Tabel 3-55 Tabel Hak Akses

Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id_hak_akses	int		*primary key*	*NotNull,autoincrement*
nama_hak_akses	varchar(30)			*NotNull*
hak_akses	varchar(30)			*NotNull*

12. Tabel Golongan

Tabel 3-56 Tabel Golongan

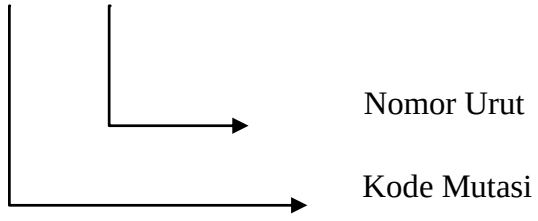
Nama Field	Tipe Data	Panjang	Kunci	Keterangan
id_golongan	int		*primary key*	*NotNull,autoincrement*
golongan	varchar(10)			*NotNull*

3.2.1.4 Perancangan Kode

Perancangan kode dilakukan untuk mengkonsepkan suatu pengkodean yang nantinya akan selalu sama dan simultan sehingga kesamaan dan keunikan dapat terjaga. Berikut adalah perancangan kode yang terdapat pada sistem informasi kepegawaian.

1. Kode Mutasi

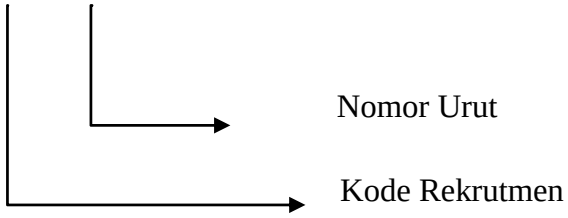
   Format rekrutmen : FFF 111



   Contoh : MUT032, data mutasi pegawai no urut 32.

2. Kode Rekrutmen

Format rekrutmen : FFF 111



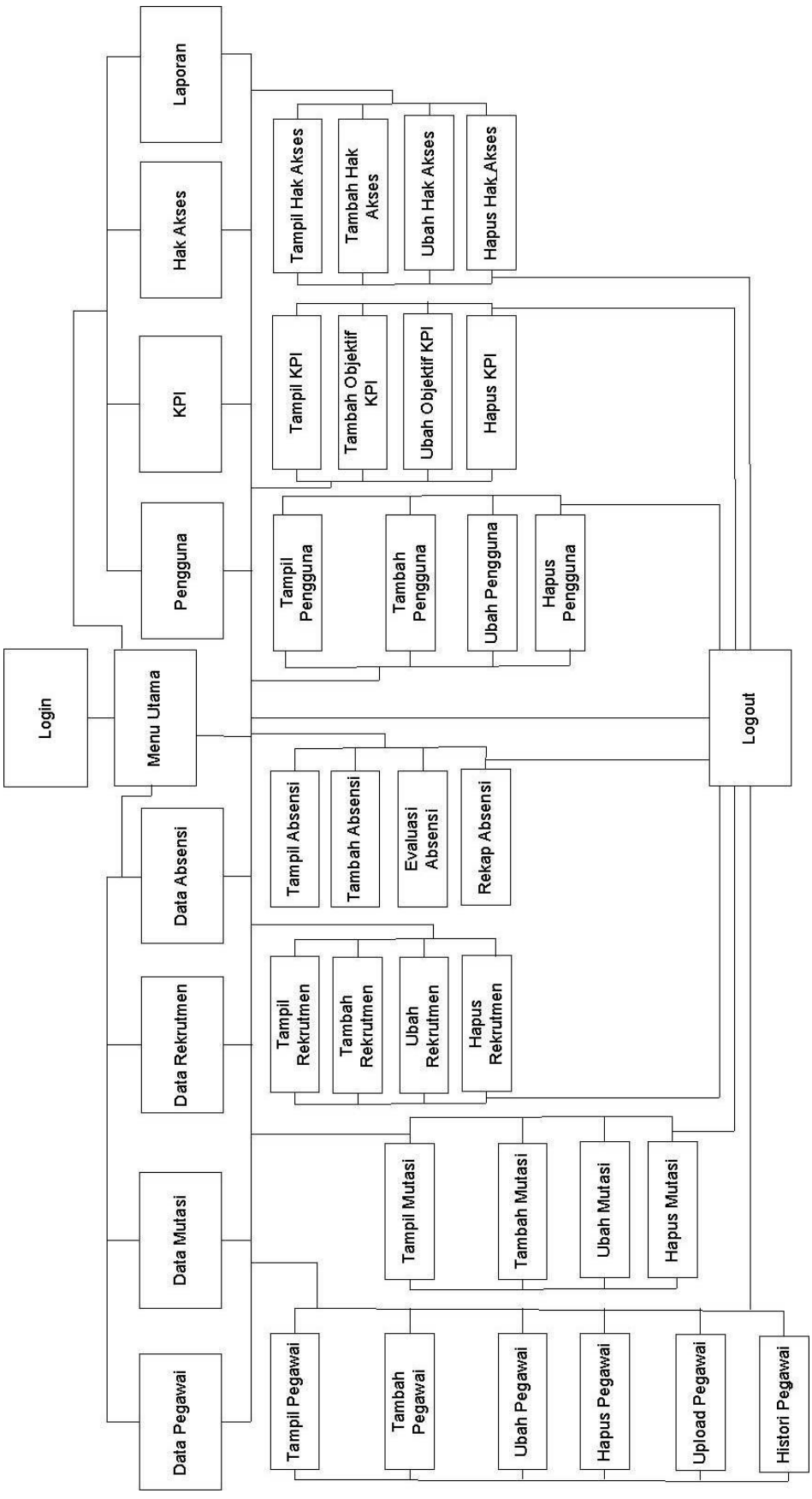
Contoh : REK014, data rekrutmen pegawai no urut 14.

3.2.1.5 Perancangan Arsitektur

Perancagan Arsitektur adalah perancangan yang menggambarkan suatu desain dari alur suatu sistem yang nantinya akan diimplementasikan. Berikut perancangan arsitektur pada sistem informasi kepegawaian akan dijelaskan mengenai perancangan arsitektur dari tiap-tiap pengguna.

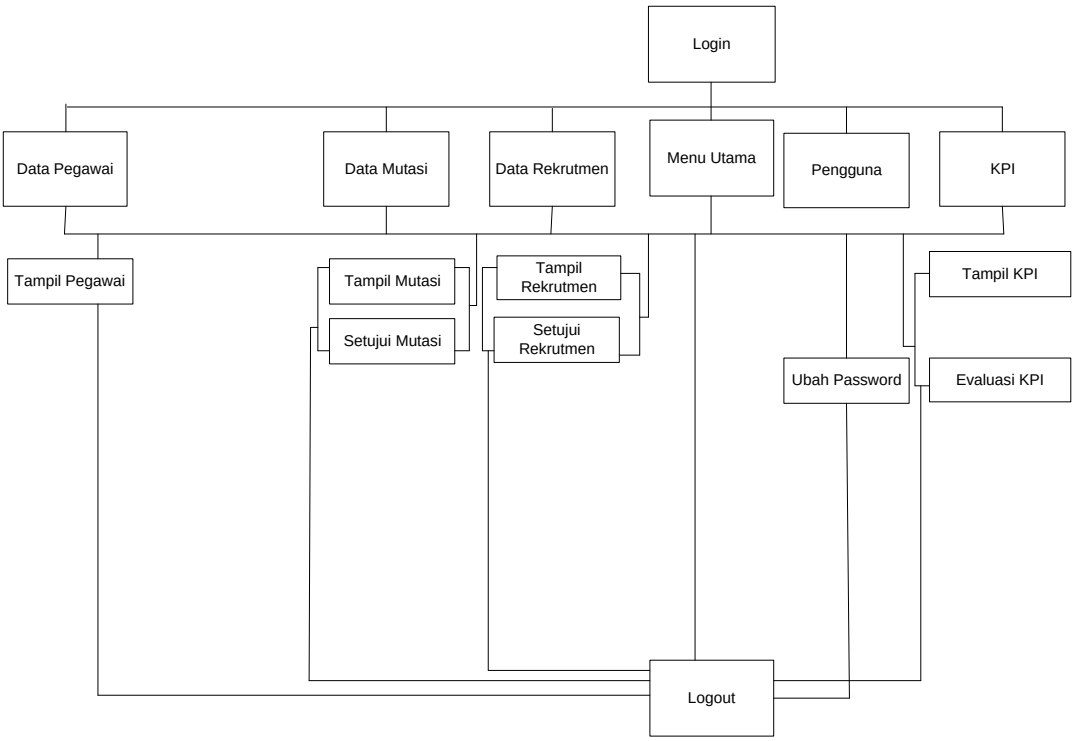
3.2.1.6 Struktur Menu Admin

Berikut pada gambar 3.51 akan digambarkan mengenai struktur menu pada sisi admin:


Login
Menu Utama
Laporan
Hak Akses
KPI
Pengguna
Data Absensi
Data Rekrutmen
Data Mutasi
Data Pegawai
Logout
Tampil Hak Akses
Tambah Hak Akses
Ubah Hak Akses
Hapus Hak Akses
Tampil KPI
Tambah Objektif KPI
Ubah Objektif KPI
Hapus KPI
Tampil Pengguna
Tambah Pengguna
Ubah Pengguna
Hapus Pengguna
Tampil Absensi
Tambah Absensi
Evaluasi Absensi
Rekap Absensi
Tampil Rekrutmen
Tambah Rekrutmen
Ubah Rekrutmen
Hapus Rekrutmen
Tampil Mutasi
Tambah Mutasi
Ubah Mutasi
Hapus Mutasi
Tampil Pegawai
Tambah Pegawai
Ubah Pegawai
Hapus Pegawai
Upload Pegawai
Histori Pegawai

3.2.1.7 Struktur Menu Asman

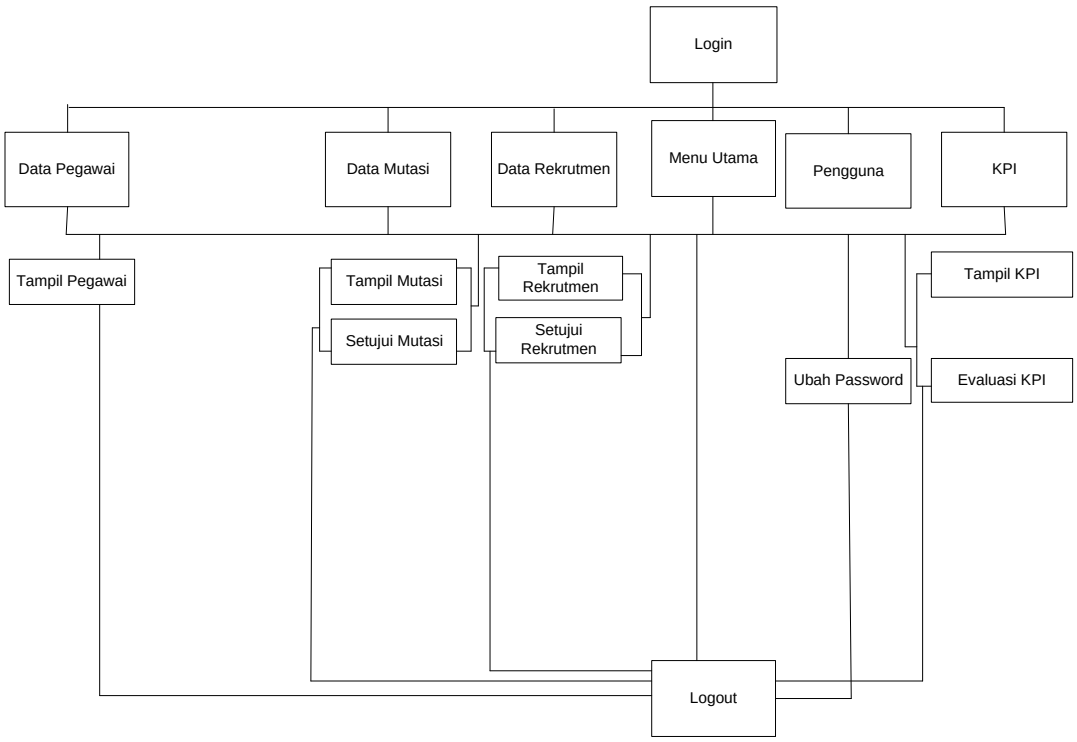
Berikut pada gambar 3.52 akan digambarkan mengenai struktur menu pada sisi asman :



Gambar 3.52 Struktur Menu Asman

3.2.1.8 Struktur Menu Supervisor

Berikut pada gambar 3.54 akan digambarkan mengenai struktur menu pada sisi supervisor :



Gambar 3.53 Struktur Menu Supervisor

3.2.2 Perancagan Antarmuka

Perancangan antarmuka dibuat untuk memberi konsep pada implementasi pembuatan suatu antarmuka nantinya pada sistem. Berikut adalah gambaran dari perancangan antarmuka di sistem informasi kepegawaian.

