BAB III
ANALISIS DAN PERANCANGAN

3.1. Analisis Masalah

Dalam pembahasan analisa masalah di Dinas Pertamanan dan Pemakaman kota Bandung dalam pengolahan datanya masih manual. Sehingga proses pengolahan data dan pembuatan laporan membutuhkan waktu yang relative lama, karena komputer yang sudah ada belum diaplikasikan secara optimal. Maka dari itu penulis menguraikan secara singkat tentang analisa masalah pada Dinas Pertamanan dan Pemakaman kota Bandung diantaranya adalah sebagai berikut :

1. Bagian kepegawaian sering terlambat mengajukan kenaikan gaji berkala bagi pegawai yang sudah berkerja diatas 2 tahun Kepada pemerintah kota bandung.
2. Bagian kepegawaian sering terlambat mengajukan kenaikan pangkat bagi pegawai yang sudah berkerja diatas 4 tahun Kepada pemerintah kota bandung.
3. Bagian kepegawaian sering terlambata mengajukan pensiun bagi pegawai yang usiannya sudah diatas 56 tahun Kepada pemerintah kota bandung.

3.2. Analisis Sistem yanng sedang berjalan

Menurut dari apa yang telah dilihat dari Analisis masalah. Sistem yang sedang berjalan di badan Kepegawaian terdapat 3 prosedur yaitu Prosedur kenaikan gaji berkala, Prosedur Kenaikan Pangkat dan pengajuan Pensiun.

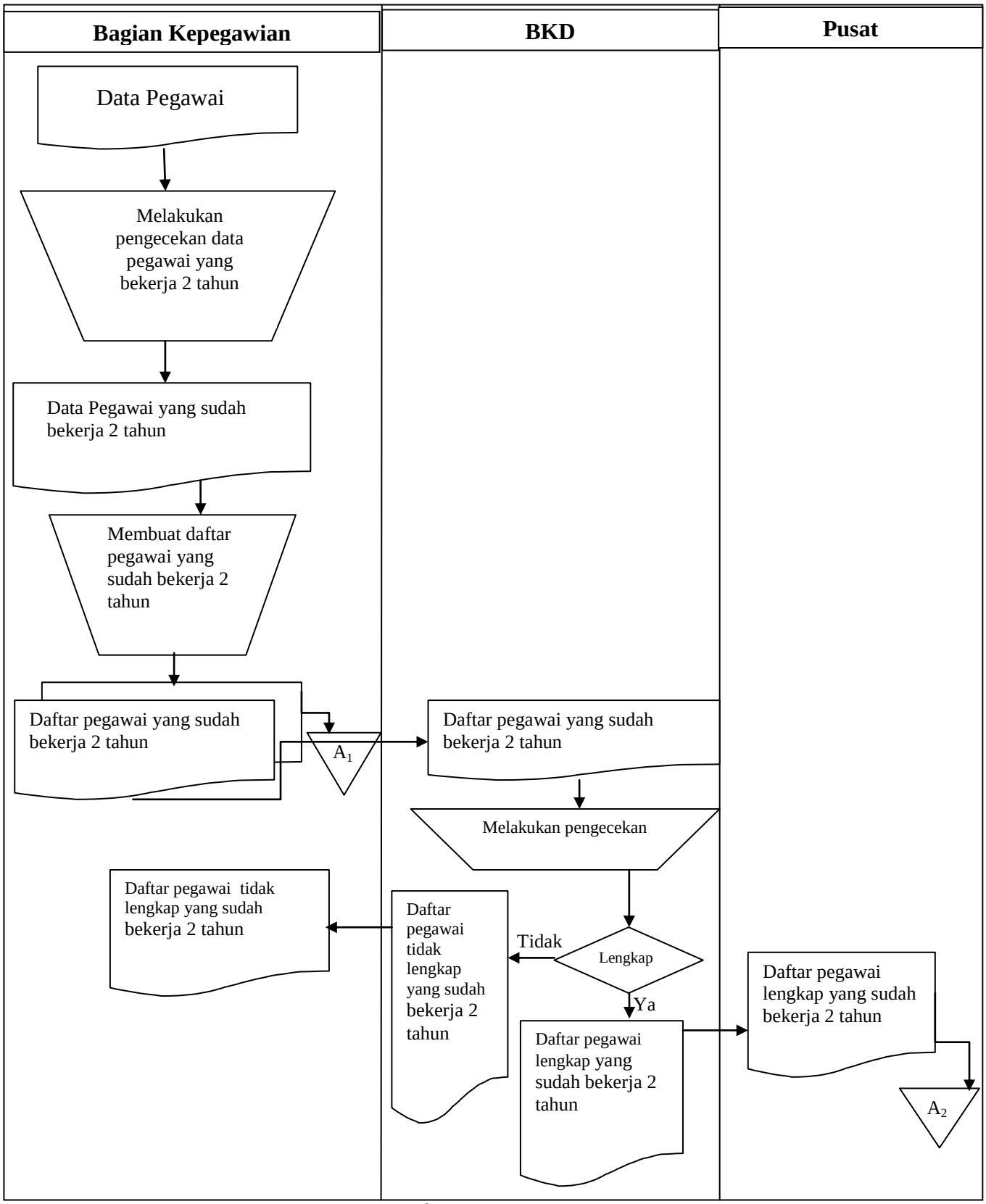
3.2.1. Prosedur Kenaikan gaji berkala

Prosedur kenaikan gaji berkala terdiri dari :

1. Bagian Kepegawaian melihat tanggal setiap pegawai dari data pegawai.
2. Bagian Kepegawaian mengecek apakah ada data pegawai yang sudah bekerja selama 2 tahun lebih.
3. Apabila terdapat data pegawai yang tepat 2 tahun lebih maka bagian kepegawaian membuat daftar pegawai yang sudah bekerja lebih dari 2 tahun lebih.

4. Daftar pegawai yang sudah bekerja 2 tahun tersebut diserahkan ke BKD
5. BKD memeriksa daftar tersebut apakah sudah lengkap atau tidak.
6. Apabila sudah lengkap maka daftar tersebut akan diserahkan ke pusat.
7. Jika belum lengkap maka akan dikembalikan lagi ke bagian kepegawaian.

3.2.2. Flowmap Kenaikan Gaji berkala



Gambar 3.1 flowmap kenaikan gaji berkala

Keterangan :

1 : Data Pegawai.

A1 : Arsip Data Pegawai.

A2 : Arsip Daftar pegawai lengkap yang sudah bekerja 2 tahun.

