BAB 3
ANALISIS DAN PERANCANGAN SISTEM

Analisis dan perancangan sistem merupakan tahapan yang sistematis untuk mendapatkan rancangan sistem yang baik dan sesuai dengan kegunaan dan tujuannya. Tahap awal dari analisis adalah menganalisa kebutuhan-kebutuhan sistem mulai dari kebutuhan pengguna, kebutuhan non fungsional, dan kebutuhan fungsional. Tahap perancangan aplikasi yaitu perancangan database dan perancangan antarmuka.

3.1 Analisis Sistem

Langkah pertama yang dilakukan dalam membangun sistem ini yaitu menganalisis sistem. Sistem yang akan dianalisis adalah sistem yang sedang berjalan (dalam hal ini sistem yang sudah ada) dan yang akan dibangun. Analisis harus dilakukan dengan seksama dan teliti untuk benar-benar mengetahui detail yang ada pada suatu sistem. Tahap berikutnya tidak akan mengalami kesulitan apabila dalam melakukan analisis dilakukan secara benar.

3.1.1 Analisis Masalah

Berdasarkan hasil pengamatan pada Divisi Administrasi dan Keuangan PT. Meta Informatika Telekomindo, didapatkan masalah yaitu proses pendistribusian surat internal perusahaan masih dilakukan secara manual sehingga berkurangnya optimalisasi kinerja karyawan.

Permasalahan yang sedang terjadi di Divisi Administrasi dan Keuangan PT. Meta Informatika Telekomindo akan diuraikan secara detil sebagai berikut :

a. Sistem yang ada di PT. Meta Informatika Telekomindo masih melakukan pembuatan surat keluar secara manual sehingga lambatnya proses pendistribusian surat.
b. Sistem yang ada belum mampu menindaklanjuti surat yang masuk ke perusahaan dengan cepat.

c. Belum adanya sarana informasi tentang agenda rapat ketika karyawan berada diluar lingkungan perusahaan sehingga karyawan dapat ketinggalan informasi yang menyebabkan lambatnya kinerja para karyawan.
d. Belum adanya sistem untuk mempermudah karyawan dalam mencari data surat dengan cepat agar karyawan tidak harus mencari secara manual di tempat penyimpanan arsip-arsip perusahaan.
e. Terhambatnya kinerja para karyawan karena surat yang bersifat penting belum ditindaklanjuti dikarenakan pihak yang berkepentingan berada diluar lingkungan perusahaan.

3.1.2 Prosedur Sistem Yang Sedang Berjalan

Ketika akan membangun sebuah program aplikasi, tahap pertama yang harus dilakukan yaitu mempelajari dan menganalisa sistem yang sedang berjalan. Berdasarkan hasil wawancara dengan bagian administrasi, yaitu Direktur Divisi Administrasi dan Keuangan, prosedur dari sistem informasi yang sedang berjalan saat ini di PT. Meta Informatika Telekomindo yaitu sebagai berikut :

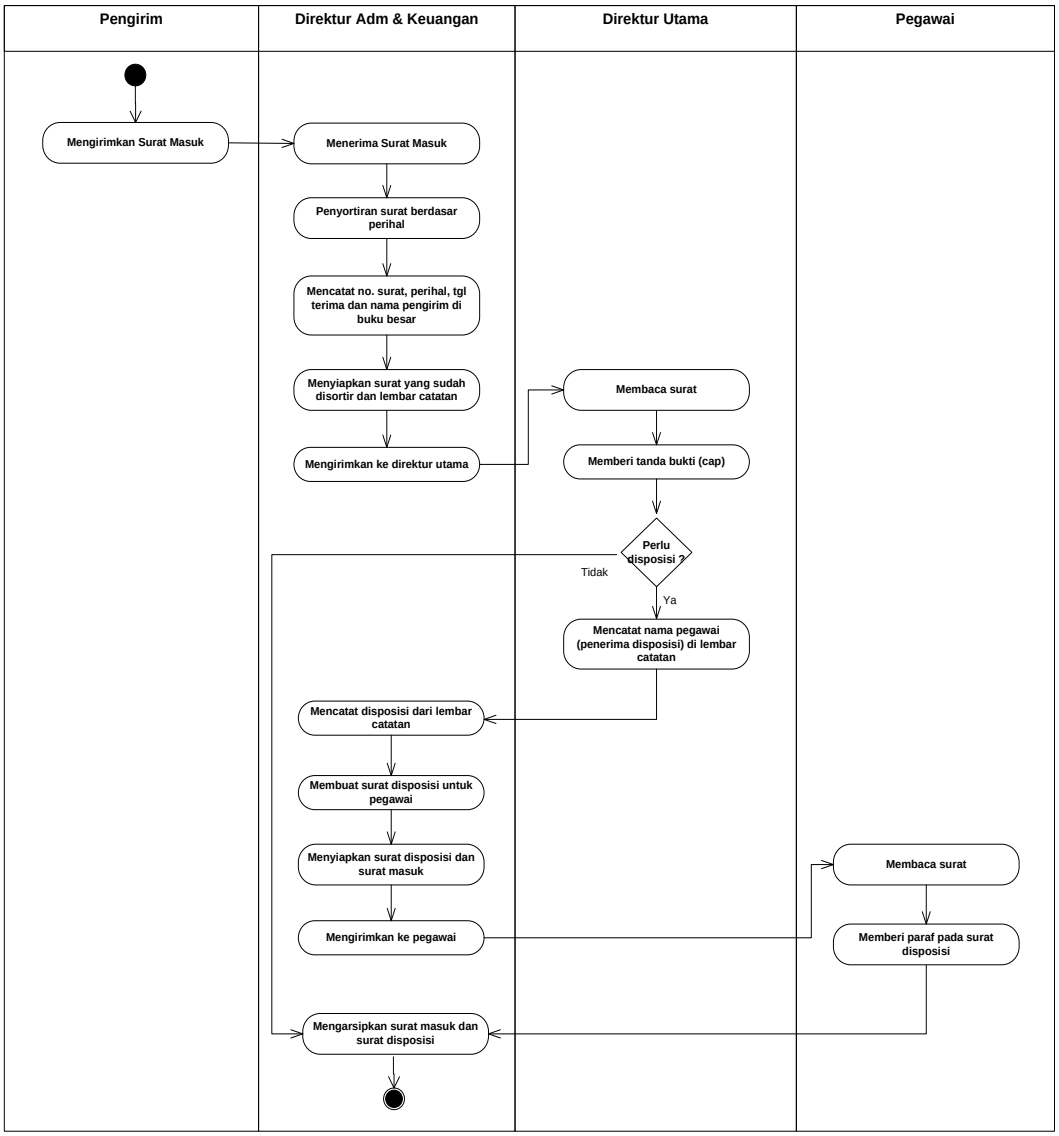
1. Prosedur Pengolahan Surat Masuk

Prosedur pengolahan surat masuk merupakan proses mengolah surat-surat yang masuk ke perusahaan yang dilakukan oleh bagian administrasi yaitu Direktur Divisi Administrasi dan Keuangan. Urutan proses-proses yang ada pada pengolahan surat masuk seperti yang terlihat pada gambar 3.1 adalah sebagai berikut :

a. Pengirim mengirimkan surat masuk ke PT. Meta Informatika Telekomindo.
b. Bagian administrasi menerima surat masuk dari pengirim di PT. Meta Informatika Telekomind.
c. Jika alamat tidak benar dikembalikan ke alamat pengirim. Jika alamat benar, surat masuk disortir berdasarkan perihalnya kemudian diberi tanda bukti.
d. Bagian administrasi mencatat nama pengirim dan tanggal masuknya surat secara manual dan disimpan dalam bentuk lembaran arsip.
e. Bagian administrasi akan mengirimkan surat masuk ke Direktur Utama.

f. Direktur Utama menerima surat masuk dan membacanya serta memberi tanda bukti pada surat.
g. Direktur Utama akan membuat instruksi tugas di lembar catatan dengan mencatat nama-nama pegawai yang berwenang sebagai penerima surat dan menyerahkannya kepada bagian administrasi.
h. Berdasarkan lembar catatan yang diberikan, bagian administrasi mengetahui nama penerima surat dan mengirimkannya kepada pegawai yang berwenang.
i. Pegawai yang berwenang akan membaca dan menindaklanjuti surat masuk dengan memberi paraf pada lembar catatan.
j. Bagian administrasi akan mengirimkann pada pegawai yang berwenang selajutnya.

Proses mengirimkan surat ini terus dilakukan dari satu pegawai ke pegawai lainnya sampai semua pegawai yang telah ditentukan oleh Direktur Utama sebagai penerima surat telah menerima dan membaca suratnya.



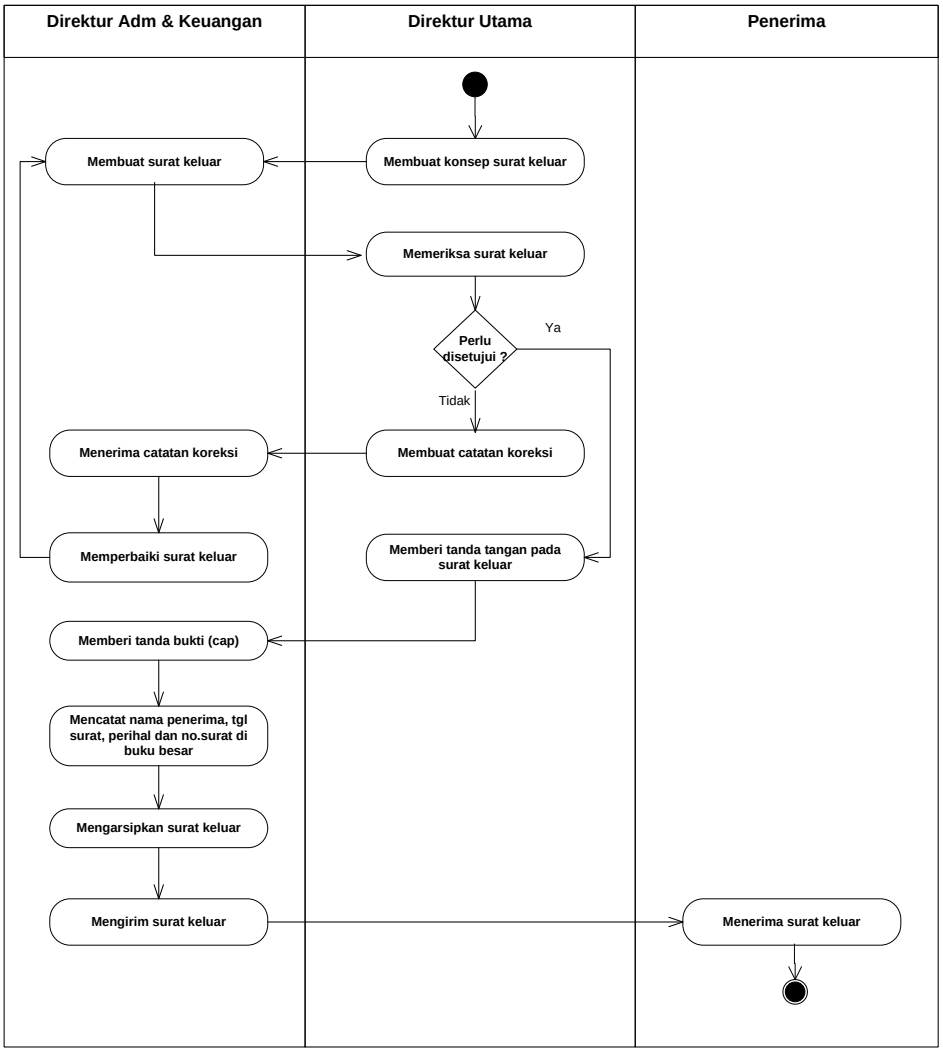
Gambar 3.1 *Activity Diagram* Pengolahan Surat Masuk

2. Prosedur Pengolahan Surat Keluar

Prosedur pengolahan surat keluar merupakan proses mengolah surat-surat yang akan dikirim keluar oleh perusahaan yang dilakukan oleh bagian administrasi yaitu Direktur Divisi Administrasi dan Keuangan. Urutan proses-proses pengolahan surat keluar seperti yang terlihat pada gambar 3.2 adalah sebagai berikut :

a. Direktur utama memberikan konsep surat keluar kepada bagian administrasi.
b. Konsep surat keluar diterima bagian administrasi dan dibuat berdasarkan konsep yang diterima.

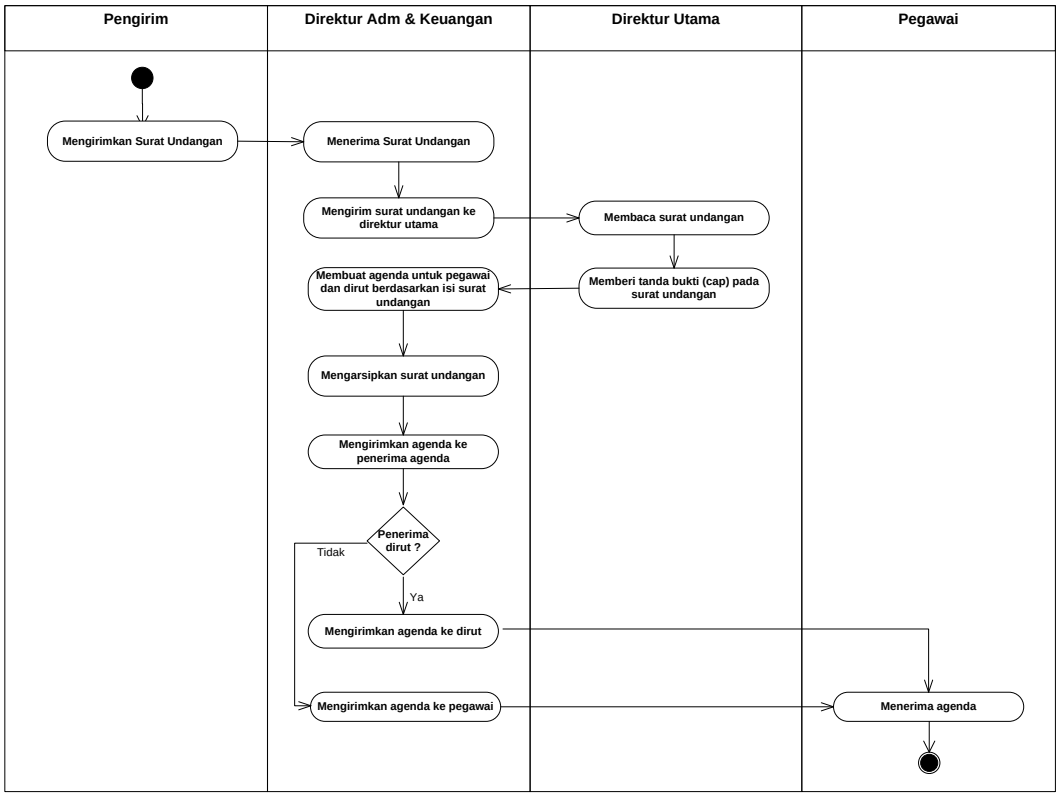
c. Surat yang telah dibuat kemudian diberikan kepada direktur utama untuk diperiksa dan disetujui, bila tidak disetujui surat akan dikembalikan lagi ke bagian administrasi untuk diperbaiki kembali. Tetapi bila disetujui maka bagian administrasi akan memberikan cap agar surat keluar menjadi sah.
d. Surat keluar diklasifikasikan berdasarkan tujuan dan perihalnya.
e. Bagian administrasi akan mencatat nama penerima dan tanggal pengiriman untuk diarsipkan.
f. Surat keluar siap untuk dikirimkan ke perusahaan yang dituju.



Gambar 3.2 *Activity Diagram* Pengolahan Surat Keluar

3. Prosedur Pengolahan Agenda

Peredaran tata surat di perusahaan bila ada pengirim yang mengirimkan surat undangan maka akan diagendakan setelah mendapat persetujuan dari Direktur Utama. Bagian administrasi akan menyebarkan kepada para pegawai yang bersangkutan. Proses-proses yang ada pada pengolahan agenda dapat dilihat pada gambar 3.3.



Gambar 3.3 *Activity Diagram* Prosedur Pengolahan Agenda

3.1.3 Analisis Aturan Bisnis

Analisis aturan bisnis berisikan analisis aturan-aturan yang berlaku pada sistem yang sedang berjalan. Analisis aturan bisnis yang berlaku di PT. Meta Informatika Telekomindo adalah sebagai berikut:

3.1.3.1 Analisis Sistem Pengolahan Surat

Sistem pengolahan surat merupakan sistem pengolahan, pengendalian, penyajian, pemeliharaan dan penyelamatan surat-surat perusahaan yang berada di lingkungan PT. Meta Informatika Telekomindo. Sistem ini intinya menangani pengiriman surat internal perusahaan. Surat-surat yang masuk ke dalam perusahaan akan dikirimkan oleh Divisi Administrasi dan Keuangan kepada Direktur Utama sebagai pimpinan tertinggi. Pimpinan akan memberikan disposisi untuk para pegawai yang berwenang dalam menindaklanjuti surat tersebut.

3.1.3.2 Penyortiran Surat

Setiap surat yang masuk ke perusahaan akan disortir terlebih dahulu sebelum masuk ke sistem pengolahan surat. Penyortiran tersebut diklasifiikasikan menjadi 2, yaitu:

1. Surat pribadi

   Surat pribadi adalah surat yang ditujukan khusus untuk pribadi, baik untuk pimpinan ataupun pegawai. Surat ini tidak masuk kedalam sistem pengolahan surat, sehingga pihak administrasi tidak berwenang untuk membukanya.
2. Surat resmi (dinas)

   Surat dinas adalah surat yang tekait dengan kegiatan perusahaan. Surat inilah yang masuk kedalam sistem pengolahan surat. Bentuknya bisa berupa undangan, penawaran, permintaan.

3.1.3.3 Proses Pengiriman Surat

Setiap surat yang masuk kedalam sistem pengolahan surat akan disortir kembali oleh bagian administrasi, diklasifikasikan berdasarkan perihal dari surat tersebut. Setelah itu surat dikirim ke Direktur Utama sebagai pimpinan tertinggi. Pimpinan menerima, membaca dan membuatkan disposisi serta menetukan nama-nama pegawai yang akan menindaklanjutinya. Surat dan disposisi dikirim ke pegawai yang bersangkutan.

Pegawai menerima dan membaca surat yang disertai disposisi. Pegawai akan menindaklanjuti surat disposisi tersebut dengan melaksanakan instruksinya.

Surat dikirim kembali ke bagian administrasi sebagai tanda bahwa surat tersebut telah ditindaklanjuti.

3.1.3.4 Surat Balasan

Aturan dari surat balasan dalam sistem pengolahan surat ini, yaitu :

1. Surat balasan adalah salah satu bentuk respon dari tindaklanjut surat yang masuk ke perusahaan.
2. Surat ini digunakan hanya untuk keperluan tertentu yaitu surat dengan perihal surat penawaran, sehingga tidak semua surat yang masuk ke perusahaan memerlukan surat balasan.
3. Surat balasan sebelum dikirim, datanya akan diarsipkan terlebih dahulu kedalam aplikasi *e-office* sehingga ketika suatu saat surat ini dibutuhkan maka bagian administrasi akan mudah mendapatkanya karena datanya telah tersimpan dalam arsip.

3.1.3.5 Pencarian Surat

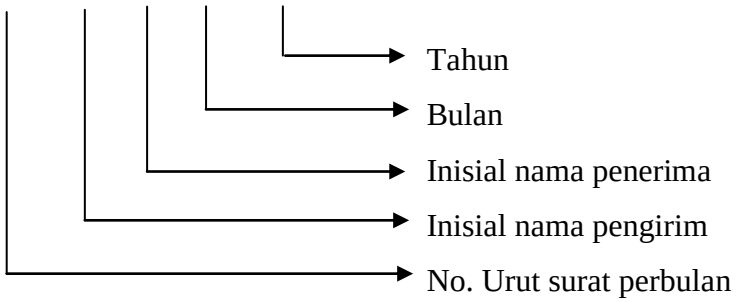
Pencarian surat adalah pencarian surat-surat yang ada didalam sistem pengolahan surat. Pencarian ini dapat membaca kata kunci walau hanya satu kata yang dimasukkan tanpa ada pengkategorian data. Keluaran dari pencarian ini adalah list data yang sesuai dengan kata kunci yang dimasukkan.

3.1.3.6 Analisis Pengkodean

Tahap analisis pengkodean akan membahas tentang pengkodean yang ada di PT Meta Informatika Telekomindo. Data pengkodean untuk nomor surat perusahaan dimana kode merupakan penyajian dalam mengklafikasikan data sehingga mudah dalam proses masukan ke dalam sistem program. Pengunaan kode biasanya untuk mengidentifikasi data, simbol kode biasanya digunakan pada hampir semua proses yang ada keterkaitannya dengan data. Sistem yang sedang berjalan memiliki kode untuk surat keluar dan NIP.

1. Pengkodean untuk surat keluar

Format : 999/AAA-AAA/AA/9999

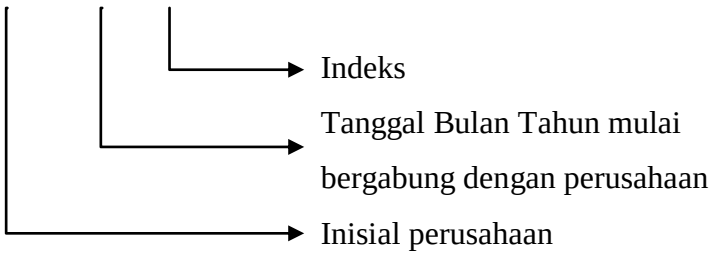


Contoh : 001/MIT-BIM/II/2013

Keterangan : tiga digit pertama menyatakan nomor urut surat perbulan, tiga digit kedua menyatakan inisial nama pengirim yaitu inisial nama perusahaan MIT, tiga digit ketiga menyatakan inisial nama penerima yaitu inisial nama dari penerima surat dimana diambil awal huruf untuk tiap kata dari nama penerima, dua digit keempat dibuat dengan angka romawi menyatakan bulan surat dibuat dan empat digit terakhir menyatakan tahun surat dibuat.

