BAB III

ANALISIS DAN PERANCANGAN

3.1. Analisis Fungsional

Analisis fungsional pada Aplikasi Surat Menyurat ini terdiri dari:

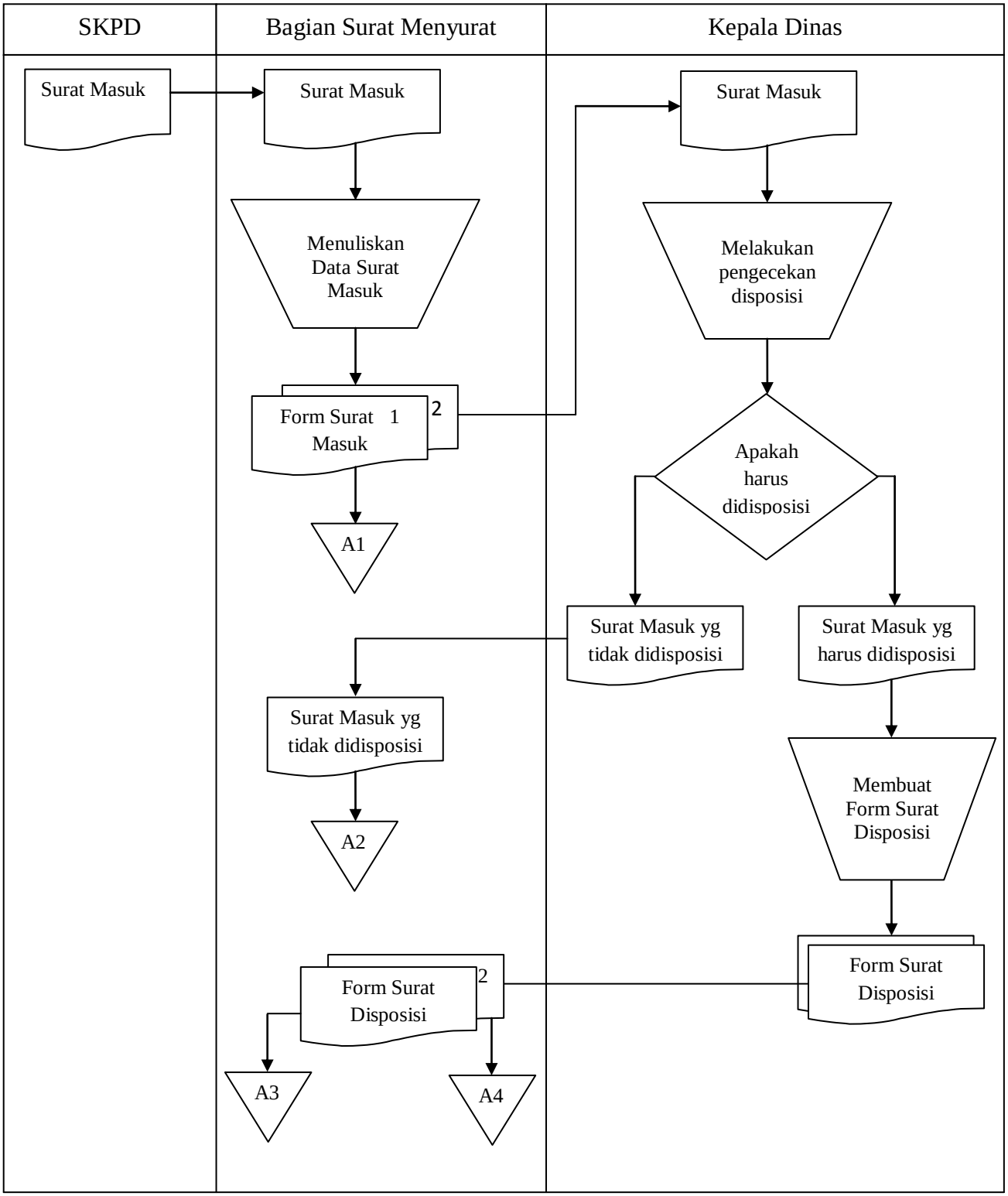
3.1.1. Prosedur dan *Flowmap*

Prosedur dan *flowmap* pada Aplikasi Surat Menyurat ini terdiri dari prosedur surat masuk serta *flowmap* untuk surat masuk dan prosedur surat keluar serta *flowmap* untuk surat keluar.

3.1.1.1. Prosedur Surat Masuk

1. Bagian surat menyurat menerima surat yang masuk dari SKPD.
2. Surat masuk tersebut dicatatkan semua datanya ke dalam form surat masuk.
3. Form surat masuk diarsipkan oleh bagian surat menyurat, sedangkan suratmasuk diserahkan ke Kepala Dinas.
4. Lalu Kepala Dinas melakukan pengecekan surat masuk, apakah harus di disposisi atau tidak.
5. Jika ya, maka Kepala Dinas membuat form surat disposisi, kemudian form surat disposisi dan surat masuk diserahkan kembali ke bagian surat menyurat untuk diserahkan kepada pihak yang ditujukan.
6. Jika tidak, surat tersebut diserahkan kembali ke bagian surat menyurat. Lalu bagian surat menyurat mengarsipkan surat tersebut unuk diserahkan kepada pihak yang ditujukan.

3.1.1.2.*Flowmap* Surat Masuk



Gambar 3.1 Flowmap Surat Masuk

Keterangan :

1 : Form Surat Masuk

2 : Surat Masuk

A1 : Arsip Form Surat Masuk

A2 : Arsip Surat Masuk yang tidak di disposisi

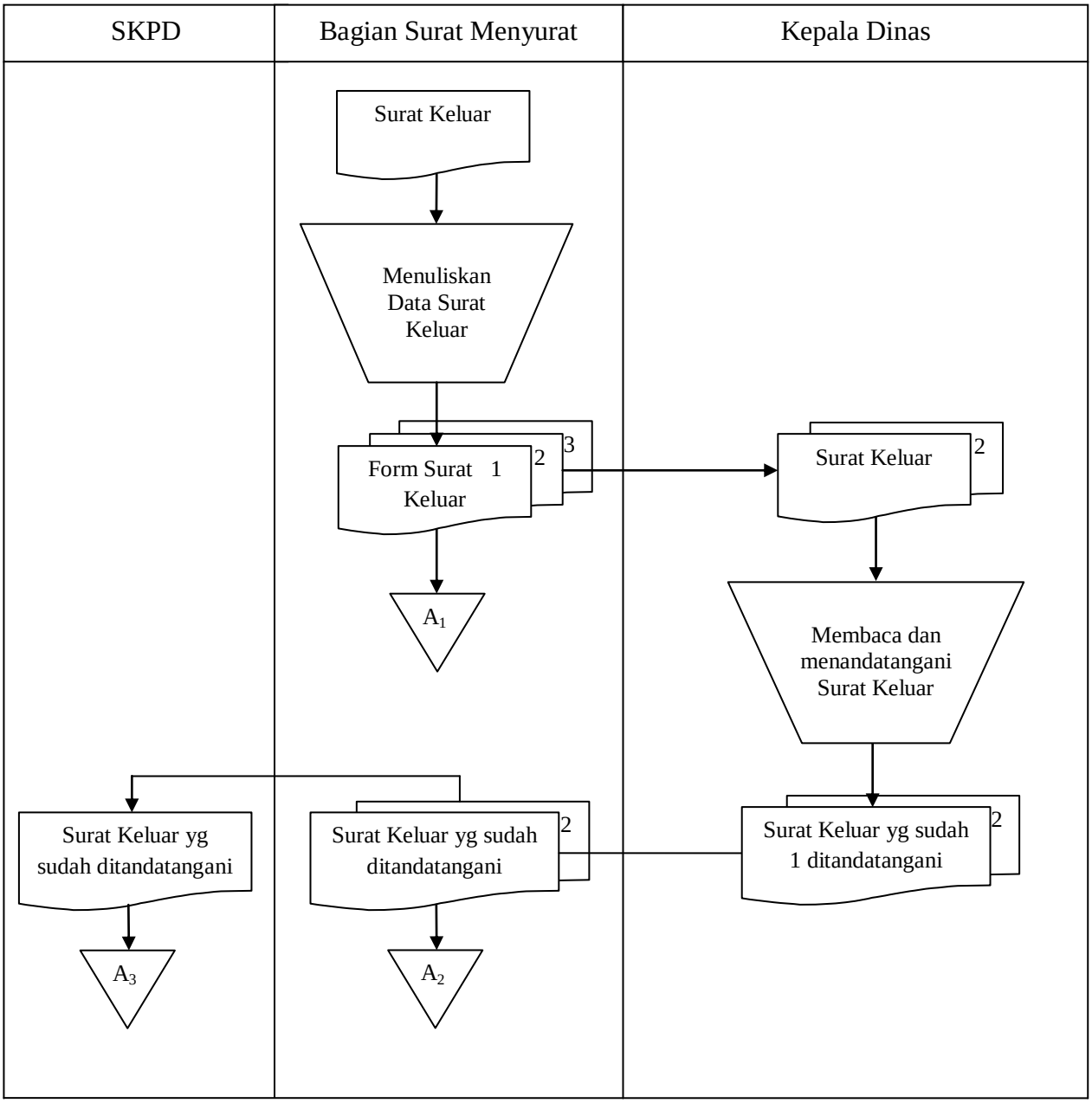
A3 : Arsip Form Surat Disposisi

A4 : Arsip Surat Masuk yang sudah di disposisi

3.1.1.3. Prosedur Surat Keluar

1. Bagian surat menyurat membuat surat keluar.
2. Bagian surat menyurat mencatat data surat keluar ke dalam form surat keluar.
3. Form surat keluar diarsipkan oleh bagian surat menyurat, sedangkan surat keluar diserahkan ke Kepala Dinas.
4. Kepala Dinas membaca dan menandatangani surat keluar.
5. Kemudian surat keluar yang sudah ditandatangani diserahkan kembali kepada bagian surat menyurat.
6. Surat keluar yang sudah ditandatangani diberikan kepada SKPD.

3.1.1.4.*Flowmap* Surat Keluar



Gambar 3.2 Flowmap Surat Masuk

Keterangan :

1 : Form Surat Keluar

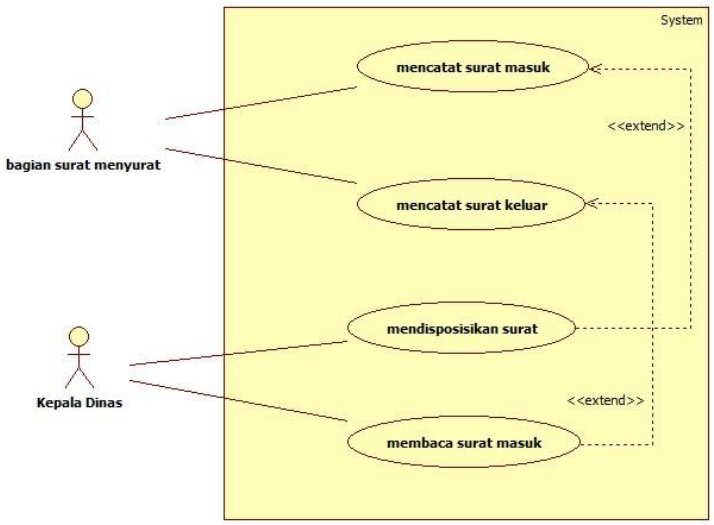
2,3 : Surat Keluar

A1 : Arsip Surat Keluar

A2,A3 : Arsip Surat Keluar yang sudah ditandatangani

3.1.2. *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.

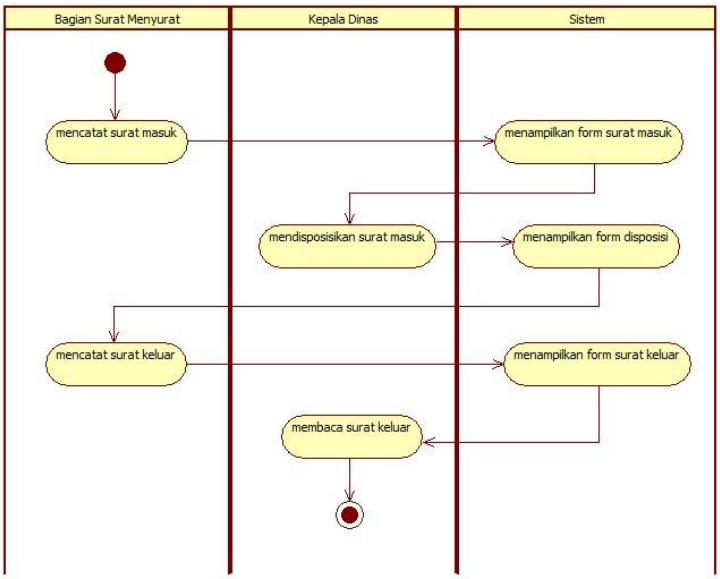


Gambar 3.3 *Use Case Diagram*

3.1.3. *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.

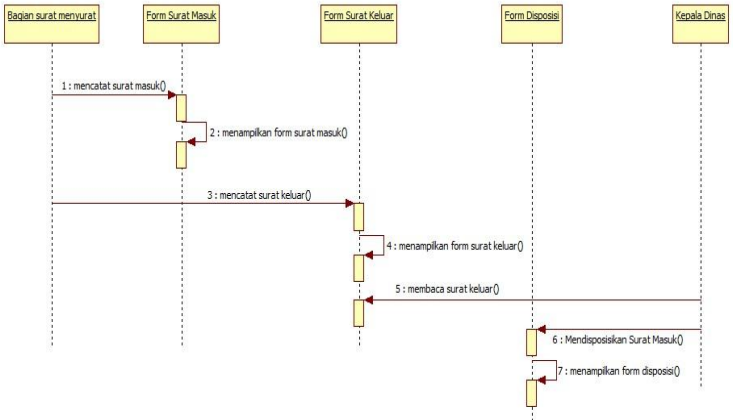
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.