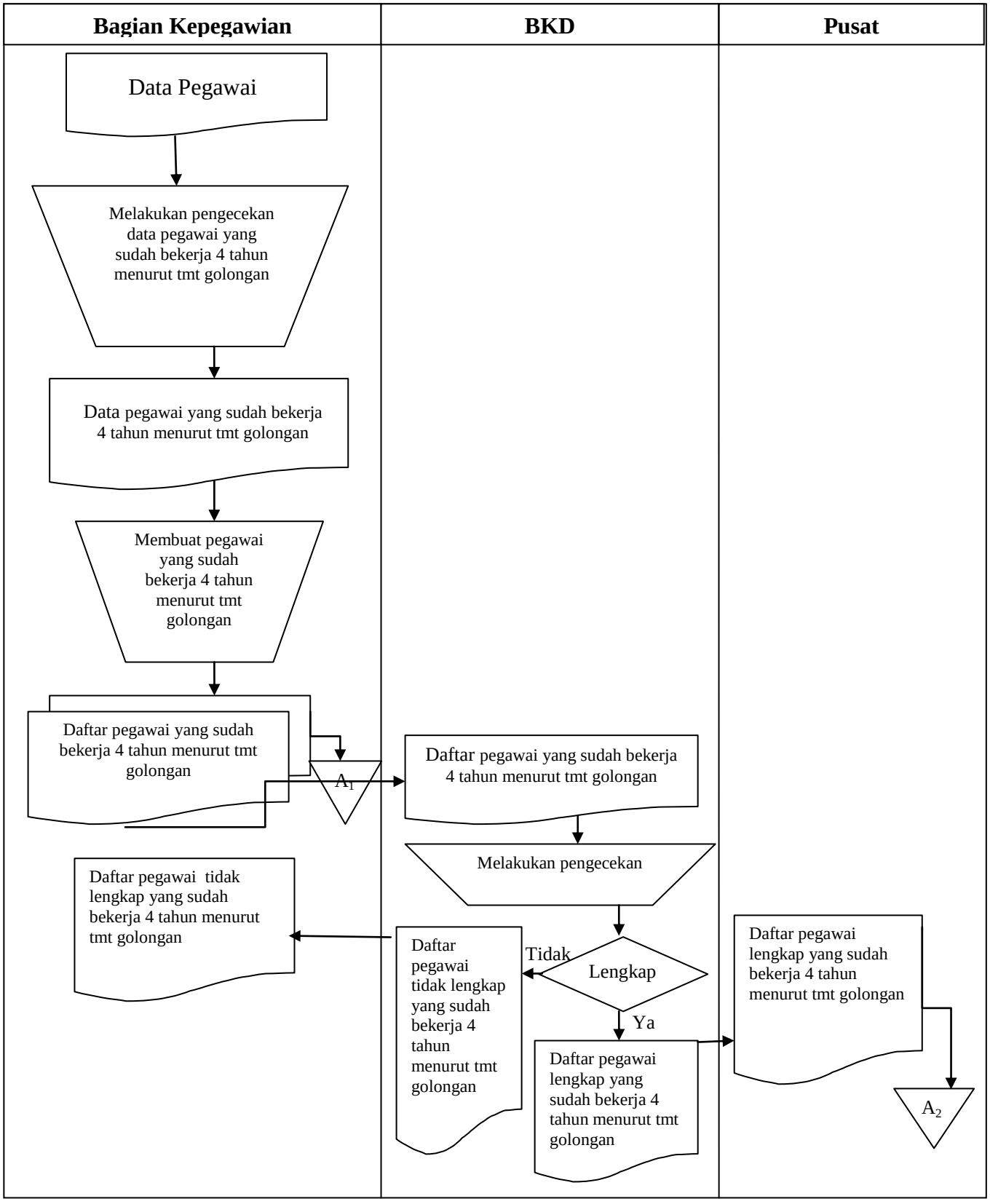
.

3.2.3. Prosedur Kenaikan Pengkat

Prosedur Kenaikan Pengkat terdiri dari :

1. Bagian Kepegawaian melihat tanggal golongan setiap pegawai dari data pegawai.
2. Bagian Kepegawaian mengecek apakah ada data pegawai yang sudah bekerja selama 4 tahun lebih yang dimulai dari tmt(terhitung mulai tanggal) golongan.
3. Apabila terdapat data pegawai yang tepat 4 tahun lebih maka bagian kepegawaian membuat daftar pegawai yang tmt golongannya sudah 4 tahun lebih.
4. Daftar pegawai yang tmt golongannya 4 tahun tersebut diserahkan ke BKD.
5. BKD memeriksa daftar tersebut apakah sudah lengkap atau tidak.
6. Apabila sudah lengkap maka daftar tersebut akan diserahkan ke pusat.
7. Jika belum lengkap maka akan dikembalikan lagi ke bagian kepegawaian.

3.2.4. Flowmap Kenaikan Pangkat



Gambar 3.2 flowmap kenaikan pangkat

Keterangan :

1 : Data Pegawai.

A1 : Arsip Data Pegawai.

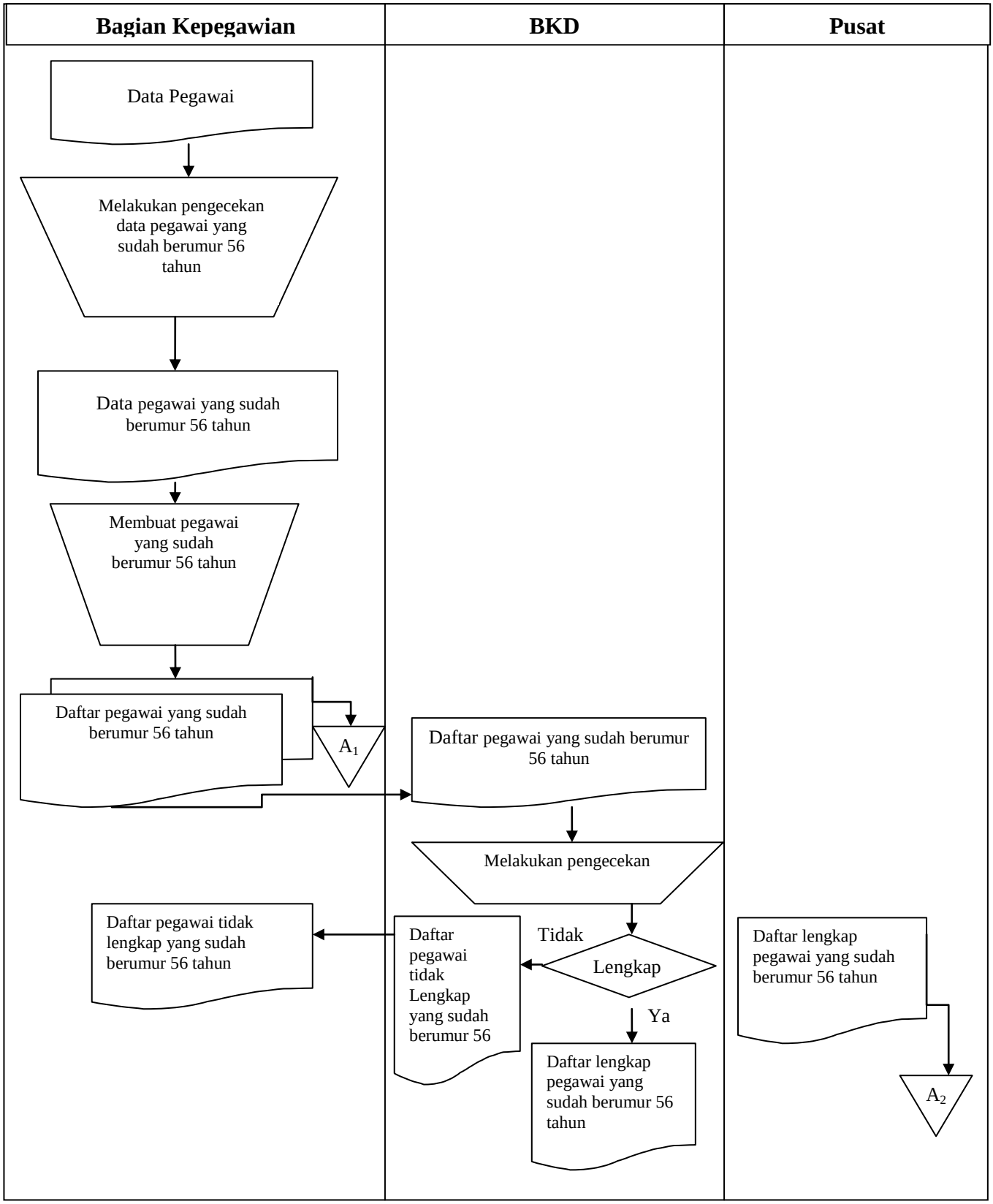
A2 : Arsip Daftar pegawai lengkap yang sudah bekerja 4 tahun menurut tmt golongan.