2. Pengkodean nomor induk pegawai (NIP)

Format : AAA 999999 9



Contoh : MIT2105092

Keterangan : tiga digit pertama menyatakan inisialperusahaan, 6 digit kedua menyatakan tanggal mulai bergabung dengan perusahaan, yaitu tanggal 21 bulan Mei tahun 2009, dan satu digit terakhir menyatakan indeks untuk nomor urut pegawai.

3.1.4 Spesifikasi Kebutuhan Perangkat Lunak (SKPL)

Perangkat lunak adalah istilah umum untuk data yang diformat dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang dibaca dan ditulis oleh komputer.

1. Perangkat lunak menyediakan fungsi login bagi pegawai yang terdaftar
2. Perangkat lunak dapat mengelola data pegawai.
3. Perangkat lunak dapat mengelola data surat masuk.
4. Perangkat lunak dapat mengelola data surat keluar.
5. Perangkat lunak dapat mengelola data agenda.
6. Perangkat lunak dapat menginputkan data disposisi.
7. Perangkat lunak dapat memfasilitasi surat masuk, surat keluar, agenda, dan disposisi bagi pimpinan.
8. Perangkat lunak dapat memfasilitasi untuk melihat agenda, disposisi dan pesan bagi pegawai yang terdaftar.
9. Perangkat lunak dapat memfasilitasi untuk mengirim pesan bagi pegawai yang terdaftar.
10. Perangkat lunak dapat memfasilitasi untuk pencarian arsip bagi administrator dan operator.

3.1.5 Analisis Kebutuhan Data

Analisis kebutuhan data bertujuan untuk menganalisis kebutuhan data yang diperlukan dan memudahkan perancangan informasi setelah sistem yang dibutuhkan telah diketahui. Analisis kebutuhan data memiliki kriteria sebagai berikut :

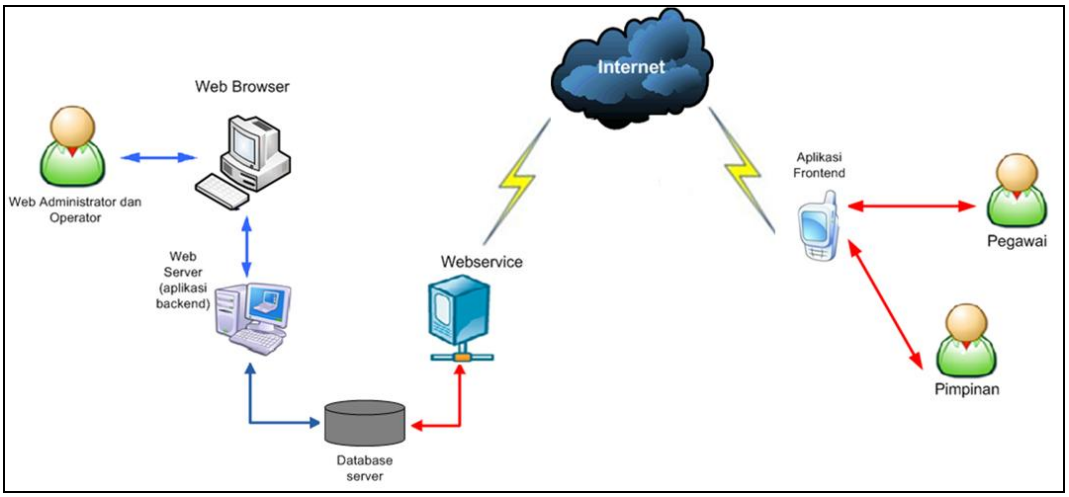
1. Data pegawai di PT. Meta Informatika Telekomindo yang meliputi nip, nama, jabatan, jenis kelamin, alamat, *e-mail*.
2. Data surat-surat yang meliputi data surat masuk dan surat keluar

3.1.6 Analisis Arsitektur Sistem

Penelitian ini merupakan penelitian lapangan dan bersifat eksperimen yaitu membangun aplikasi perangkat lunak *e-office* dengan objek penelitian pada PT.

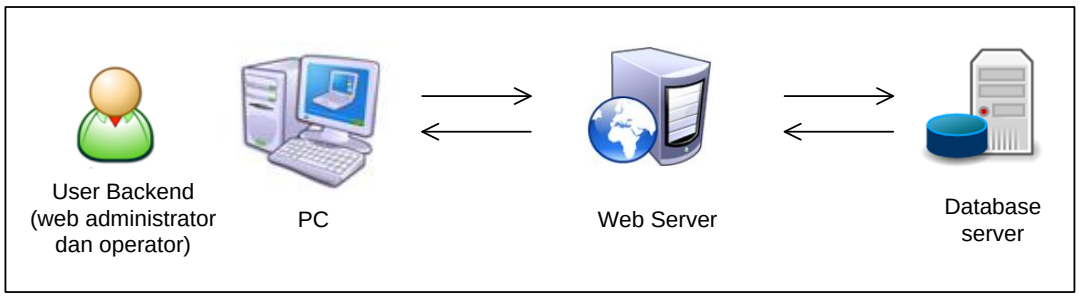
Meta Informatika Telekomindo khususnya pada bagian administrasi perusahaan yang tugasnya mengelola semua sistem tata surat yang terjadi pada perusahaan.

Arsitektur fisik sistem yang dipakai dalam penelitian ini adalah menggunakan model arsitektur aplikasi *two tier* dimana arsitektur fisik sistem ini mempunyai tiga komponen utama yaitu *Client (Frontend), application server* (*Web admin dan Web Service*) dan *database server*.



Gambar 3. 4 Arsitektur sistem

Gambar 3.4 menjelaskan alur sistem yang akan dibangun. Sistem memiliki dua aplikasi yang akan dibangun yaitu aplikasi *backend* dan aplikasi *Frontend.* Penjelasan untuk masing-masing aplikasi dapat dilihat pada gambar 3.5 dan 3.6.

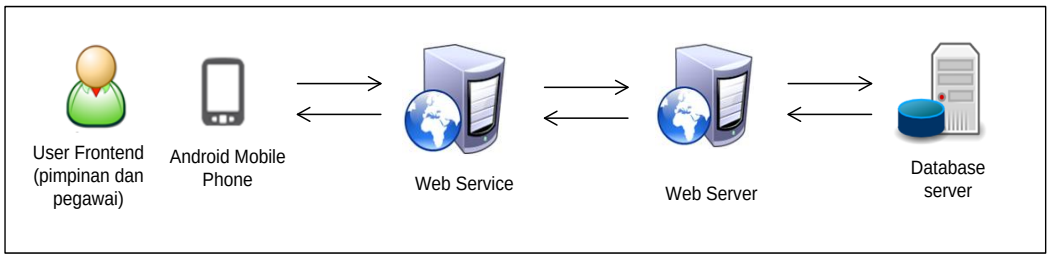


Gambar 3.5 Arsitektur Sistem Aplikasi *Backend*

Gambar 3.5 menjelaskan alur sistem aplikasi *e-office backend* yang akan dibangun. Aplikasi *Backend* mempunyai beberapa sistem pendukung diantaranya yaitu *user backend* (web administrator dan operator) , *web server* dan *database*

server. Aplikasi *backend* ini berupa aplikasi web yang akan berfungsi sebagai penghubung antara *aplikasi frontend* dengan database *server* dan melakukan proses pengolahan sumber basis data. Pada aplikasi *Backend* ini diperlukan koneksi internet untuk proses komunikasi data. Proses yang dilakukan yaitu pengguna mengakses *website* yang terdapat pada *web server*, jika terjadi *request* dari *user backend* untuk mengakses *database*, maka *web server* akan meneruskannya kepada *server database* yang menggunakan MySql dan menampilkan hasilnya pada *website* yang diakses oleh pengguna. *Web server* berisi file–file html dan php yang berfungsi untuk menampilkan *website.*

Bentuk arsitektur sistem aplikasi *frontend* dapat dilihat pada gambar 3.6. Aplikasi *frontend* digunakan pada SmartPhone bersistem operasi Android. Aplikasi *frontend* merupakan perangkat SmartPhone yang bertindak sebagai *client* yang akan mengakses *web service* yang terdapat pada *server.* Aplikasi Android disini akan memanggil fungsi-fungsi yang berada di dalam *web service.* Aplikasi android dapat dijalankan bila perangkat SmartPhone telah diinstal aplikasi dalam bentuk format .apk untuk menampilkan data–data yang terdapat pada *database server*. Penjembatan antara aplikasi klien dan *database server*, digunakan perantara file php yang terdapat pada *server web service*. File–file php berfungsi untuk menangkap *request* dari aplikasi java dan berfungsi untuk memberikan *response* kepada aplikasi java. File php tersebut merupakan penjembatan antara *database* MySql dengan aplikasi java.



Gambar 3.6 Arsitektur Sistem Aplikasi *Frontend*

Komunikasi data JSON digunakan untuk *request* dan *response* data yang cukup banyak. Data dari *server web service* diolah ke dalam php sehingga membentuk suatu obyek, kemudian obyek tersebut di-*encode* dan dikirimkan

kepada aplikasi java. Proses pada aplikasi java, yaitu *response* yang diberikan oleh *server web service* di-*decode* menjadi suatu obyek yang nantinya dapat diproses ke dalam aplikasi android (*frontend*). Agar semua *request* dan *response* data yang terjadi antara aplikasi android dengan *web service* berjalan baik maka digunakan penghubung akses internet dengan protokol HTTP.

Aplikasi *frontend* ini memiliki semua informasi mengenai surat masuk, surat keluar, agenda, disposisi dan pesan yang ada dilingkup internal perusahaan. Pengguna aplikasi *frontend* ini dibagi menjadi dua yaitu pimpinan dan pegawai. Pengguna tersebut memiliki akses sistem yang berbeda, yaitu pimpinan dapat melihat semua informasi mengenai surat masuk, surat keluar, agenda, disposisi dan pesan sedangkan pegawai hanya memiliki akses sistem untuk melihat infromasi agenda, disposisi dan pesan saja. Selain itu pada aplikasi ini juga tersedia fitur notifikasi yang berfungsi untuk memberikan tanda dari data baru yang masuk ke aplikasi *frontend*. Akses koneksi jaringan yang digunakan dari *frontend* untuk berintegrasi dengan *backend* melalui perangkat *mobile* dapat menggunakan jaringan *mobile* GPRS/EDGE/UMTS.

3.1.7 Analisis Kebutuhan Non Fungsional

Tahap ini meliputi elemen-elemen apa saja yang dibutuhkan untuk sebuah sistem yang akan dibangun, spesifikasi masukan yang diperlukan sistem, keluaran yang dihasilkan sistem dan proses yang dibutuhkan sampai sistem tersebut diimplementasi. Kebutuhan non-fungsional terbagi menjadi tiga bagian yaitu kebutuhan perangkat lunak, perangkat keras dan pengguna.

3.1.7.1 Analisis Perangkat Keras

Komputer dan *smartphone* terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat keras atau *hardware* merupakan peralatan fisik dari komputer yang dapat kita lihat dan rasakan. *Hardware* atau perangkat keras ini adalah salah satu perangkat yang sangat penting, karena tanpa *hardware* yang memenuhi syarat, sebuah aplikasi yang dibangun tidak akan berjalan tanpa adanya perangkat keras.

1. Spesifikasi umum perangkat keras yang ada pada PT. Meta Informatika Telekomindo, yaitu :
   a) Processor intel Core 1.6 GHz
   b) *Harddisk* 250 GB
   c) RAM 1 GB
   d) *VGA* 256 MB
   e) Monitor 14”
   f) *Keyboard* dan *mouse*
   g) *Scanner*

2. Spesifikasi minimum perangkat keras yang dibutuhkan untuk dapat menjalankan aplikasi, yaitu :
   a. Aplikasi *backend* (web)
      1) PC yang terhubung dengan jaringan (internet)
      2) *Monitor*, *Keyboard* dan *Mouse*
      3) *Scanner*
      4) Selebihnya aplikasi ini tidak membutuhkan antarmuka perangkat keras yang spesifik
   b. Aplikasi *frontend* (*mobile*)
      Aplikasi *frontend* dijalankan pada sebuah perangkat *SmartPhone* yaitu *SmartPhone* dengan sistem operasi berbasis Android.

Spesifikasi perangkat keras yang ada di PT. Meta Informatika Telekomindo sudah memenuhi syarat untuk pembangunan aplikasi *e-office.* Terutama untuk aplikasi *frontend* karena sudah banyaknya pengguna *smartphone* android di PT. Meta Informatika.

3.1.7.2 Analisis Perangkat Lunak

Perangkat lunak adalah istilah umum untuk data yang diformat dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang dibaca dan ditulis oleh komputer.

1. Spesifikasi perangkat lunak pada sistem yang sedang berjalan

   PT. Mitra Informatika Telekomindo menggunakan sistem operasi *Windows XP*, aplikasi perkantoran office. Untuk penggunaan web sudah menggunakan web browser google dan Mozilla firefox serta telah disediakan juga fasilitas akses internet oleh perusahaan.
2. Spesifikasi minimum perangkat lunak untuk dapat menjalankan aplikasi, yaitu :
   a. Aplikasi *backend (web)*
      1) Sistem Operasi Windows XP
      2) *Browser* Mozilla Firefox.
   b. Aplikasi *frontend (mobile)*, yaitu smartphone dengan sistem operasi Android versi 2.2 (froyo).

Spesifikasi perangkat lunak yang ada di PT. Meta Informatika Telekomindo sudah memenuhi syarat untuk pembangunan aplikasi *e-office* karena perangkat lunak minimum yang dibutuhkan hanya sistem operasi windows XP dan web browser dengan akses internet. Sedangkan untuk pengguna perangkat *smartphone* android di perusahaan sudah banyak yang memiliki *smartphone* android versi 2.2 keatas.

3.1.7.3 Analisis Pengguna *(User)*

Analisis pengguna sistem dimaksudkan untuk mengetahui siapa saja *actor* yang terlibat dalam menjalankan sistem. Sistem aplikasi *e-office* menggunakan platform teknologi *mobile* dan web.

Berdasarkan hasil pengumpulan data, diketahui bahwa tingkat pendidikan terendah yang dimiliki pengguna adalah D3, namun semua pengguna terbiasa menggunakan komputer dan menjalankan aplikasi *office* serta pernah mendapatkan pelatihan penggunaan aplikasi *Microsoft Office*.

Penggunaan sistem yang akan dibangun agar maksimal maka pengguna perangkat lunak ini dibagi menjadi dua, yaitu:

1. Pengguna aplikasi *frontend* yaitu khusus hanya untuk pimpinan dan para karyawan di PT. Meta Informatika Telekomindo. Pimpinan disini yaitu

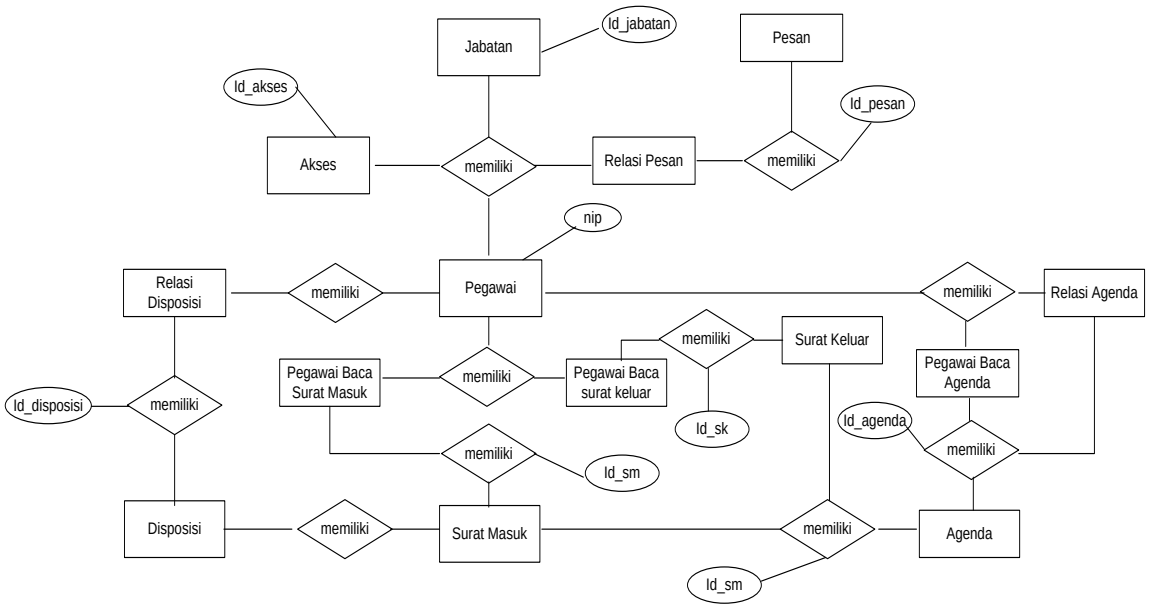
direktur utama yang mana dapat melakukan aktivitas melihat surat masuk, surat keluar, agenda, pesan, membuat disposisi, menyetujui surat keluar dan memberikan catatan koreksi terhadap surat keluar. Pegawai dapat melihat agenda, disposisi dan pesan. Para pengguna menggunakan *smartphone* dengan *platform* android versi 2.2 atau lebih tinggi. Penggunasistem ini, diharuskan terhubung dengan koneksi jaringan internet.

2. Pengguna aplikasi *backend* sebagai pengelola konten yaitu Direktur Divisi Administrasi dan Keuangan yang bertugas sebagai administrator dan operator yang dapat mengelola *database* seperti menambah, mengubah, dan menghapus data pegawai, surat masuk, surat keluar dan agenda.

Spesifikasi pengguna yang ada di PT. Meta Informatika Telekomindo sudah memenuhi syarat untuk pembangunan aplikasi *e-office* karena *user backend* di PT. Meta Informatika Telekomindo yaitu direktur administrasi dan keuangan sudah dapat menggunakan komputer dan *web browser* dalam melaksanakan aktifitas pekerjaan sehari-hari. *User frontend* juga sudah memenuhi persyaratan untuk penggunaan aplikasi karena pimpinan dan pegawai sudah memiliki perangkat *mobile* yang berbasis android sehingga sudah terbiasa menggunakan fitur-fitur pada *smartphone* android.

3.1.7.4 Analisis Basis Data

Memodelkan data dan hubungan-hubungan data yang ada pada aplikasi *e-office* digunakan sebuah alat bantu pemodelan yaitu *Entity Relationship Diagram* (ERD). Hubungan antar entitas yang terjadi dalam sistem aplikasi *e-office* yang akan dirancang dapat dilihat pada gambar 3.7.



Gambar 3.7 *Entity Relationship Diagram* (ERD) Aplikasi *E-office*

3.1.8 Analisis Kebutuhan Fungsional

Analisis kebutuhan sistem merupakan salah satu usaha untuk mengetahui apa saja yang dibutuhkan oleh sistem yang akan dibangun, tentunya hal ini sangat penting untuk dipenuhi karena selain kebutuhan non fungsional ada kebutuhan fungsional yang harus dipenuhi agar sistem *e-office mobile* dapat berjalan dengan baik. Analisis kebutuhan fungsional akan membahas proses informasi yang mengalir melalui perangkat lunak. Analisis kebutuhan fungsional ini meliputi pemodelan sistem. Pemodelan sistem dimodelkan dengan menggunakan UML (*Unified Modelling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain *Use Case Diagram, Class Diagram, Activity Diagra* dan *Sequence Diagram,*

3.1.8.1 Pemodelan Sistem

Pemodelan sistem berfungsi untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem pada aplikasi serta menentukan kelas yang dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

3.1.8.1.1 *Use Case Diagram*

Use case diagram digunakan untuk memodelkan proses bisnis dan merepresentasikan sebuah interaksi antara aktor dengan sistem. Mendiskripsikan fungsi dari sebuah sistem dari perspektif pengguna, use case bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebauh sistem dipakai. *Use case* terdiri dari tiga bagian yaitu identifikasi *actor*, identifikasi *use case*, dan *scenario use case*.

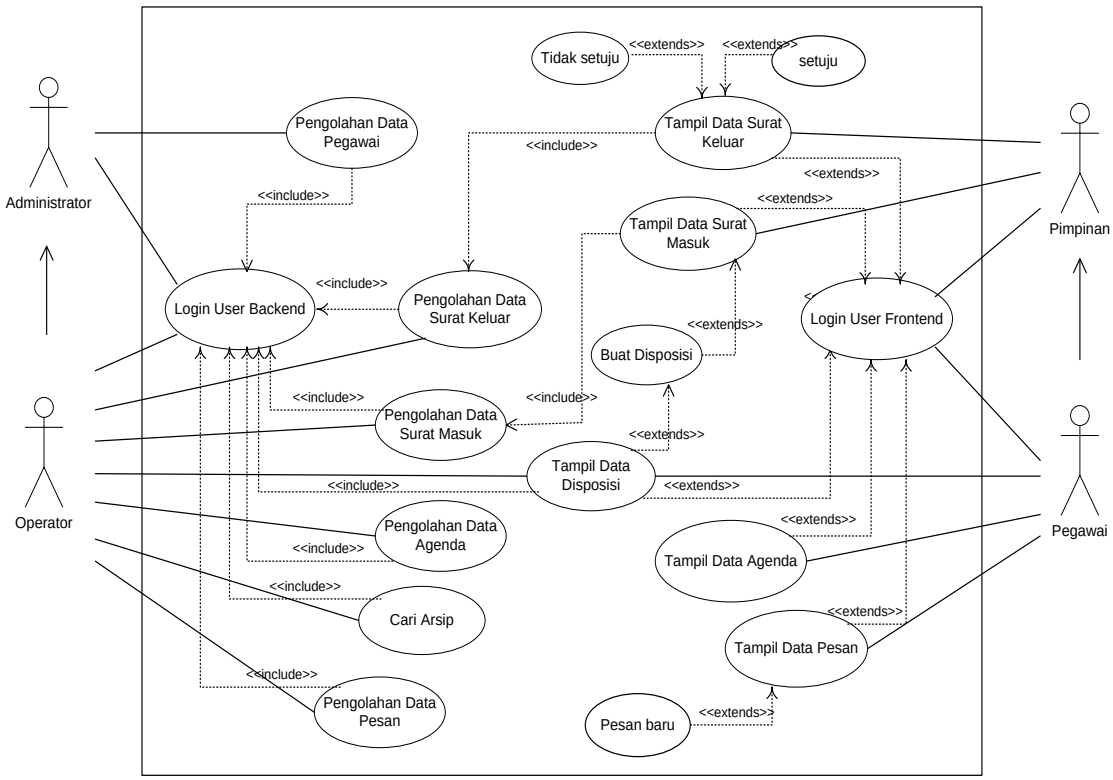
a. Identifikasi *Actor*

   Aktor adalah abstraksi dari orang dan sistem yang lain yang mengaktifkan fungsi dari target sistem. Pada tabel 3.1 dapat dilihat aktor-aktor yang berperan dalam menjalan sistem yang dibangun.