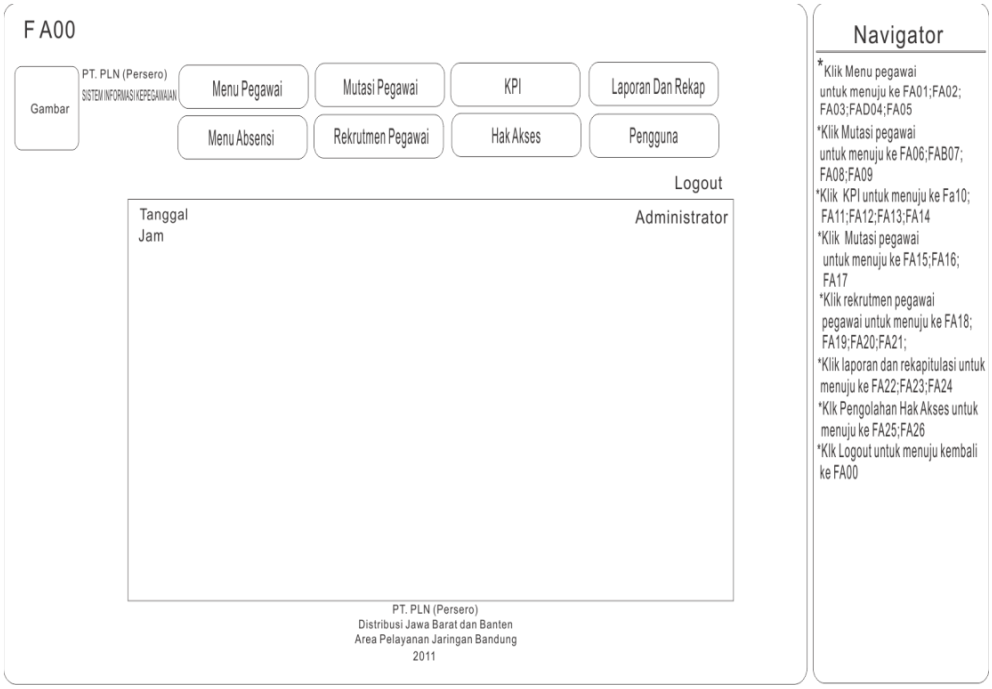
3.2.2.1 Perancangan Form

1. Form Login



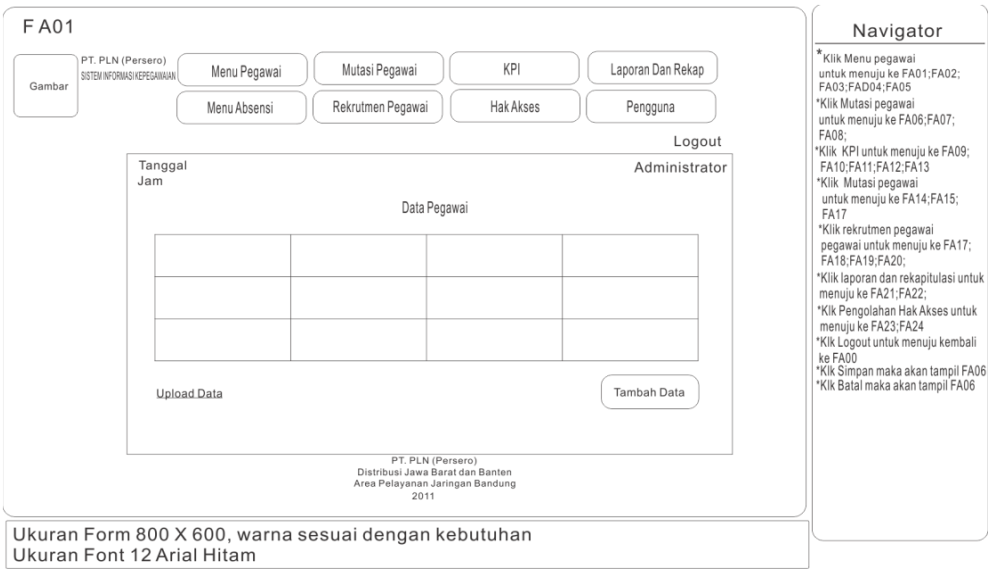
Gambar 3.54 Form Login

2. Form Menu Utama



Gambar 3.55 Form Menu Utama

3. Form Pengolahan Data Pegawai



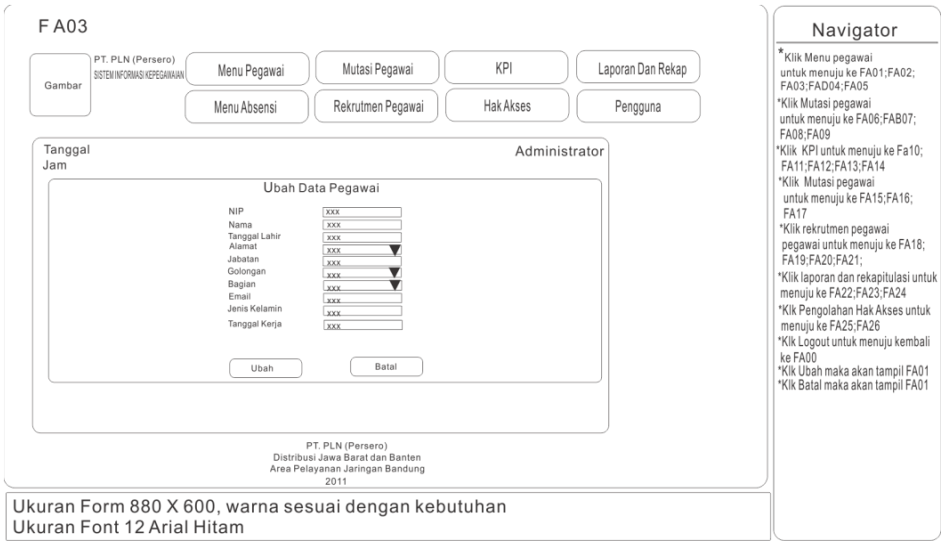
Gambar 3.56 Form Pengolahan Data Pegawai