3.2.5. Prosedur Pensiun

Prosedur Pensiun terdiri dari :

1. Bagian Kepegawaian melihat tanggal Lahir setiap pegawai dari data pegawai.
2. Bagian Kepegawaian mengecek apakah ada data pegawai yang sudah berumur 56 tahun lebih.
3. Apabila terdapat data pegawai yang berumur 56 tahun lebih maka bagian kepegawaian membuat daftar pegawai yang berumur 56 tahun lebih.
4. Daftar pegawai yang umurnya sudah 56 tahun lebih tersebut diserahkan ke BKD.
5. BKD memeriksa daftar tersebut apakah sudah lengkap atau tidak.
6. Apabila sudah lengkap maka daftar tersebut akan diserahkan ke pusat.
7. Jika belum lengkap maka akan dikembalikan lagi ke bagian kepegawaian.

3.2.6. Flowmap Pengajuan Pensiun



Gambar 3.3 flowmap pengajuan pensiun

Keterangan :

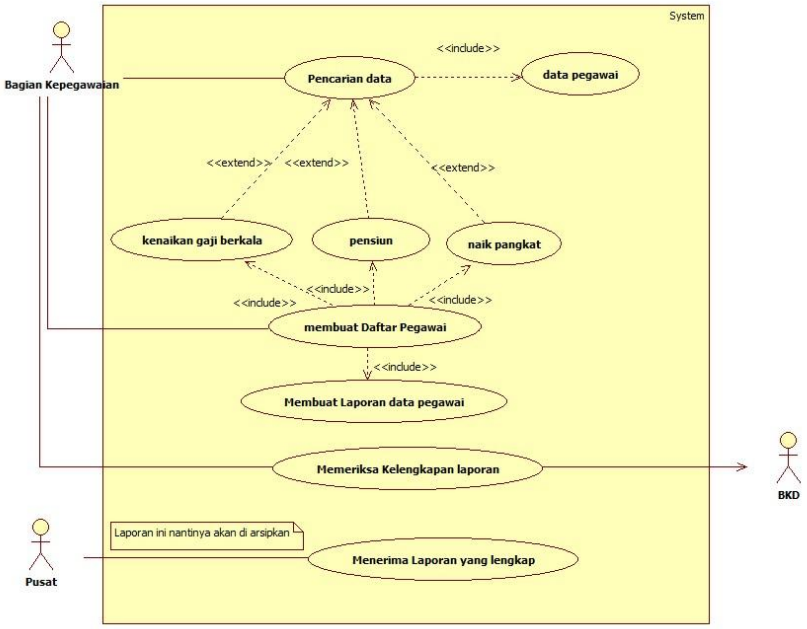
1 : Data Pegawai.

A1 : Arsip Data Pegawai.

A2 : Arsip Daftar lengkap pegawai yang sudah berumur 56 tahun.

3.2.7. *Use case diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

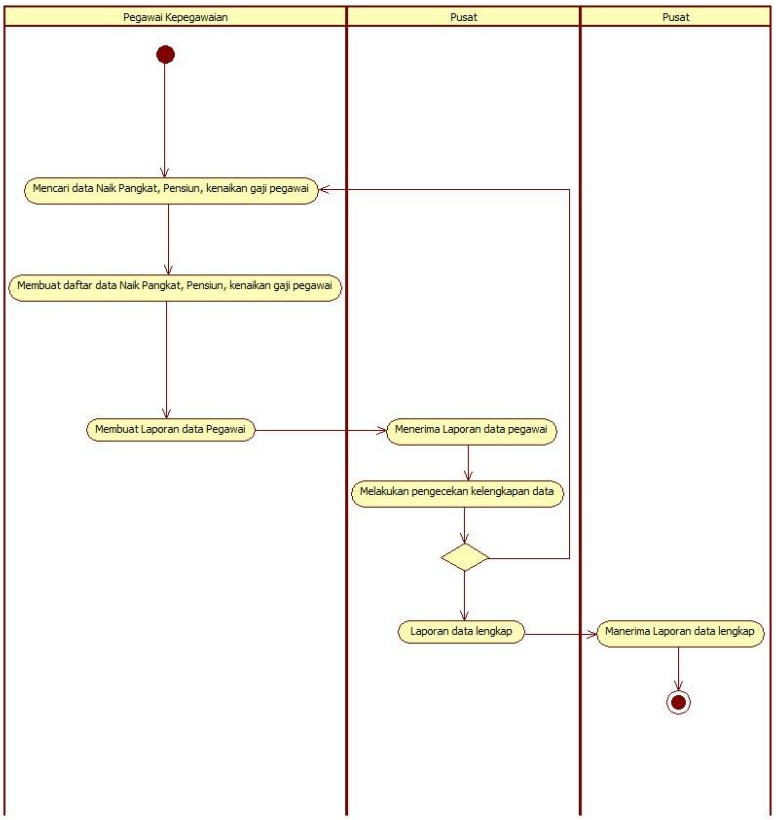


Gambar 3.4 *Use case diagram*

3.2.8. *Activity diagram*

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi ditrigger oleh selesainya state sebelumnya *(internal processing)*. Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.



Gambar 3.5 *Activity diagram*

3.2.9. *Sequence diagram*

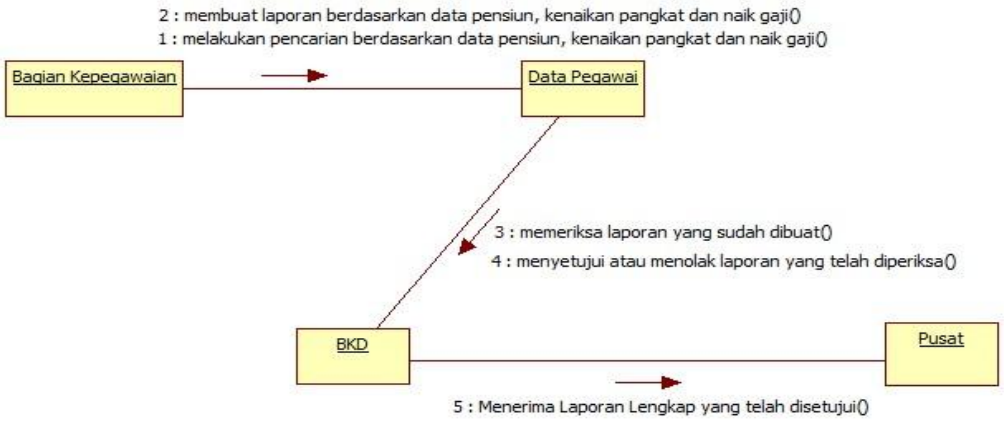
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.



Gambar 3.6 *Sequence diagram*

3.2.10. *Collaboration diagram*

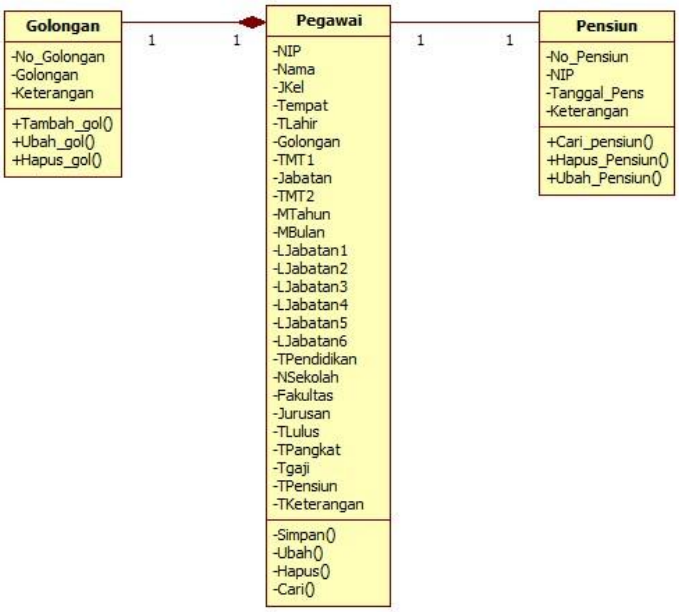
Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*.