Tabel 3.1 Tabel *Actor Use Case*

No	Aktor	Deskripsi
A-01	Pengguna *Frontend*: Pegawai dan Pimpinan	Merupakan *actor* atau pengguna dari aplikasi yang dibangun, aplikasi *frontend* (*e-office mobile*)
A-02	Pengguna *Backend*: Administrator dan Operator	Merupakan *actor* yang berperan mengelola data informasi dalam *database server* melalui CMS berbasis web.

Penggambaran *use case diagram* pada aplikasi *e-office* dapat dilihat pada gambar 3.8.



Gambar 3.8 *Use Case Diagram* Aplikasi *E-Office*

b. Skenario *use case*

Skenario *Use Case* adalah alur cerita atau proses-proses yang terjadi pada sistem antara aktor dengan *Use Case*. Skenario *Use Case* dapat dilihat pada tabel 3.2 sampai tabel 3.14.

Tabel 3.2 Skenario *Use Case Login User Backend*

Identifikasi	
No	01
Nama Use Case	*Login User Backend*
Aktor	Administrator Operator
Tujuan	Memastikan hak akses user backend
Deskripsi	Aplikasi akan memeriksa apakah pengguna memiliki hak akses terhadap aplikasi dengan memasukan nip dan password

Keadaan awal	Menampilan Halaman Login
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memasukkan nip dan password	-
2. Memilih tombol masuk	3. Mengautentikasi nip dan password pada database server
-	4. Jika cocok maka menu utama ditampilkan, jika tidak maka muncul pesan kesalahan login
Kondisi Akhir	Menampilkan Halaman utama aplikasi

Tabel 3.3 Skenario *Use Case* Pengolahan Data Pegawai

Identifikasi	
No	02
Nama Use Case	Pengolahan Data pegawai
Aktor	Administrator (*user backend*)
Tujuan	Mengelola data pegawai
Deskripsi	Sistem menyediakan fungsionalitas untuk menampilkan dan mengolah data pegawai
Keadaan awal	Sistem menampilkan halaman kelola data pegawai
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu daftar pegawai	2. Menampilkan halaman kelola pegawai
3. Memilih tombol pengolahan data pegawai (tambah, edit dan hapus)	4. Melakukan proses pengolahan data pegawai
-	5. Menyimpan proses hasil pengolahan data pegawai
Kondisi Akhir	Menampilkan data pegawai

Tabel 3.4 Skenario *Use Case* Pengolahan Data Surat Masuk

Identifikasi	
No	03
Nama Use Case	Pengolahan Data Surat masuk
Aktor	Administrator dan Operator (*user backend*)
Tujuan	Mengelola data surat masuk
Deskripsi	Sistem menyediakan fungsionalitas untuk menampilkan dan mengolah data surat masuk
Keadaan awal	Sistem menampilkan halaman kelola data surat masuk
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu surat masuk	2. Menampilkan halaman kelola surat masuk
2. Memilih tombol pengolahan data surat masuk (tambah, edit dan hapus)	3. Melakukan proses pengolahan data surat masuk
-	4. Menyimpan proses hasil pengolahan data surat masuk
Kondisi Akhir	Menampilkan data surat masuk

Tabel 3.5 Skenario *Use Case* Tampil Data Disposisi

Identifikasi	
No	04
Nama Use Case	Tampil Data Disposisi
Aktor	Administrator dan Operator (*user backend*) Pimpinan dan Pegawai (*user frontend*)
Tujuan	Melihat data disposisi
Deskripsi	Sistem menyediakan halaman untuk melihat data disposisi
Keadaan awal	Sistem menampilkan halaman utama
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu disposisi	2. Menampilkan list data disposisi

3. Memilih salah satu disposisi	4. Menampilkan detail data disposisi
Kondisi Akhir	Menampilkan detail data disposisi

Tabel 3.6 Skenario *Use Case* Pengolahan Data Surat Keluar

Identifikasi	
No	05
Nama Use Case	Pengolahan Data Surat keluar
Aktor	Administrator dan Operator (*user backend*)
Tujuan	Mengelola data surat keluar
Deskripsi	Sistem menyediakan fungsionalitas untuk menampilkan dan mengolah data surat keluar
Keadaan awal	Sistem menampilkan halaman kelola data surat keluar
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu surat keluar	2. Menampilkan halaman kelola surat keluar
3. Memilih tombol pengolahan data surat keluar (tambah, edit dan hapus)	4. Melakukan proses pengolahan data surat keluar
-	5. Menyimpan proses hasil pengolahan data surat keluar
Kondisi Akhir	Menampilkan data surat keluar

Tabel 3.7 Skenario *Use Case* Pengolahan Data Agenda

Identifikasi	
No	06
Nama Use Case	Pengolahan Data Agenda
Aktor	Administrator dan Operator (*user backend)*
Tujuan	Mengelola data agenda
Deskripsi	Sistem menyediakan fungsionalitas untuk menampilkan dan mengolah data agenda
Keadaan awal	Sistem menampilkan halaman kelola data agenda
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**

1. Memilih menu agenda	2. Menampilkan halaman kelola data agenda
3. Memilih tombol pengolahan data agenda (tambah, edit dan hapus)	4. Melakukan proses pengolahan data agenda
-	5. Menyimpan proses hasil pengolahan data agenda
Kondisi Akhir	Menampilkan data agenda

Tabel 3.8 Skenario *Use Case* Cari Arsip

Identifikasi	
No	07
Nama Use Case	Cari Arsip
Aktor	Administrator dan Operator (*user backend*)
Tujuan	Mencari arsip surat
Deskripsi	Sistem menyediakan fungsionalitas untuk mencari arsip
Keadaan awal	Sistem menampilkan halaman cari arsip
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memasukkan query untuk arsip yang akan dicari	-
-	2. Melakukan proses pencarian data berdasarkan *query* yang dimasukan
	3. Menampilkan data arsip yang dicari
Kondisi Akhir	Menampilkan data arsip yang dicari

Tabel 3.9 Skenario *Use Case* Pengolahan Data Pesan

Identifikasi	
No	08
Nama Use Case	Pengolahan Data Pesan
Aktor	Administrator dan Operator (*user backend*)
Tujuan	Mengelola pesan

Deskripsi	Sistem menyediakan halaman untuk mengelola pesan
Keadaan awal	Sistem menampilkan menu pesan
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu pesan	2. Menampilkan halaman pesan
3. Memilih tombol pengolahan data pesan (tambah, edit dan hapus)	4. Melakukan proses pengolahan data pesan
-	5. Menyimpan proses hasil pengolahan data pesan
Kondisi Akhir	Menampilkan data pesan

Tabel 3.10 Skenario *Use Case Login User Frontend*

Identifikasi	
No	09
Nama Use Case	*Login User Frontend*
Aktor	Pimpinan Pegawai
Tujuan	Memastikan hak akses user frontend
Deskripsi	Aplikasi akan memeriksa apakah pengguna memiliki hak akses terhadap aplikasi dengan memasukan nip dan password
Keadaan awal	Menampilan Halaman Login
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memasukkan nip dan password	-
2. Memilih tombol masuk	3. Memparsing data ke dalam format JSON
-	4. Mengautentikasi nip dan password pada database server
-	5. Jika cocok maka menu utama ditampilkan, jika tidak maka muncul pesan kesalahan login

Kondisi Akhir	Menampilkan Halaman utama aplikasi

Tabel 3.11 Skenario *Use Case* Tampil Data Surat Masuk

Identifikasi	
No	10
Nama Use Case	Tampil Data Surat masuk
Aktor	Pimpinan *(user frontend)*
Tujuan	Melihat data surat masuk
Deskripsi	Sistem menyediakan halaman untuk melihat data surat masuk
Keadaan awal	Sistem menampilkan halaman utama
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu surat masuk	2. Menampilkan list data surat masuk
3. Memilih salah satu list surat masuk	4. Menampilkan detail data surat masuk
5. Memilih tombol buat disposisi	6. Menampilkan halaman tambah data disposisi
7. Memasukkan data disposisi dan penerima, lalu menekan tombol kirim	8. Menyimpan data disposisi yang telah dimasukkan
	9. Menampilkan pesan "data telah terkirim ke server" dan kembali ke halaman utama
Kondisi Akhir	Menampilkan data surat masuk

Tabel 3.12 Skenario *Use Case* Tampil Data Surat Keluar

Identifikasi	
No	11
Nama Use Case	Tampil Data Surat keluar
Aktor	Pimpinan *(user frontend)*
Tujuan	Melihat data surat keluar
Deskripsi	Sistem menyediakan halaman untuk melihat data surat

	keluar
Keadaan awal	Sistem menampilkan halaman utama
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu surat keluar	2. Menampilkan list data surat keluar
3. Memilih salah satu list surat keluar	4. Menampilkan detail data surat keluar
5. Memilih tombol tidak setuju	6. Menampilkan halaman catatan
7. Memasukkan data catatatan perbaikan surat keluar dan menekan tombol kirim	8. Menyimpan data catatan perbaikan ke server
-	9. Menampilkan pesan "data telah terkirim ke server" dan kembali ke halaman utama
10. Memilih tombol setuju	11. Menyimpan data setuju ke server
-	12. Menampilkan pesan "data telah terkirim ke server" dan kembali ke halaman utama
Kondisi Akhir	Menampilkan data surat keluar

Tabel 3.13 Skenario *Use Case* Tampil Data Agenda

Identifikasi	
No	12
Nama Use Case	Tampil Data Agenda
Aktor	Pimpinan dan Pegawai (*user frondend*)
Tujuan	Melihat data agenda
Deskripsi	Sistem menyediakan halaman untuk melihat data agenda
Keadaan awal	Sistem menampilkan halaman utama
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu agenda	2. Menampilkan list data agenda
3. Memilih salah satu list agenda	4. Meminta detail data agenda yang

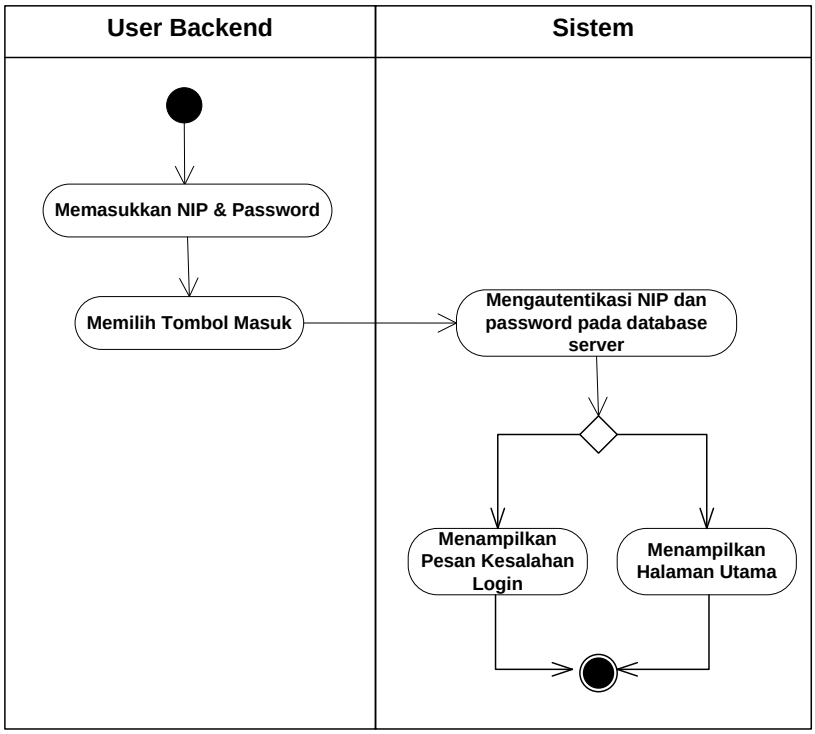
	dipilih ke server
-	5. Menampilkan detail agenda
Kondisi Akhir	Menampilkan detail data agenda

Tabel 3.14 Skenario *Use Case* Tampil Data Pesan

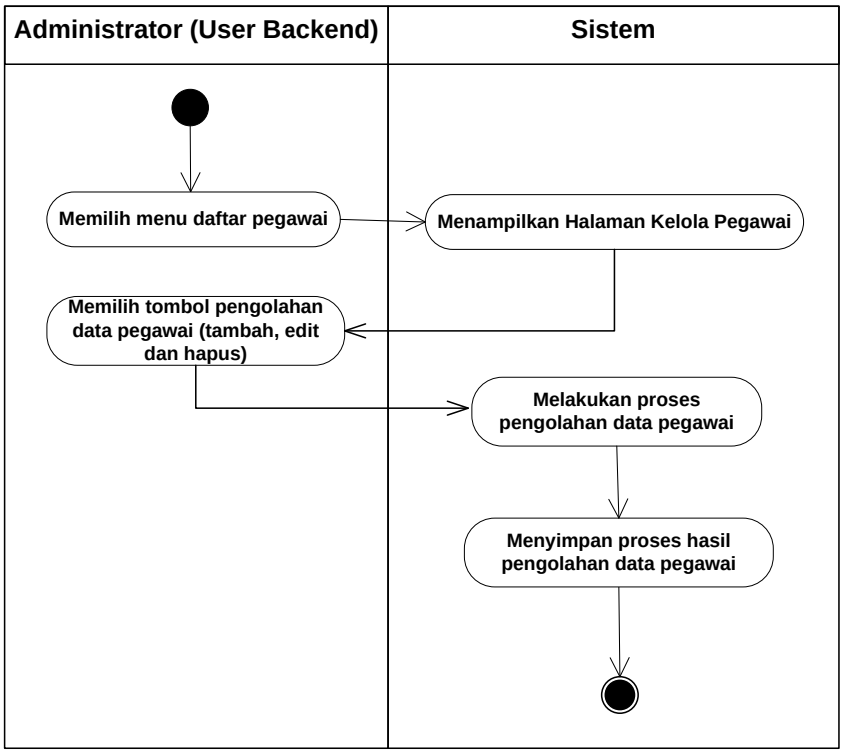
Identifikasi	
No	13
Nama Use Case	Tampil Data Pesan
Aktor	Pimpinan dan Pegawai (*user frondend*)
Tujuan	Melihat data Pesan dan membuat pesan
Deskripsi	Sistem menyediakan halaman untuk melihat data pesan
Keadaan awal	Sistem menampilkan halaman utama
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu pesan	2. Menampilkan list data pesan
3. Memilih salah satu list pesan	4. Meminta detail data pesan yang dipilih ke server
-	5. Menampilkan detail data pesan
6. Memilih tombol pesan baru	7. Menampilkan halaman buat pesan
7. Mengisi data pesan dan penerima pesan, menekan tombol kirim	8. Mengrimkan dan menyimpan data ke server
-	9. Kembali ke halaman utama
Kondisi Akhir	Menampilkan data pesan

3.1.8.1.2 *Activity Diagram*

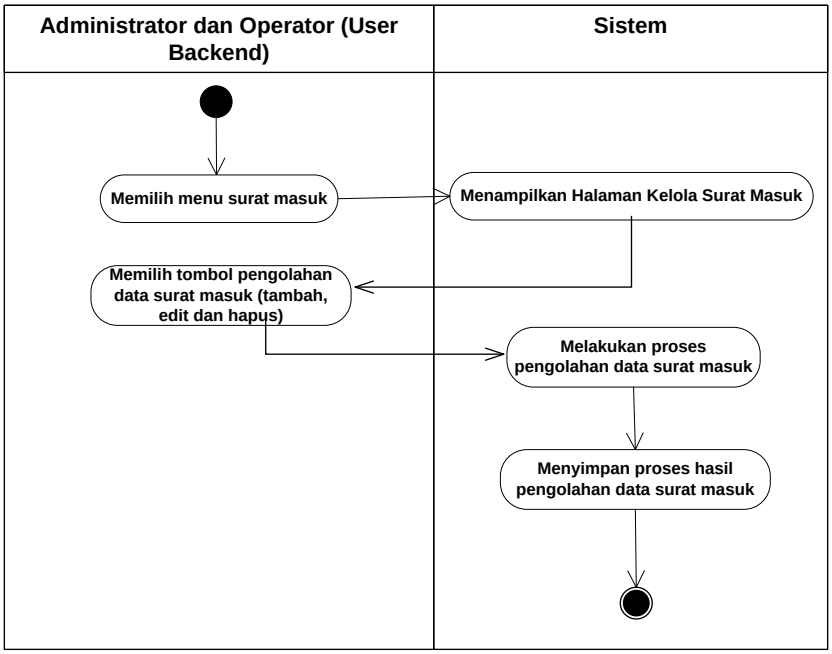
Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. Penggambaran *activity* diagram memiliki kemiripan dengan *flowchart diagram. Activity diagram* memodelkan *event-event* yang terjadi pada *Use Case* dan digunakan untuk pemodelan aspek dinamis dari sistem.



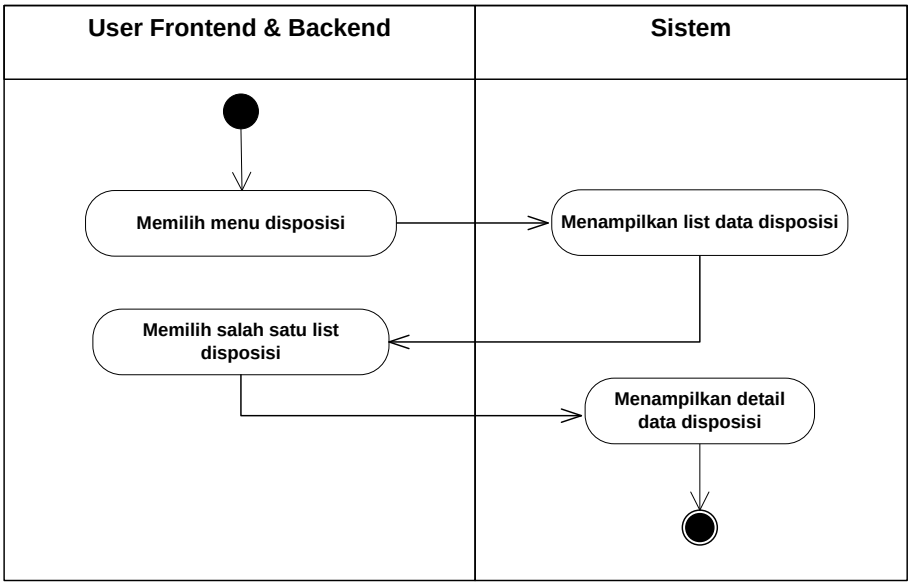
Gambar 3.9 ***Activity diagram*** **proses** ***login user backend***



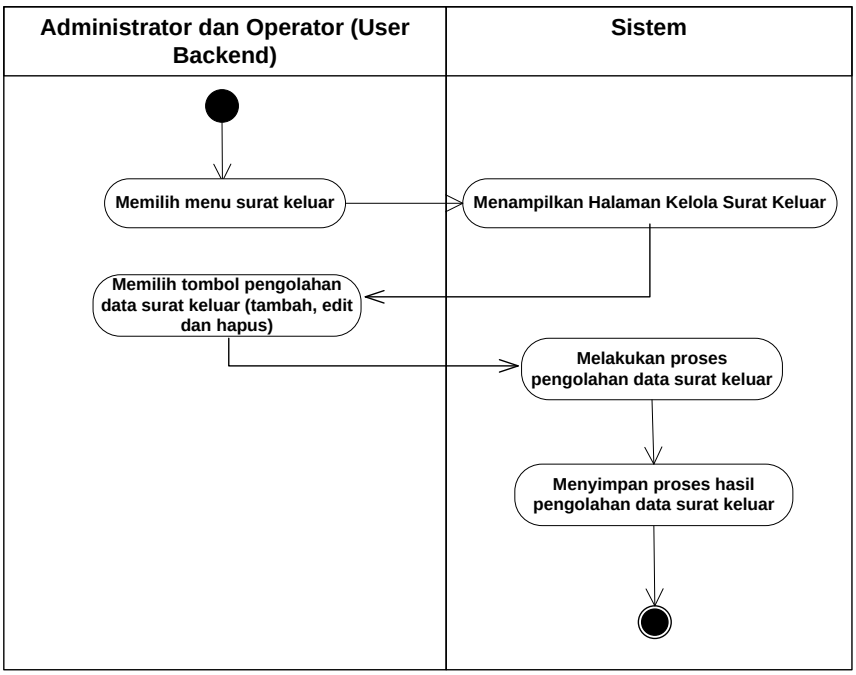
Gambar 3.10 ***Activity diagram*** **Pengolahan Data Pegawai**