Gambar 3.4 *Activity Diagram*

3.1.4. *Sequence Diagram*

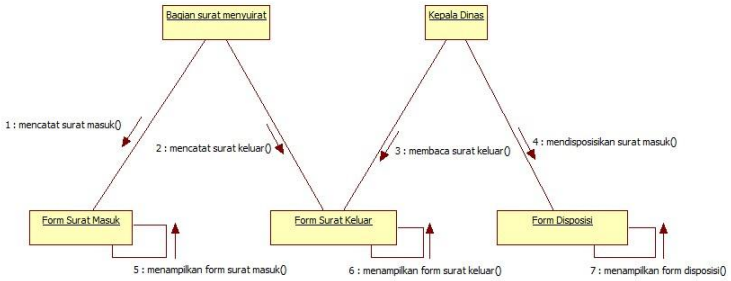
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* 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 3.5 *Sequence Diagram*

3.1.5. *Collaboration Diagram*

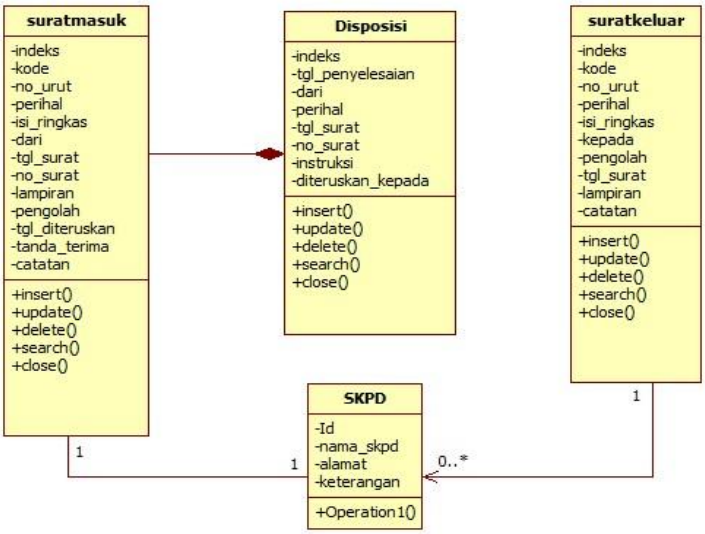
Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*.



Gambar 3.6 *Collaboration Diagram*

3.1.6. *Class Diagram*

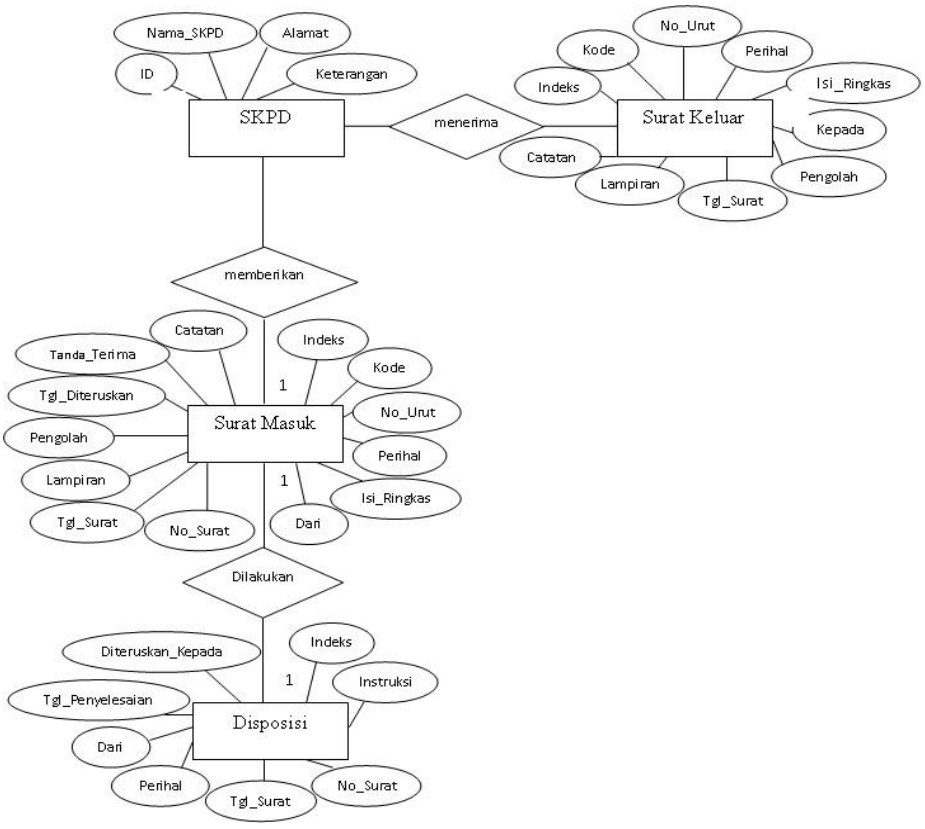
Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class Diagram* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).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



Gambar 3.7 *Class Diagram*

3.1.7. *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 3.8 ERD

3.2. Analisis Non Fungsional

3.2.1. Analisis Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan dalam Aplikasi Surat Menyurat adalah komputer.

3.2.2. Analisis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan dalam Aplikasi Surat Menyurat adalah:

1. *Windows XP, Windows 7* atau *Linux*
2. *MySQL*

3. *NetBeans IDE 6.9.1*
4. *jdk-6u24-windows-i586*

3.2.3. Analisis Kebutuhan Pengguna

Kebutuhan pengguna dalam Aplikasi Surat Menyurat adalah:

1. Bagian Surat Menyurat
2. Kepala Dinas

3.3. Perancangan Tabel

Perancangan tabel yang digunakan dalam aplikasi surat menyurat ini adalah :

Tabel 3.1 Perancangan Tabel Login

Nama	Email	Alamat	Username	Userpass	jabatan

Tabel 3.2 Perancangan Tabel Surat Masu**k**

Indeks	Kode	No.Urut	Perihal	Isi Ringkas	Dari	Tgl.Surat	No.Surat	Lampiran	Pengolah	Tgl.Diteruskan	Tanda Terima	Catatan

Tabel 3.3 Perancangan Tabel Surat Keluar

Indeks	Kode	No.Urut	Perihal	Isi Ringkas	Kepada	Pengolah	Tgl.Surat	Lampiran	Catatan

Tabel 3.4 Perancangan Tabel Disposisi

Indeks	Tgl.Penyelesaian	Dari	Perihal	Kepada	Tgl.Surat	No.Surat	Instruksi	Diteruskan Kepada

Tabel 3.5 Perancangan Tabel SKPD

ID	Nama SKPD	Alamat	Keterangan

Tabel 3.6 Perancangan Tabel Kode Surat

Kode	Nama

3.4. Struktur Tabel

Struktur tabel yang digunakan dalam aplikasi surat menyurat ini yaitu:

Tabel 3.7 Struktur Tabel Login

Field	Type	Collation	Attributes	Null	Default	Axtra
nama	varchar(30)	*latin1_swedish_ci*		No	*None*	
email	varchar(50)	*latin1_swedish_ci*		No	*None*	
alamat	varchar(150)	*latin1_swedish_ci*		No	*None*	
username	varchar(20)	*latin1_swedish_ci*		No	*None*	
userpass	varchar(100)	*latin1_swedish_ci*		No	*None*	
jabatan	enum('Sudah','Belum')	*latin1_swedish_ci*		No	*None*	

Tabel 3.8 Struktur Tabel Surat Masuk

Field	Type	Collation	Attributes	Null	Default	Axtra
indeks	varchar(50)	*latin1_swedish_ci*		No	*None*	
kode	varchar(20)	*latin1_swedish_ci*		No	*None*	
no_urut	int(11)			No	*None*	auto_increment
perihal	varchar(100)	*latin1_swedish_ci*		No	*None*	
isi_ringkas	varchar(500)	*latin1_swedish_ci*		No	*None*	
dari	varchar(100)	*latin1_swedish_ci*		No	*None*	
tgl_surat	Date			No	*None*	
no_surat	varchar(50)	*latin1_swedish_ci*		No	*None*	
lampiran	varchar(20)	*latin1_swedish_ci*		No	*None*	
pengolah	varchar(50)	*latin1_swedish_ci*		No	*None*	
tgl_diteruskan	date			No	*None*	
tanda_terima	varchar(20)	*latin1_swedish_ci*		No	*None*	

catatan	varchar(200)	*latin1_swedish_ci*		No	*None*	
disposisi	enum('Sudah','Belum')	*latin1_swedish_ci*		No	*None*	

Tabel 3.9 Struktur Tabel Surat Keluar

Field	Type	Collation	Attributes	Null	Default	Axtra
indeks	varchar(100)	*latin1_swedish_ci*		No	*None*	
kode	varchar(50)	*latin1_swedish_ci*		No	*None*	
no_urut	int(10)			No	*None*	
perihal	varchar(200)	*latin1_swedish_ci*		No	*None*	
isi_ringkas	varchar(500)	*latin1_swedish_ci*		No	*None*	
kepada	varchar(1000)	*latin1_swedish_ci*		No	*None*	
pengolah	varchar(100)	*latin1_swedish_ci*				
tgl_surat	Date			No	*None*	
lampiran	varchar(1000)	*latin1_swedish_ci*		No	*None*	
catatan	varchar(200)	*latin1_swedish_ci*		No	*None*	
tembusan	varchar(300)	*latin1_swedish_ci*		No	*None*	

Tabel 3.10 Struktur Tabel Disposisi

Field	Type	Collation	Attributes	Null	Default	Axtra
indeks	varchar(100)	*latin1_swedish_ci*		No	*None*	
tgl_penyelesaian	date			No	*None*	
dari	varchar(100)	*latin1_swedish_ci*		No	*None*	
perihal	varchar(200)	*latin1_swedish_ci*		No	*None*	
tgl_surat	date			No	*None*	
no_surat	varchar(50)	*latin1_swedish_ci*		No	*None*	
instruksi	varchar(200)	*latin1_swedish_ci*		No	*None*	

diteruskan_ kepada	varchar(200)	*latin1_swedish_ci*		No	*None*	

Tabel 3.10 Struktur Tabel SKPD

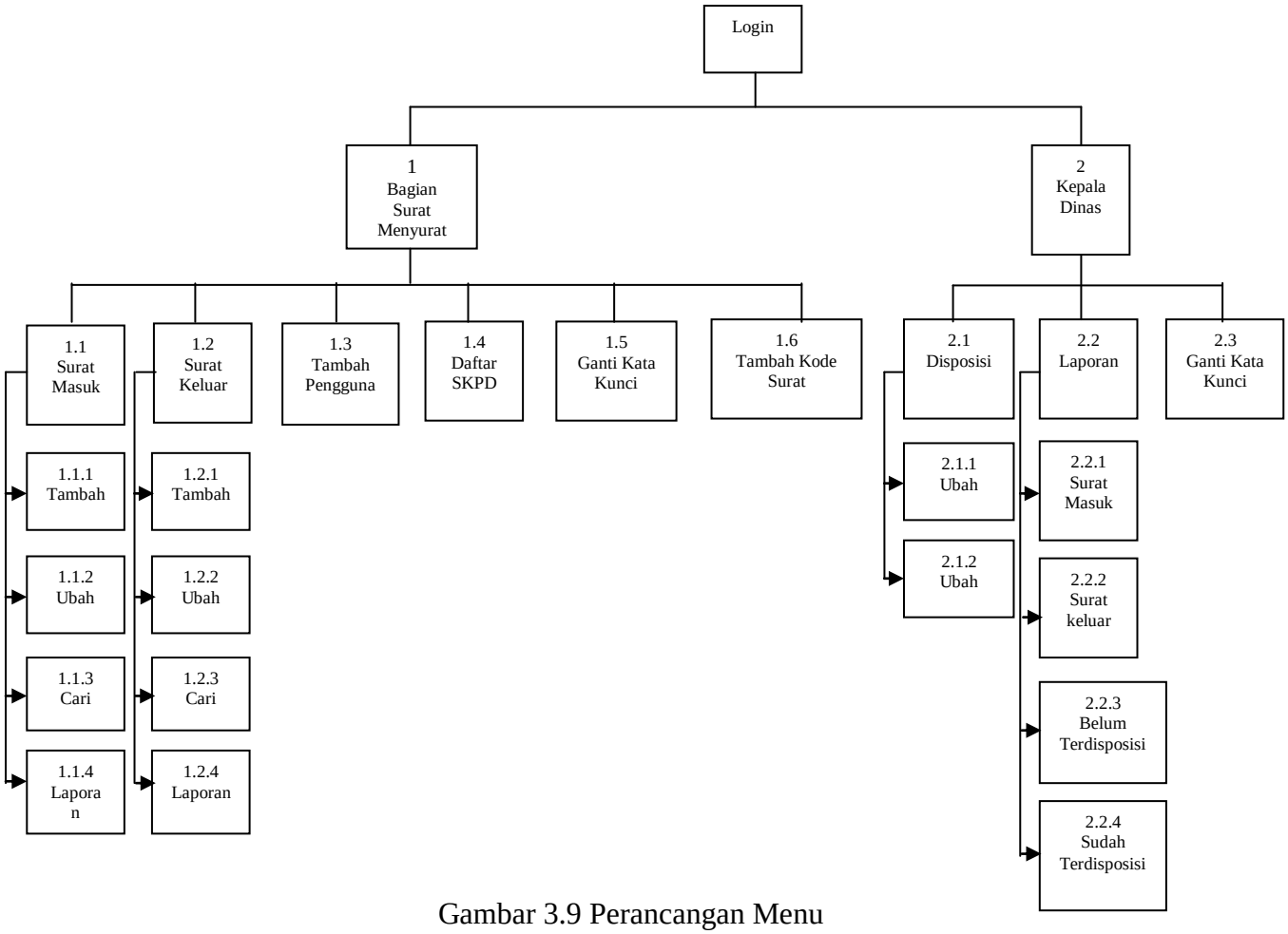
Field	Type	Collation	Attributes	Null	Default	Axtra
id_skpd	int(11)	*latin1_swedish_ci*		No	*None*	
nama	varchar(100)			No	*None*	
alamat	varchar(400)	*latin1_swedish_ci*		No	*None*	
ket	varchar(400)			No	*None*	

Tabel 3.11 Struktur Tabel Kode Surat

Field	Type	Collation	Attributes	Null	Default	Axtra
kode	varchar(20)	*latin1_swedish_ci*		No	*None*	
nama	varchar(200)	*latin1_swedish_ci*		No	*None*	