4. Form Tambah Data Pegawai



Gambar 3.57 Form Tambah Data Pegawai

5. Form Ubah Data Pegawai



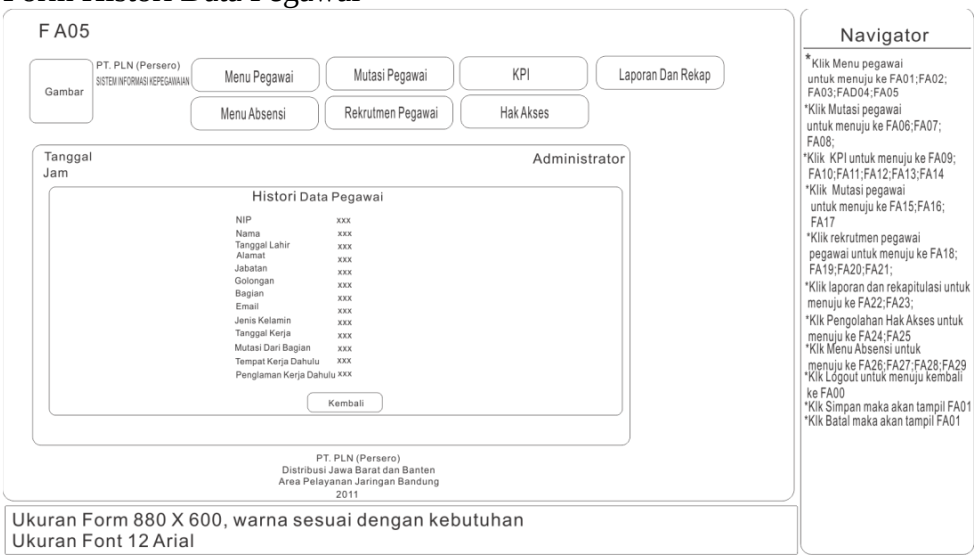
Gambar 3.58 Form Ubah Data Pegawai

6. Form *Upload* Data Pegawai



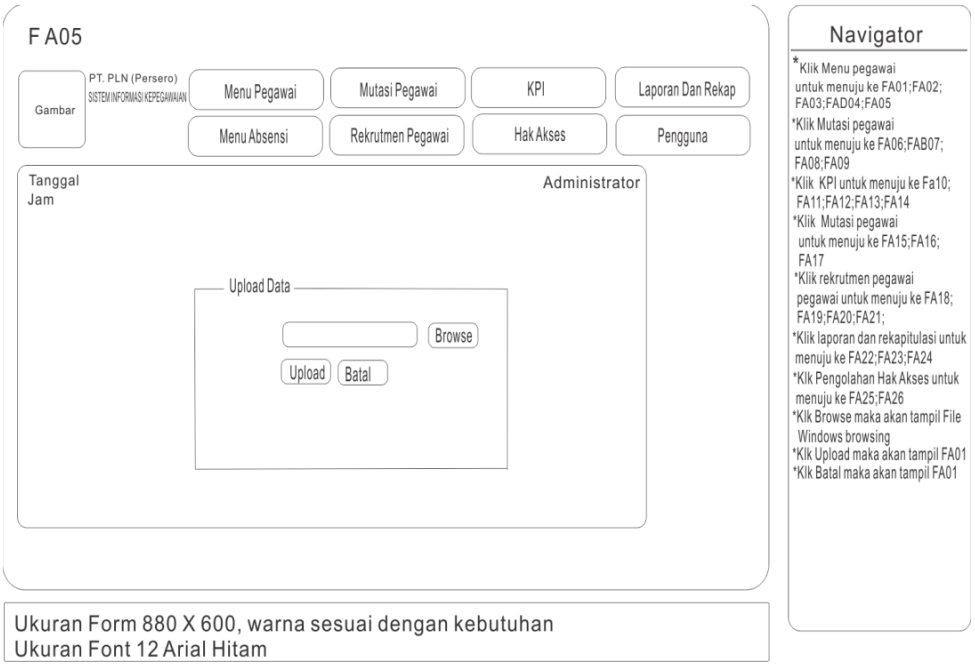
Gambar 3.59 Form Upload Data Pegawai

7. Form Histori Data Pegawai



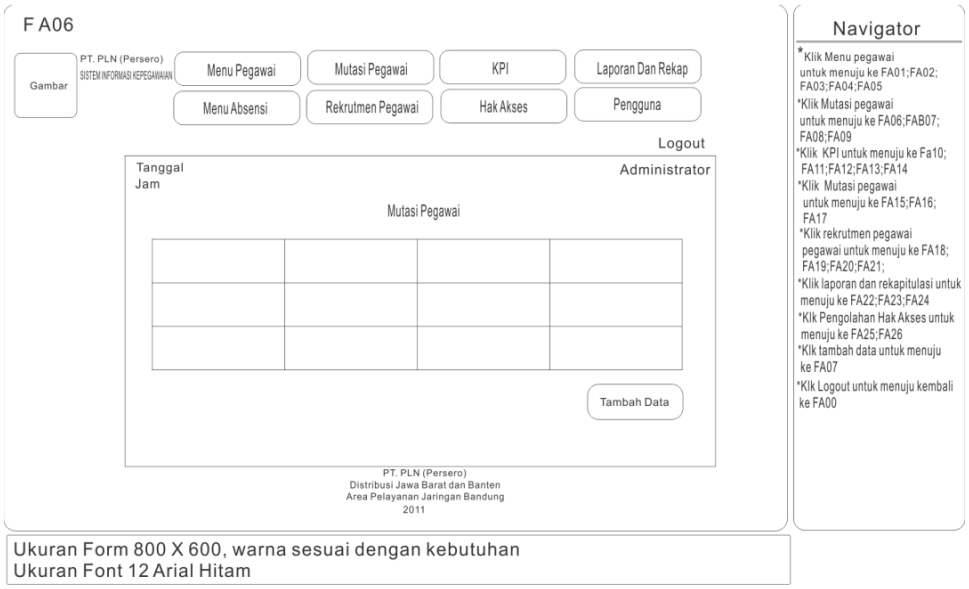
Gambar 3.60 Form Histori Data Pegawai

8. Form Upload Data Pegawai



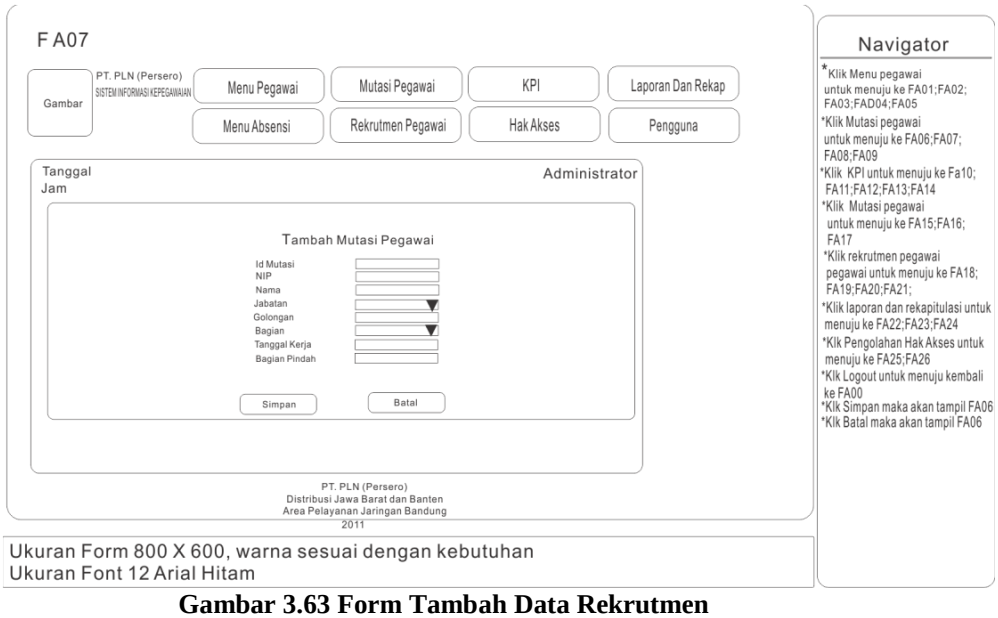
Gambar 3.61 Form Upload Data Pegawai

9. Form Pengolahan Data Rekrutmen



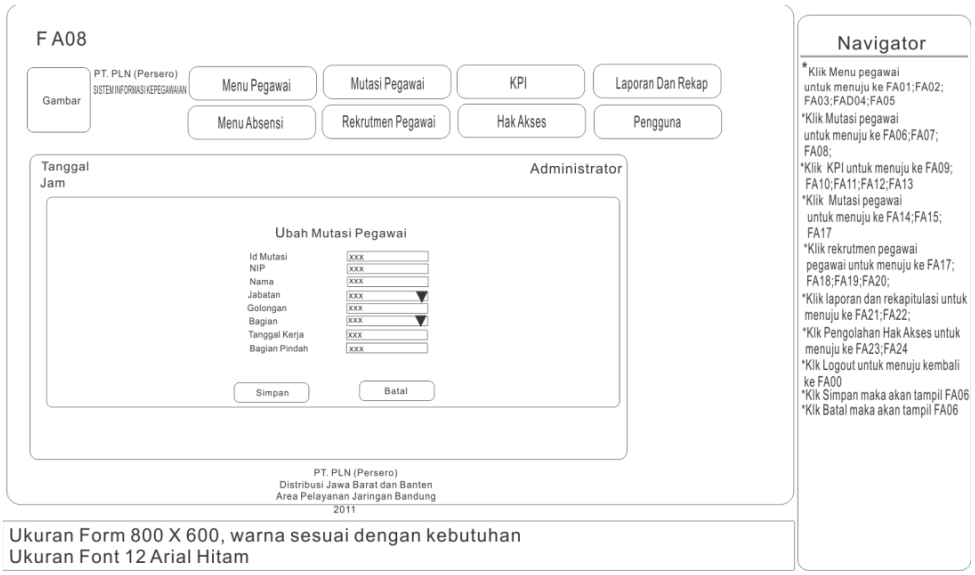
Gambar 3.62 Form Pengolahan Rekrutmen Pegawai

10. Form Tambah Data Rekrutmen



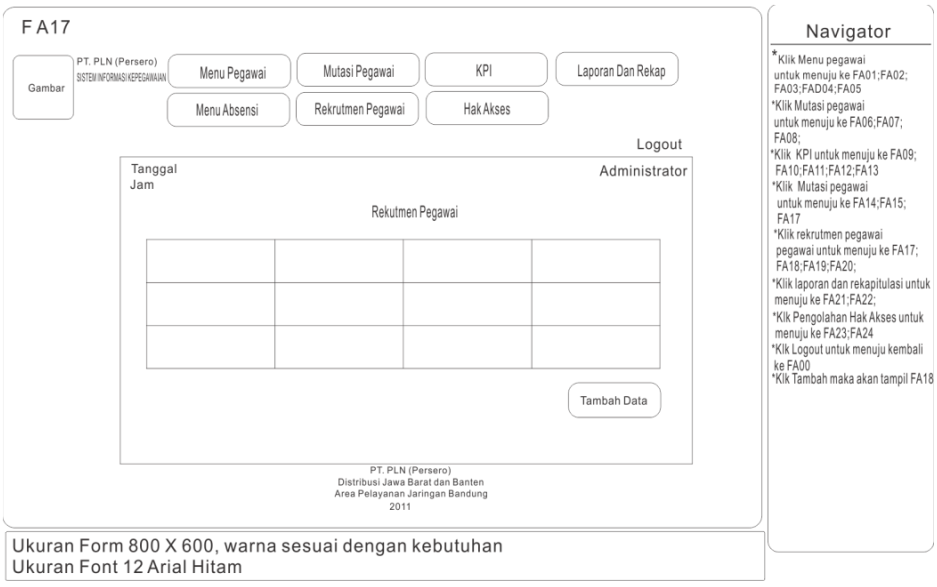
Gambar 3.63 Form Tambah Data Rekrutmen

11. Form Ubah Data Rekrutmen



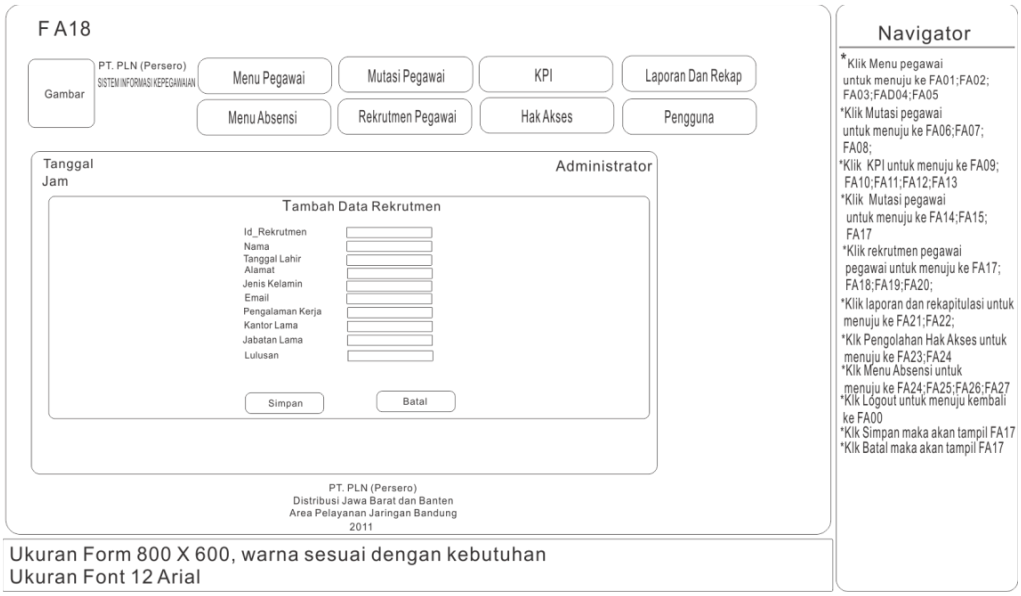
Gambar 3.64 Form Ubah Data Rekrutmen

12. Form Pengolahan Data Rekrutmen



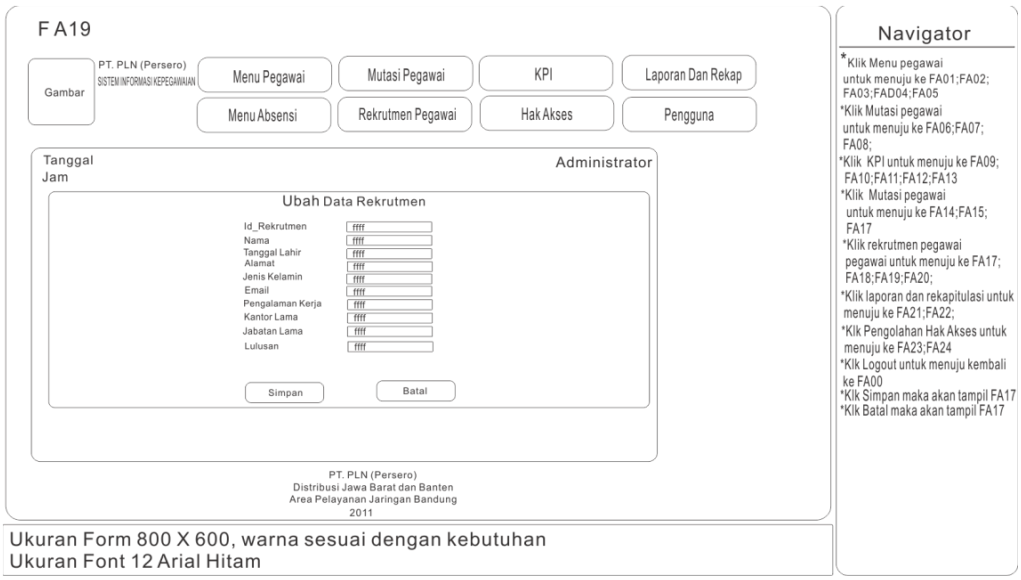
Gambar 3.65 Form Pengolahan Data Rekrutmen