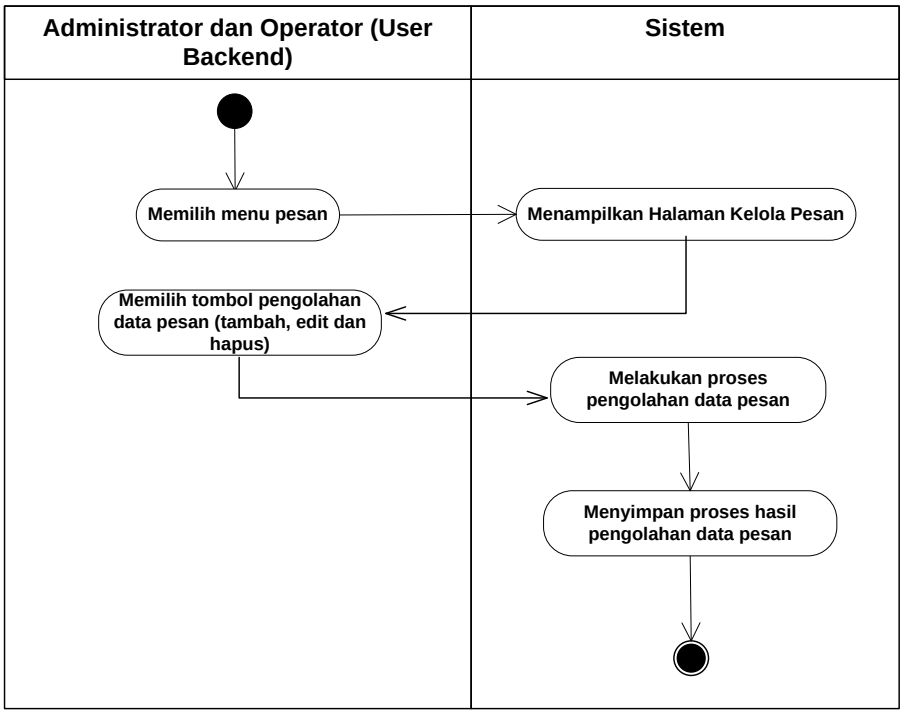
Gambar 3.11 *Activity diagram* Pengolahan Data Surat Masuk



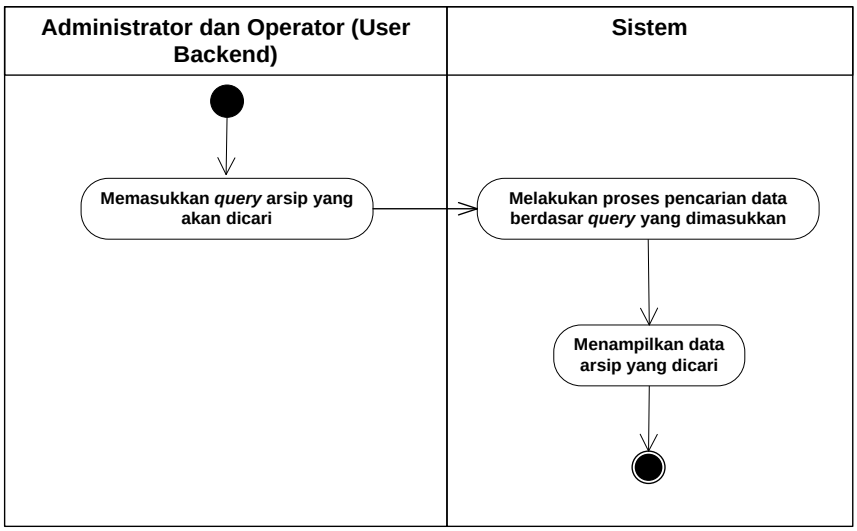
Gambar 3.12 *Activity diagram* Tampil Data Disposisi



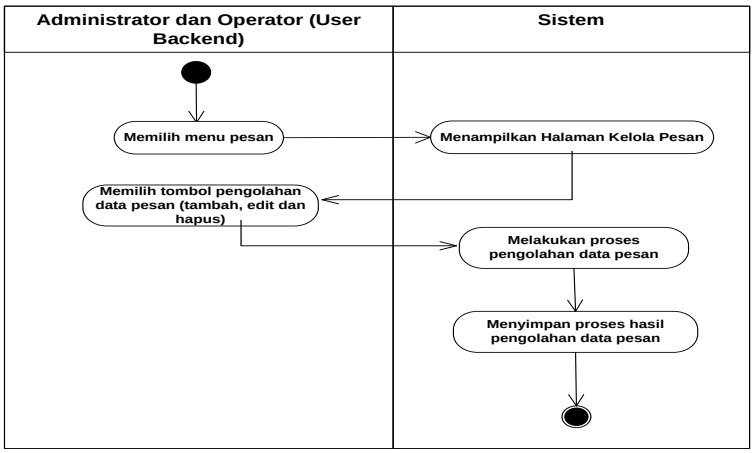
Gambar 3.13 *Activity diagram* Pengolahan Data Surat Keluar



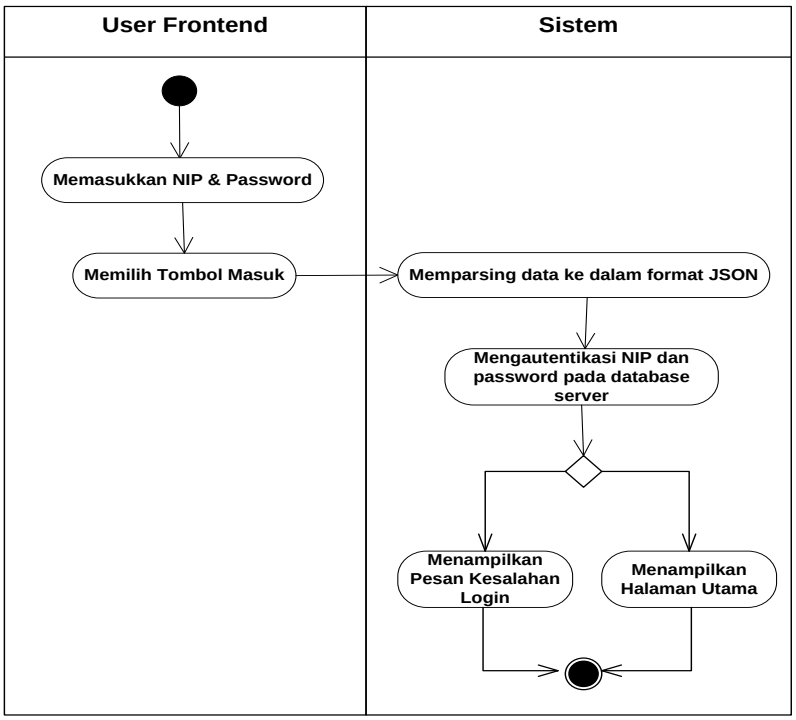
Gambar 3.14 *Activity diagram* Pengolahan Data Agenda



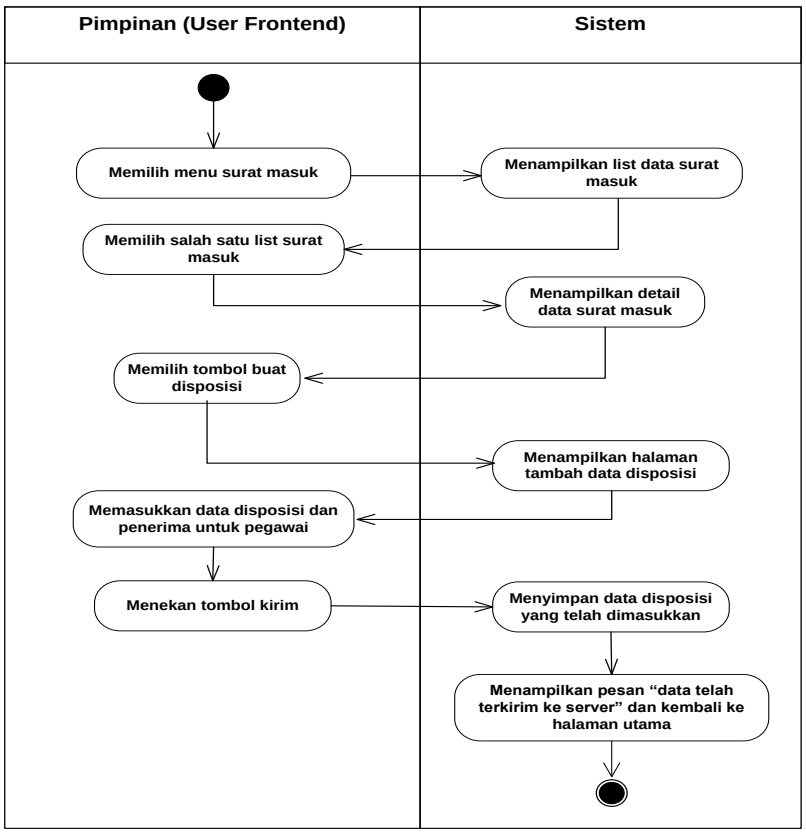
Gambar 3.15 *Activity diagram* Cari Arsip



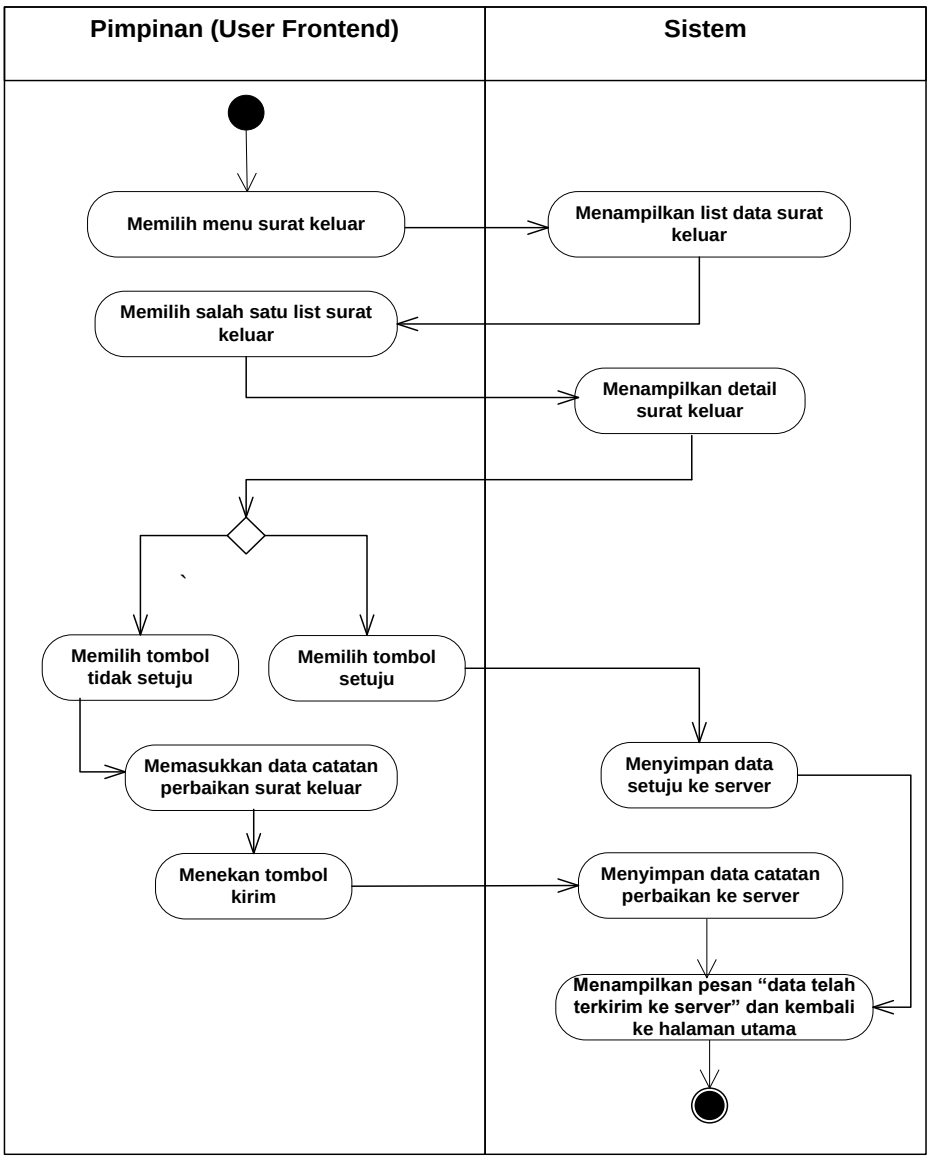
Gambar 3.16 *Activity diagram* Pengolahan Data Pesan



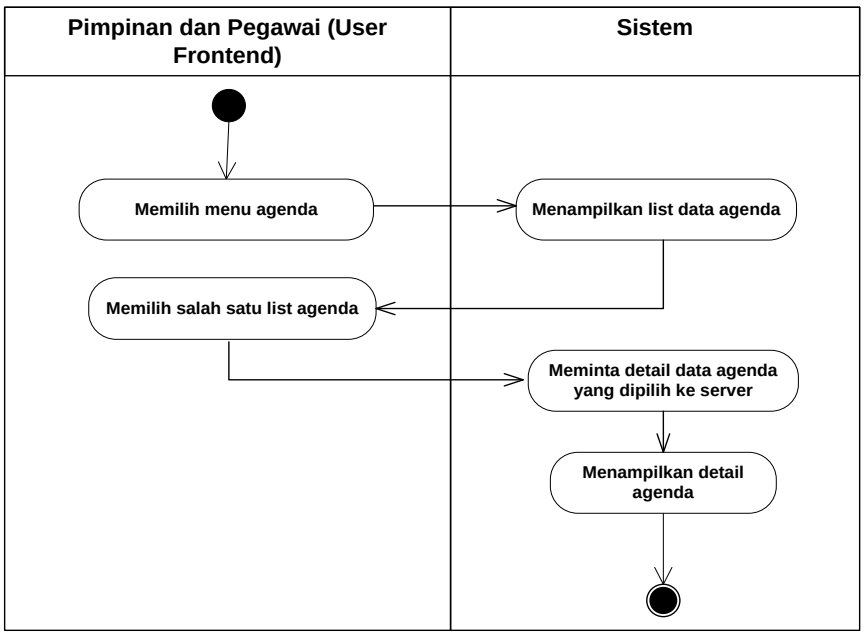
Gambar 3.17 ***Activity diagram*** **proses** ***login user frontend***



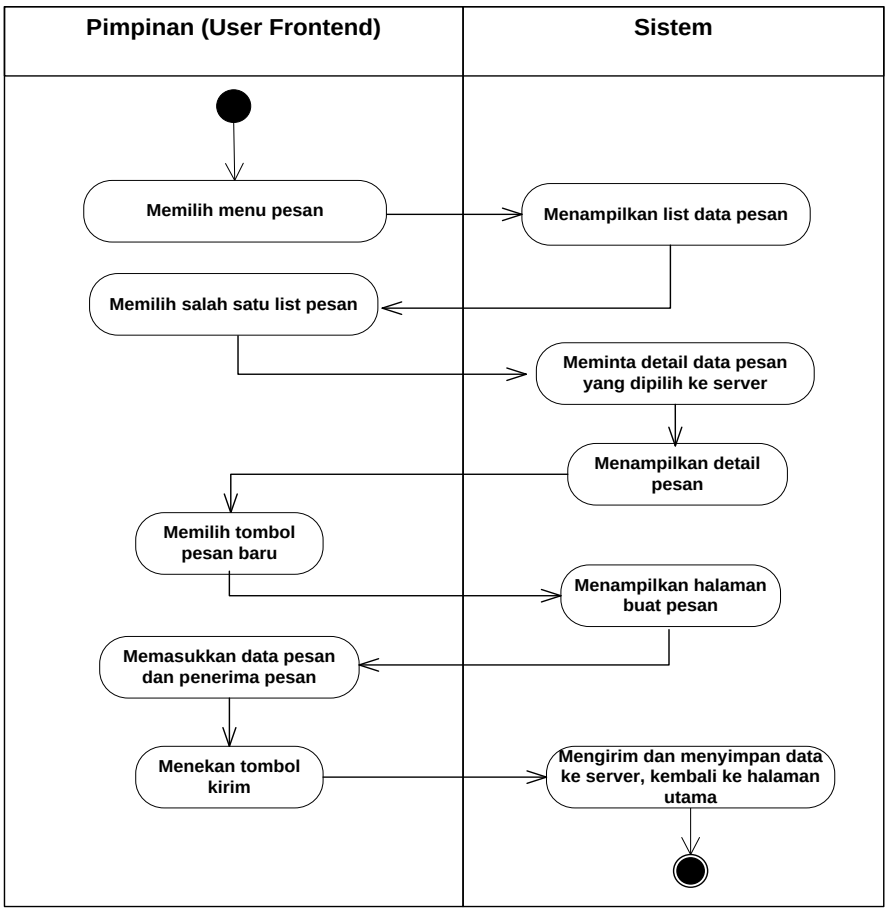
Gambar 3.18 ***Activity diagram*** **Tampil Data Surat Masuk**



Gambar 3.19 *Activity diagram* Tampil Data Surat Keluar



Gambar 3.20 *Activity diagram* Tampil Data Agenda

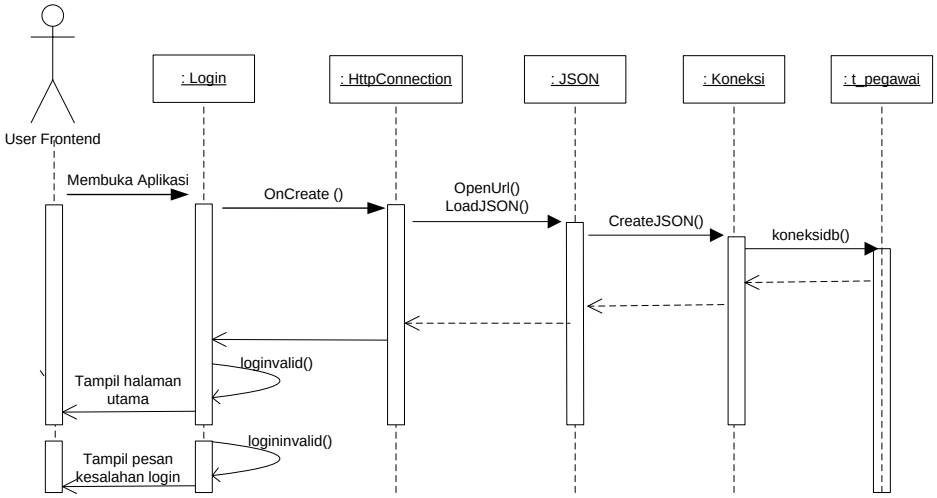


Gambar 3.21 *Activity diagram* Tampil Data Pesan

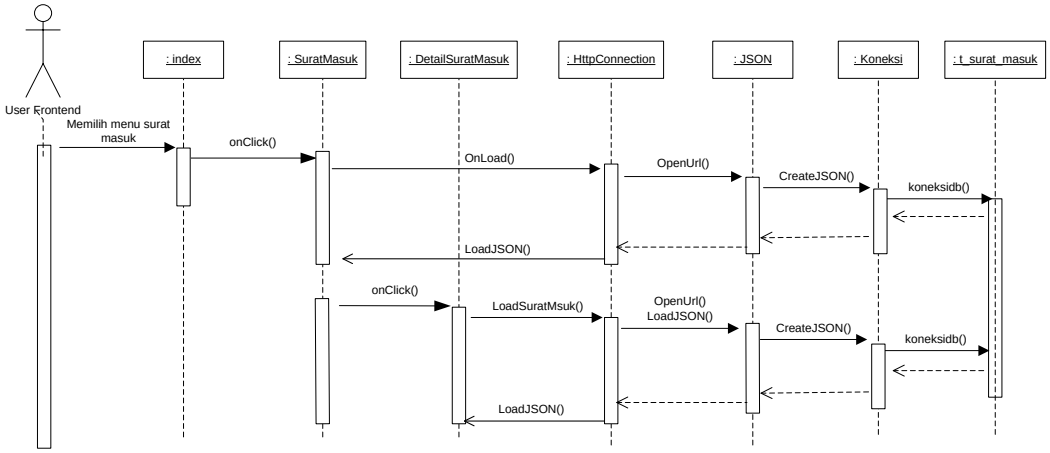
3.1.8.1.3 *Sequence Diagram*

Sequence diagram menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam use case. Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progres vertikal.