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 :

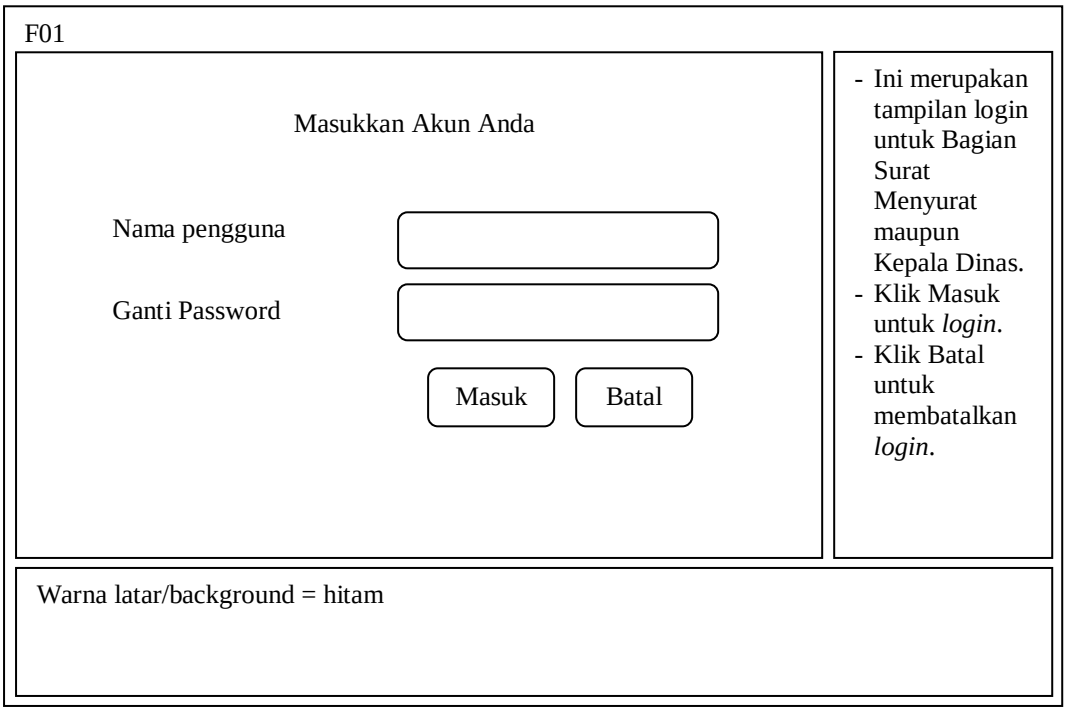


Gambar 3.9 Perancangan Menu

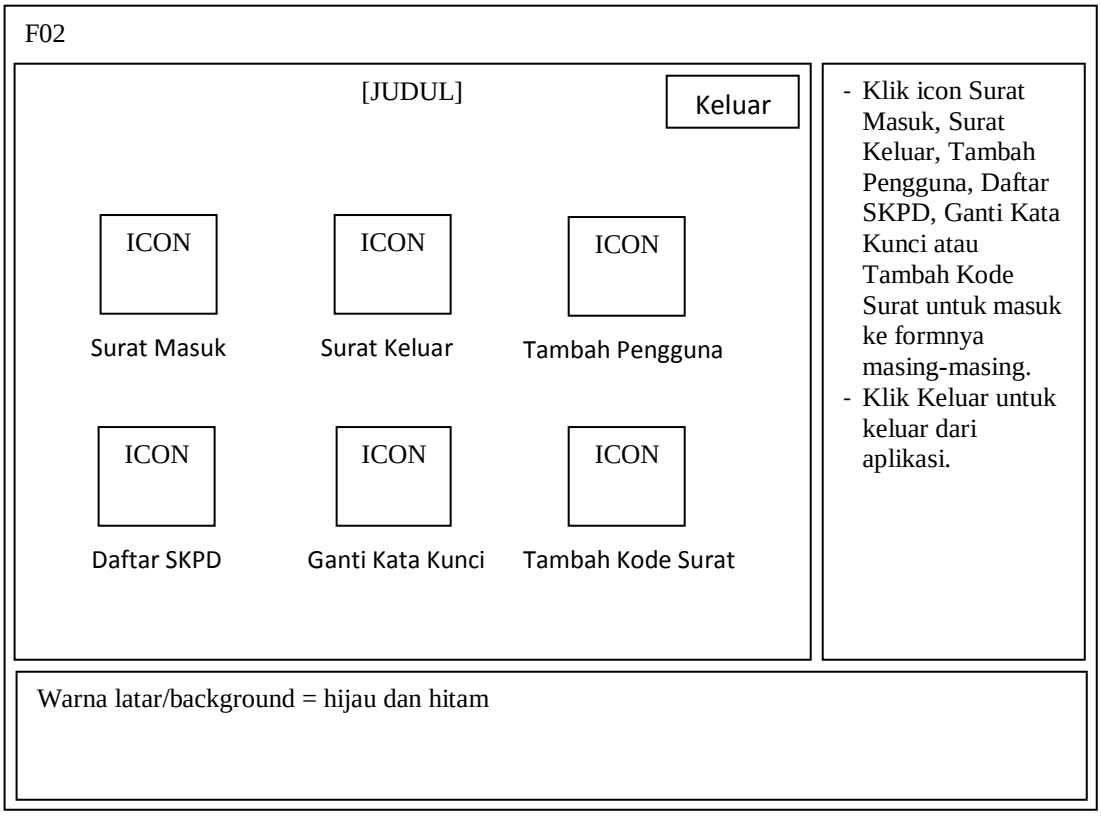
3.6. Perancangan Antarmuka

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.

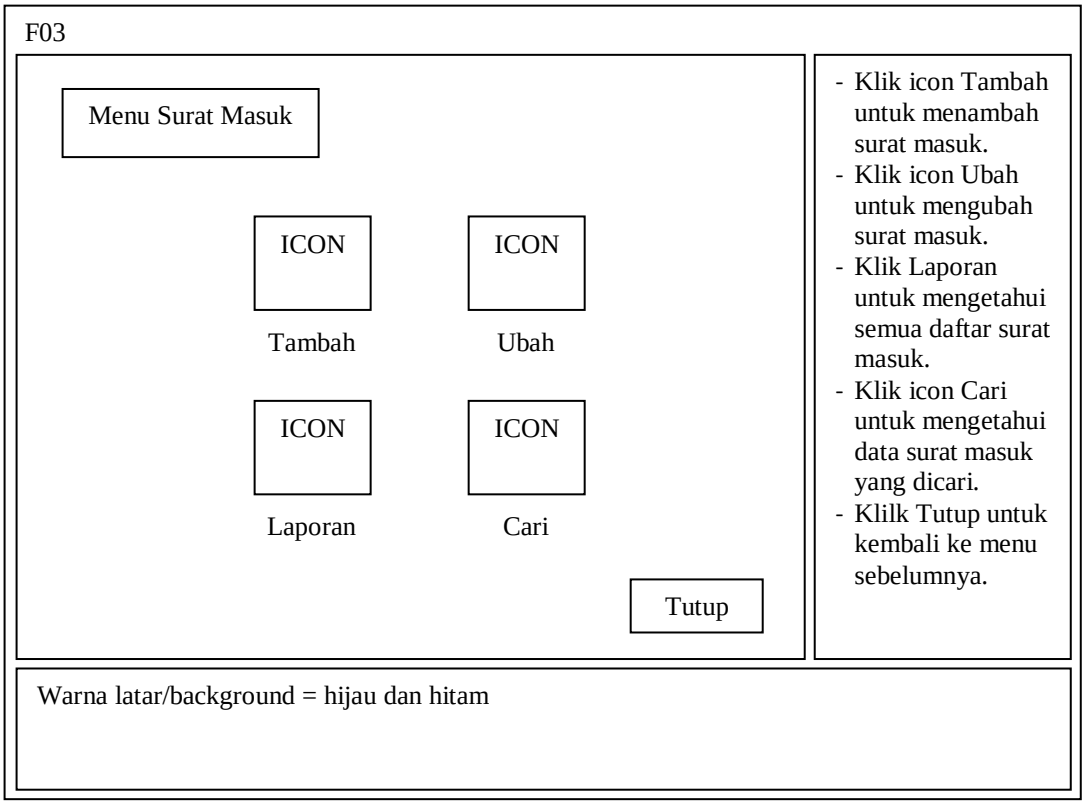
Perancangan antarmuka pada dasarnya adalah proses penggambaran dari bagaimana sebuah bagian sistem dibentuk. Maka dari itu pada saat akan melakukan tahap perancangan antarmuka diperlukan untuk mendefinisikan kebutuhan-kebutuhan fungsional dari sistem yang akan dibangun serta mengetahui sejauh mana performansi profile pengguna pada saat akan menggunakan sistem tersebut.



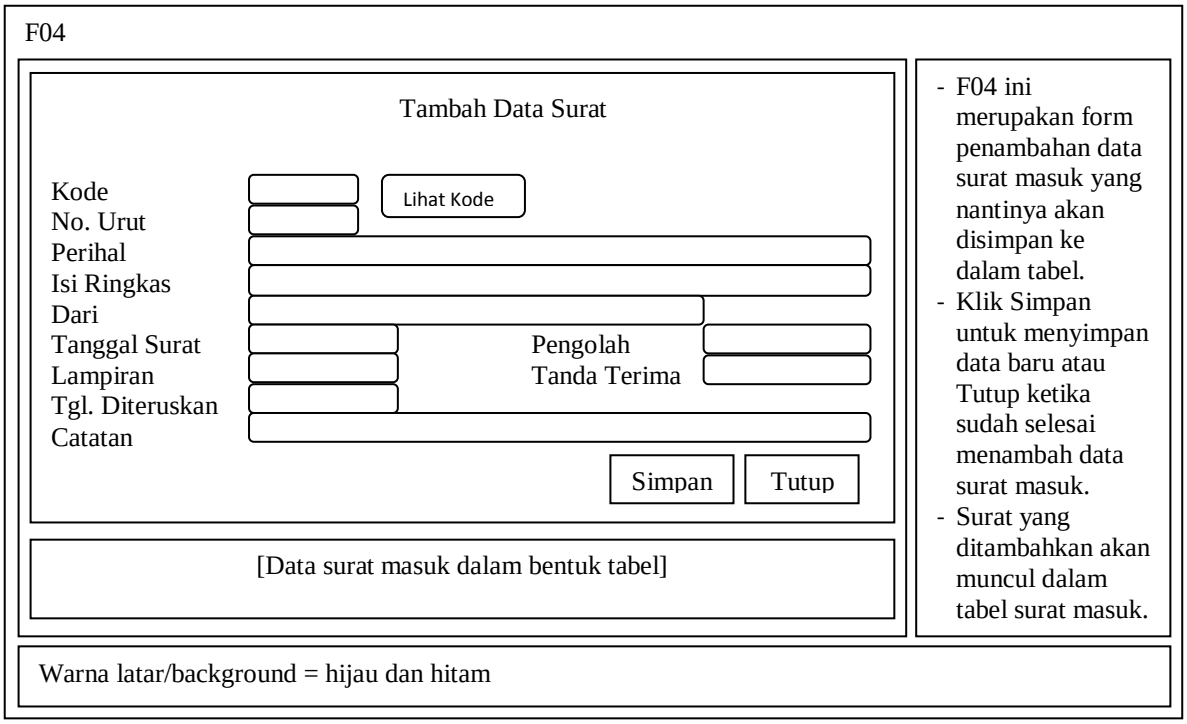
Gambar 3.10 Perancangan Form Login



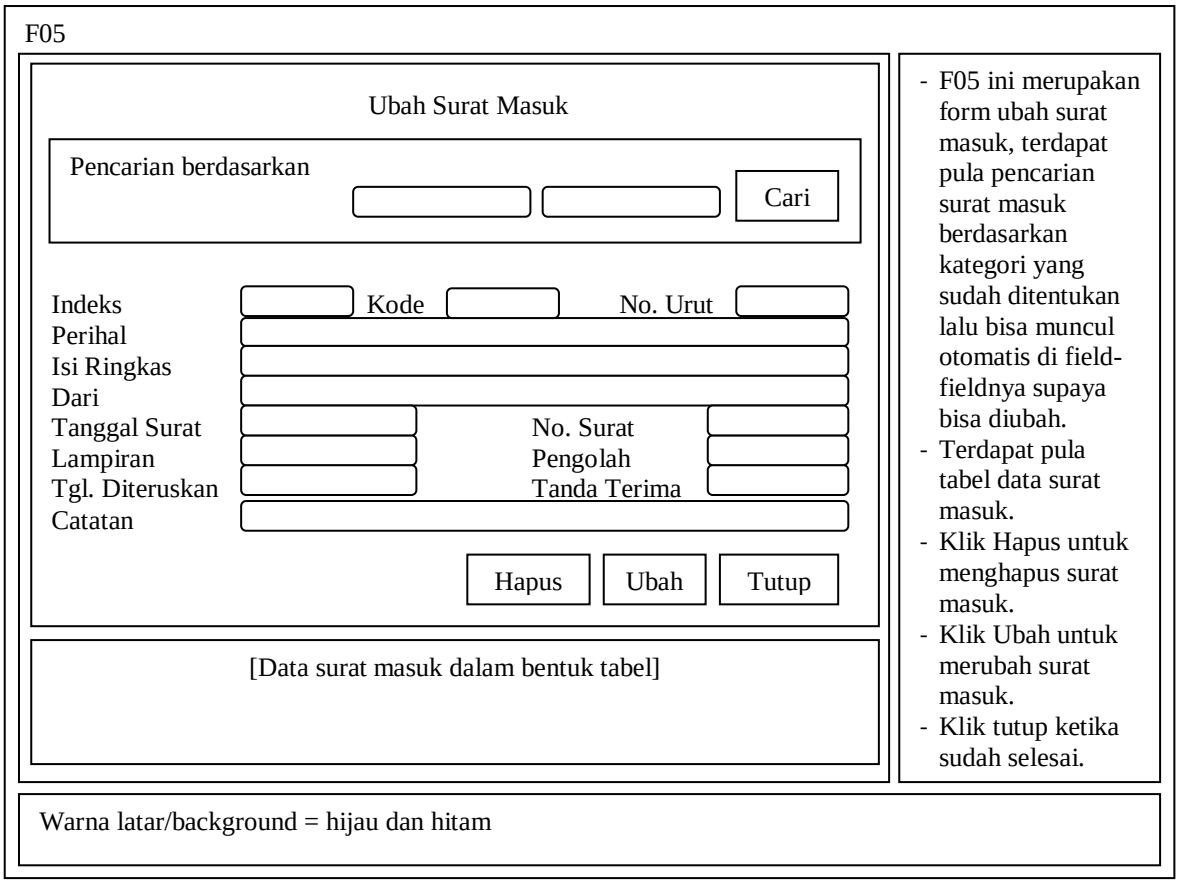
Gambar 3.11 Perancangan Form Menu untuk Bagian Surat Menyurat