13. Form Tambah Rekrutmen Pegawai



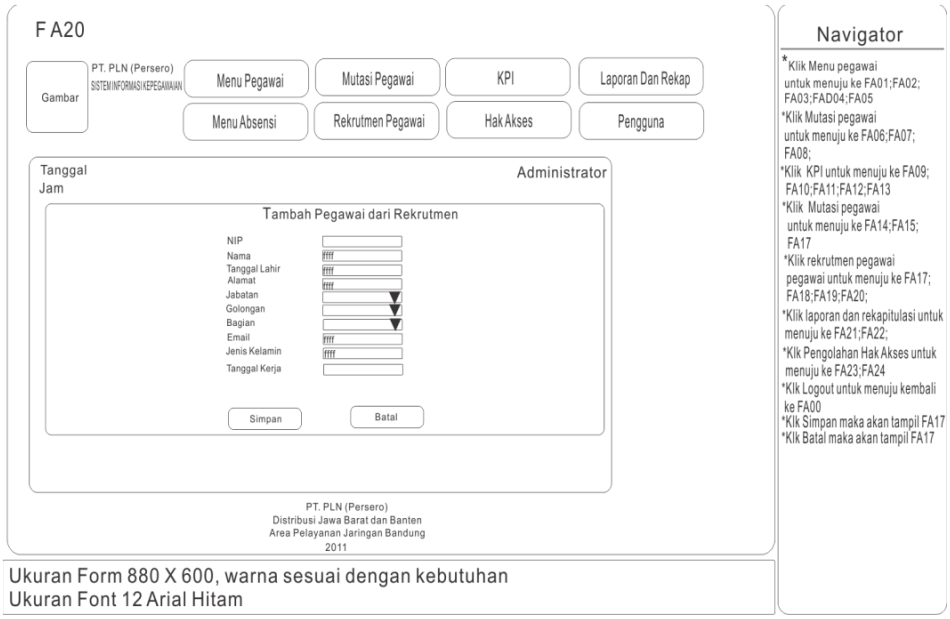
Gambar 3.66 Form Tambah Data Rekrutmen

14. Form Ubah Rekrutmen Pegawai



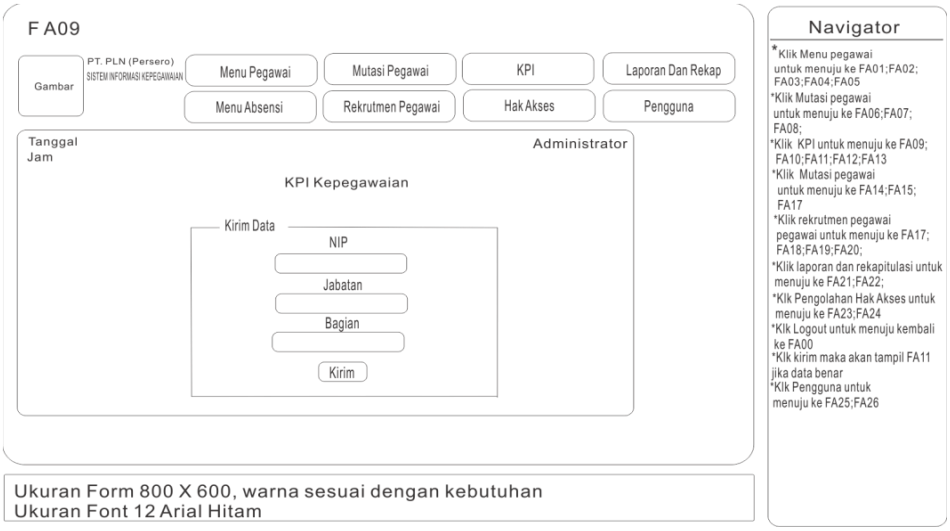
Gambar 3.67 Form Ubah Rekrutmen Pegawai

15. Form Terima Pegawai dari Rekrutmen



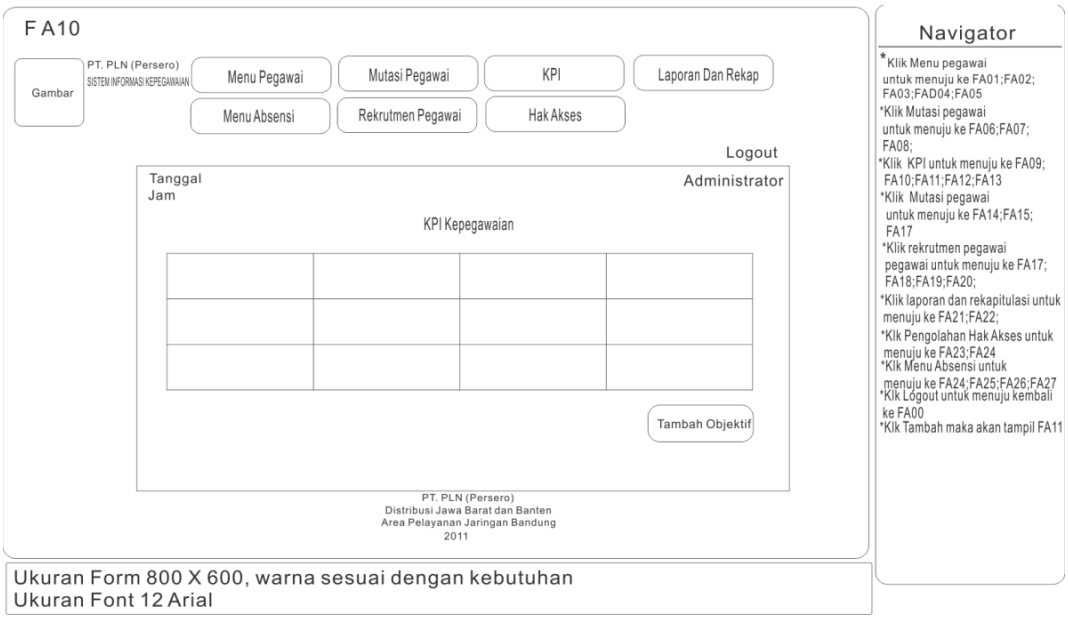
Gambar 3.68 Form Terima Pegawai dari Rekrutmen