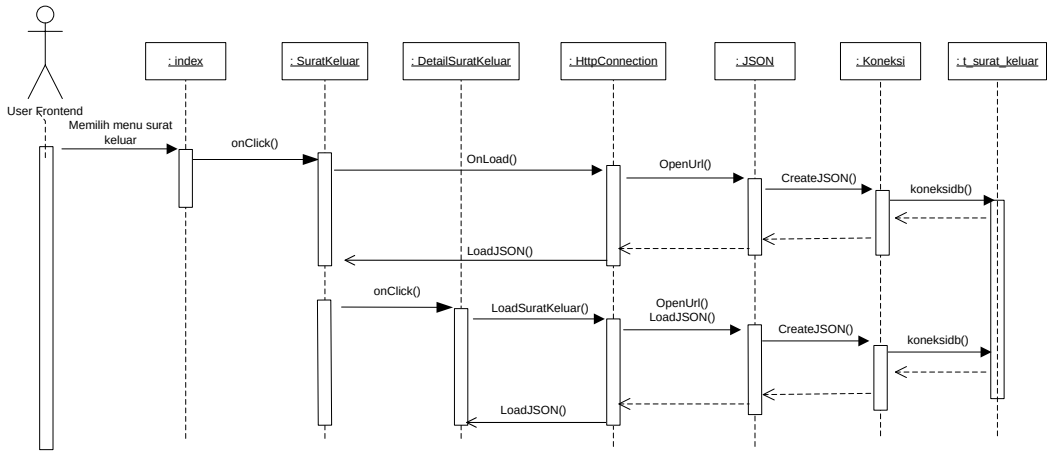
1. *Sequence Diagram Frontend Application*



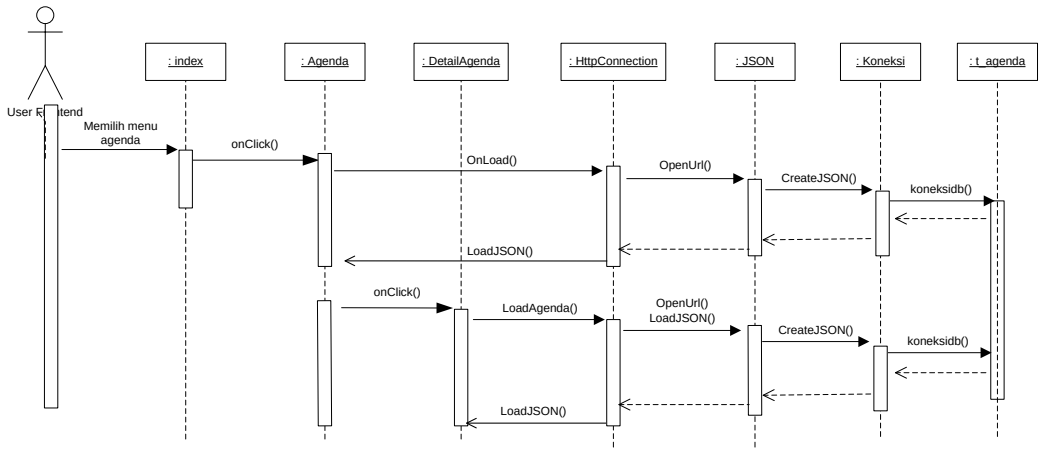
Gambar 3.22 *Sequence Diagram Login User Frontend*



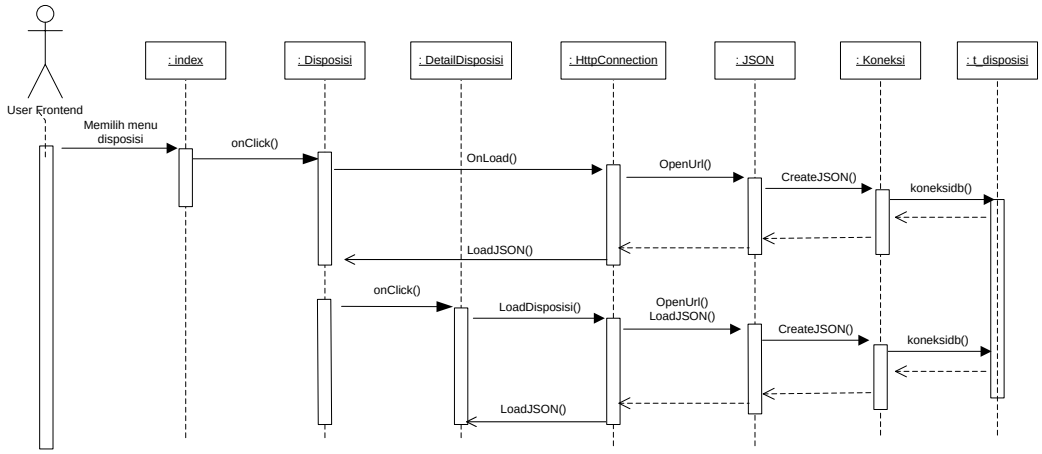
Gambar 3.23 *Sequence Diagram* Tampil Surat Masuk



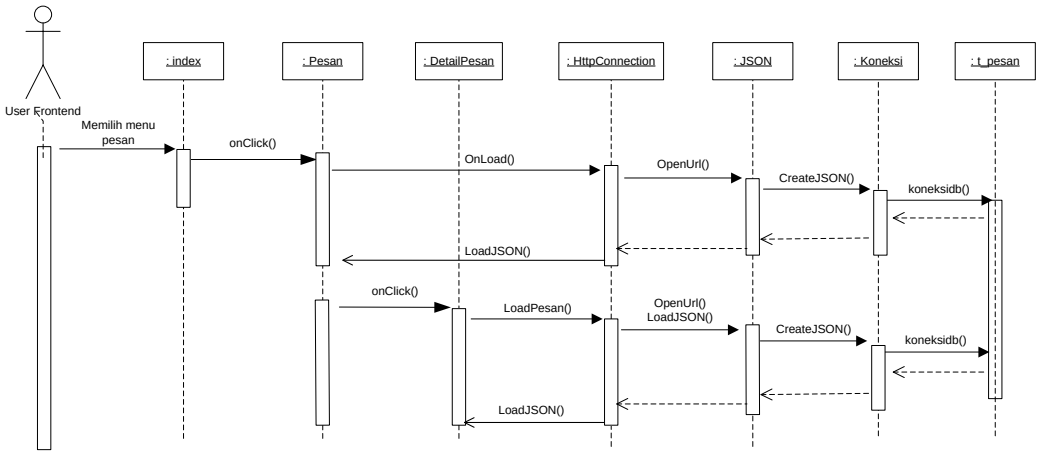
Gambar 3.24 *Sequence Diagram* Tampil Surat Keluar



Gambar 3.25 *Sequence Diagram* Tampil Agenda

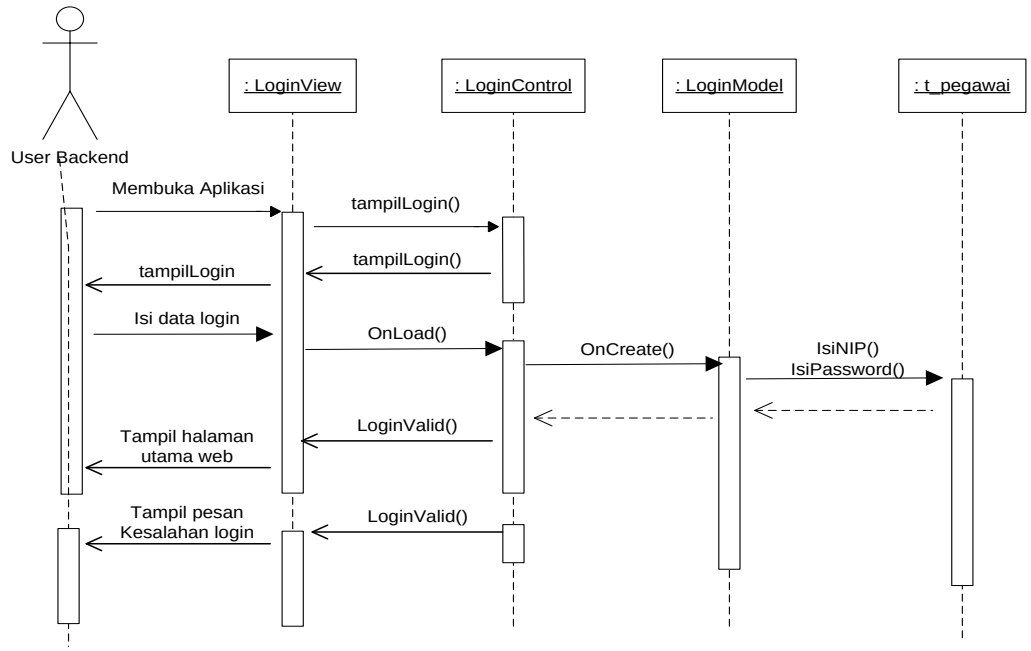


Gambar 3.26 *Sequence Diagram* Tampil Disposisi

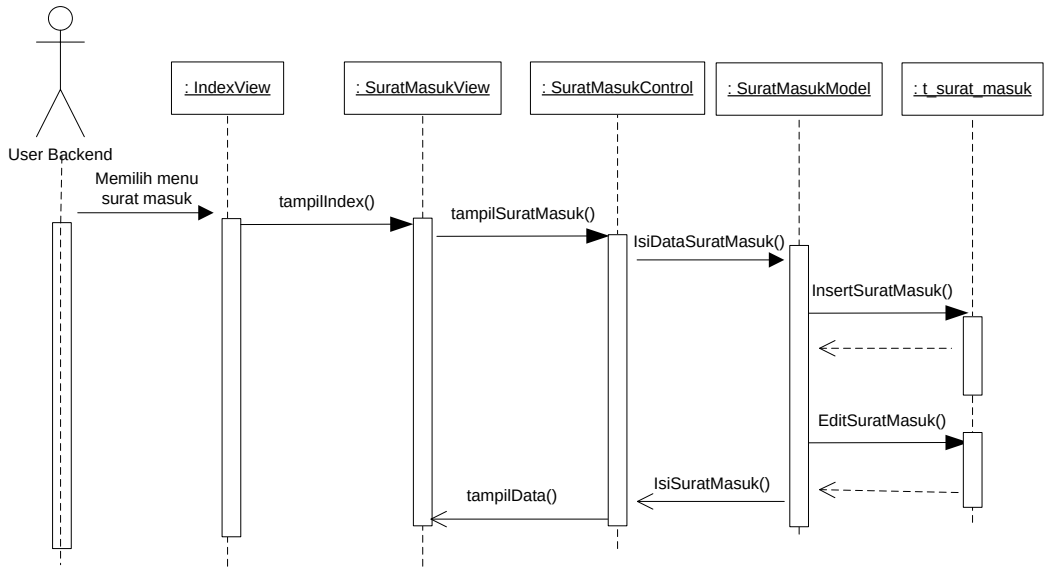


Gambar 3.27 *Sequence Diagram* Tampil Pesan

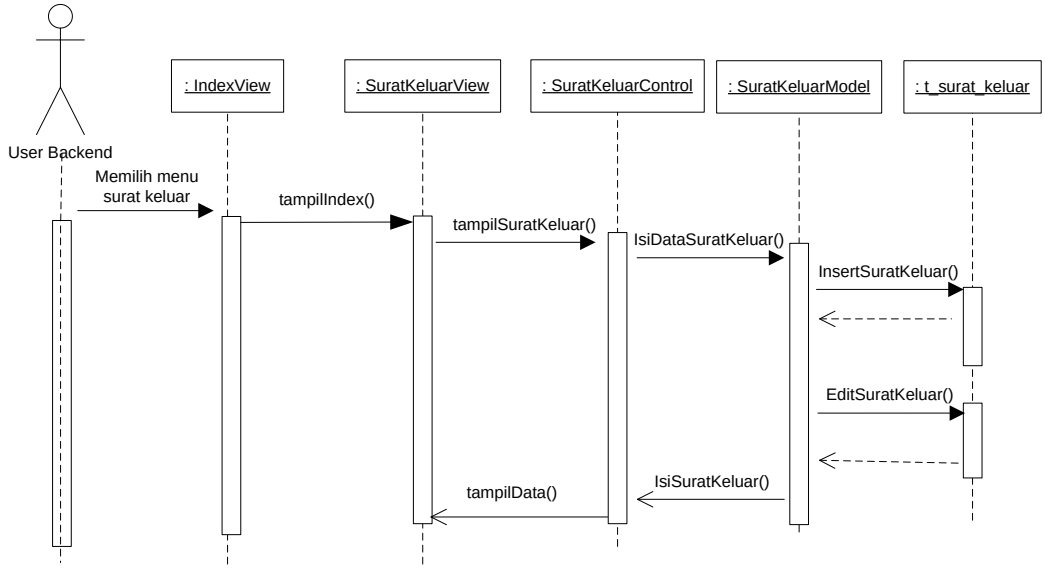
2. *Sequence Diagram Backend Application*



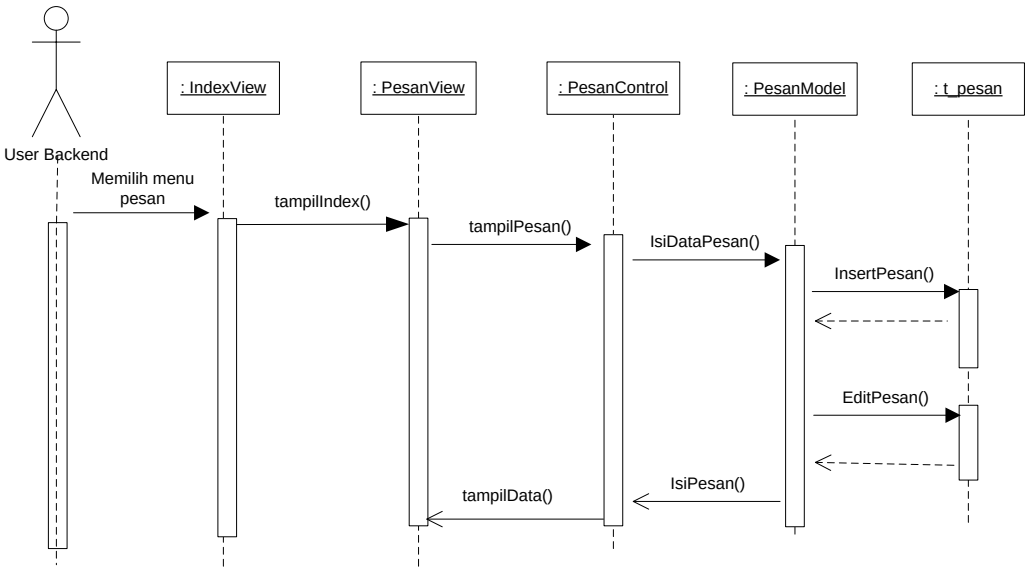
Gambar 3.28 *Sequence Diagram Login User Backend*



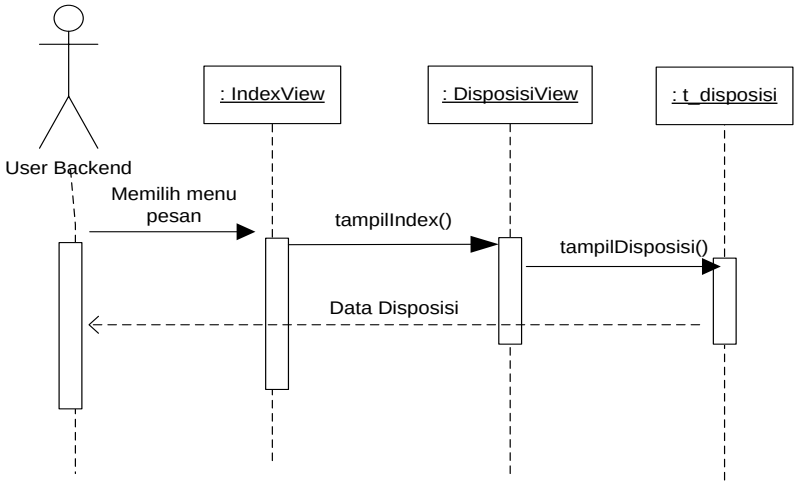
Gambar 3.29 *Sequence Diagram* Pengolahan Data Surat Masuk



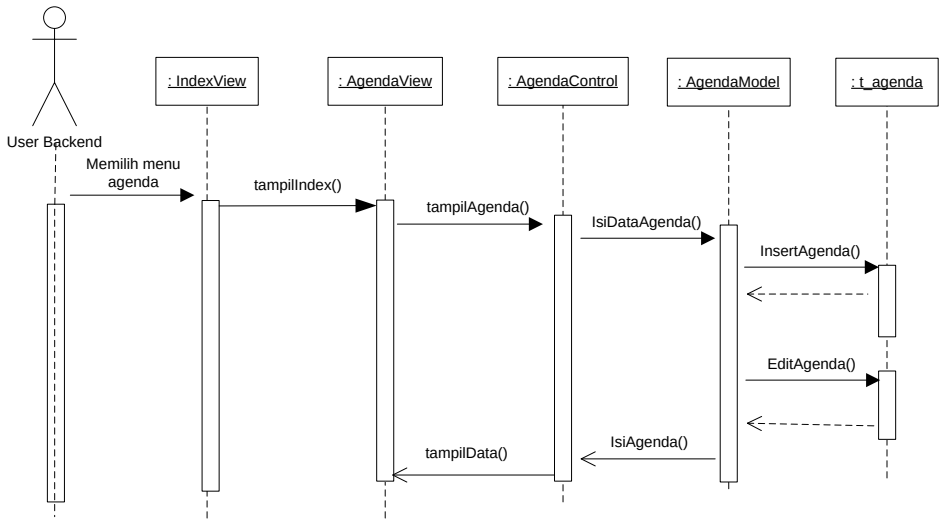
Gambar 3.30 *Sequence Diagram* Pengolahan Data Surat Keluar



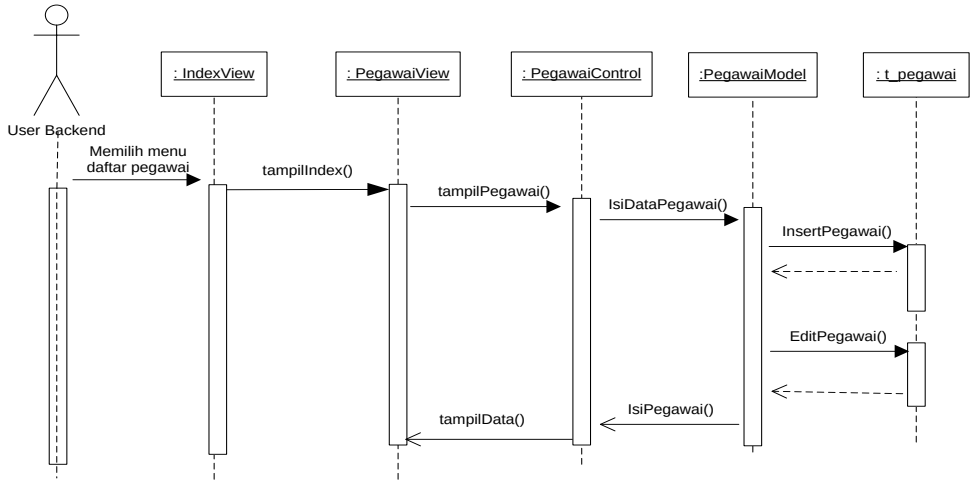
Gambar 3.31 *Sequence Diagram* Pengolahan Data Pesan



Gambar 3.32 *Sequence Diagram* Lihat Disposisi



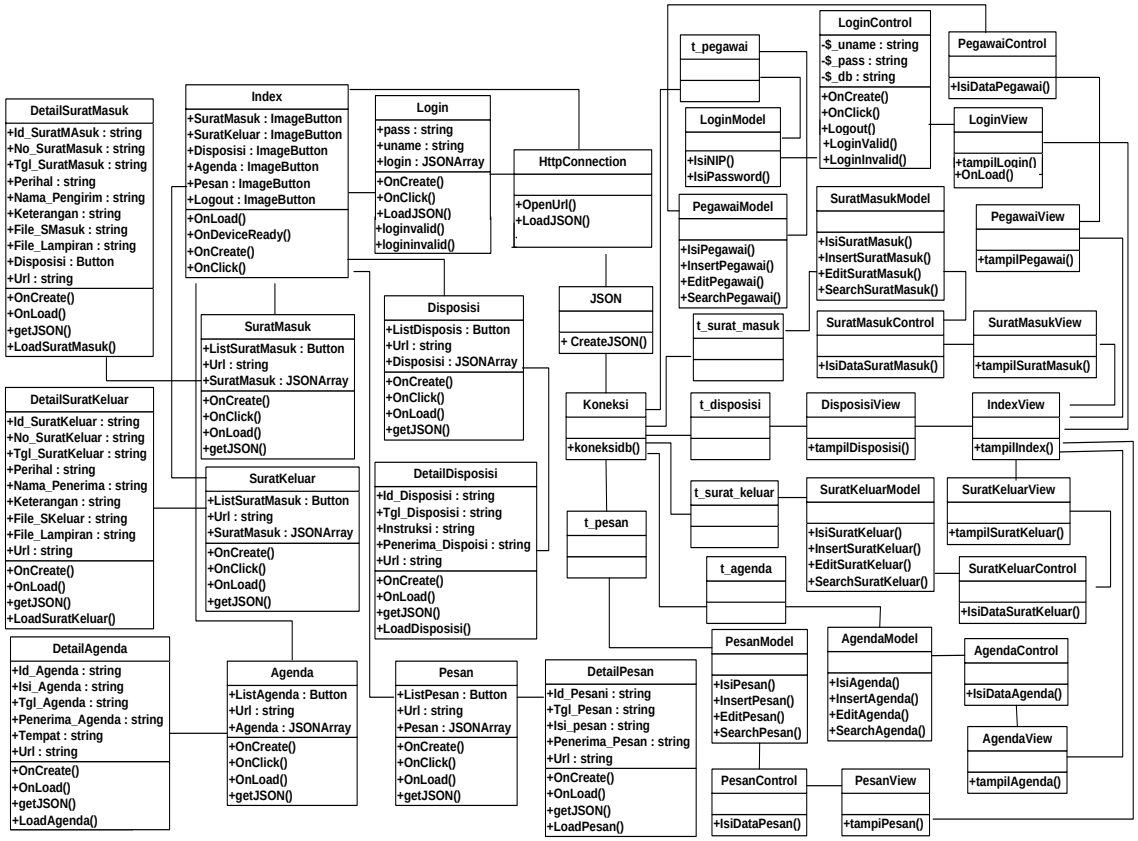
Gambar 3.33 *Sequence Diagram* Pengolahan Data Agenda



Gambar 3.34 *Sequence Diagram* Pengolahan Data Pegawai

3.1.8.1.4 *Class Diagram*

Class diagram mendeskripsikan jenis–jenis obyek dalam sistem dan berbagai macam hubungan statis yang terjadi. *Class diagram* juga menunjukkan *property* dan operasi sebuah *Class* dan batasan yang terdapat dalam hubungan dengan objek. *Class* menggambarkan keadaan (*attribut* atau *property*) suatu sistem sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi). *Class diagram* aplikasi *e-office* dapat dilihat pada gambar 3.35.