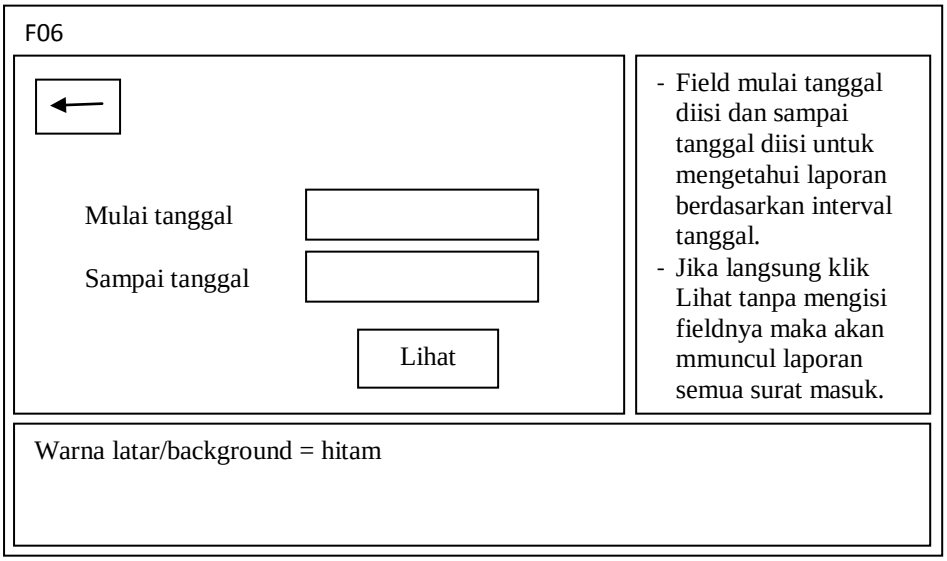
Gambar 3.12 Perancangan Form Menu Surat Masuk



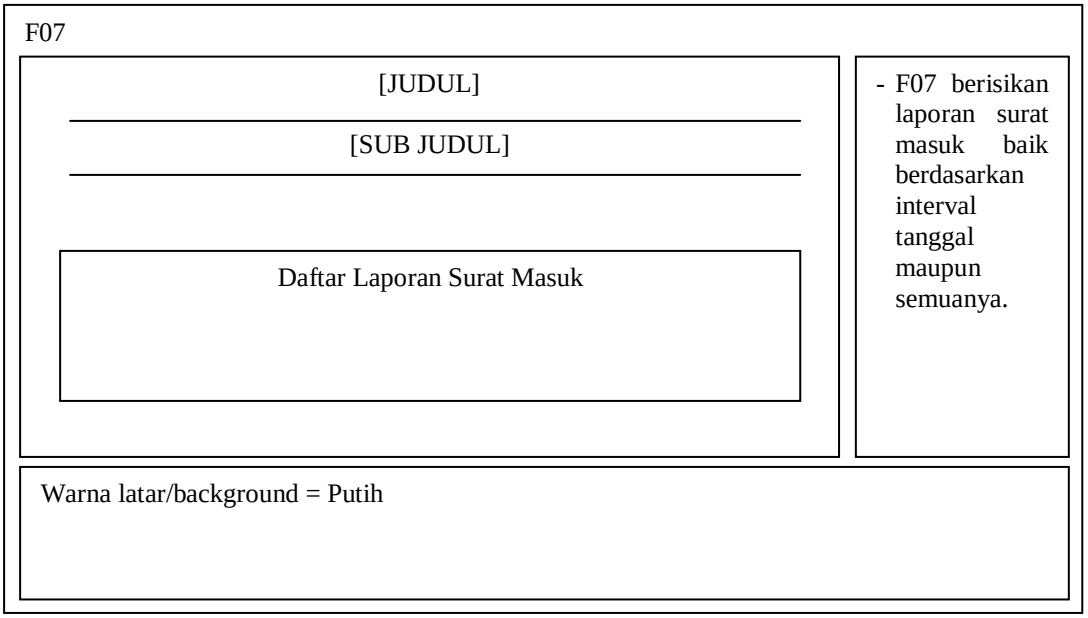
Gambar 3.13 Perancangan Form Tambah Surat Masuk



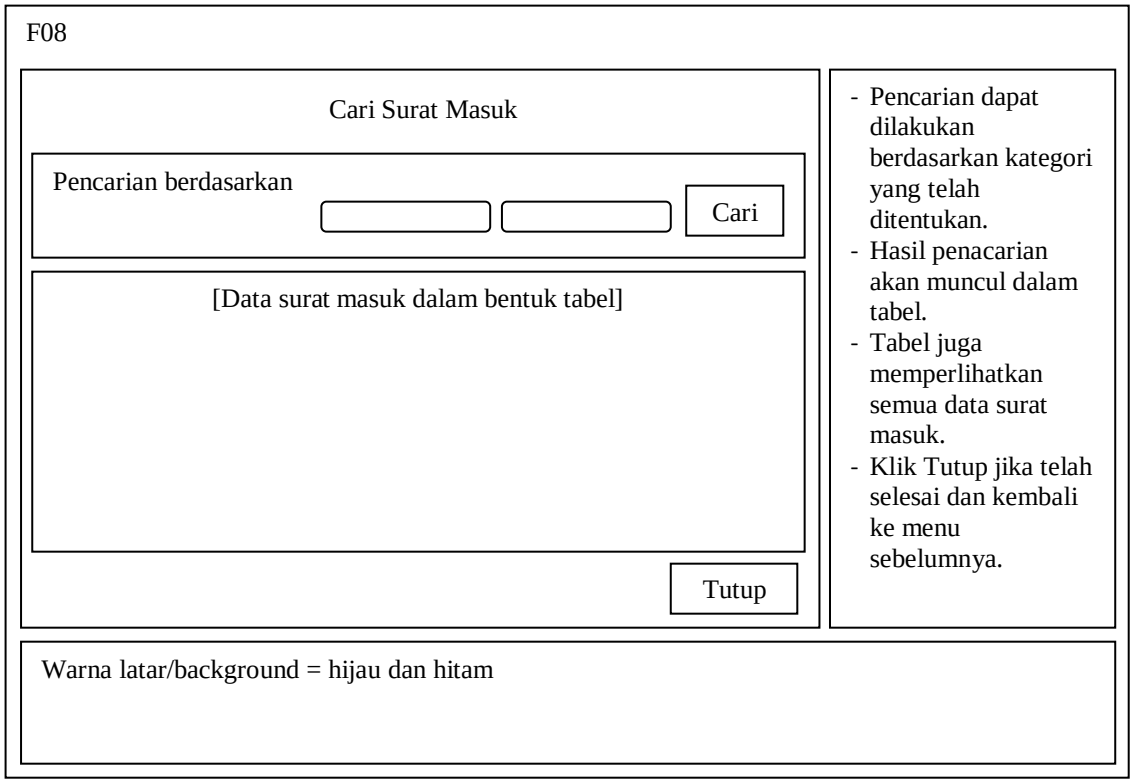
Gambar 3.14 Perancangan Form Ubah Surat Masuk



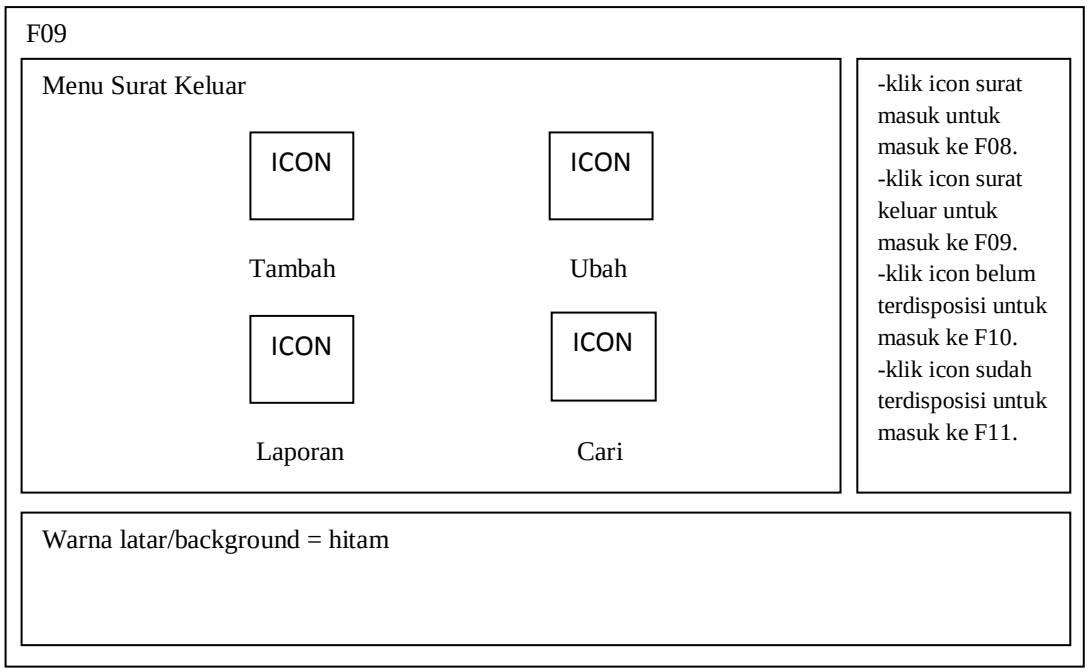
Gambar 3.15 Perancangan Form Laporan Berdasarkan Tanggal Surat Masuk



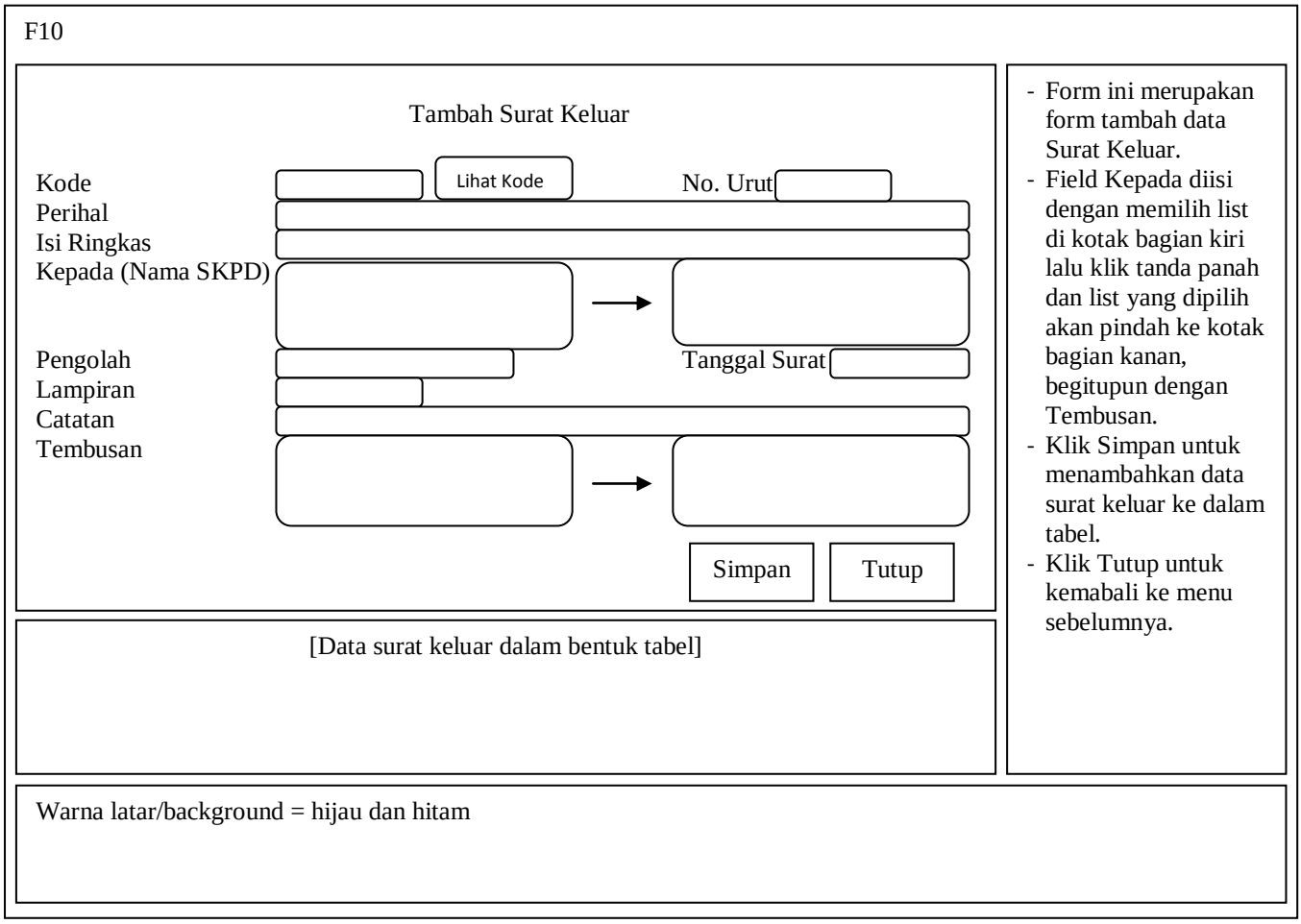
Gambar 3.16 Perancangan Form Laporan Surat Masuk