Gambar 3.7 *Collaboration diagram*

3.2.11. *Class Diagram*

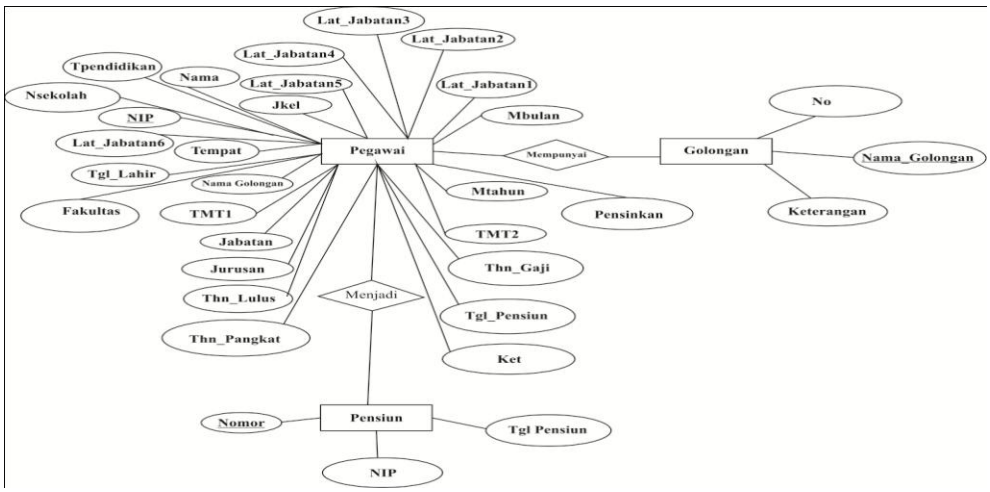
Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class Diagram* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class Diagram* menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.



Gambar 3.8 *Class diagram*

3.2.12. Entity Relational Diagram (ERD)

ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam system secara abstrak. Model data E-R (Entity Relationship) didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang dissebut entitas dan relasi.



Gambar 3.9 *entity relational diagram*

3.3. Analisis Non Fungsional

3.3.1. Analisis Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan dalam Sistem informasi pengambilan keputusan adalah :

1. Processor dengan kapasitas minimum 1800Mhz,
2. Minimum Ram 512 Mb,
3. Minimumj Harddisk 40 Gb,
4. Cd Rom drive,
5. Monitor 15‘,
6. Keyboard,
7. Mouse.

3.3.2. Analisis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan dalam Aplikasi Surat Menyurat adalah:

1. Windows XP, windows 7
2. *MySQL*
3. *NetBeans IDE 6.9.1*
4. *JDK 6u24*

3.3.3. Analisis Kebutuhan Pengguna

Kebutuhan pengguna dalam Sistem informasi pengambilan keputusan adalah:

1. Bagian Kepegawaian

3.4. Perancangan Tabel

3.4.1. Perancangan Tabel Pegawai

Field	Type	Collation	Attributes	Null	Default	Extra
NIP	varchar(50)	*latin1_swedish_ci*		No	*None*	
Nama	varchar(200)	*latin1_swedish_ci*		No	*None*	
JKel	enum('L', 'P')	*latin1_swedish_ci*		No	*None*	
Tempat	varchar(100)	*latin1_swedish_ci*		No	*None*	
Tgl_Lahir	date			No	*None*	
Gol	varchar(10)	*latin1_swedish_ci*		No	*None*	
TMT1	date			No	*None*	

Jabatan	varchar(150)	*latin1_swedish_ci*		No	*None*	
TMT2	date			Yes	*NULL*	
MTahun	int(11)			No	*None*	
MBulan	int(11)			No	*None*	
Lat_Jabatan1	varchar(50)	*latin1_swedish_ci*		No	*None*	
Lat_Jabatan2	varchar(50)	*latin1_swedish_ci*		No	*None*	
Lat_Jabatan3	varchar(50)	*latin1_swedish_ci*		No	*None*	
Lat_Jabatan4	varchar(50)	*latin1_swedish_ci*		No	*None*	
Lat_Jabatan5	varchar(50)	*latin1_swedish_ci*		No	*None*	
Lat_Jabatan6	varchar(50)	*latin1_swedish_ci*		No	*None*	
TPendidikan	varchar(20)	*latin1_swedish_ci*		No	*None*	
NSekolah	varchar(150)	*latin1_swedish_ci*		No	*None*	
Fakultas	varchar(100)	*latin1_swedish_ci*		No	*None*	
Jurusan	varchar(150)	*latin1_swedish_ci*		No	*None*	
Thn_Lulus	int(11)			No	*None*	
Thn_Pangkat	date			Yes	*NULL*	
Thn_Gaji	date			Yes	*NULL*	
Tgl_Pensiun	date			Yes	*NULL*	
ket	varchar(250)	*latin1_swedish_ci*		No	*None*	
Pensiunkan	enum('TIDAK', 'IYA')	*latin1_swedish_ci*		No	*None*	

Tabel 3.1 Perancangan tabel pegawai

3.4.2. Perancangan Tabel Golongan

Field	Type	Collation	Attributes	Null	Default	Extra
No	int(11)			No	*None*	AUTO_INCREMENT
Nama_Golongan	varchar(15)	*latin1_swedish_ci*		No	*None*	
Keterangan	varchar(250)	*latin1_swedish_ci*		No	*None*	

Tabel 3.2 Perancangan tabel golongan

3.4.3. Perancangan Tabel Pensiun

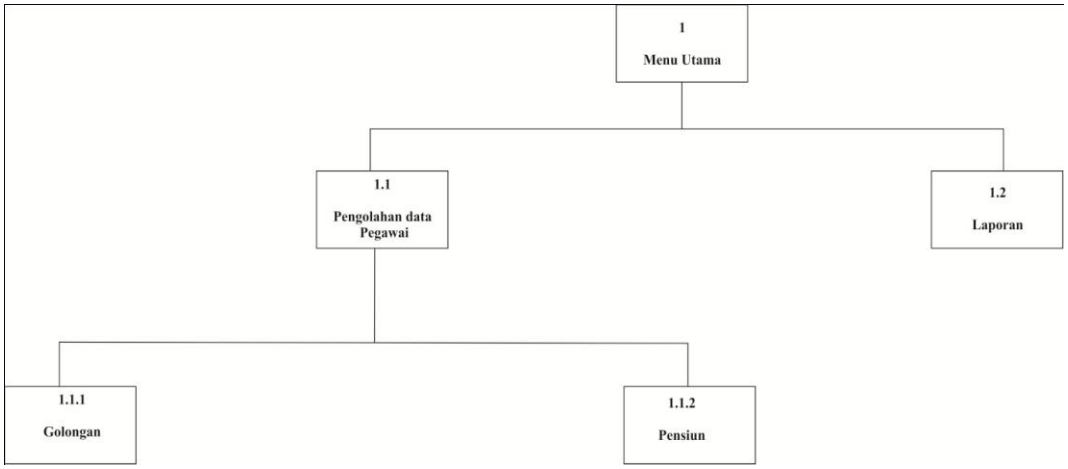
Field	Type	Collation	Attributes	Null	Default	Extra
Nomor	int(11)			No	*None*	AUTO_INCREMENT
NIP	varchar(50)	*latin1_swedish_ci*		No	*None*	
Tanggal_Pensiun	date			No	*None*	