Gambar 3.35 *Class Diagram* Aplikasi *E-Office*

Tabel 3.15 Deskripsi *Class Diagram* Aplikasi *E-Office*

Backend Application		
Class	**Jenis *Class***	**Deskripsi**
SuratMasukModel	Model	*Class* yang berisikan data surat masuk
SuratKeluarModel	Model	*Class* yang berisikan data surat keluar
AgendaModel	Model	*Class* yang berisikan data agenda
PesanModel	Model	*Class* yang berisikan data pesan
PegawaiModel	Model	*Class* yang berisikan data pegawai
LoginView	*View*	*Class* yang berisikan tampilan *login web*
IndexView	*View*	*Class* yang berisikan tampilan halaman utama *web*
SuratMasuk*View*	*View*	*Class* yang berisikan tampilan data surat

		masuk
SuratKeluar*View*	*View*	*Class* yang berisikan tampilan data surat keluar
Agenda*View*	*View*	*Class* yang berisikan tampilan data agenda
Pesan*View*	*View*	*Class* yang berisikan tampilan data pesan
Pegawai*View*	*View*	*Class* yang berisikan tampilan data pegawai
Disposisi*View*	*View*	*Class* yang berisikan tampilan data disposisi
Login*Control*	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class interface login*
SuratMasuk*Control*	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class interface* surat masuk
SuratKeluar*Control*	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class interface* surat keluar
Agenda*Control*	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class interface* agenda
Pesan*Control*	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class interface* pesan
Pegawai*Control*	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class interface* pegawai
t_pegawai	*Entity*	Tabel *database* pegawai
t_surat_mauk	*Entity*	Tabel *database* surat masuk
t_surat_keluar	*Entity*	Tabel *database* surat keluar
t_agenda	*Entity*	Tabel *database* agenda
t_disposisi	*Entity*	Tabel *database* disposisi
t_pesan	*Entity*	Tabel *database* pesan
Frontend Application		
Class	**Jenis *Class***	**Deskripsi**
Index	*View*	*Class* yang berisikan *method-method* untuk fungsionalitas tampilan utama aplikasi

HTTPConnection	*Control*	*Class* yang berisikan *method-method* untuk menghubungkan aplikasi dengan *webservice*
JSON	*Control*	*Class* berisikan *method-method* untuk memparsing data berformat JSON
SuratMasuk	*Control*	*Class* yang berisikan *method-method* untuk menampilkan list surat masuk
DetailSuratMasuk	*Control*	*Class* yang berisikan *method-method* untuk menampilkan rincian data surat masuk
SuratKeluar	*Control*	*Class* yang berisikan *method-method* untuk menampilkan list surat keluar
DetailSuratKeluar	*Control*	*Class* yang berisikan *method-method* untuk menampilkan rincian data surat keluar
Agenda	*Control*	*Class* yang berisikan *method-method* untuk menampilkan list agenda
DetailAgenda	*Control*	*Class* yang berisikan *method-method* untuk menampilkan rincian data agenda
Disposisi	*Control*	*Class* yang berisikan *method-method* untuk menampilkan list disposisi
DetailDisposisi	*Control*	*Class* yang berisikan *method-method* untuk menampilkan rincian data disposisi
Pesan	*Control*	*Class* yang berisikan *method-method* untuk menampilkan list pesan
DetailPesan	*Control*	*Class* yang berisikan *method-method* untuk menampilkan rincian data pesan

3.2 Perancangan Sistem

Perancangan sistem merupakan metodologi pengembangan suatu perangkat lunak Perubahan data kategori sesuai dengan data yang dihapus.yang dilakukan setelah melalui tahapan analisis.

3.2.1 Perancangan Fitur Aplikasi *E-Office*

Perancangan fitur-fitur pada apikasi dimulai dari penentuan pengguna. Pengguna pada aplikasi akan dibagi manjadi beberapa level yang membedakan hak akses masing-masing pengguna dalam aplikasi *e-office.* Level-level pada aplikasi *e-office* adalah : level 1 (administrator), level 2 (operator), level 3 (pimpinan) dan level 4 (pegawai). Berdasarkan perancangan level ini maka akan dibentuk fitur-fitur yang diimplementasikan kedalam aplikasi *e-office.* Fitur-fitur pengguna berdasarkan levelnya dapat dilihat pada tabel 3.16.

Tabel 3.16 Fitur Menu Aplikasi *E-Office*

Fitur	Level Pengguna			
	Level 1	Level 2	Level 3	Level 4
Mengolah User	√	-	-	-
Lihat User	√	-	-	-
Setup Surat Masuk	√	√	-	-
Lihat Surat Masuk	√	√	√	-
Setup Surat Keluar	√	√	-	-
Lihat Surat Keluar	√	√	√	-
Setup Agenda	√	√	-	-
Lihat Agenda	√	√	√	√
Isi Disposisi	-	-	√	-
Lihat Disposisi	√	√	√	√
Setup Pesan	√	√	√	√
Lihat Pesan	√	√	√	√
Isi Catatan Koreksi	-	-	√	-

Keterangan :

Level 1 = Administrator

Level 2 = Operator

Level 3 = Pimpinan

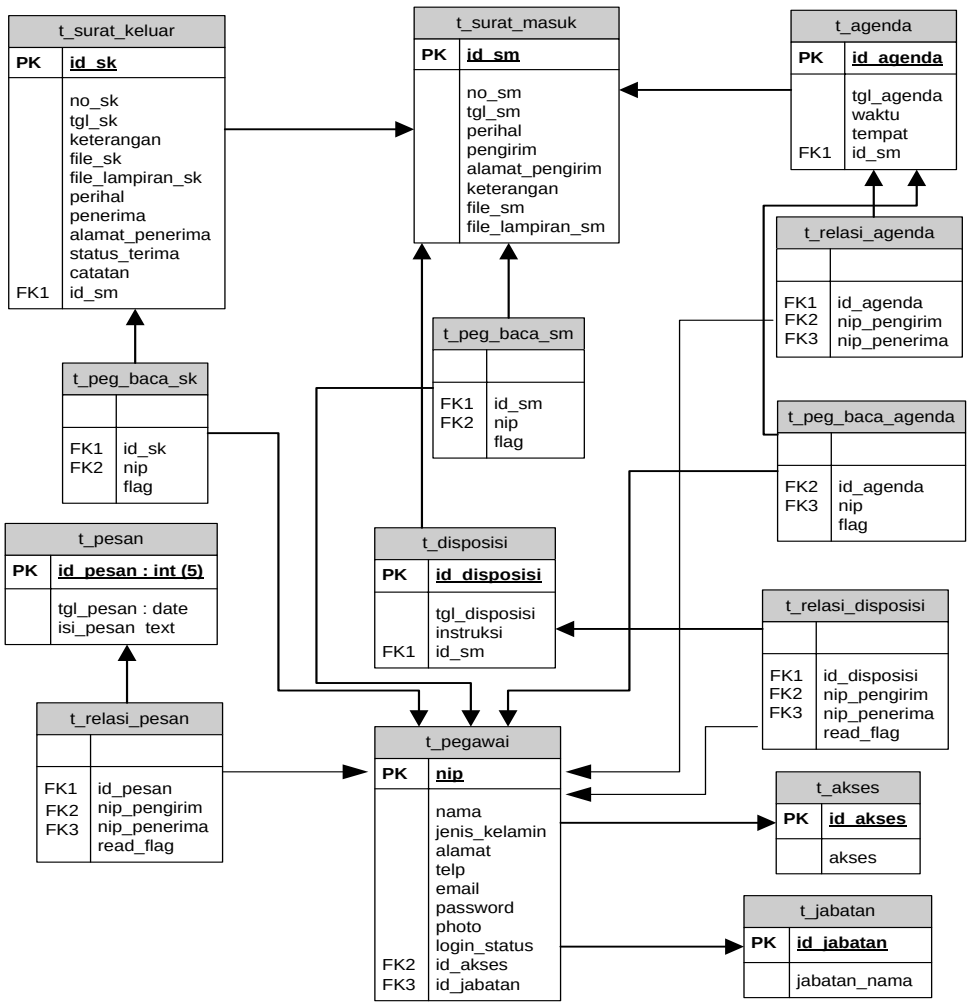
Level 4 = Pegawai

3.2.2 Perancangan Basis Data

Perancangan data yaitu menciptakan atau merancang data yang terhubung dan disimpan secara bersama-sama. Untuk menggambarkannya digunakanlah skema relasi dan struktur tabel. Dari dua hasil tersebut, implementasi basis data akan bisa dikerjakan.

1. Skema Relasi

Proses relasi antar file merupakan gabungan antar file yang mempunyai kunci utama yang sama, sehingga file-file tersebut menjadi satu kesatuan yang dihubungkan oleh *filed* kunci tersebut. Pada proses ini elemen-elemen data dikelompokan menjadi satu file database beserta entitas dan hubungannya. Skema relasi aplikasi *e-office* dapat dilihat pada gambar 3.36.



Gambar 3.36 Skema Relasi Aplikasi *E-Office*

2. Struktur Tabel

Tabel adalah kumpulan data atau informasi spesifik tentang subjek tertentu yang disusun dalam bentuk kolom dan baris. Tabel adalah komponen utama dan pertama dari sebuah *database*. Struktur tabel untuk setiap tabel yang terdapat dalam database aplikasi ini adalah sebagai berikut.

Tabel 3.176 Tabel Surat Keluar (t_surat_keluar)

Nama kolom	Tipe	Ukuran	Keterangan
id_sk	Int	11	*Primary key*
id_sm	Int	11	*Foreign Key reference* t_surat_masuk (id_sm)
no_sk	varchar	19	Nomor surat keluar
tgl_sk	Date		Tanggal surat keluar
keterangan	varchar	256	
file_sk	varchar	256	File surat keluar
file_lampiran_sk	varchar	256	File lampiran surat keluar
Perihal	varchar	50	
penerima	varchar	50	
alamat_penerima	varchar	256	
status_terima	Char	1	
Catatan	Text		

Tabel 3.18 Tabel Surat Masuk (t_surat_masuk)

Nama kolom	Tipe	Ukuran	Keterangan
id_sm	Int	11	*Primary key*
no_sm	Char	15	Nomor surat masuk
tgl_sm	Date		Tanggal surat masuk
Keterangan	varchar	256	
file_sm	varchar	256	File surat masuk

file_lampiran_sm	varchar	256	File lampiran surat masuk
Perihal	varchar	50	
Pengirim	varchar	50	
alamat_pengirim	varchar	256	

Tabel 3.19 Tabel Pegawai Baca Surat Keluar (t_peg_baca_sk)

Nama kolom	Tipe	Ukuran	Keterangan
id_sk	Int	11	*Foreign Key reference* t_surat_keluar (id_sk)
Nip	varchar	11	*Foreign Key reference* t_pegawai (nip)
Flag	Int	1	

Tabel 3.20 Tabel Pegawai Baca Surat Masuk (t_peg_baca_sm)

Nama kolom	Tipe	Ukuran	Keterangan
id_sm	Int	11	*Foreign Key reference* t_surat_masuk (id_sm)
Nip	varchar	11	*Foreign Key reference* t_pegawai (nip)
Flag	Int	1	

Tabel 3.21 Tabel Agenda (t_agenda)

Nama kolom	Tipe	Ukuran	Keterangan
id_agenda	Int	10	*Primary key*
id_sm	Int	11	*Foreign Key reference* t_surat_masuk (id_sm)
tgl_agenda	Date		Tanggal agenda
Waktu	Char	5	

Tempat	varchar	256	

Tabel 3.22 Tabel Relasi Agenda (t_relasi_agenda)

Nama kolom	Tipe	Ukuran	Keterangan
Id_agenda	Int	10	*Foreign Key reference* t_agenda (id_agenda)
nip_pengirim	varchar	11	*Foreign Key reference* t_pegawai (nip)
nip_penerima	varchar	11	*Foreign Key reference* t_pegawai (nip)

Tabel 3.23 Tabel Pegawai Baca Agenda (t_peg_baca_agenda)

Nama kolom	Tipe	Ukuran	Keterangan
id_agenda	Int	10	*Foreign Key reference* t_agenda (id_agenda)
Nip	varchar	11	*Foreign Key reference* t_pegawai (nip)
Flag	Int	1	

Tabel 3.24 Tabel Pesan (t_pesan)

Nama kolom	Tipe	Ukuran	Keterangan
id_pesan	Int	10	*Primery key*
tgl_pesan	Date		
isi_pesan	Text		

Tabel 3.25 Tabel Relasi Pesan (t_relasi_pesan)

Nama kolom	Tipe	Ukuran	Keterangan
Id_pesan	Int	10	*Foreign Key reference* t_pesan (id_pesan)

nip_pengirim	varchar	11	*Foreign Key reference* t_pegawai (nip)
nip_penerima	varchar	11	*Foreign Key reference* t_pegawai (nip)
read_flag	Int	1	

Tabel 3.26 Tabel Disposisi (t_disposisi)

Nama kolom	Tipe	Ukuran	Keterangan
id_disposisi	Int	10	*Primery key*
Id_sm	Int	11	*Foreign Key reference* t_surat_masuk (id_sm)
tgl_disposisi	Date		Tanggal disposisi
Instruksi	Varchar	256	

Tabel 3.27 Tabel Relasi Disposisi (t_relasi_disposisi)

Nama kolom	Tipe	Ukuran	Keterangan
Id_disposisi	Int	10	*Foreign Key reference* t_disposisi (id_disposisi)
nip_pengirim	varchar	11	*Foreign Key reference* t_pegawai (nip)
nip_penerima	varchar	11	*Foreign Key reference* t_pegawai (nip)
read_flag	Int	1	

Tabel 3.28 Tabel Pegawai (t_pegawai)

Nama kolom	Tipe	Ukuran	Keterangan
nip	Varchar	11	*Primery key*
id_akses	Int	3	*Foreign Key reference*

			t_akses (id_akses)
id_jabatan	Int	2	*Foreign Key reference* t_jabatan (id_jabatan)
nama	Varchar	50	
jenis_kelamin	Char	1	
alamat	Varchar	256	
telp	Int	200	
email	varchar	256	
password	varchar	10	
photo	varchar	256	
login_status	Char	1	

Tabel 3.29 Tabel Akses (t_akses)

Nama kolom	Tipe	Ukuran	Keterangan
Id_akses	Int	3	*Primery key*
akses	Varchar	20	

Tabel 3.309 Tabel Jabatan (t_jabatan)

Nama kolom	Tipe	Ukuran	Keterangan
Id_jabatan	Int	2	*Primery key*
Jabatan_nama	varchar	60	

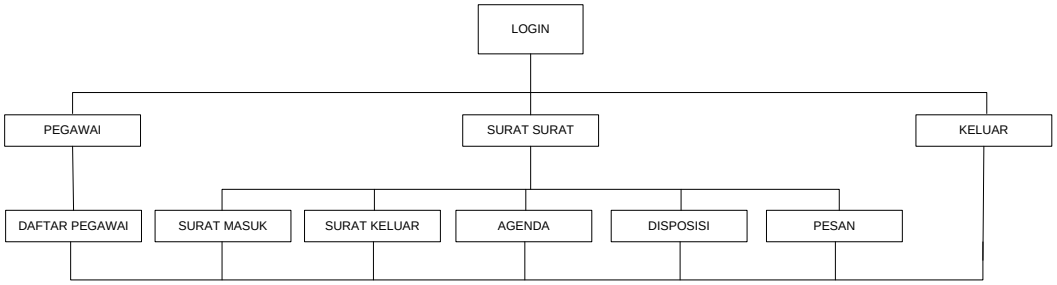
3.2.3 Perancangan Struktur Menu

Perancangan struktur menu merupakan jalur pemakai (*konsumen interface*) yang mudah dipahami dan mudah digunakan. Struktur menu dari aplikasi *e-office* dapat dilihat pada gambar 3.37 sampai 3.40.

1. Struktur menu Aplikasi *Backend*

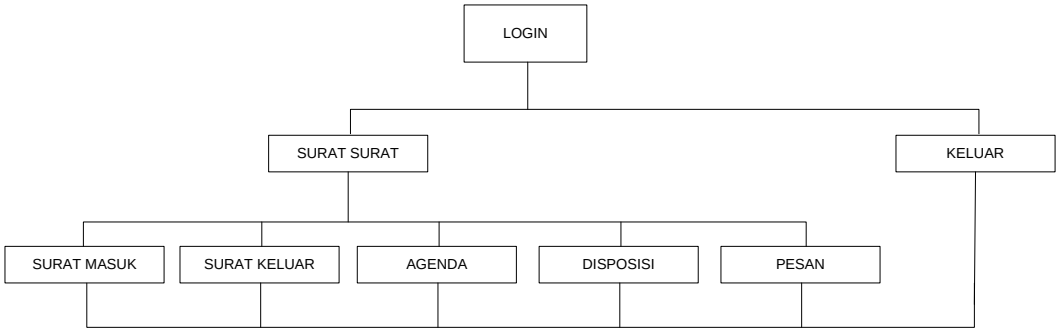
   Struktur menu aplikasi *backend* terdiri dari dua, yaitu :

a. Struktur menu untuk hak akses Administrator



Gambar 3.37 Struktur Menu Administrator

b. Struktur menu untuk hak akses Operator

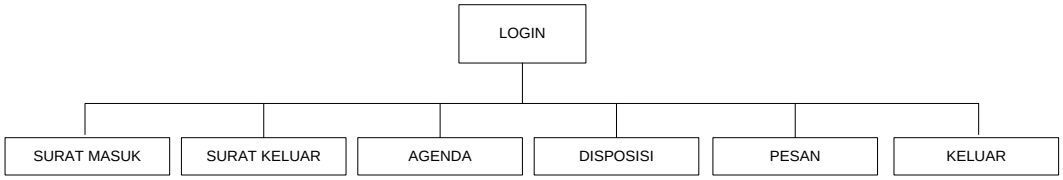


Gambar 3.38 Struktur Menu Operator

2. Struktur menu Aplikasi *Frontend*

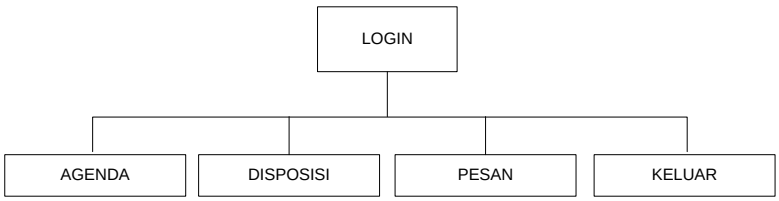
   Struktur menu aplikasi *frontend* terdiri dari dua, yaitu :

   a. Struktur menu untuk hak akses Pimpinan



Gambar 3.39 Struktur Menu Pimpinan

b. Struktur menu untuk hak akses Pegawai

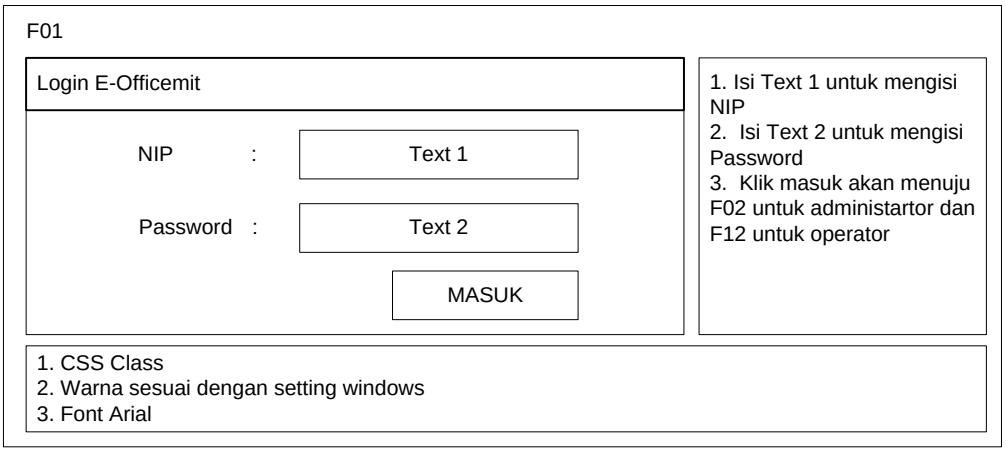


Gambar 3.40 Struktur Menu Pegawai

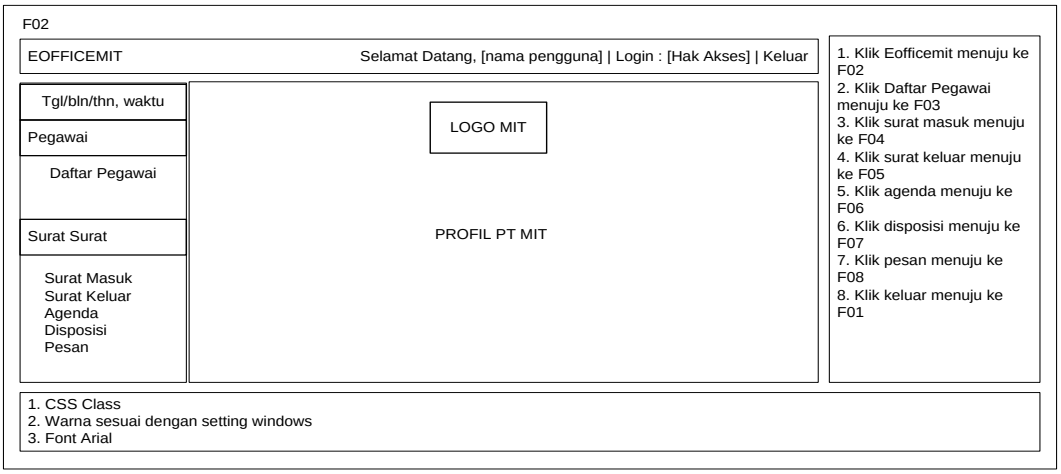
3.2.4 Perancangan Antar Muka

Perancangan antar muka ini terdiri dari dua bagian, yaitu perancangan antar muka *user* yang menggunakan PC (*Backend*) yang terdiri dari antarmuka administrator dan operator sedangkan perancangan antar muka *user* yang menggunakan *handphone* (*frontend*) terdiri dari antarmuka pimpinan dan pegawai.