16. Form Awal KPI (*Key Performance Indicators*)



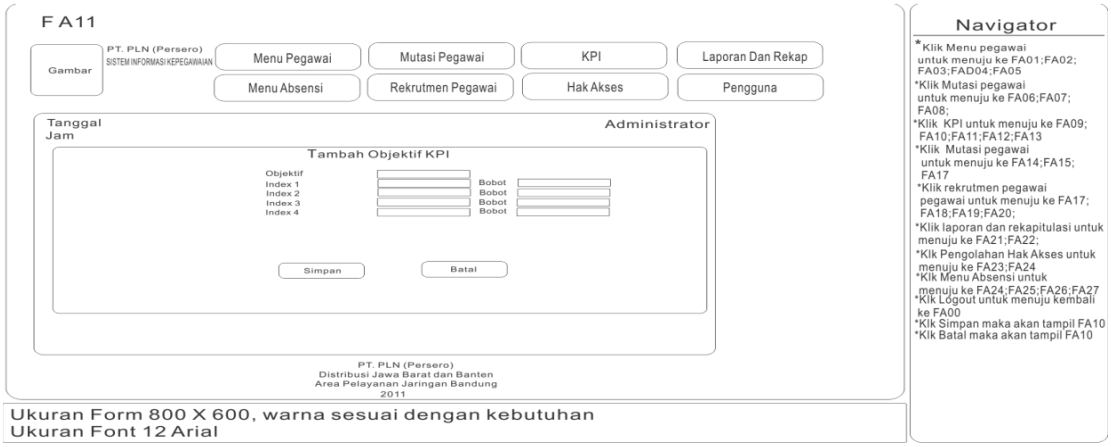
Gambar 3.69 Form Awal KPI

17. Form Pengolahan KPI



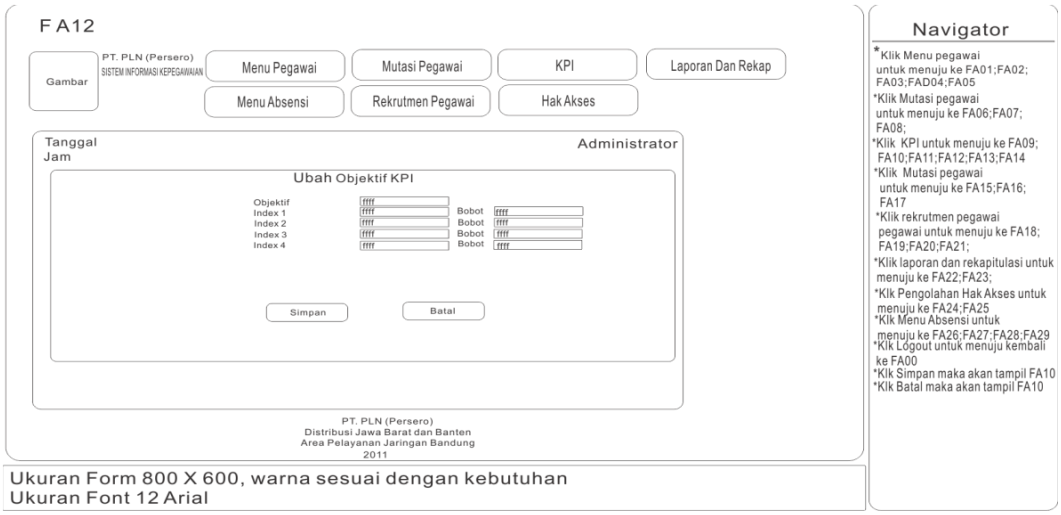
Gambar 3.70Form Pengolahan KPI

18. Form Tambah Objektif KPI



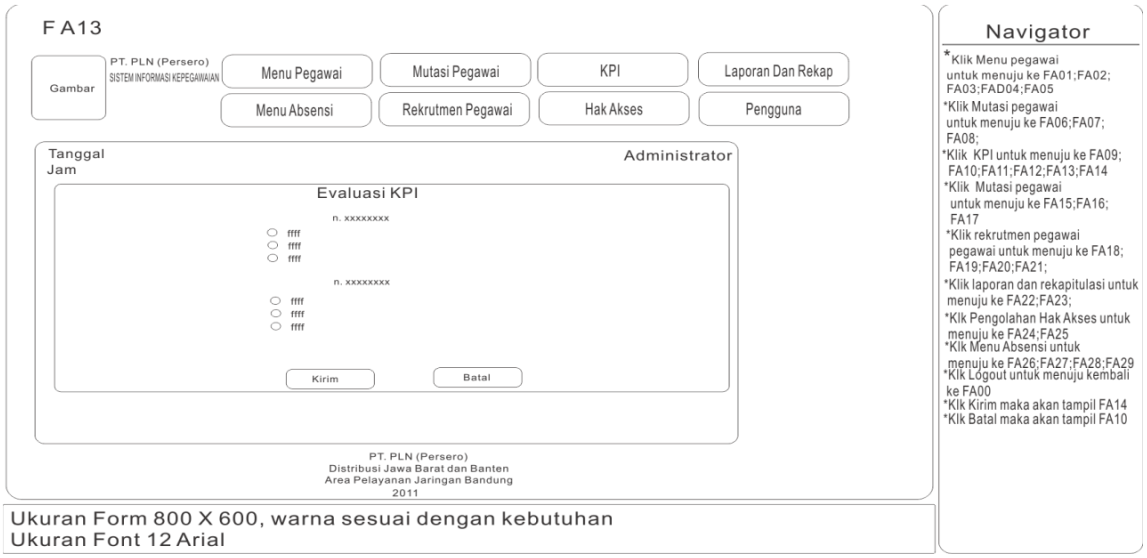
Gambar 3.71 Form Tambah Objektif

19. Form Ubah Objektif KPI



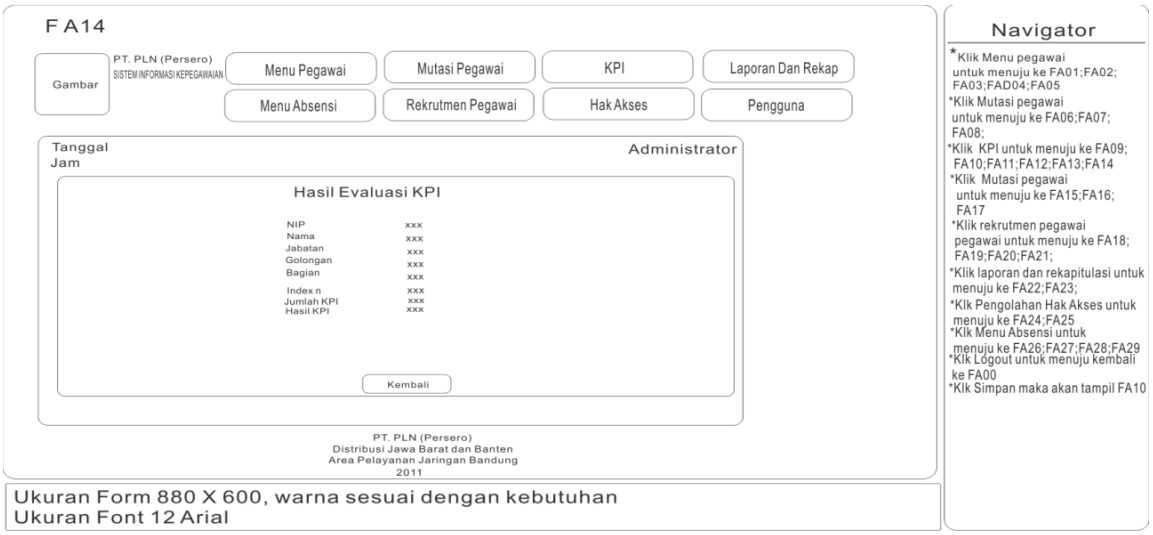
Gambar 3.72 Form Ubah Objektif KPI

20. Form Evaluasi KPI



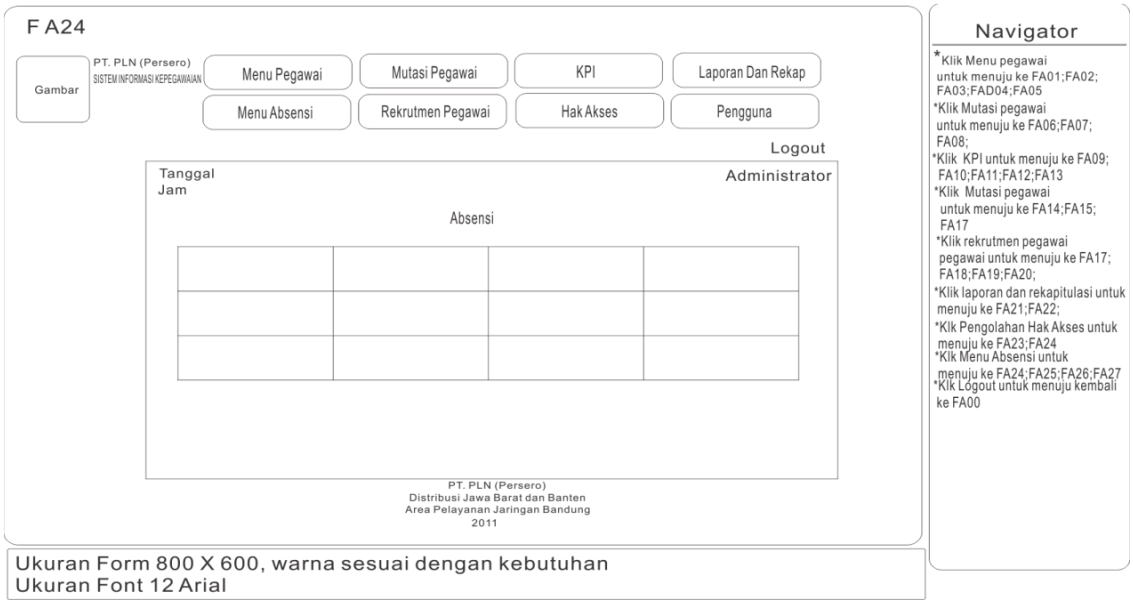
Gambar 3.73 Form Evaluasi KPI

21. Form Hasil KPI



Gambar 3.74 Form Hasil KPI

22. Form Pengolahan Absensi



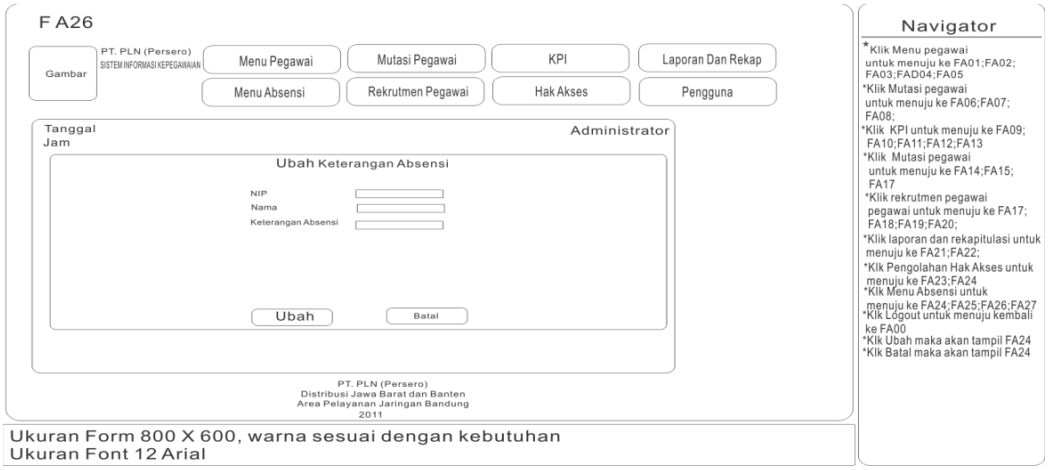
Gambar 3.75 Form Pengolahan Absensi

23. Form Tambah Keterangan Absensi



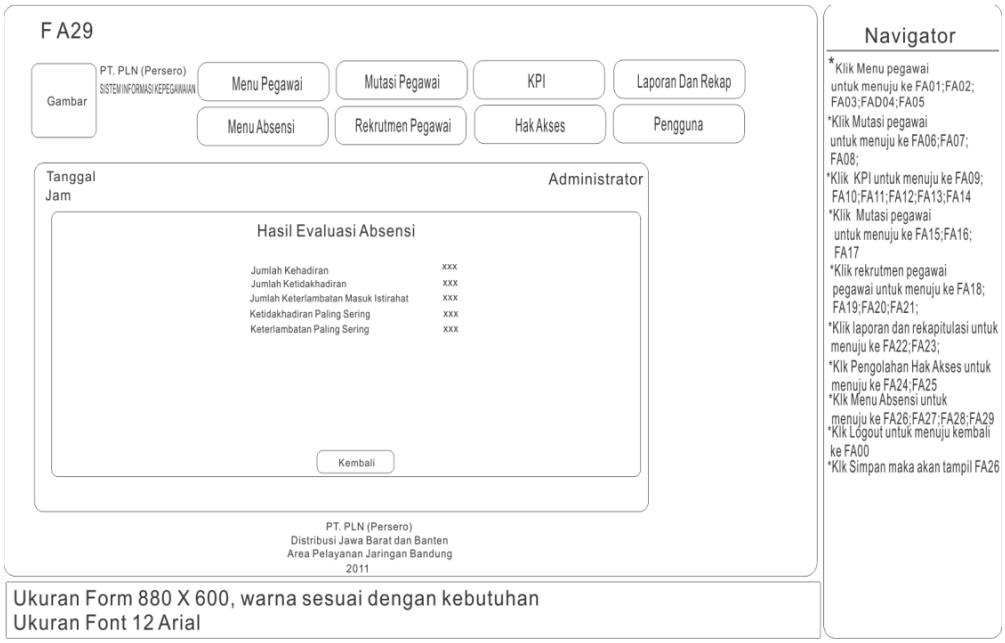
Gambar 3.76 Form Tambah Keterangan Absensi

24. Form Ubah Keterangan Absensi



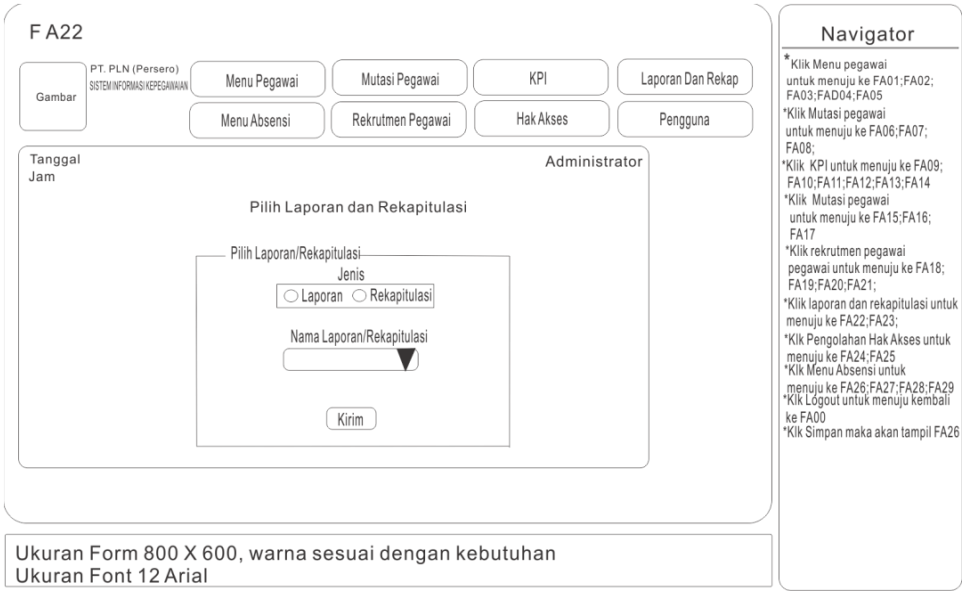
Gambar 3.77 Form Ubah Keterangan Absensi

25. Form Hasil Evaluasi Absensi



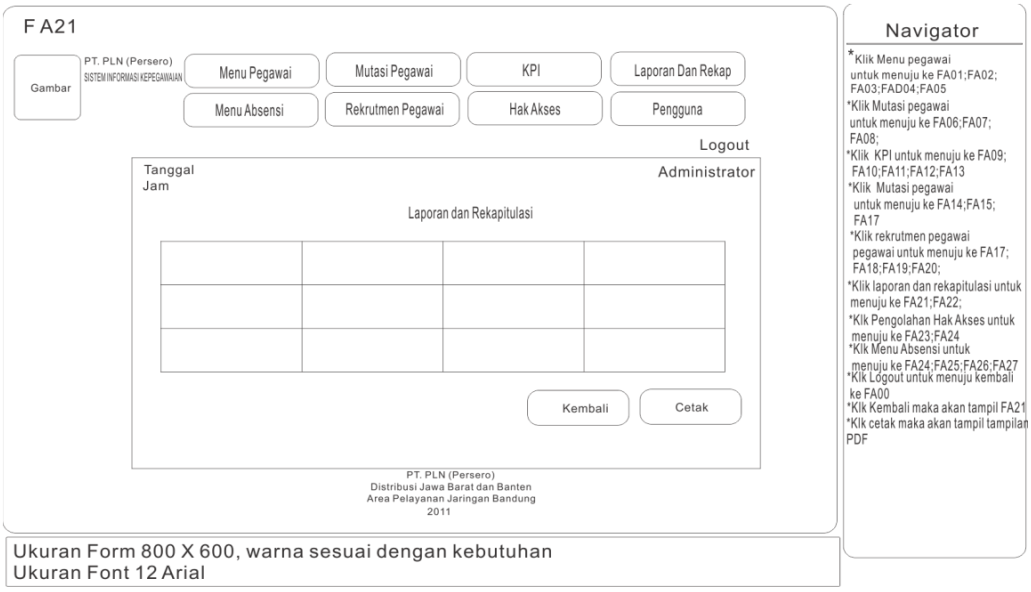
Gambar 3.78 Form Hasil Evaluasi Absensi

26. Form Pilih Laporan dan Rekapitulasi



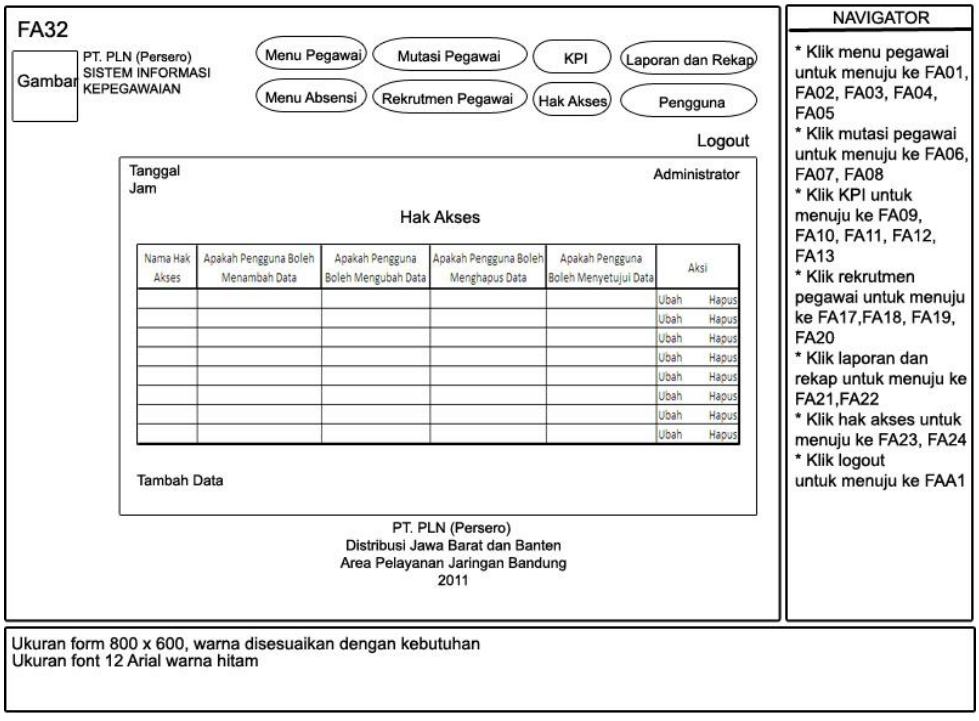
Gambar 3.79 Form Pilih Laporan dan Rekapitulasi