Tabel 3.3 Perancangan table pensiun

3.5. Perancangan Menu

Perancangan menu adalah perancangan antar muka pilihan perintah pada program aplikasi untuk mengoprasikan dan memudahkan pemakai

dalam menjalankan program. Struktur menu aplikasi yang di usulkan dapat di lihat pada gambar berikut :

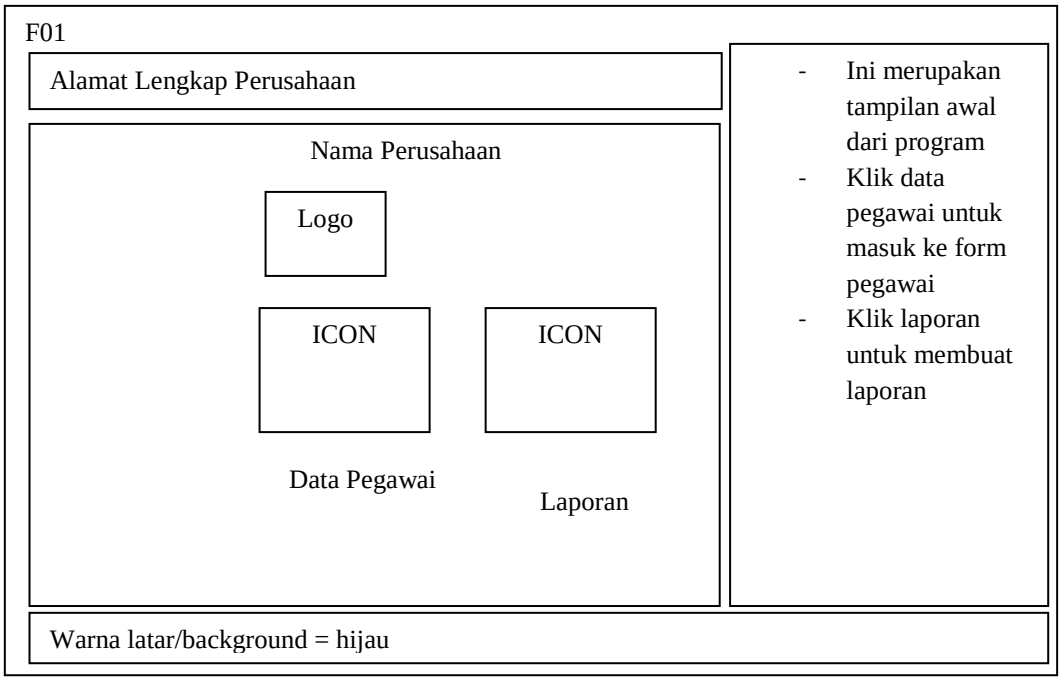


Gambar 3.10 Perancangan Menu

3.6. Perancangan Atarmuka

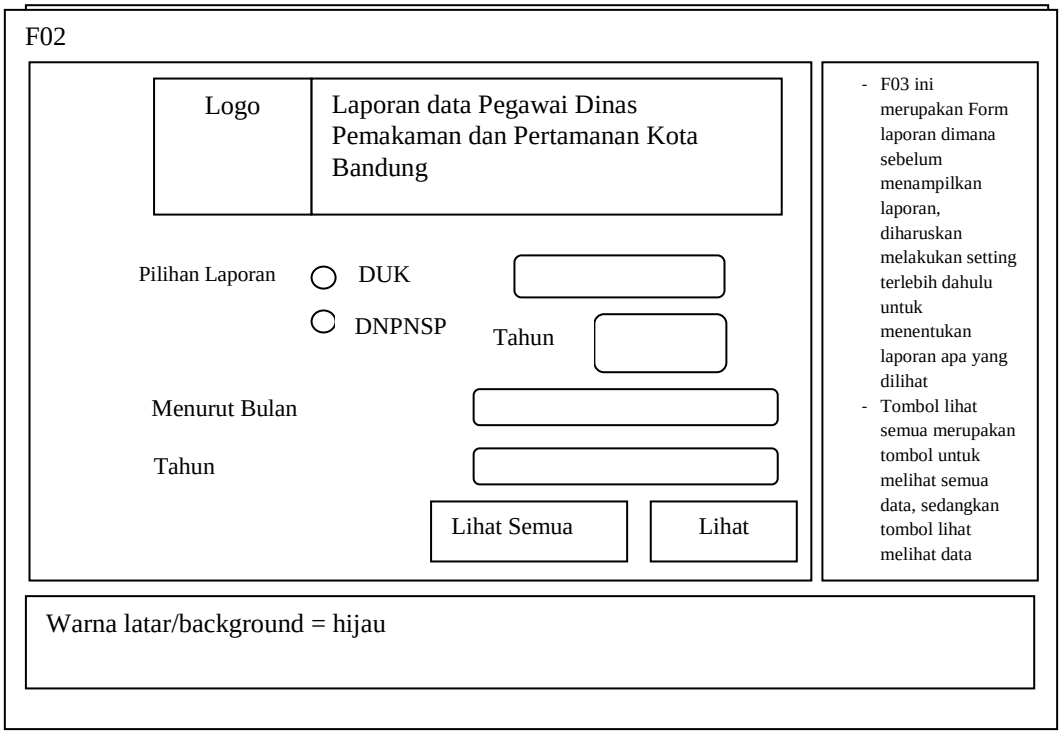
Antarmuka perangkat lunak merupakan media komunikasi pengguna saat akan berinteraksi dengan sistem. Oleh karena itu agar proses interaksi pengguna dengan sistem berjalan dengan baik maka perlu diketahui terlebih dahulu sejauhmana performansi profile pengguna antarmuka, sehingga antarmuka yang dihasilkan dapat dengan mudah dipelajari, mudah dan nyaman pada saat digunakan. Performansi pengguna dapat diukur dari beberapa *profile* pengguna, dimana setiap *profile* akan memiliki implikasi terhadap perancangan antarmuka.