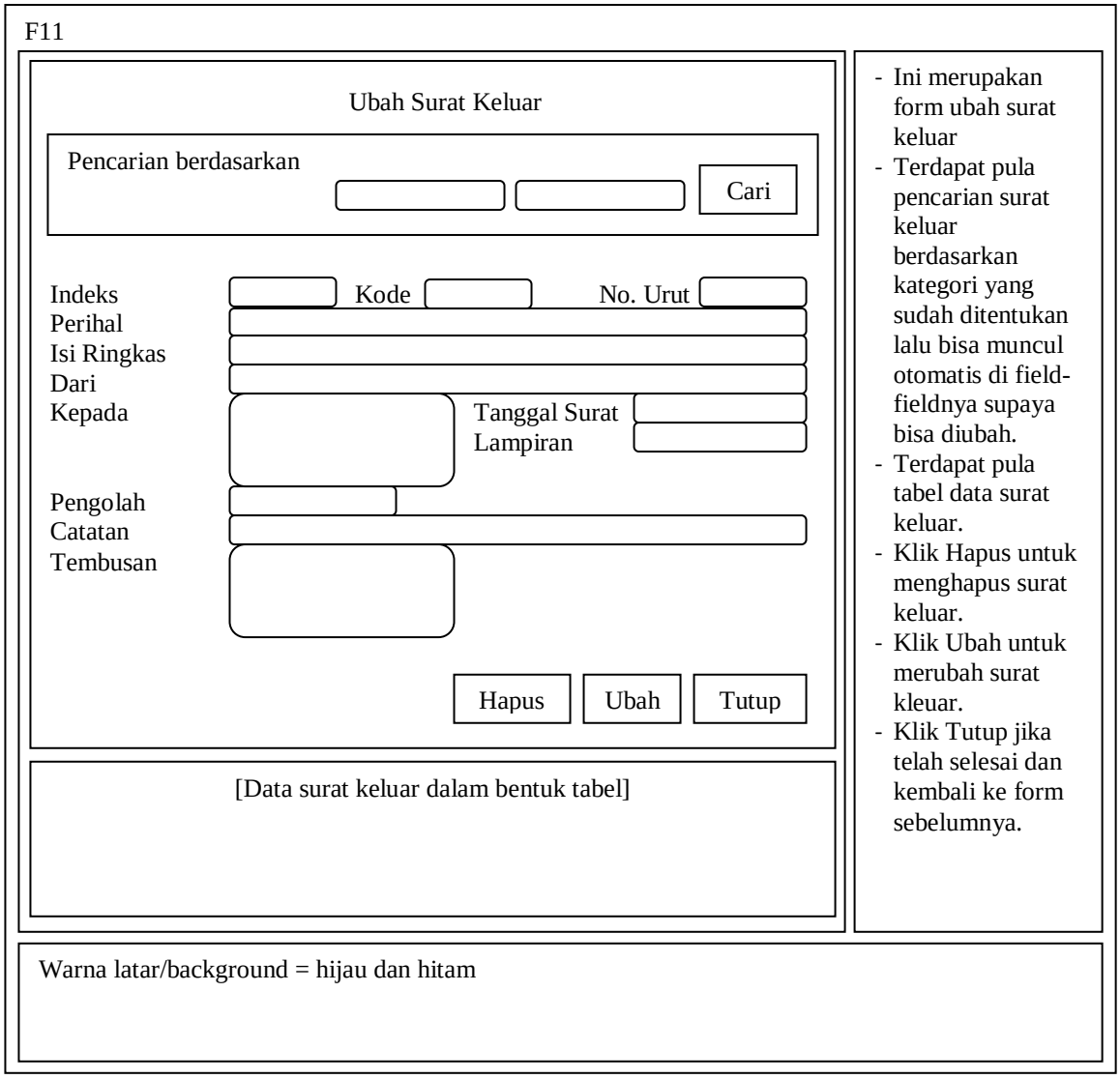
Gambar 3.17 Perancangan Form Cari Surat Masuk



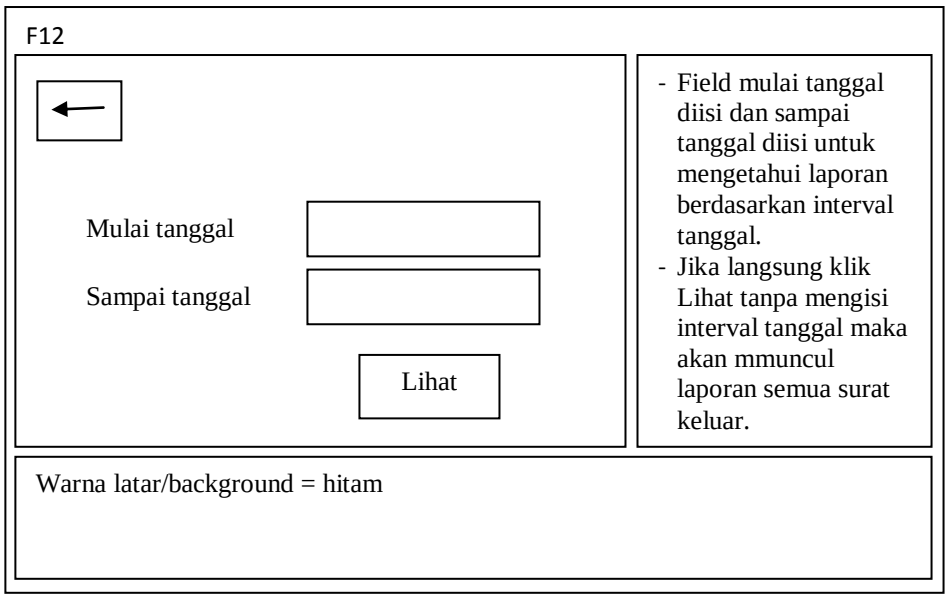
Gambar 3.18 Perancangan Form Menu Surat Keluar



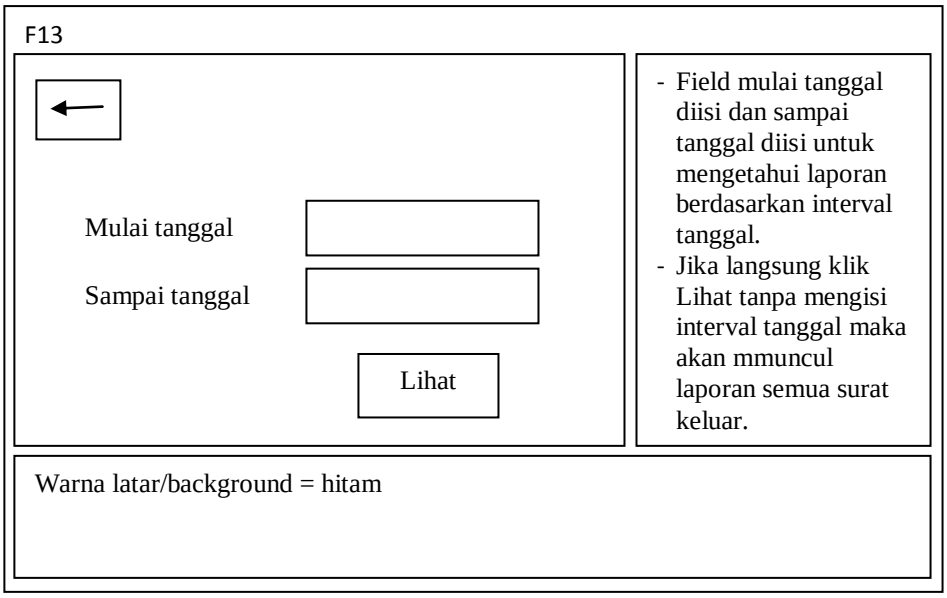
Gambar 3.19 Perancangan Form Tambah Surat Keluar



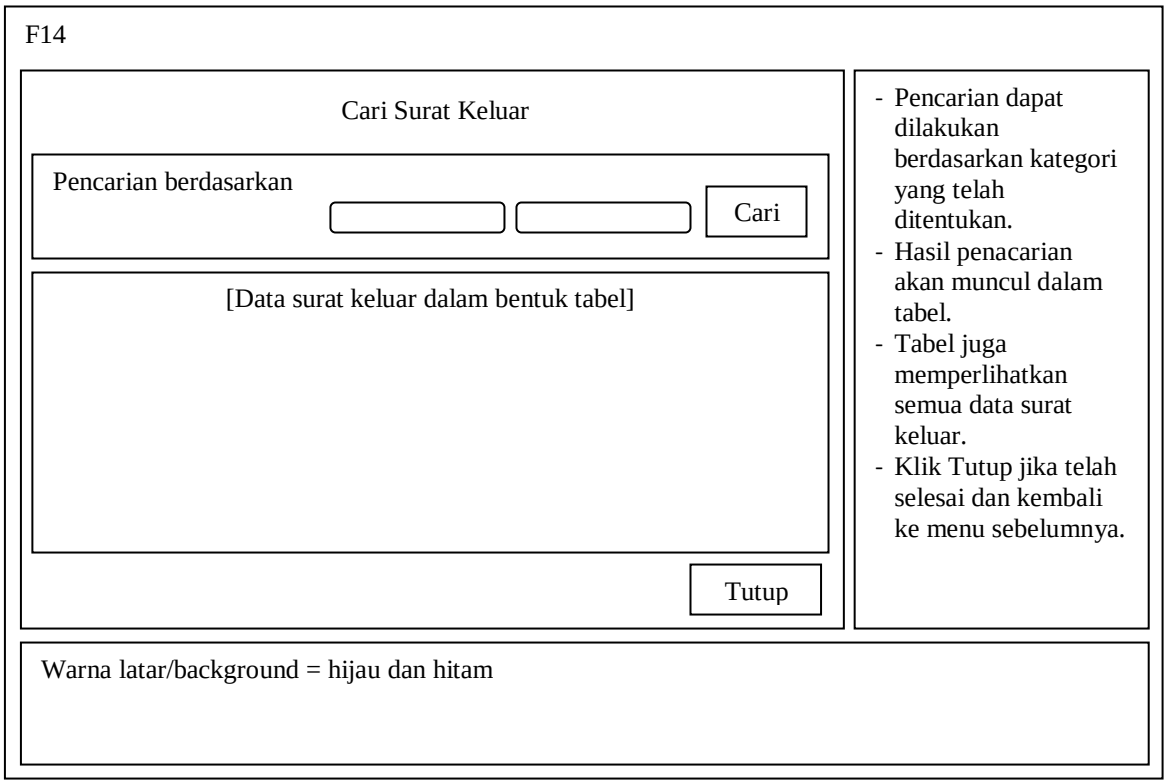
Gambar 3.20 Perancangan Form Ubah Surat Keluar



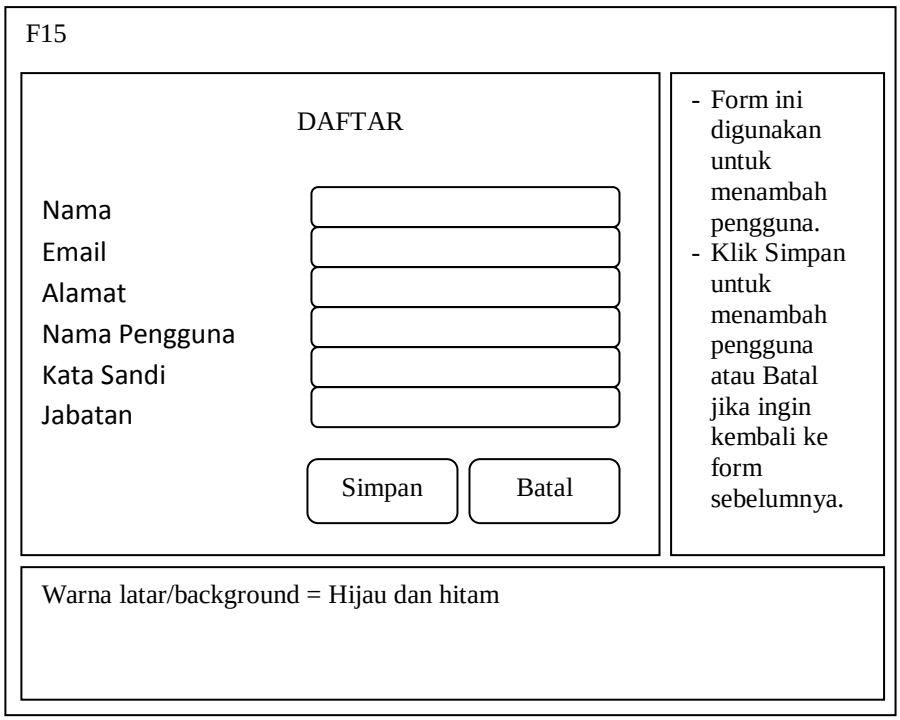
Gambar 3.21 Perancangan form Laporan Berdasarkan Tanggal Surat Keluar