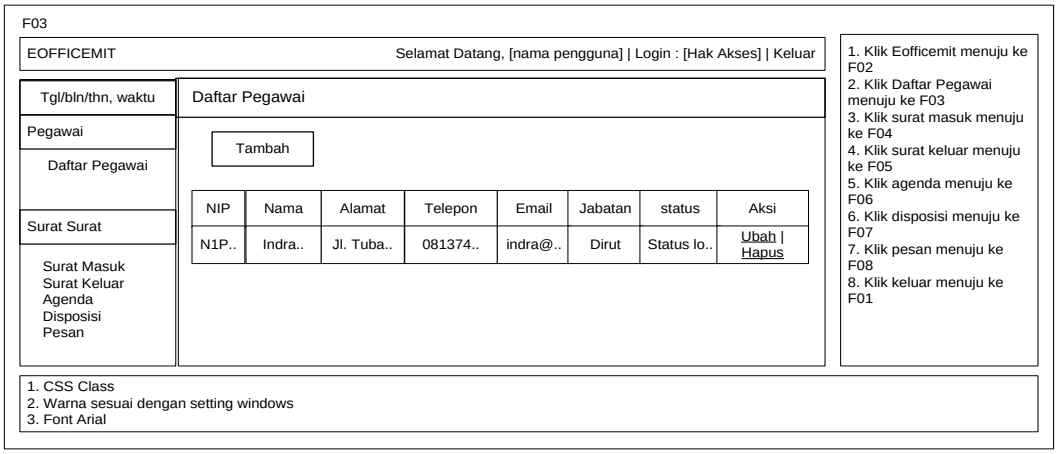
1. Perancangan Antarmuka *Backend*



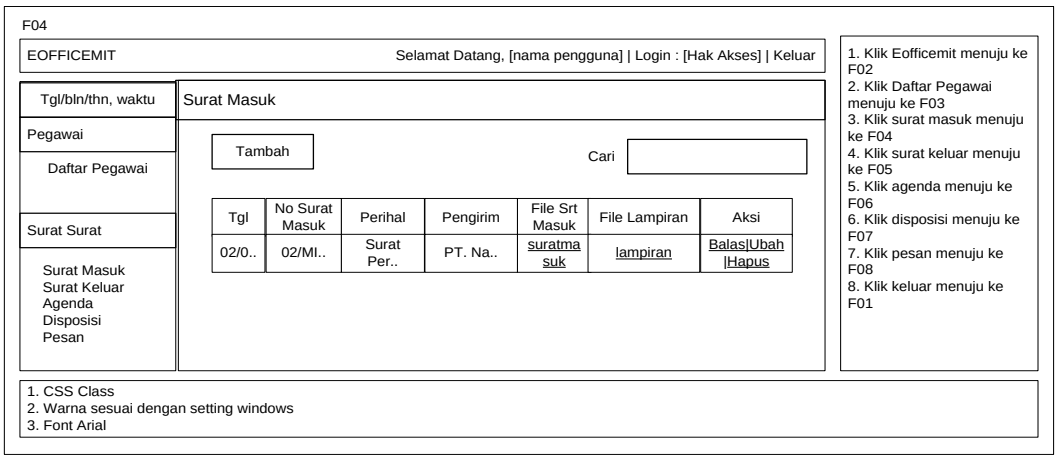
Gambar 3.37 *Form Login*



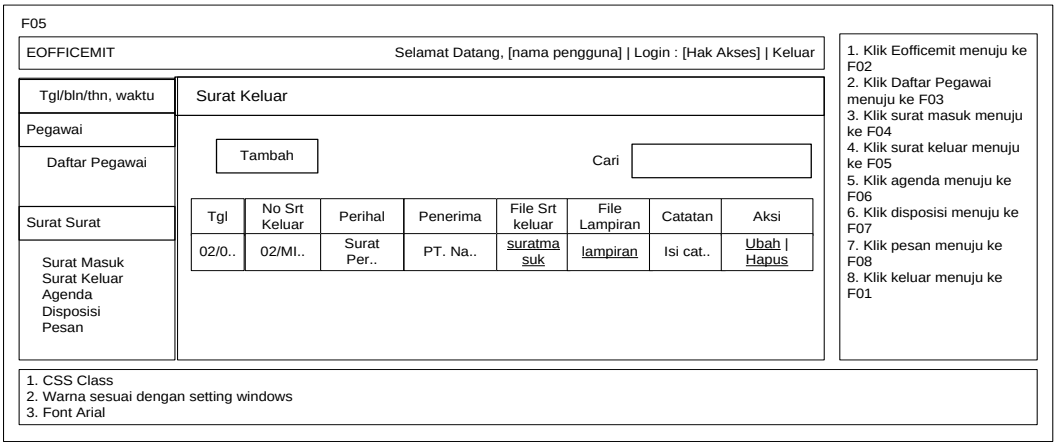
Gambar 3.38 *Form* Halaman Utama Administrator



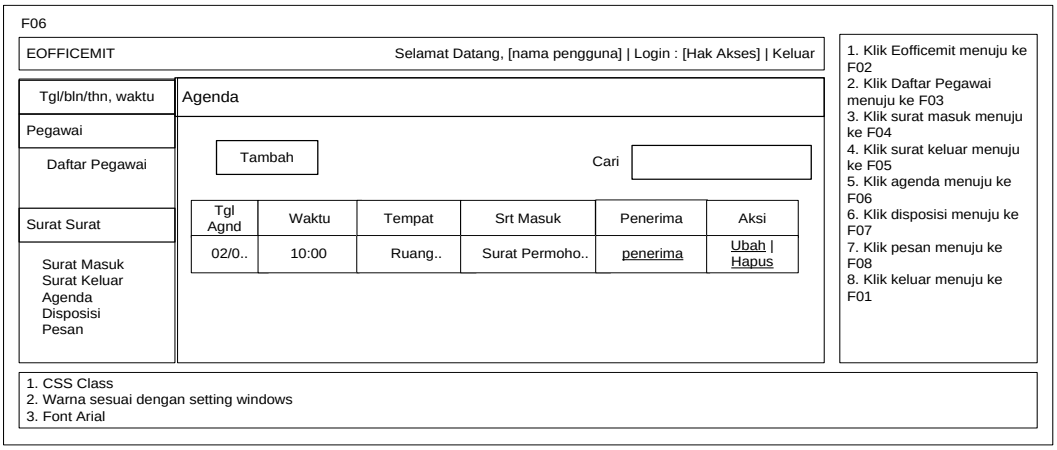
Gambar 3.39 *Form* Daftar Pegawai



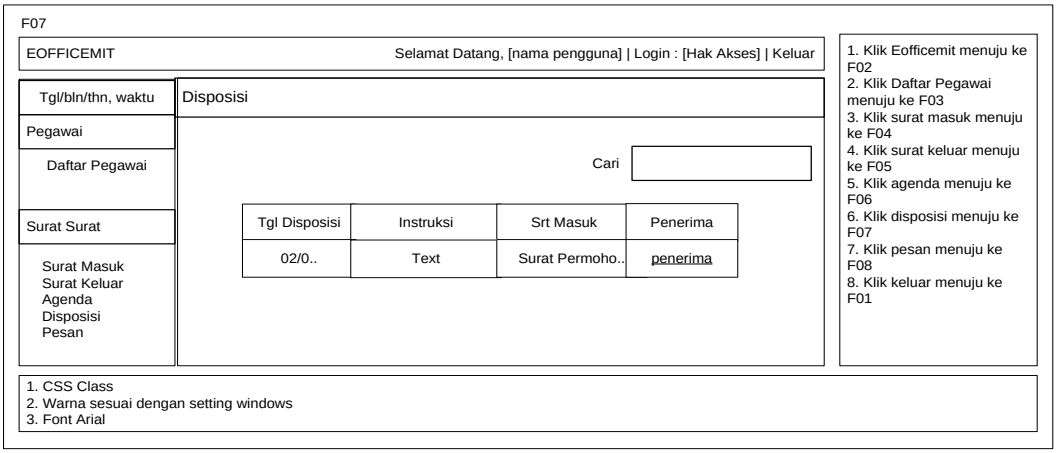
Gambar 3.40 *Form* Surat Masuk



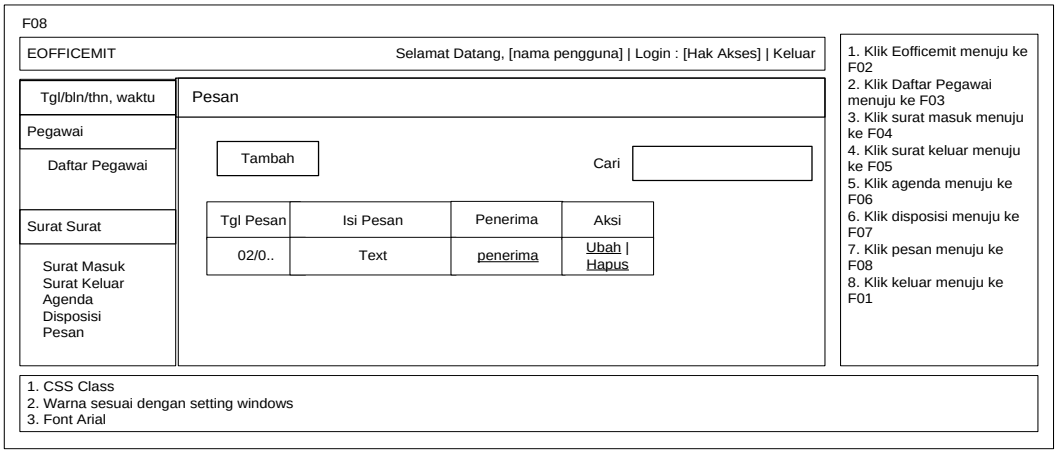
Gambar 3.41 *Form* Surat Keluar



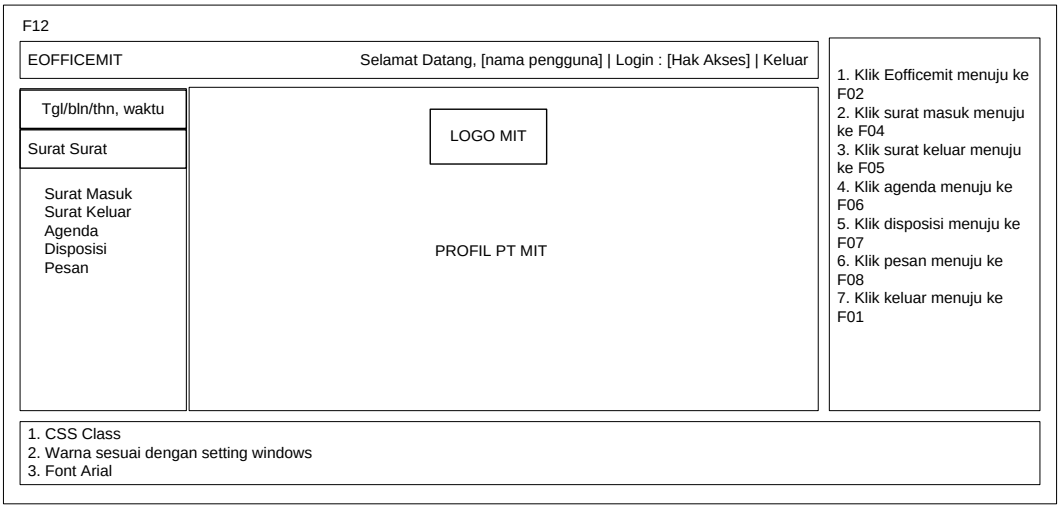
Gambar 3.42 *Form* Agenda



Gambar 3.43 *Form* Disposisi

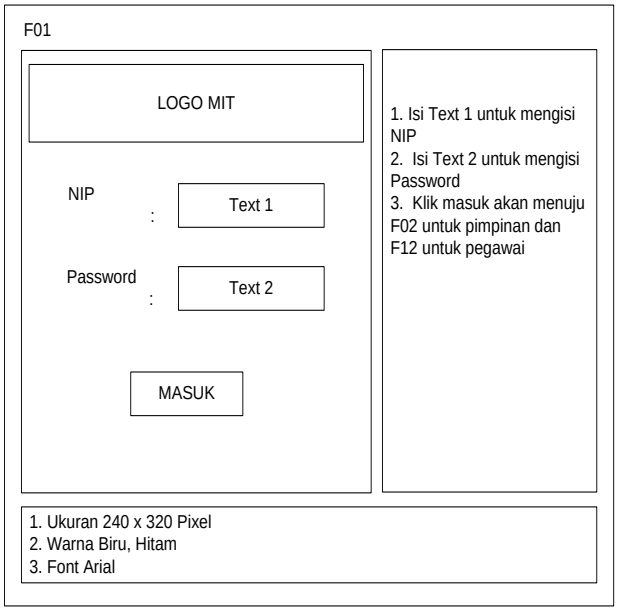


Gambar 3.44 *Form* Pesan

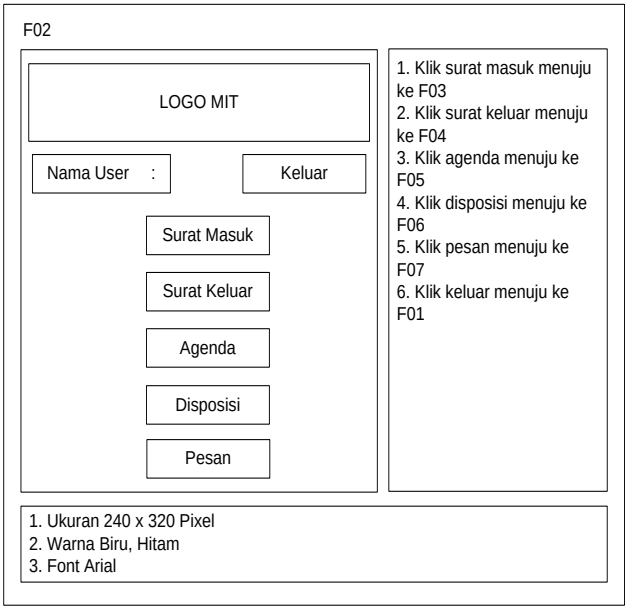


Gambar 3.45 *Form* Halaman Utama Operator

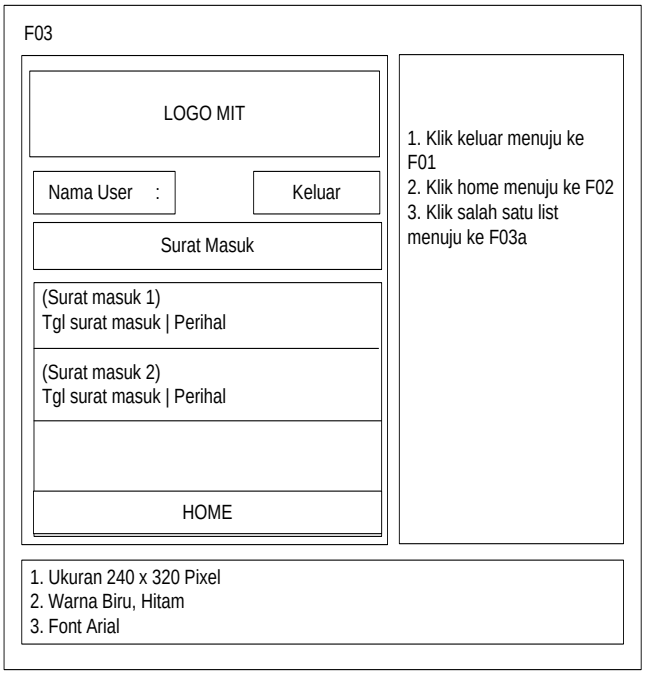
2. Perancangan Antar Muka pada *Frontend*