3.6.1. Perancangan Form Menu



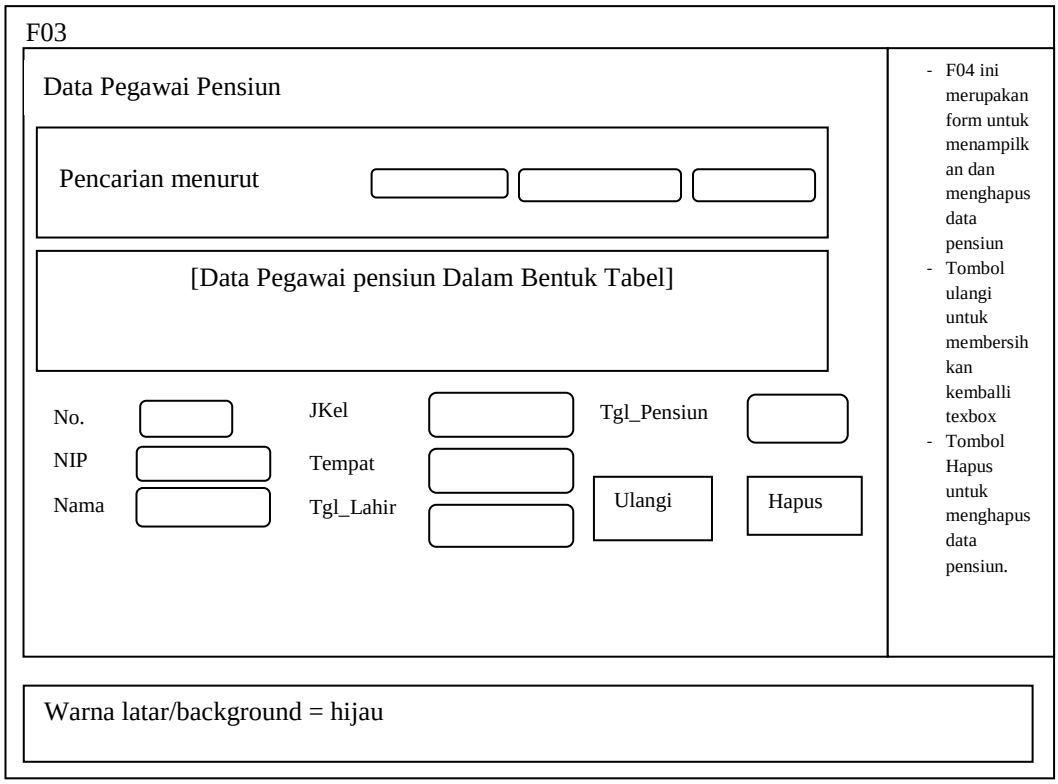
Gambar 3.11 Perancangan form menu

3.6.2. Perancangan Form Menu Laporan



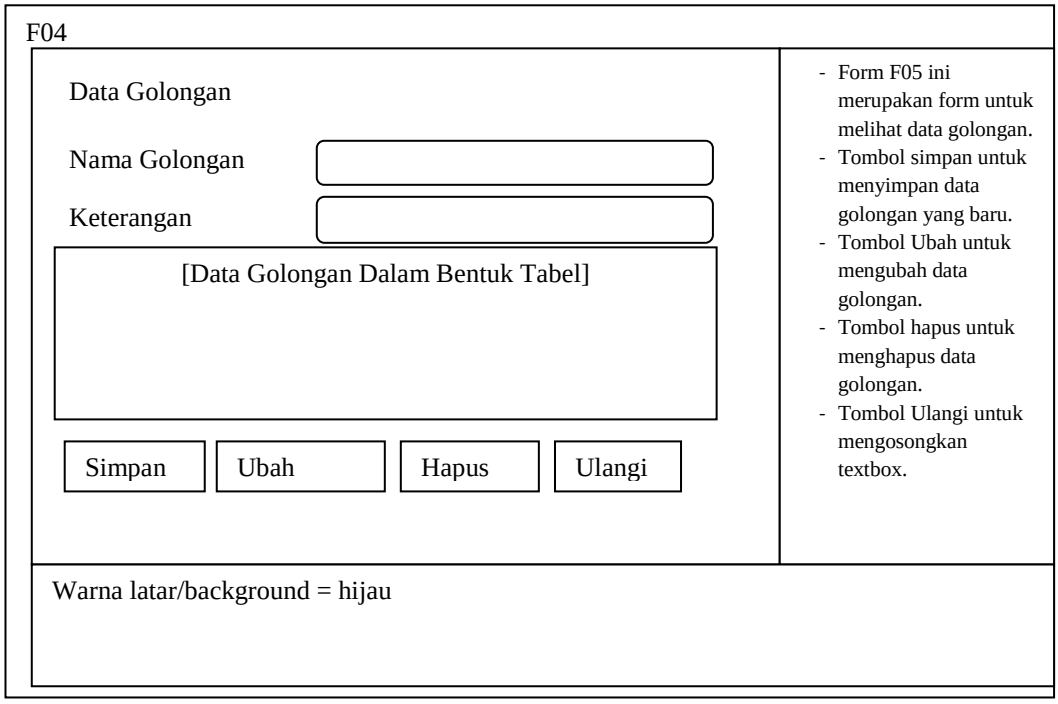
Gambar 3.12 Perancangan form menu laporan

3.6.3. Perancangan Form Menampilkan dan menghapus Data Pensiun



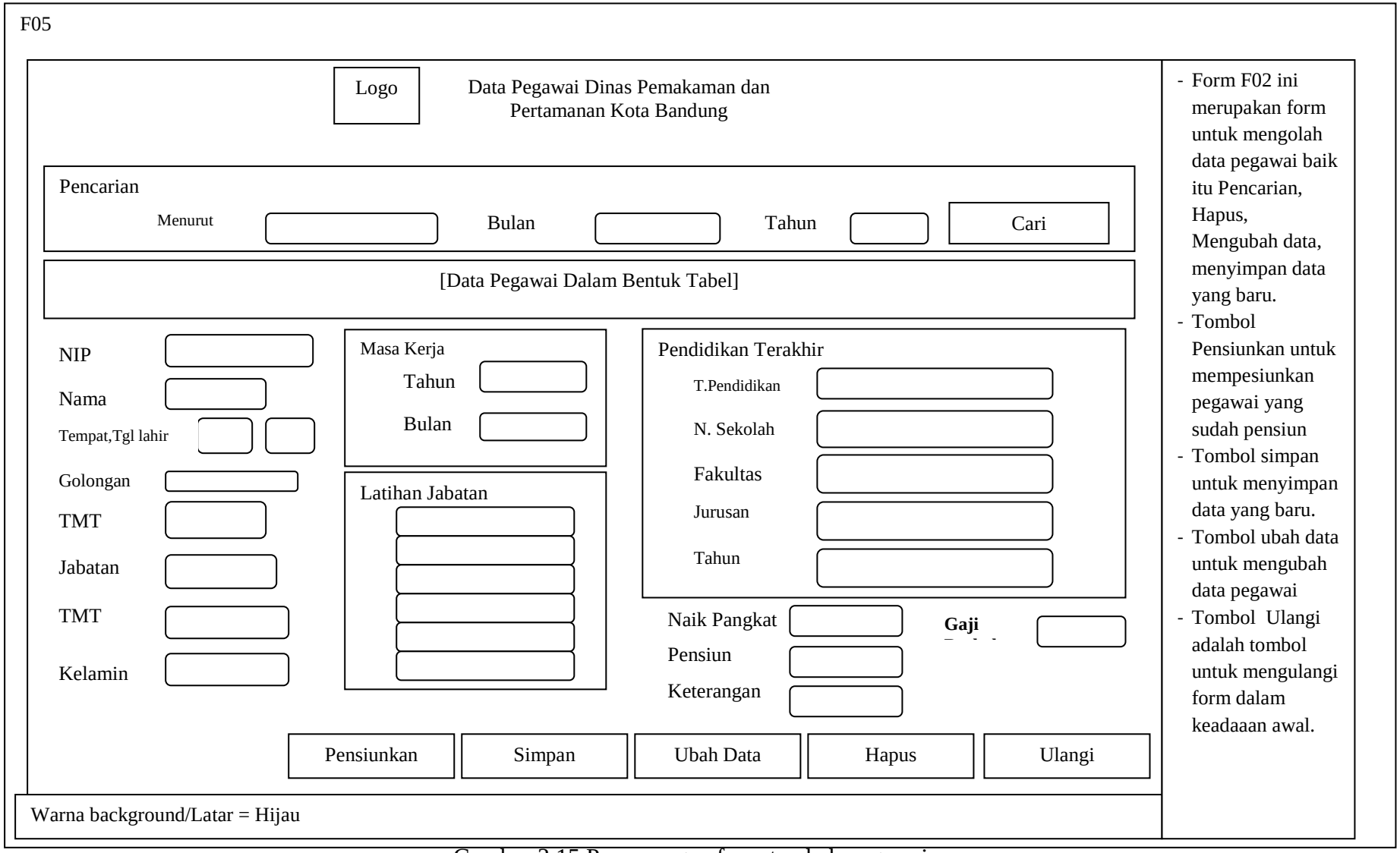
Gambar 3.13 Perancangan form menampilkan dan menghapus data pensiun