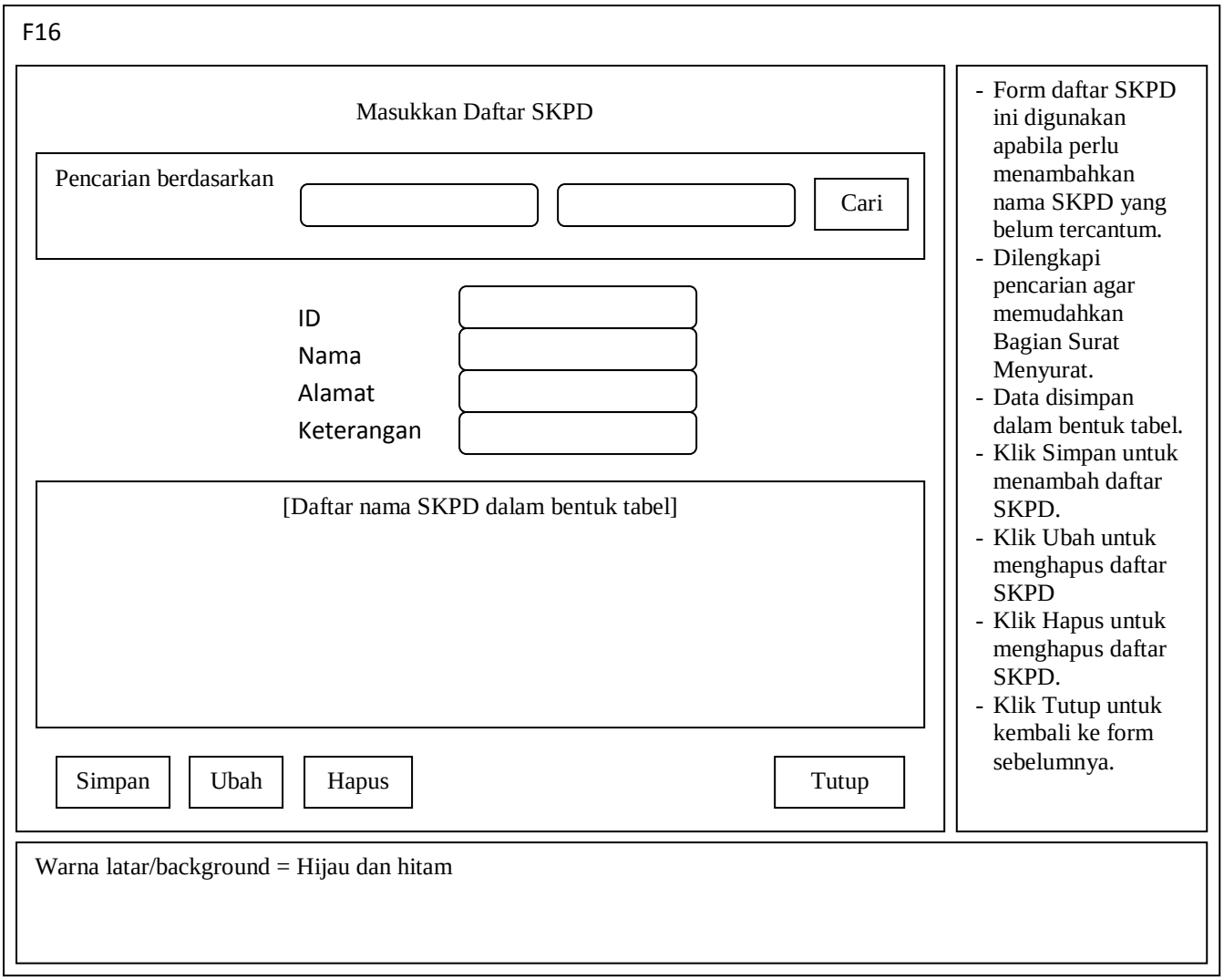
Gambar 3.22 Perancangan Laporan Surat Keluar



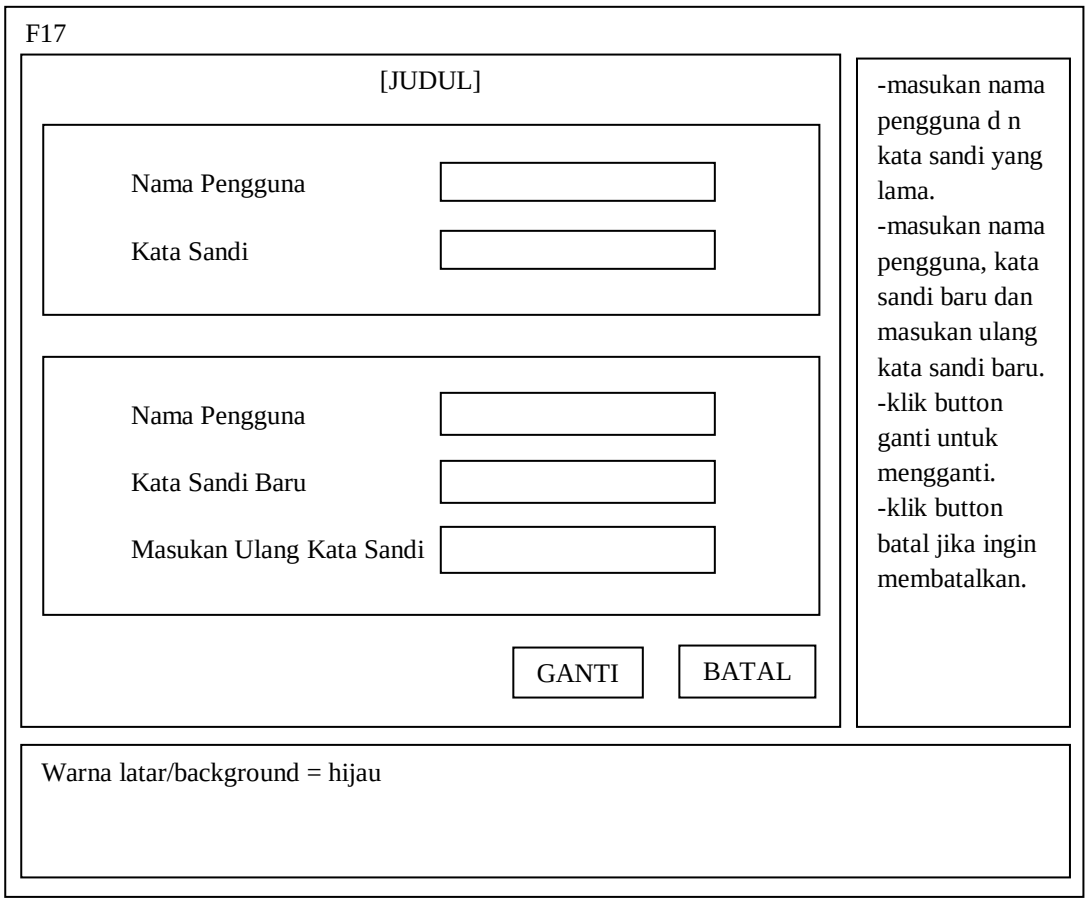
Gambar 3.23 Perancangan Form Cari Surat Keluar



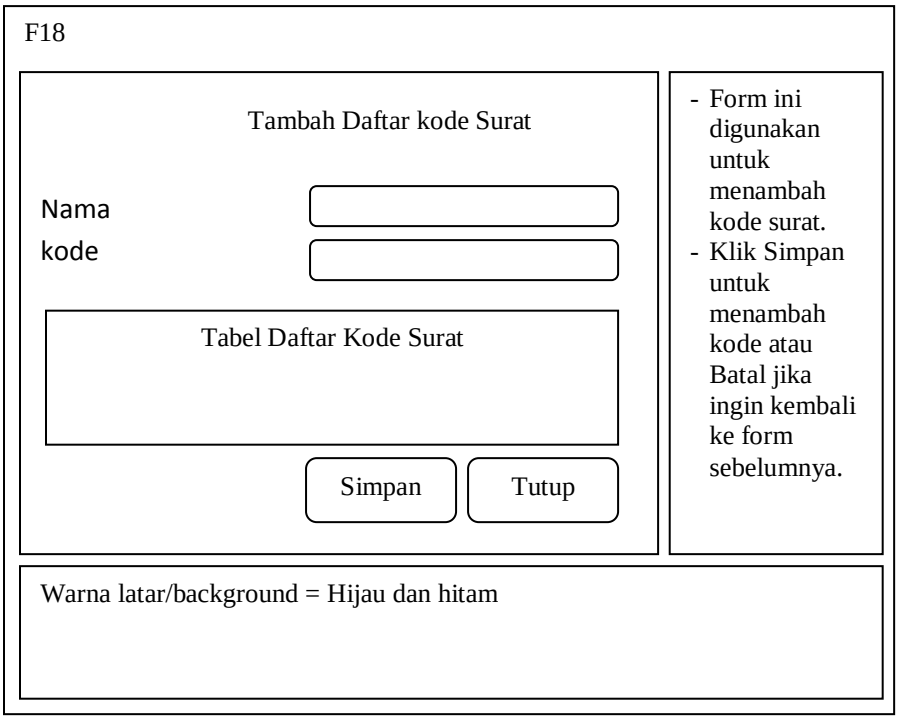
Gambar 3.24 Perancangan Form Tambah Pengguna



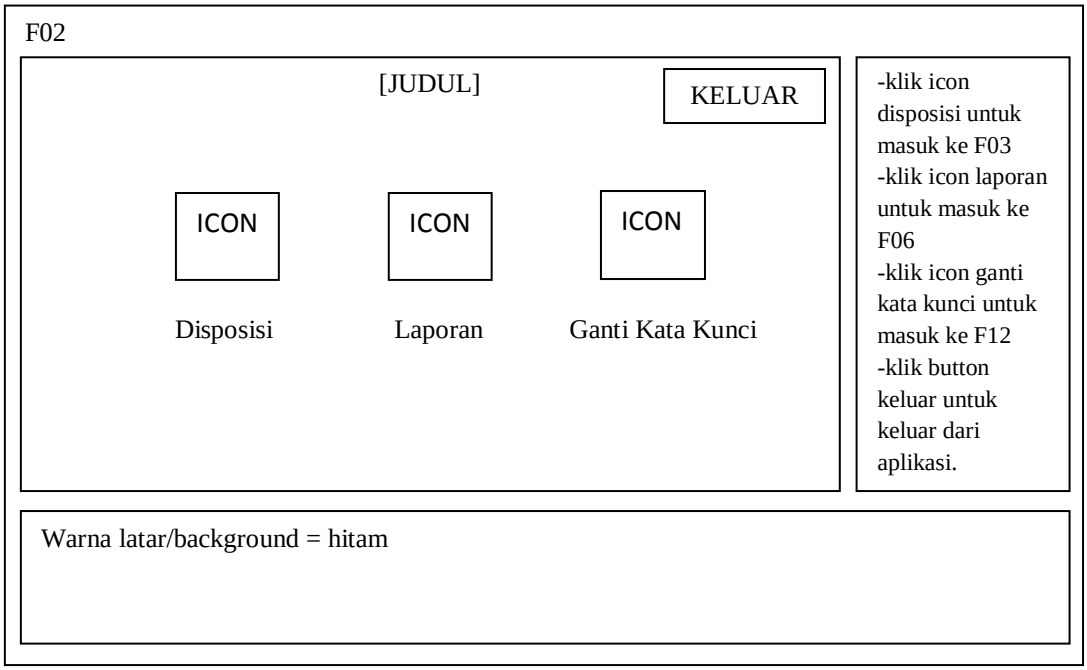
Gambar 3.25 Perancangan Form Daftar SKPD



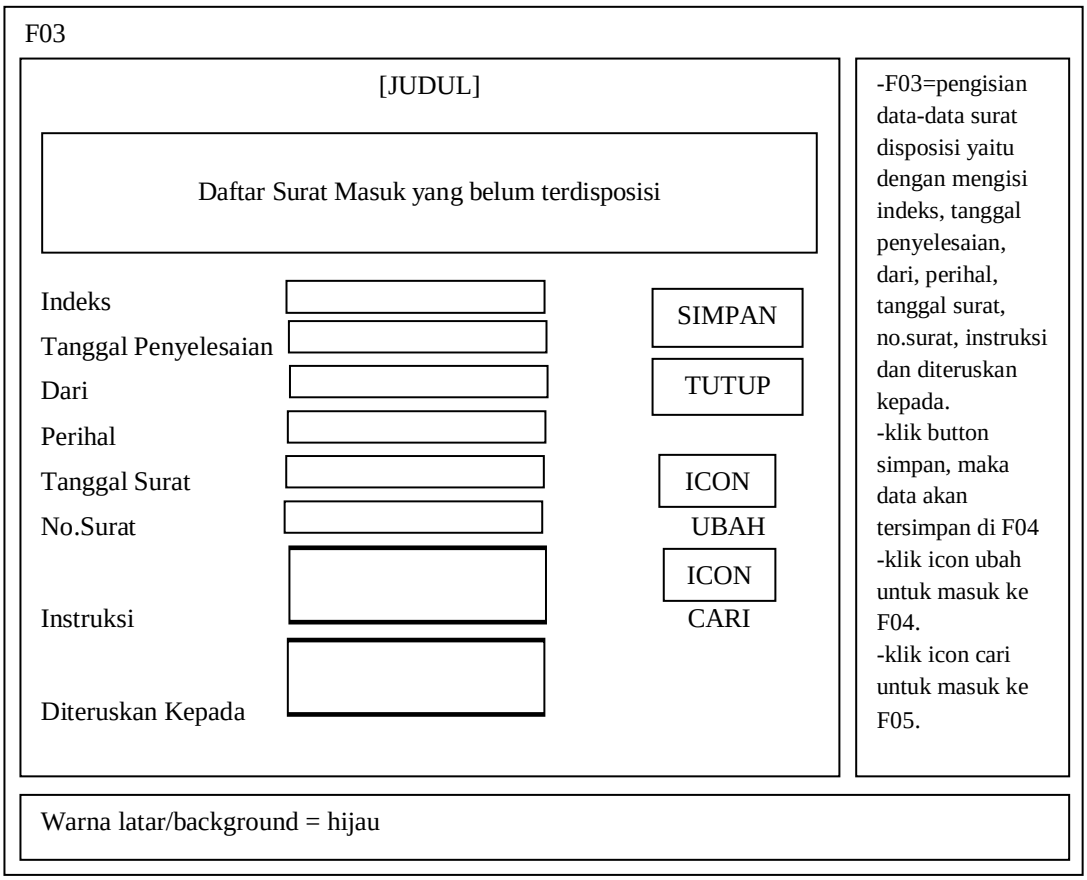
Gambar 3.26 Perancangan Form Ganti Kata Kunci