27. Form Laporan dan Rekapitulasi



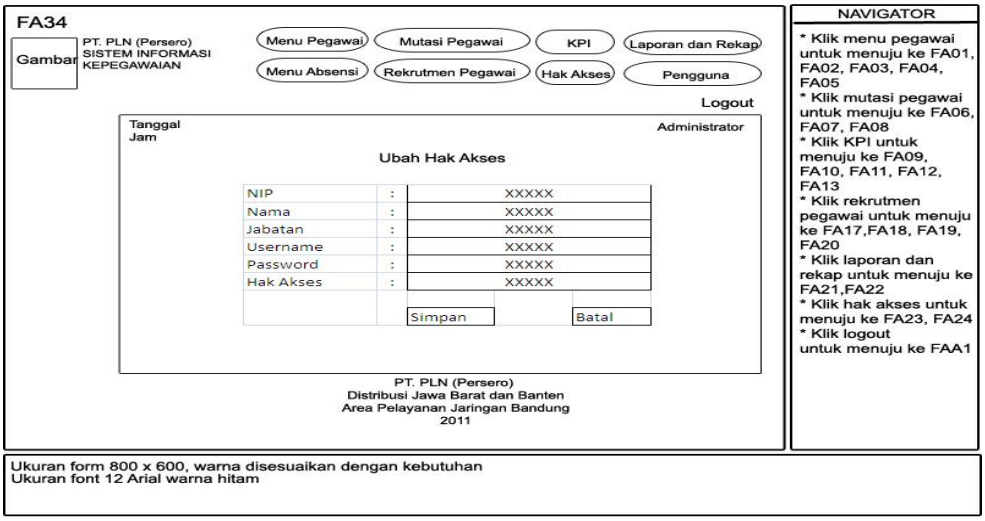
Gambar 3.80 Form Laporan dan Rekapitulasi