3.6.4. Perancangan Form Menampilkan Data Golongan



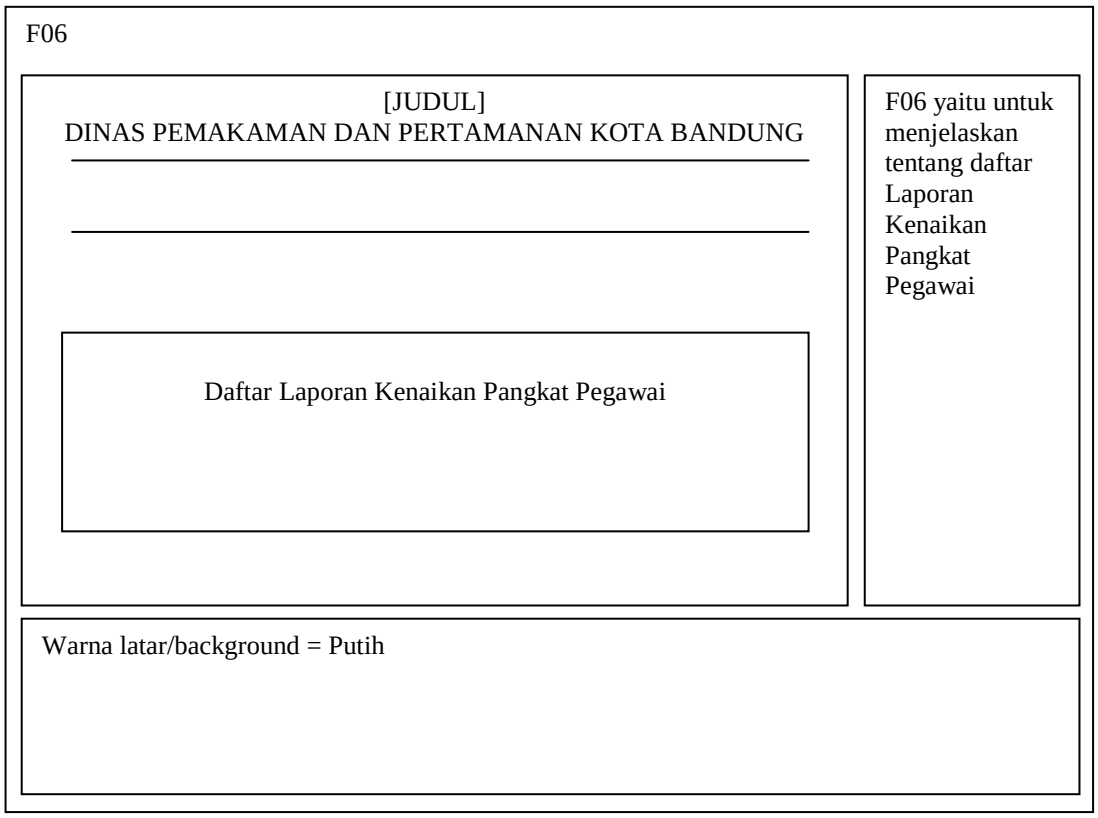
Gambar 3.14 perancangan form menampilkan data golongan

3.6.5. Perancangan Form Tambah Pegawai



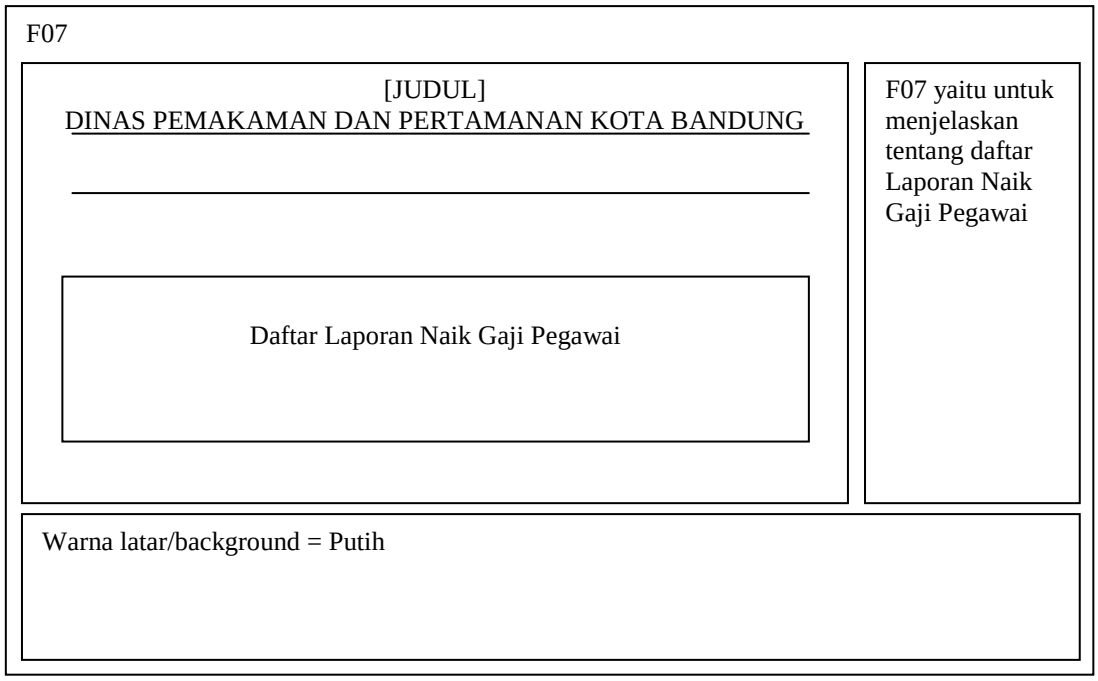
Gambar 3.15 Perancangan form tambah pegawai

3.6.6. Perancangan form Laporan Kenaikan Pangkat



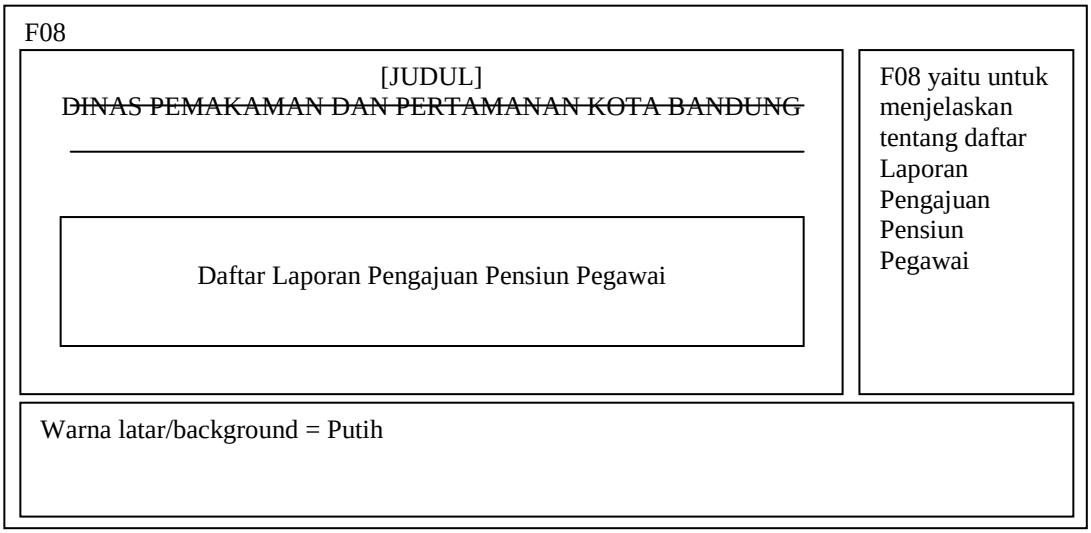
Gambar 3.16 Perancangan form laporan kenaikan pangkat