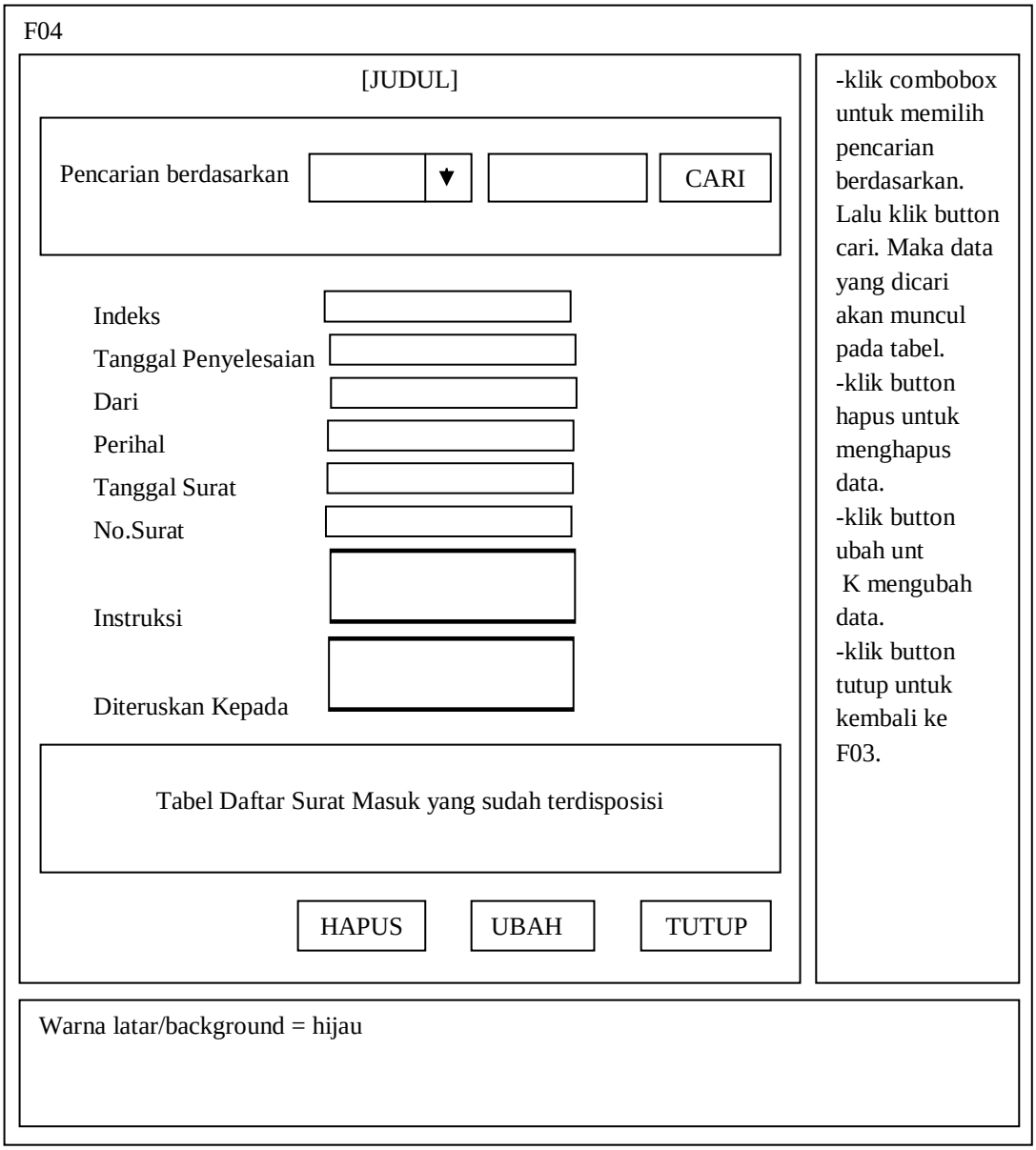
Gambar 3.27 Perancangan Form Tambah Kode Surat



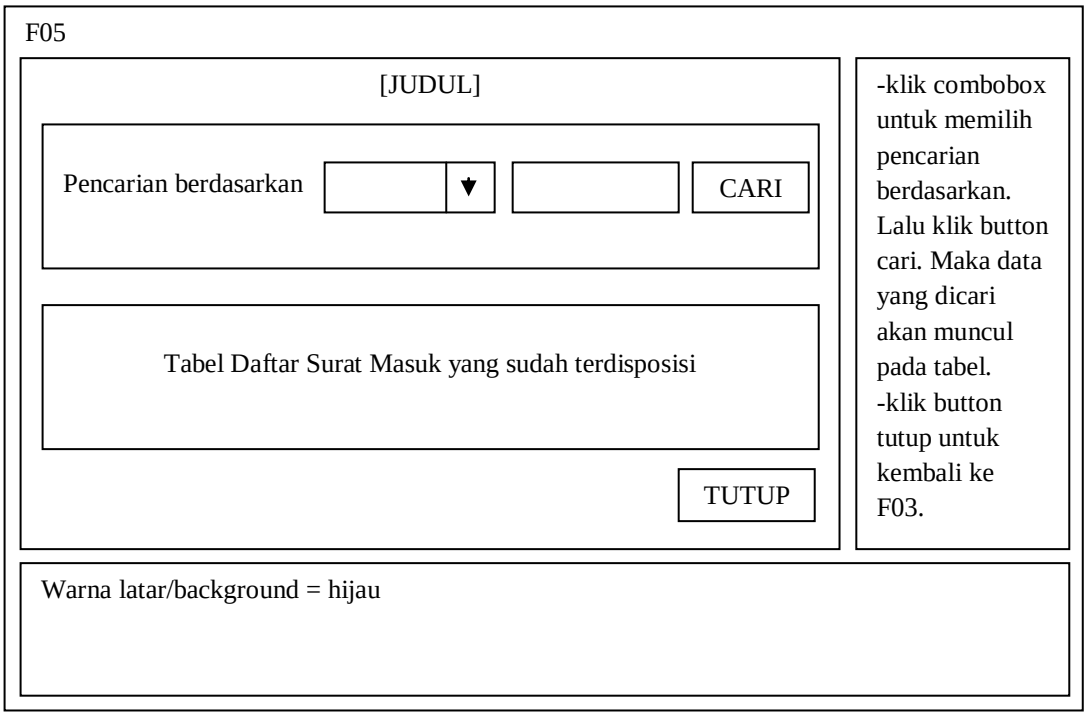
Gambar 3.28 Perancangan Form Menu untuk Kepala Dinas



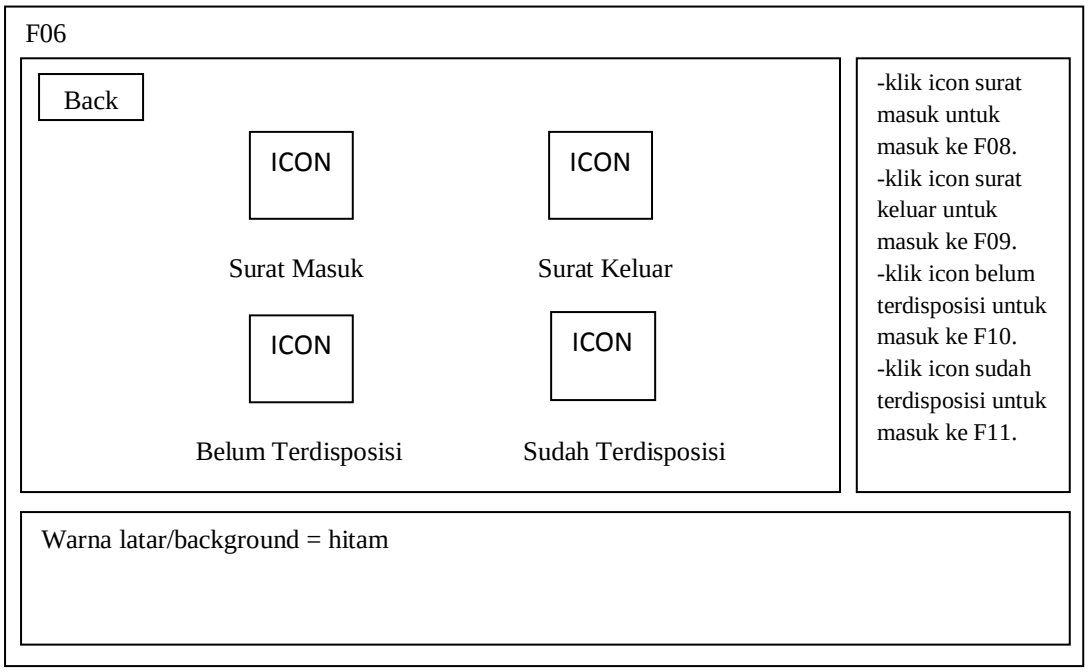
Gambar 3.29 Perancangan Form Tambah Disposisi



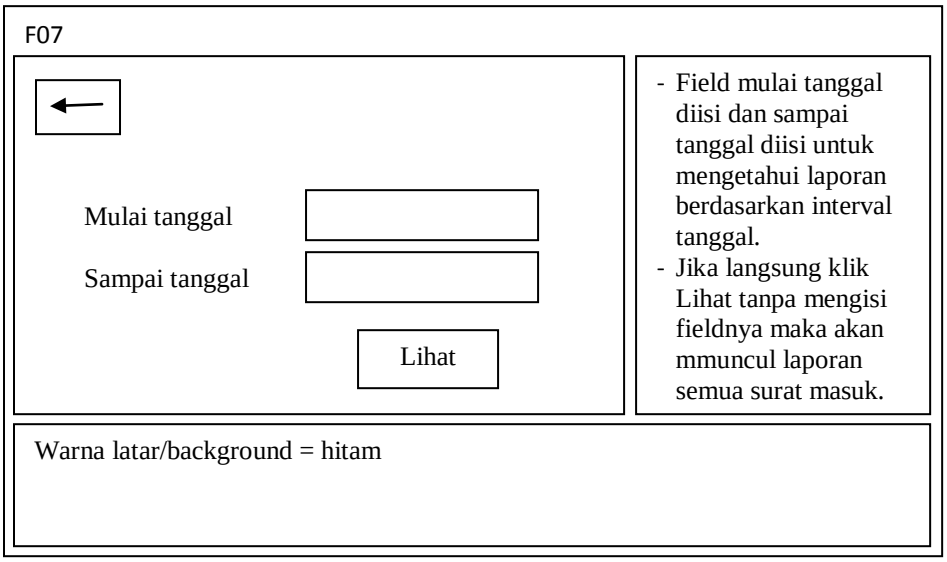
Gambar 3.30 Perancangan Form Ubah Disposisi



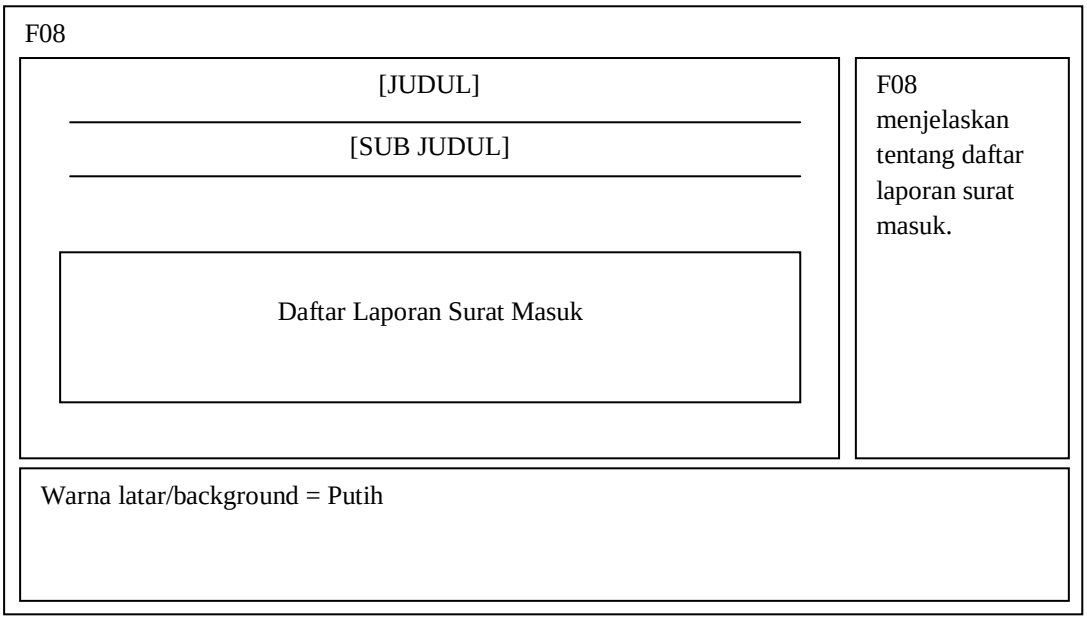
Gambar 3.31 Perancangan Form Cari Disposisi



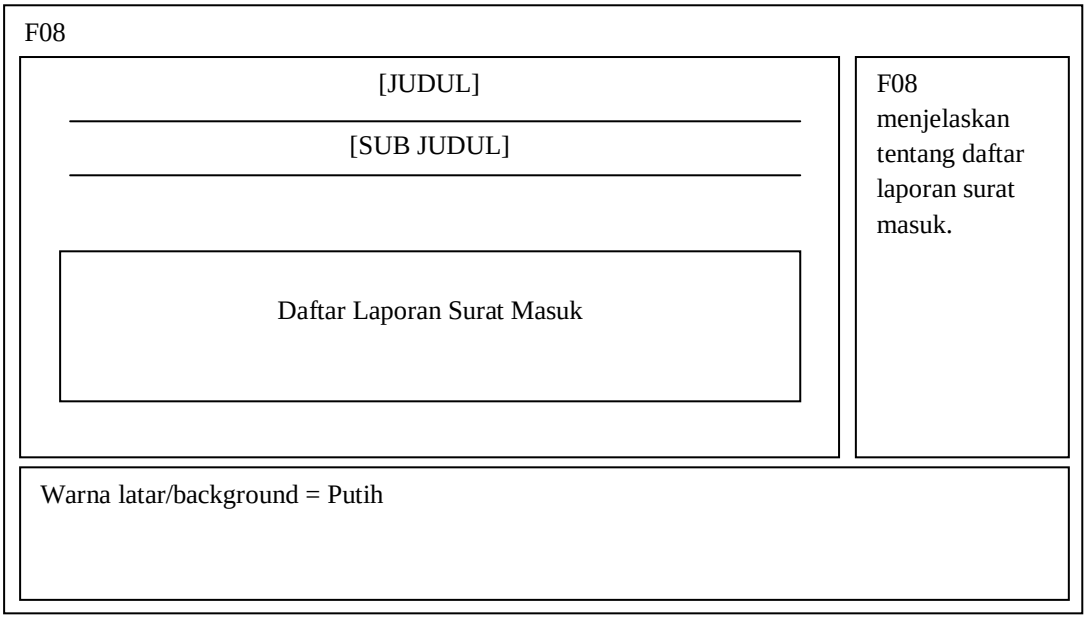
Gambar 3.32 Perancangan Form Menu Laporan Kepala Dinas