28. Form Hak Akses



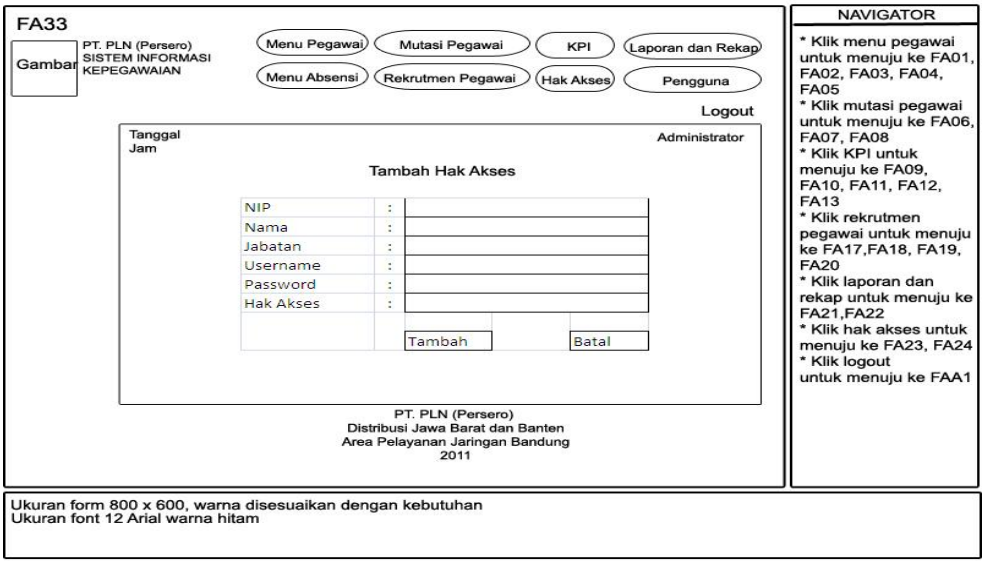
Gambar 3.81 Form Hak Akses

29. Form Ubah Hak Akses



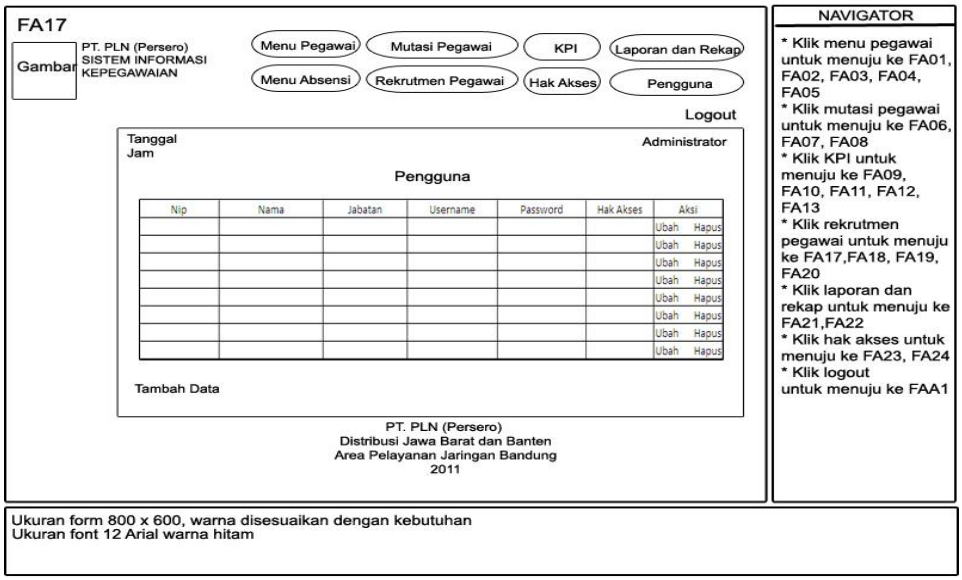
Gambar 3.82 Form Ubah Hak Akses

30. Form Tambah Hak Akses



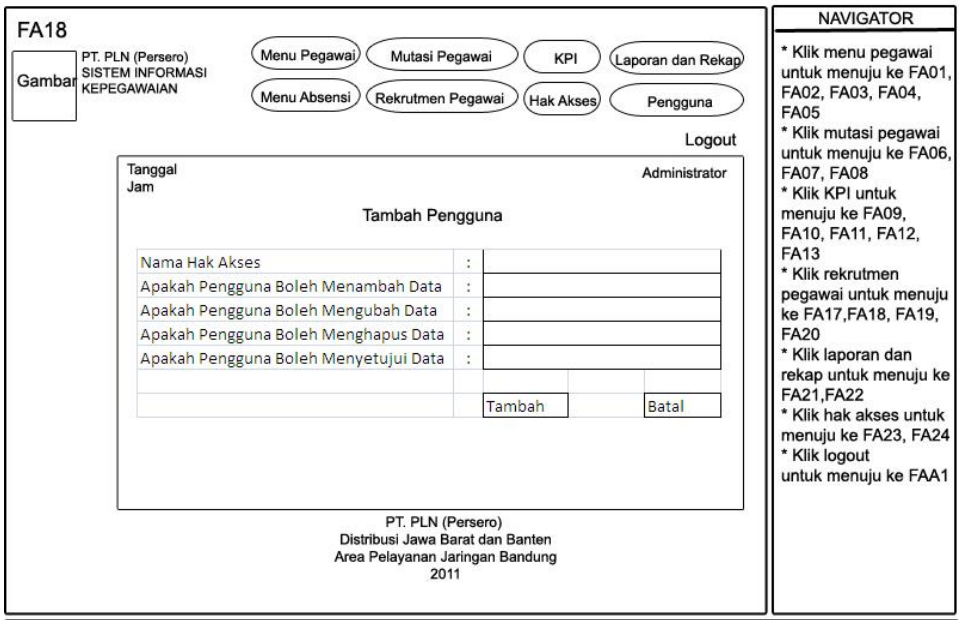
Gambar 3.83 Form Tambah Hak Akses

31. Form Pengguna



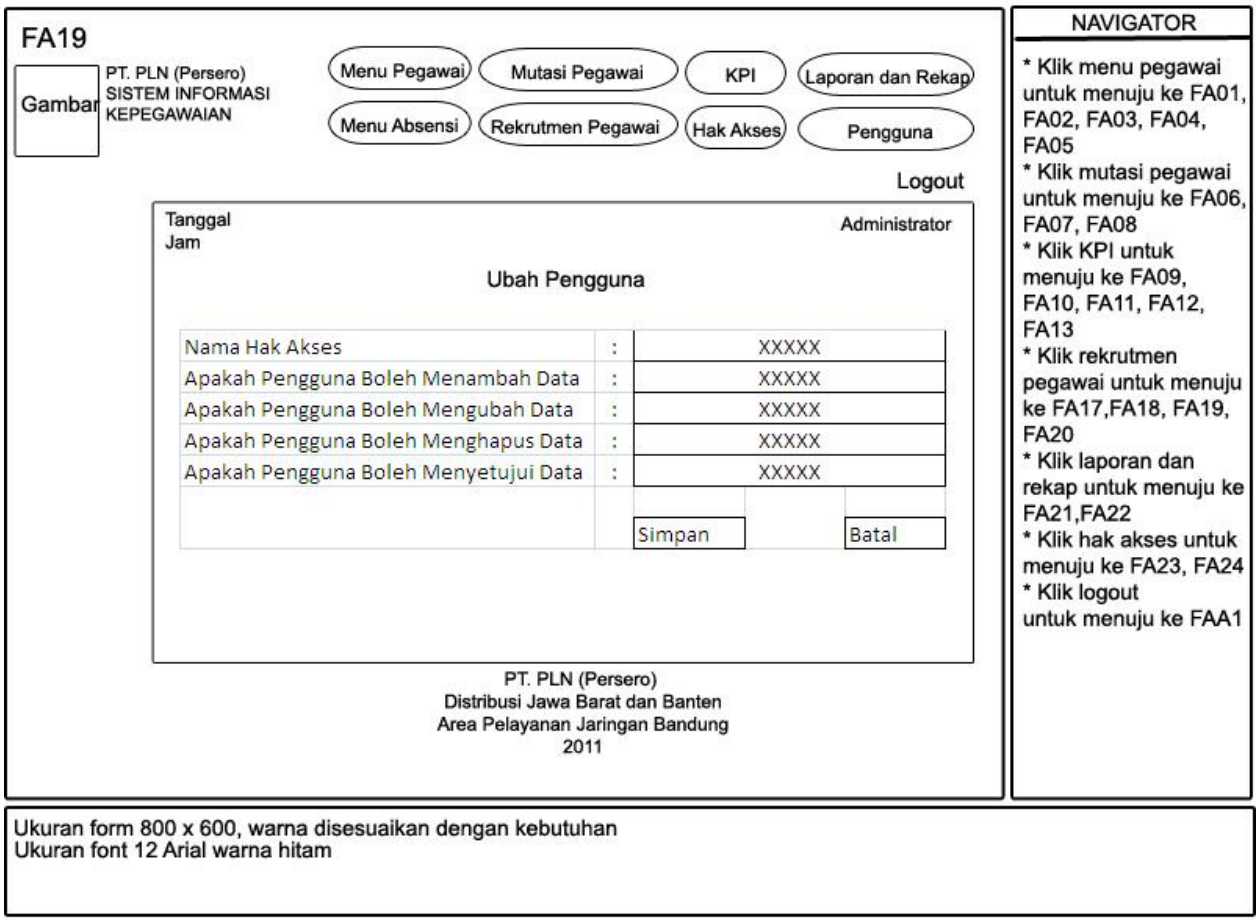
Gambar 3.84 Form Pengguna

32. Form Tambah Pengguna



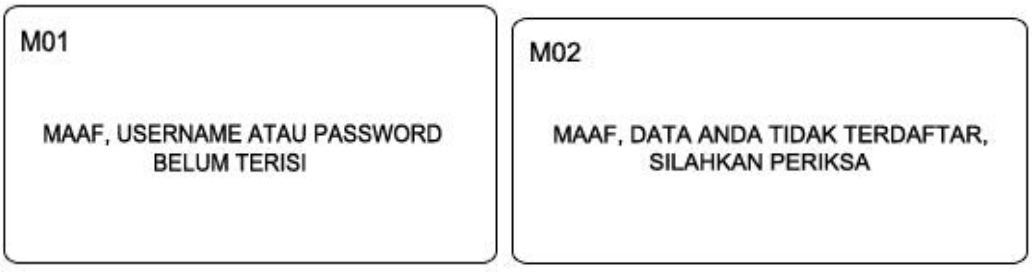
Gambar 3.85 Form Tambah Pengguna

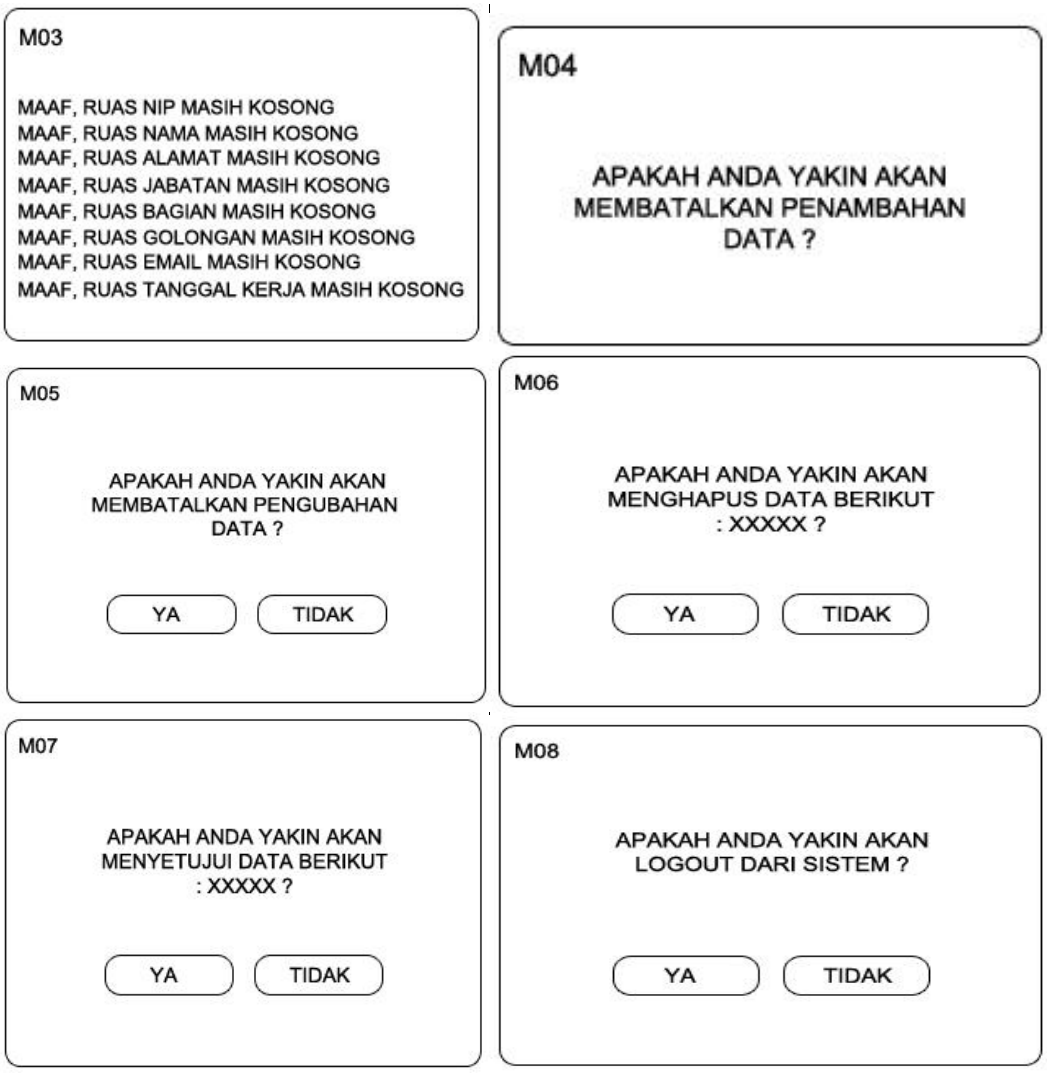
33. Form Ubah Pengguna



Gambar 3.86 Form Ubah Pengguna

3.2.2.2 Perancagan Pesan




M03
MAAF, RUAS NIP MASIH KOSONG
MAAF, RUAS NAMA MASIH KOSONG
MAAF, RUAS ALAMAT MASIH KOSONG
MAAF, RUAS JABATAN MASIH KOSONG
MAAF, RUAS BAGIAN MASIH KOSONG
MAAF, RUAS GOLONGAN MASIH KOSONG
MAAF, RUAS EMAIL MASIH KOSONG
MAAF, RUAS TANGGAL KERJA MASIH KOSONG
M04
APAKAH ANDA YAKIN AKAN MEMBATALKAN PENAMBAHAN DATA ?
M05
APAKAH ANDA YAKIN AKAN MEMBATALKAN PENGUBAHAN DATA ?
YA
TIDAK
M06
APAKAH ANDA YAKIN AKAN MENGHAPUS DATA BERIKUT : XXXXX ?
YA
TIDAK
M07
APAKAH ANDA YAKIN AKAN MENYETUJUI DATA BERIKUT : XXXXX ?
YA
TIDAK
M08
APAKAH ANDA YAKIN AKAN LOGOUT DARI SISTEM ?
YA
TIDAK


M09
MAAF, RUAS ID MASIH KOSONG
MAAF, RUAS NAMA MASIH KOSONG
MAAF, RUAS ALAMAT MASIH KOSONG
MAAF, RUAS JABATAN MASIH KOSONG
MAAF, RUAS BAGIAN MASIH KOSONG
MAAF, RUAS GOLONGAN MASIH KOSONG
MAAF, RUAS EMAIL MASIH KOSONG
MAAF, RUAS TANGGAL KERJA MASIH KOSONG