3.6.7. Perancangan form Laporan Kenaikan Gaji



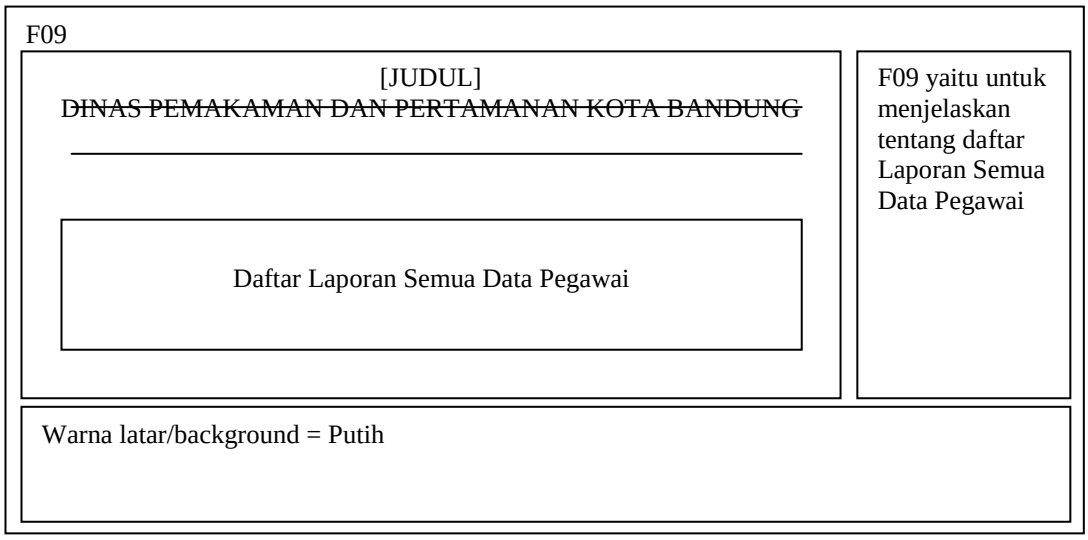
Gambar 3.17 Perancangan form laporan kenaikan gaji

3.6.8. Perancangan form Laporan Pengajuan Pensiun



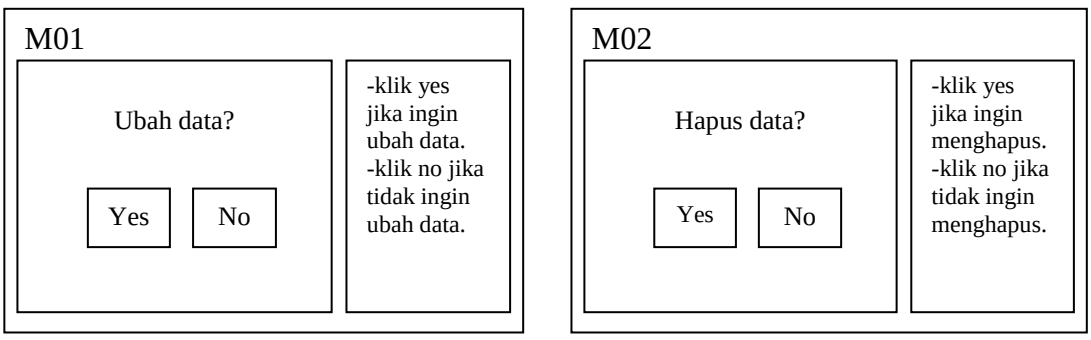
Gambar 3.18 perancangan form laporan pengajuan pensiun

3.6.9. Perancangan form laporan Semua Data Pegawai



Gambar 3.19 perancangan form laporan semua data pegawai

3.6.10. Perancangan form Pesan

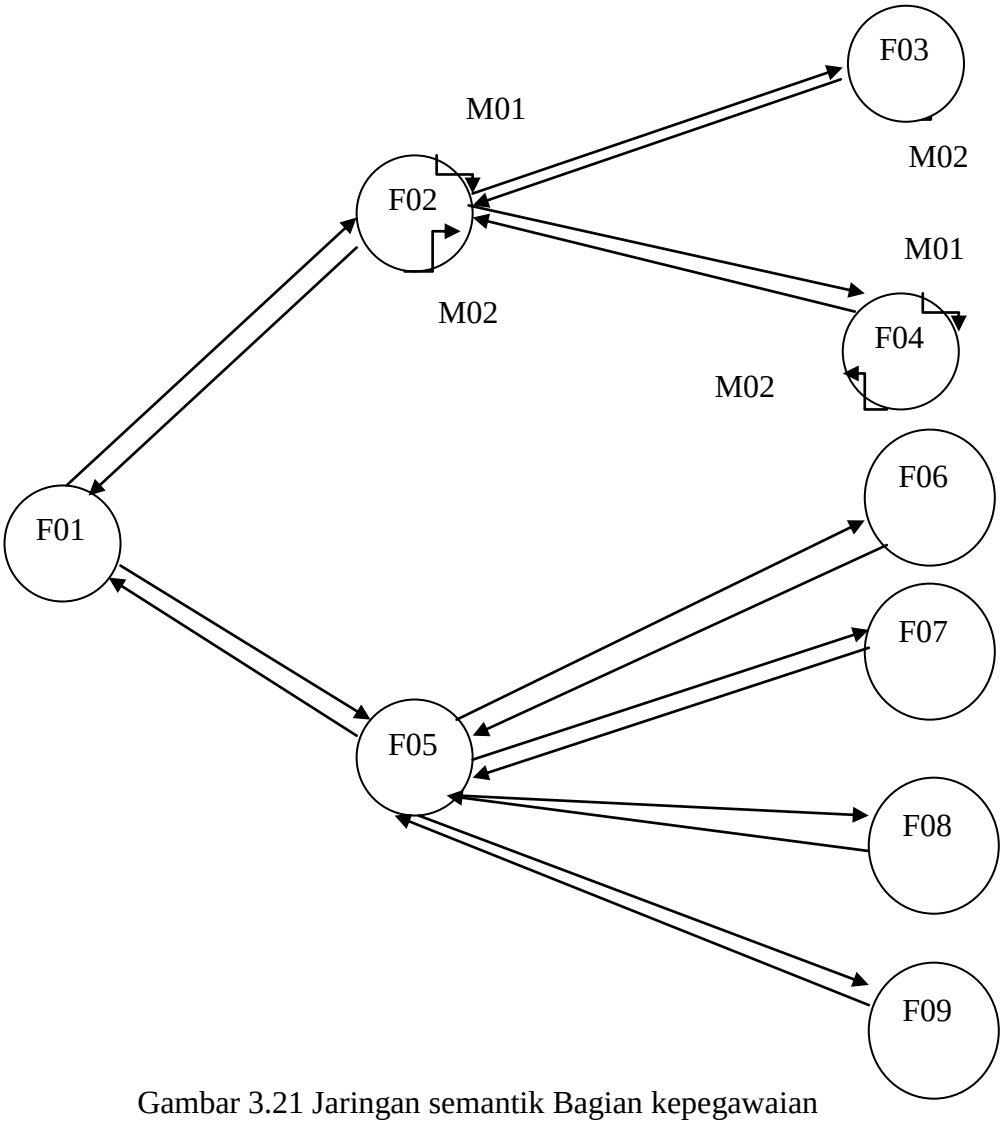


Gambar 3.20 Perancangan form pesan

3.7. Jaringan Semantik

Jaringan semantik merupakan jaringan data dan informasi, yang menunjukan hubungan antar berbagai objek dimana informasi yang terhubung tersebut adalah informasi yang proporsional (suatu pernyataan yang dapat bernilai benar atau salah). Dalam matematika, istilah jaringan semantik merupakan suatu label atau graph berarah. Struktur jaringan semantik terdiri dari *node* atau simpul dan busur atau *arc* yang menghubungkannya. Simpul menyatakan objek sedangkan busur menyatakan *links. Links* dari jaringan semantik digunakan untuk menunjukkan hubungan (*relationship*) antar simpul-simpul tersebut.

3.7.1. Jaringan Semantik Untuk Bagian Kepegawaian



Gambar 3.21 Jaringan semantik Bagian kepegawaian