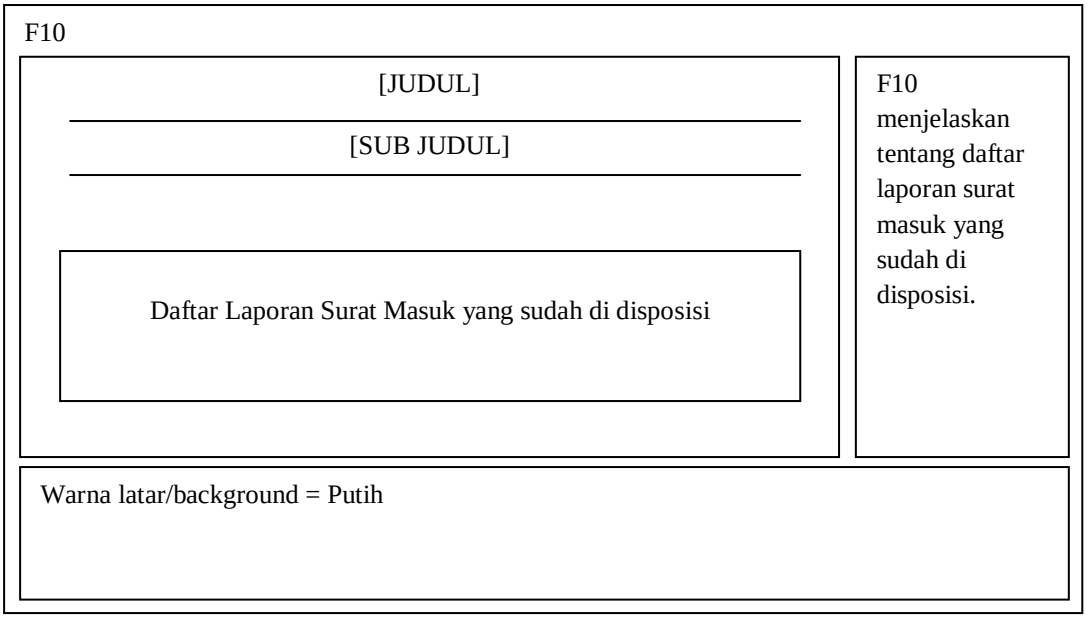
Gambar 3.33 Perancangan Form Tanggal Surat



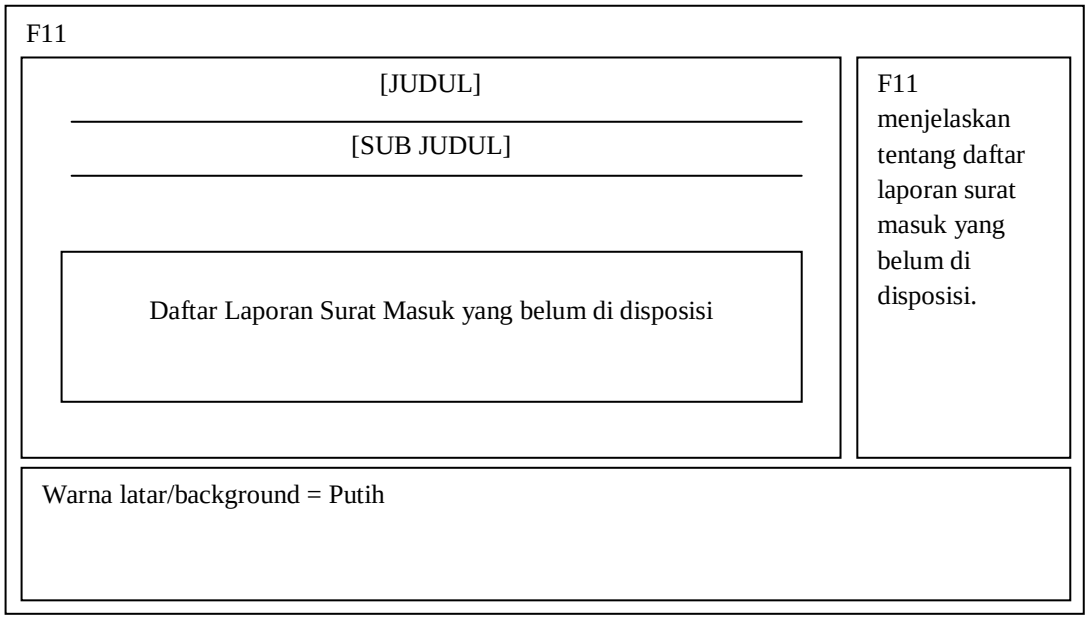
Gambar 3.34 Perancangan Form Laporan Surat Masuk



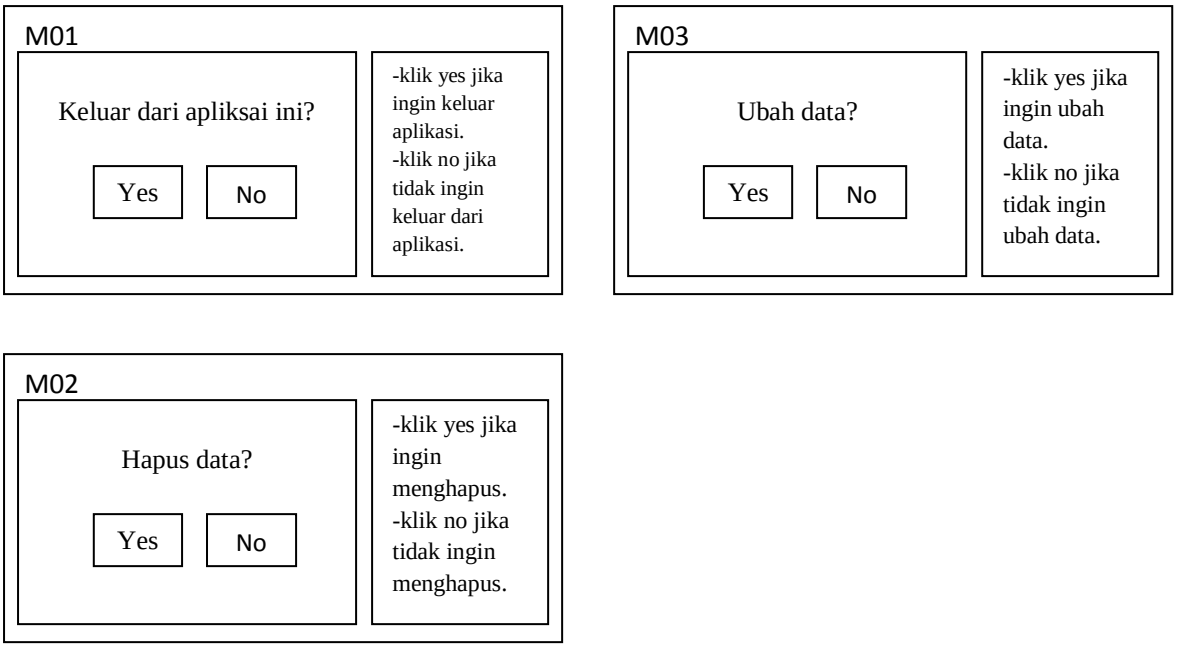
Gambar 3.35 Perancangan Form Laporan Surat Keluar



Gambar 3.36 Perancangan Form Laporan Surat Masuk yang Sudah di Disposisi



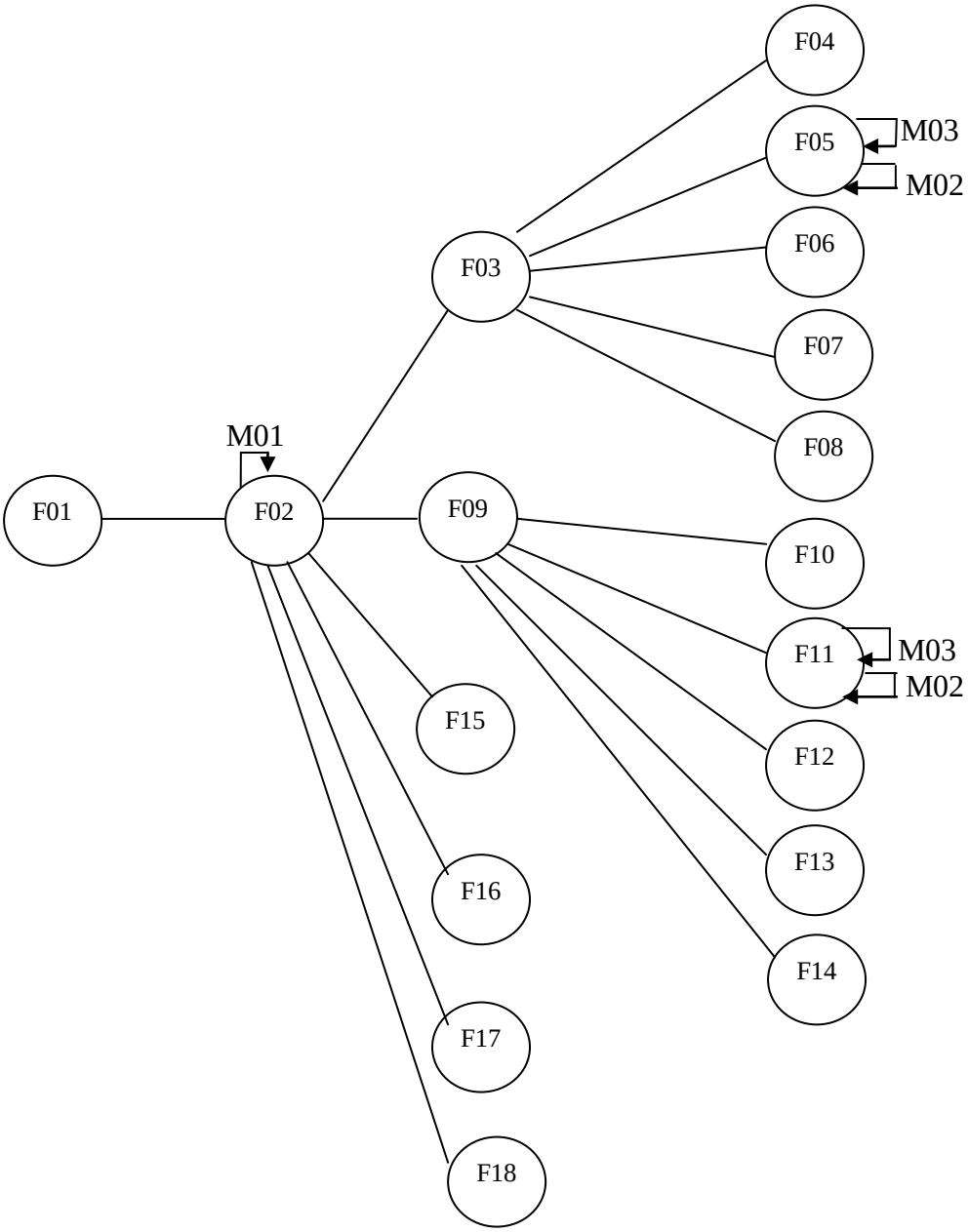
Gambar 3.37 Perancangan Form Laporan Surat Masuk yang Belum di Disposisi



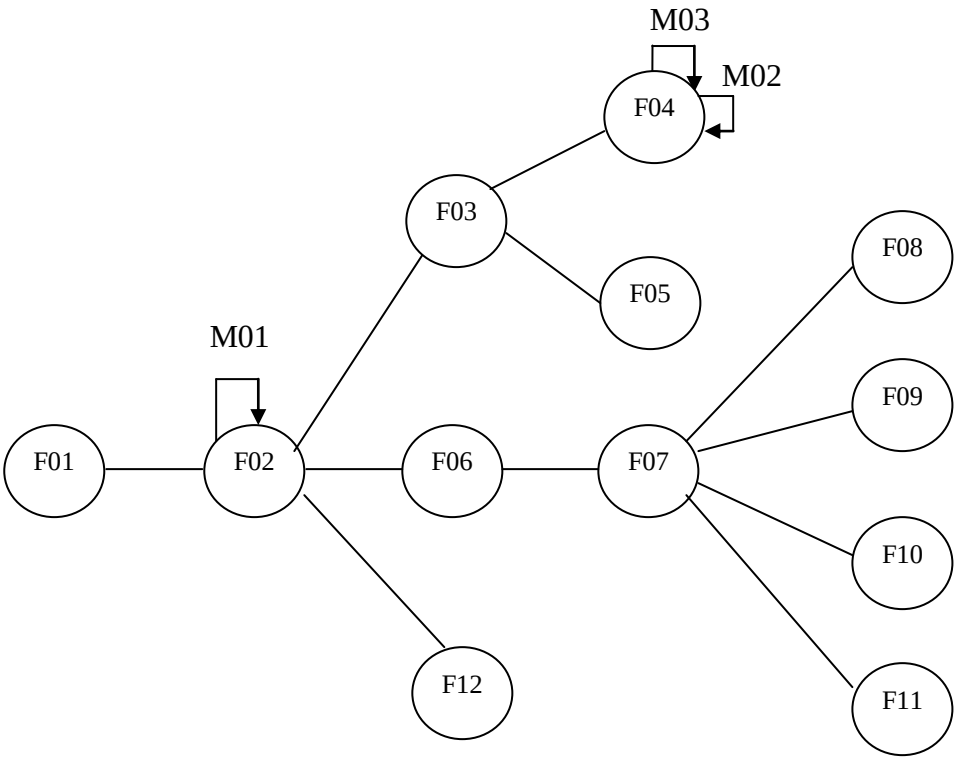
Gambar 3.38 Perancangan Form Pesan

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). 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.



Gambar 3.39 Jaringan Semantik untuk Bagian Surat Menyurat



Gambar 3.40 Jaringan Semantik untuk Kepala Dinas