M10
APAKAH ANDA YAKIN AKAN
MENGHAPUS DATA BERIKUT ?
YA
TIDAK

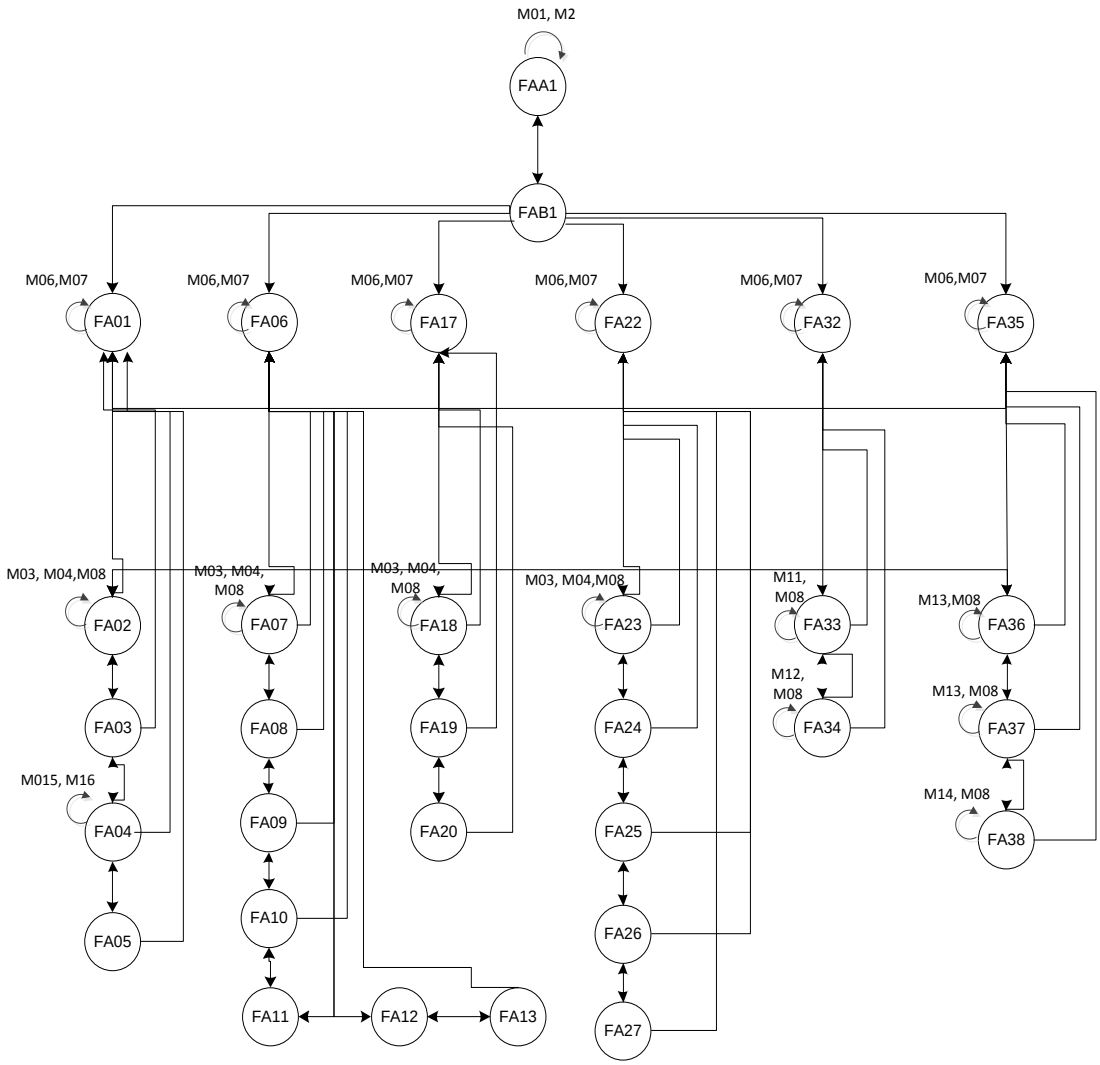

M11
APAKAH ANDA YAKIN AKAN
MENAMBAH DATA BERIKUT ?
YA
TIDAK


M12
APAKAH ANDA YAKIN AKAN
MENGHAPUS DATA BERIKUT ?
YA
TIDAK


M13
APAKAH ANDA YAKIN AKAN
MENAMBAH DATA BERIKUT ?
YA
TIDAK

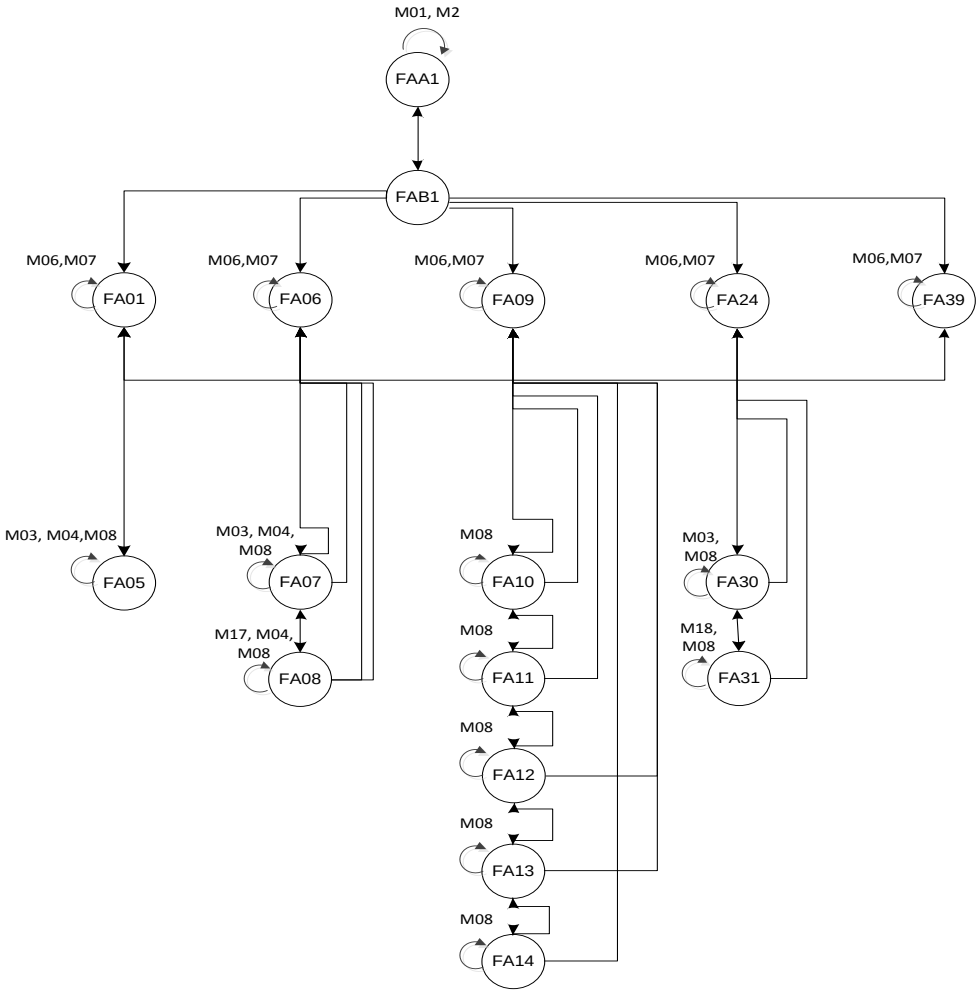

M14
APAKAH ANDA YAKIN AKAN
MENGHAPUS DATA BERIKUT ?
YA
TIDAK

3.2.2.3 Jaringan Semantik Admin



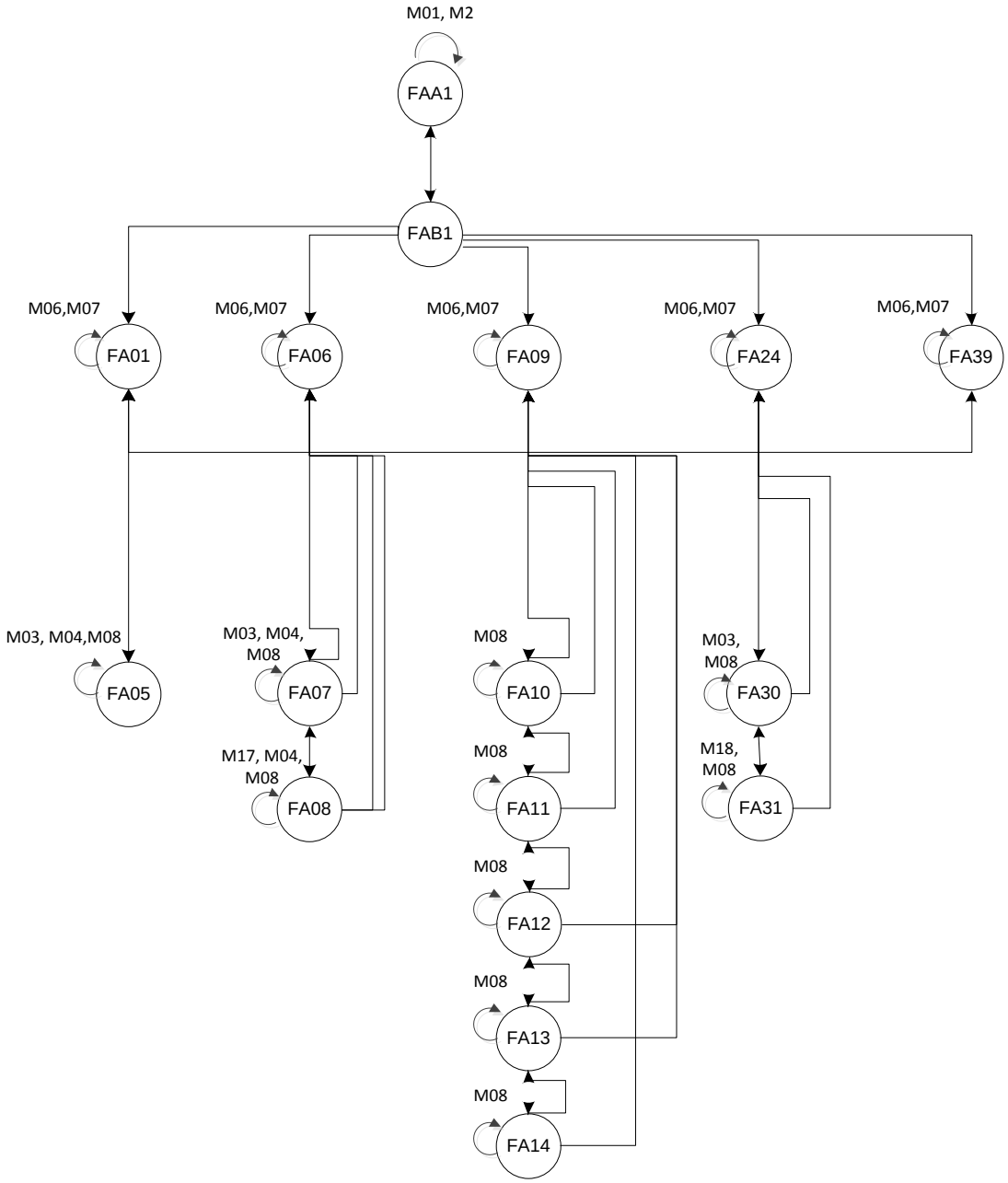
Gambar 3.87 Jaringan Semantik Admin

3.2.2.4 Jaringan Semantik Supervisor



Gambar 3.88 Jaringan Semantik Supervisor

3.2.2.5 Jaringan Semantik Asman



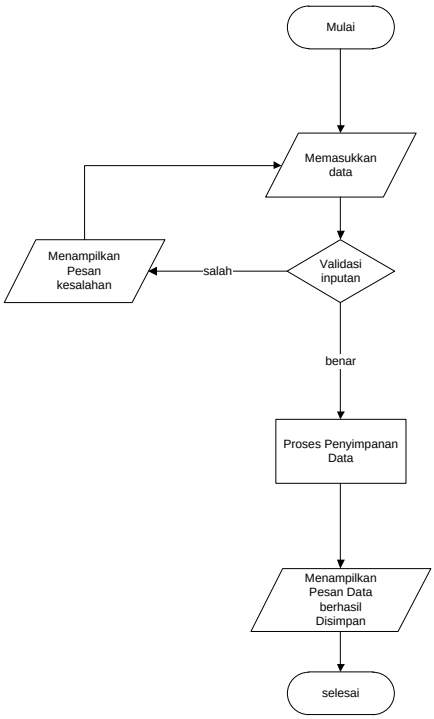
Gambar 3.89 Jaringan Semantik Asman

3.2.3 Perancangan Prosedural

Perancangan prosedural menggunakan *Flowchart* merupakan representasi algoritma dari perangkat lunak sistem informasi *inventory control*. Detail *Flowchart* sistem informasi *inventory control* adalah sebagai berikut :

3.2.3.1 *Flowchart* Penambahan Data

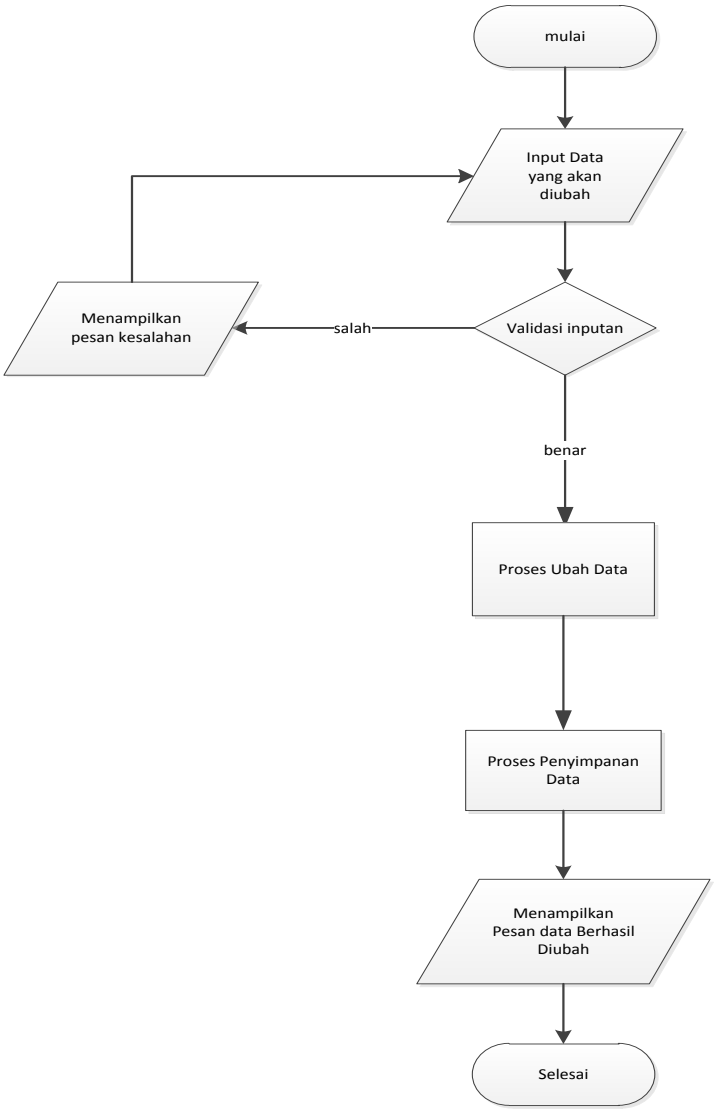
Flowchart penambahan data ini menggambarkan langkah-langkah yang dilakukan untuk penambahan data pada sistem yang dikembangkan.



Gambar 3.90 *Flowchart* Penambahan Data

3.2.3.2 *Flowchart* Perubahan Data

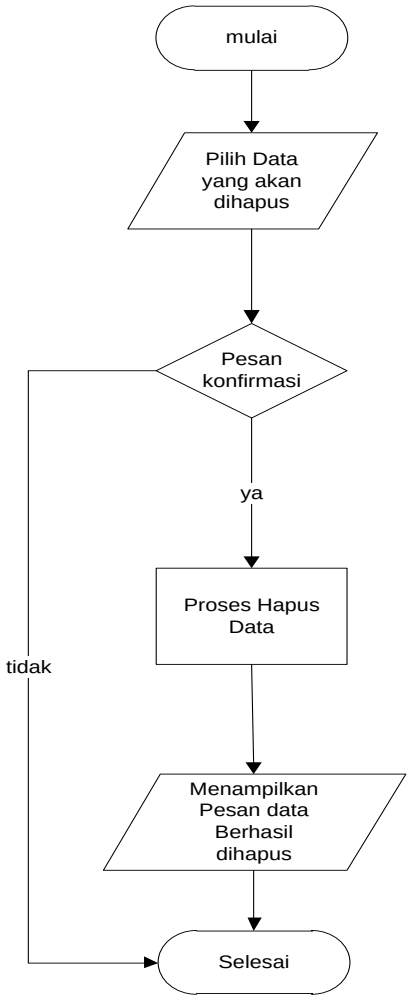
Flowchart pengubahan data ini menggambarkan langkah-langkah yang dilakukan untuk mengubah data pada sistem yang dikembangkan.



Gambar 3.91 *Flowchart* Perubahan Data

3.2.3.3 *Flowchart* Penghapusan Data

Flowchart penghapusan data ini menggambarkan langkah-langkah yang dilakukan untuk menghapus data pada sistem yang dikembangkan.



Gambar 3.92 *Flowchart* Penghapusan Data