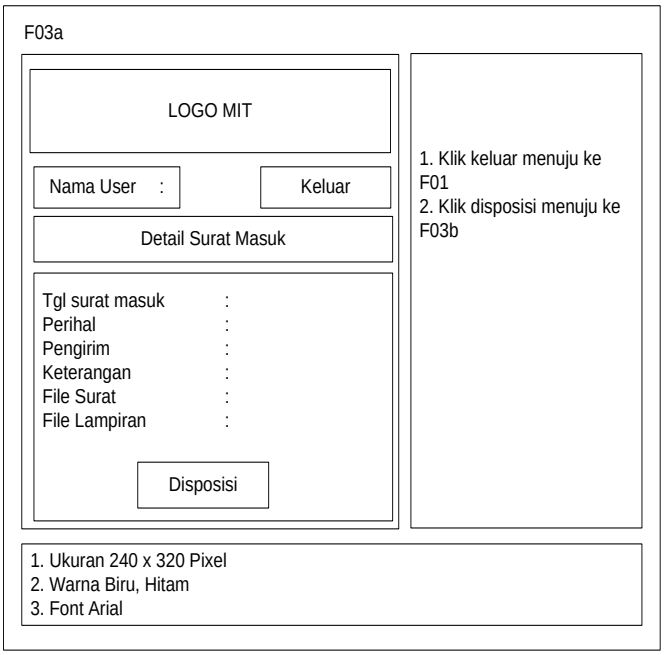
Gambar 3.46 *Form Login*



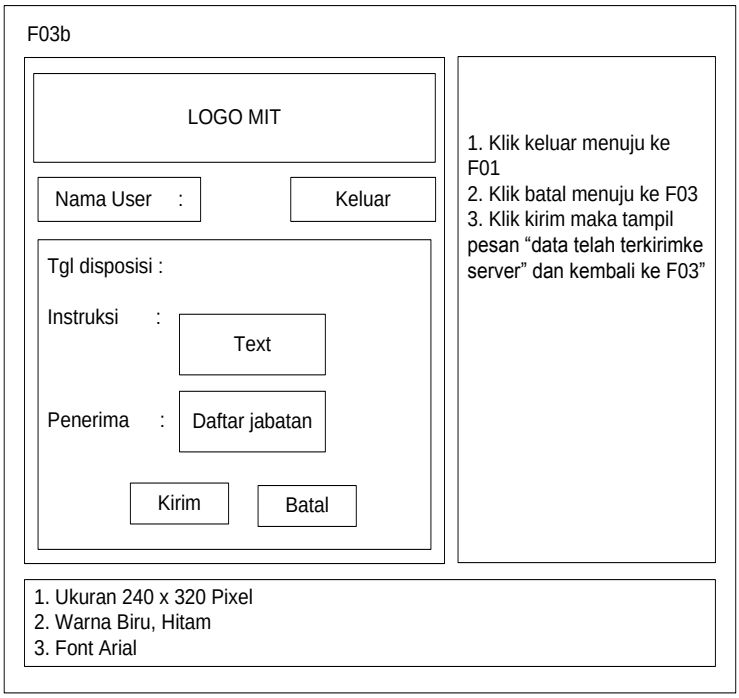
Gambar 3.47 *Form* Menu Utama Pimpinan



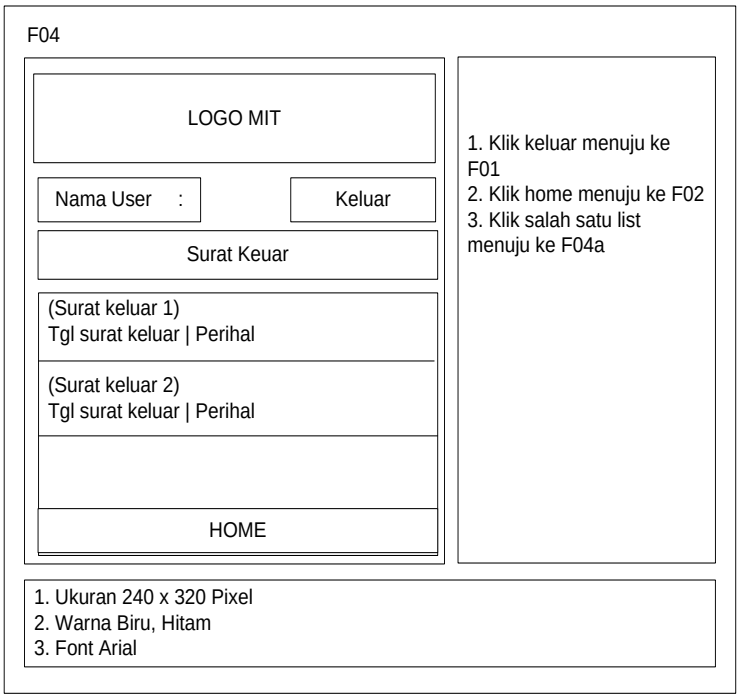
Gambar 3.48 *Form* Surat Masuk



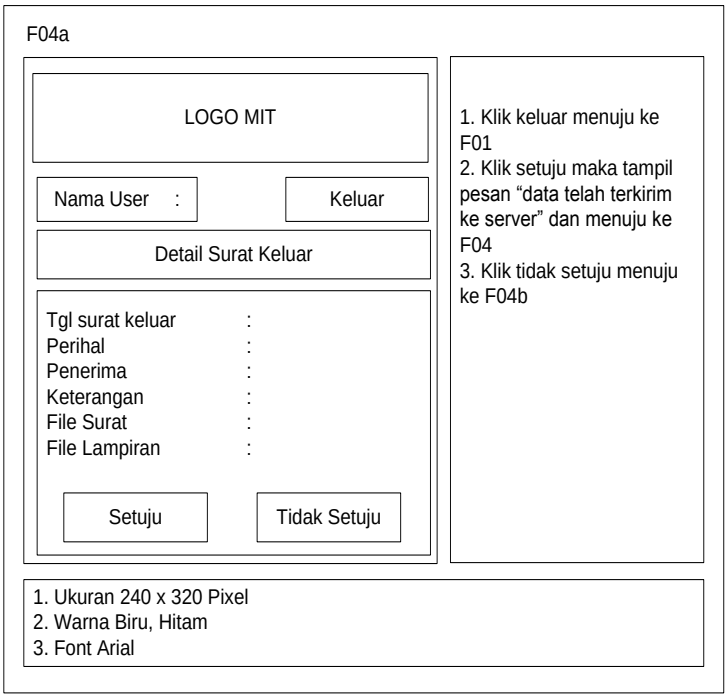
Gambar 3.49 *Form* Detail Surat Masuk



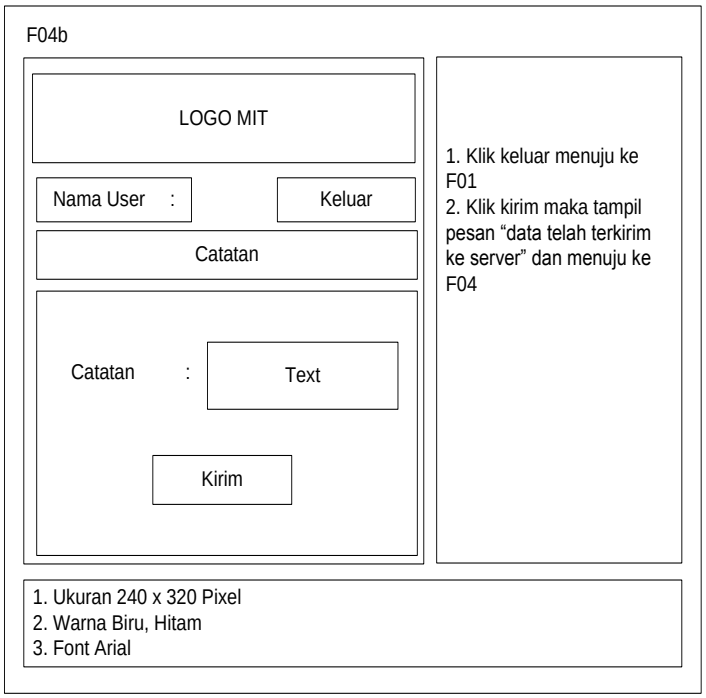
Gambar 3.50 *Form* Disposisi Baru



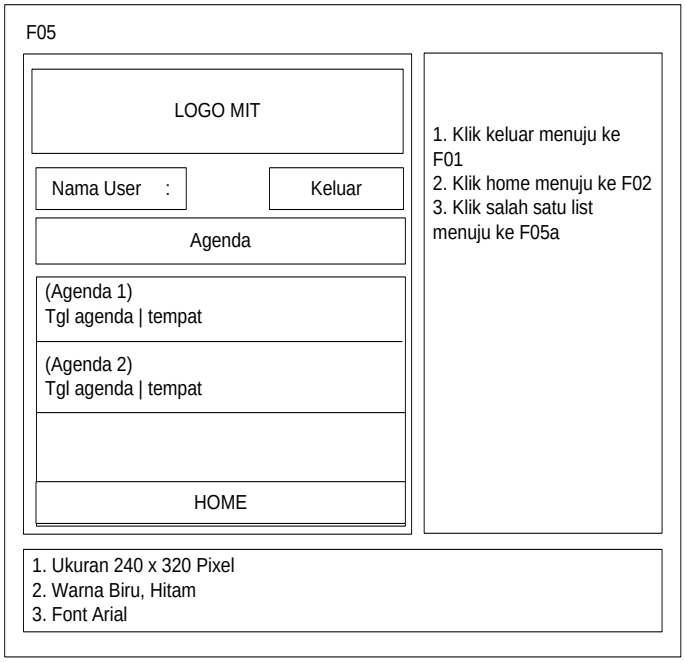
Gambar 3.51 *Form* Surat Keluar



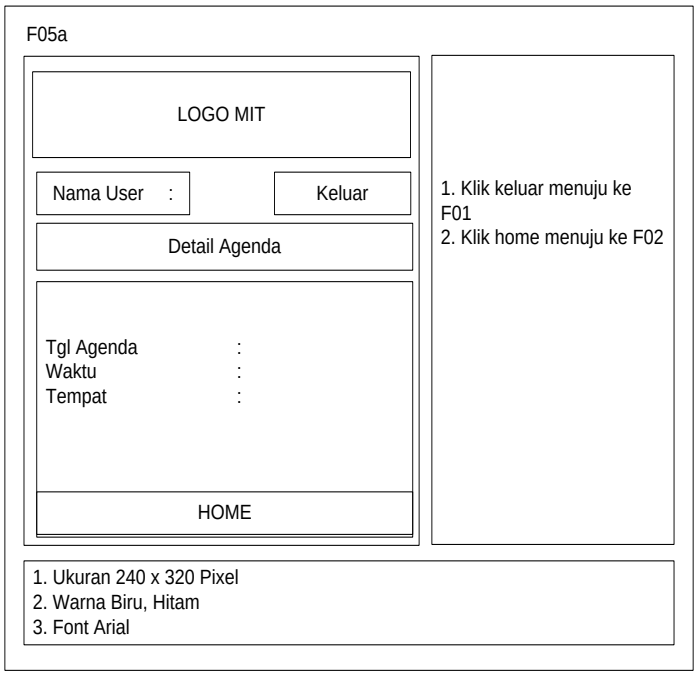
Gambar 3.52 *Form* Detail Surat Keluar



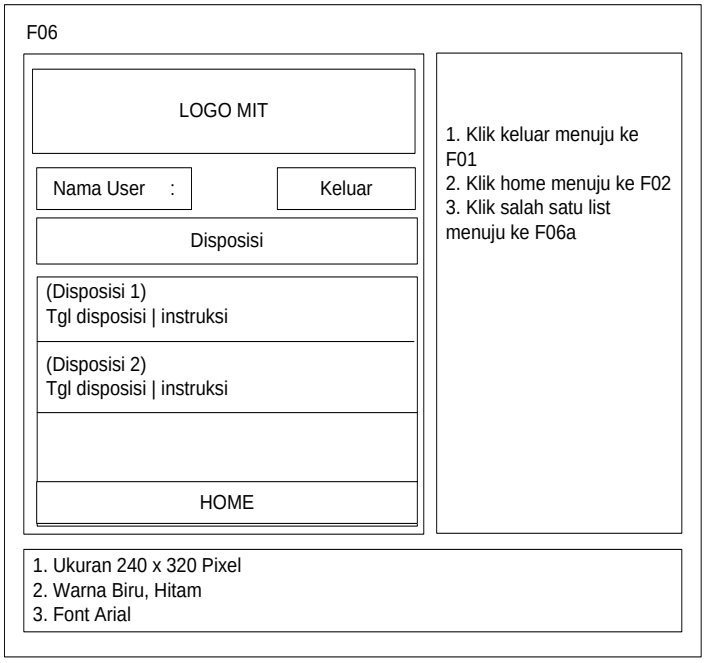
Gambar 3.53 *Form* Catatan Koreksi



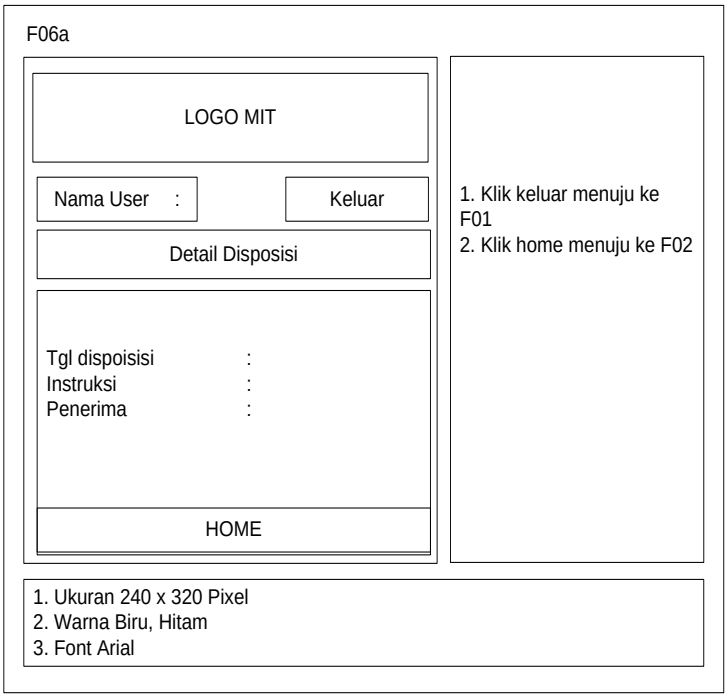
Gambar 3.54 *Form* Agenda



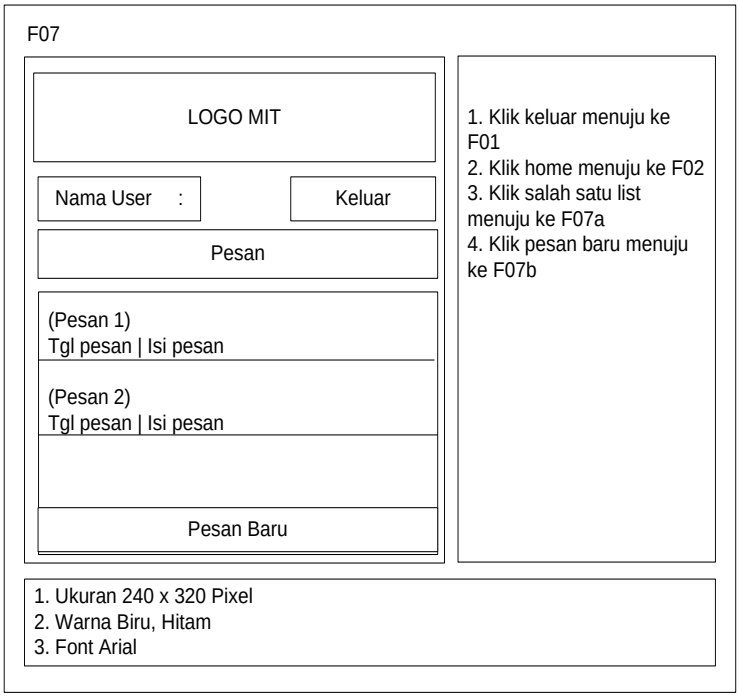
Gambar 3.55 *Form* Detail Agenda



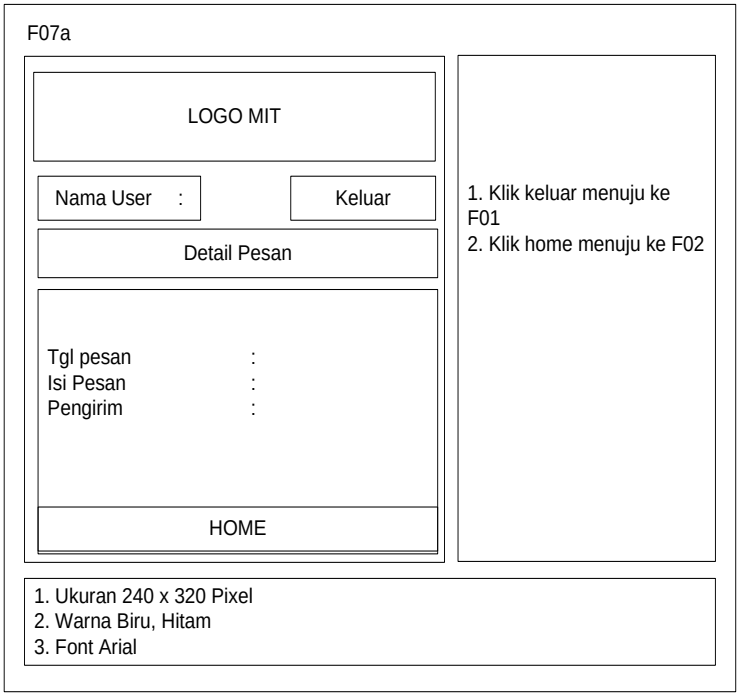
Gambar 3.56 *Form* Disposisi



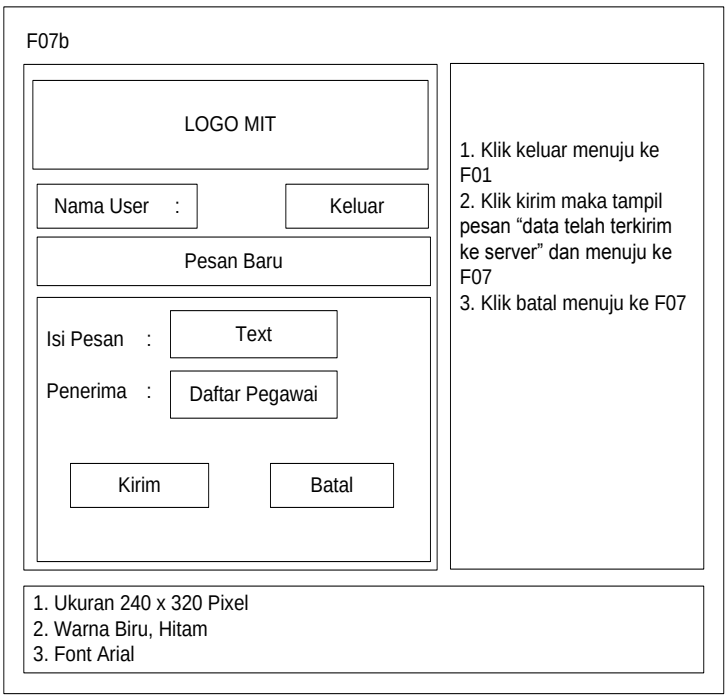
Gambar 3.57 *Form* Detail Disposisi



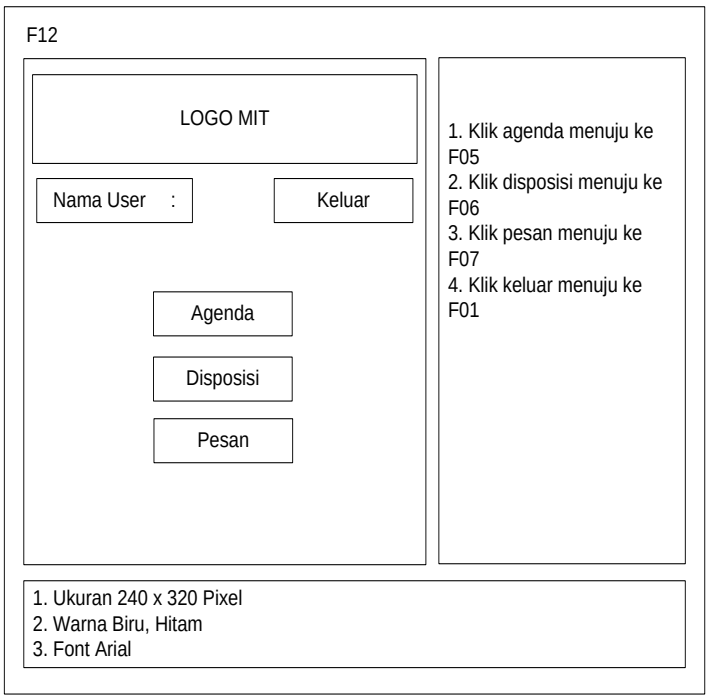
Gambar 3.58 *Form* Pesan



Gambar 3.59 *Form* Detail Pesan



Gambar 3.60 *Form* Pesan Baru



Gambar 3.61 *Form* Menu Utama Pegawai

3.2.5 Perancangan Pesan

Pada tabel 3.31 dapat dilihat perancangan pesan pada aplikasi *e-office* yang akan dibangun.

Tabel 3.31 Perancangan Pesan Pada Aplikasi *E-Office*

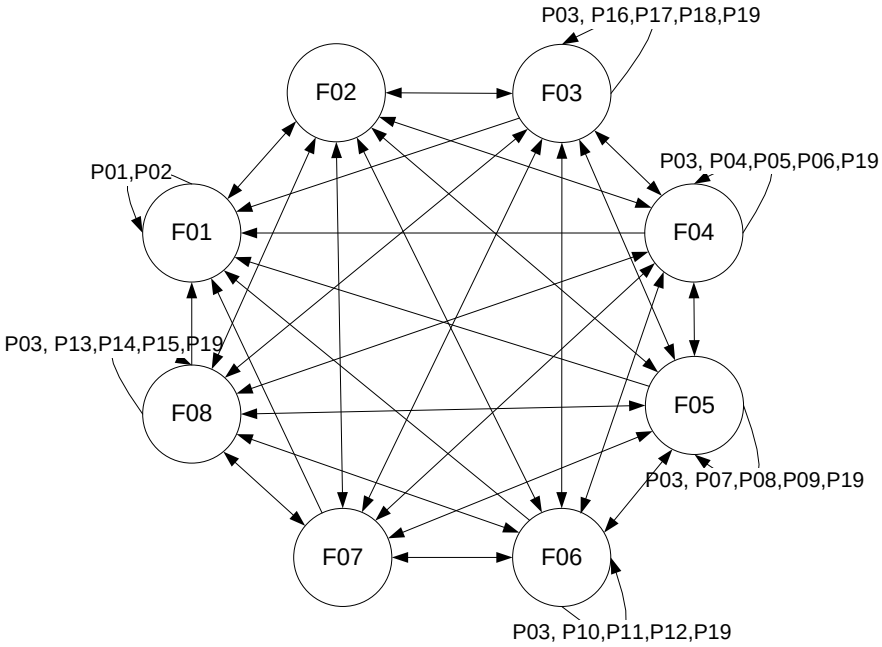
Kode	Pesan
P01	Maaf NIP anda salah
P02	Maaf Password anda salah
P03	Field ini harus diisi
P04	Data surat masuk berhasil disimpan
P05	Data surat masuk berhasil diubah
P06	Data surat masuk berhasil dihapus
P07	Data surat keluar berhasil disimpan
P08	Data surat keluar berhasil diubah
P09	Data surat keluar berhasil dihapus
P10	Data agenda berhasil disimpan
P11	Data agenda berhasil diubah
P12	Data agenda berhasil dihapus
P13	Data pesan berhasil disimpan
P14	Data pesan berhasil diubah
P15	Data pesan berhasil dihapus
P16	Data pegawai berhasil disimpan
P17	Data pegawai berhasil diubah
P18	Data pegawai berhasil dihapus
P19	Apakah anda yakin ingin menghapus data ini?
P20	Pesan telah dikirim ke server

3.2.6 Jaringan Semantik

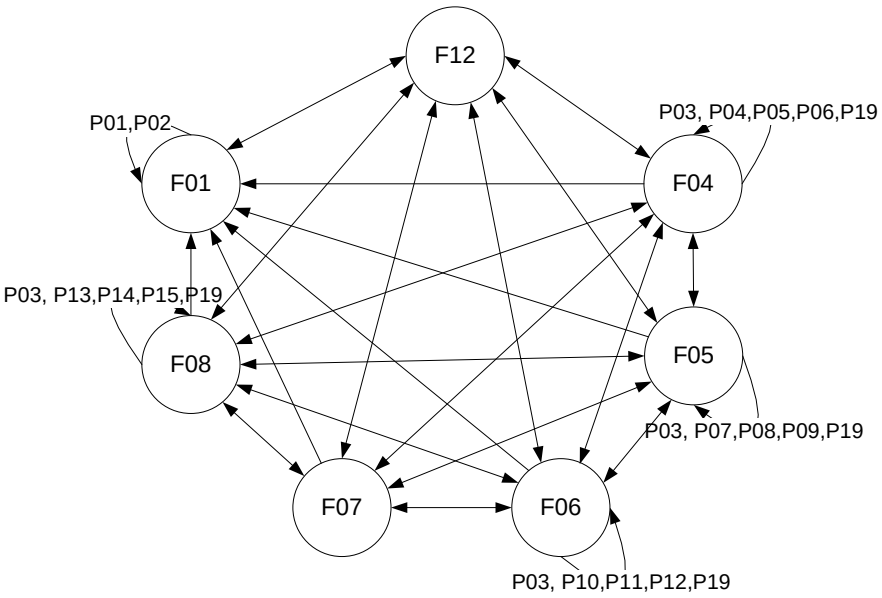
Setelah melakukan perancangan antarmuka, maka dilakukanlah perancangan terhadap aliran dari menu-menu yang akan digambarkan dalam sebuah jaringan semantik.

Jaringan semantik yang akan dijelaskan pada gambar, terdiri dari jaringan semantik menu *frontend* yang terdiri dari jaringan semantik menu pimpinan dan pegawai sedangkan *backend* yang terdiri dari jaringan semantik menu administrator dan operator pada aplikasi *e-office*.

1. Jaringan Semantik Pada Menu Aplikasi *Backend*

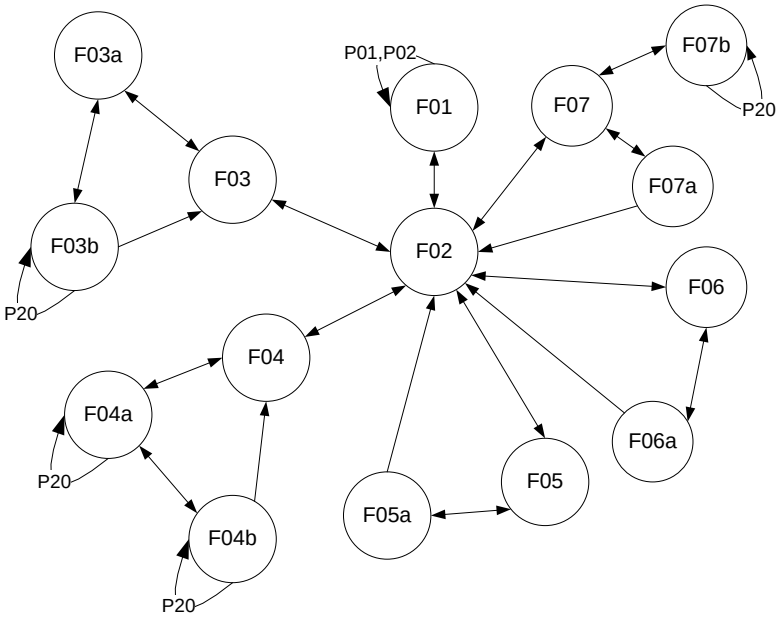


Gambar 3.62 Jaringan Semantik Pada Menu Adminiatrator

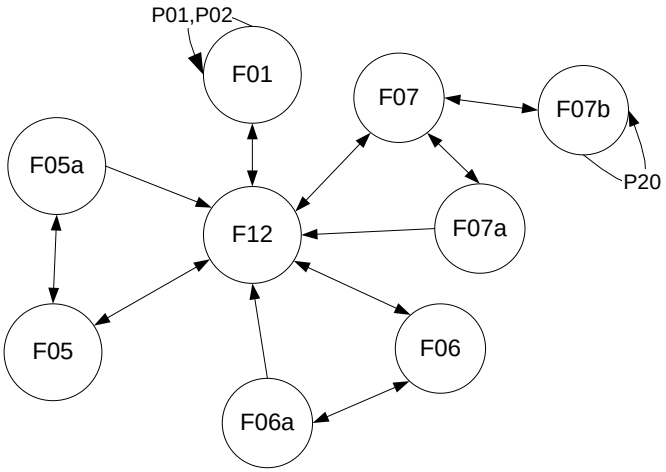


Gambar 3.63 Jaringan Semantik Pada Menu Operator

2. Jaringan Semantik Pada Menu Aplikasi *Frontend*



Gambar 3.64 Jaringan Semantik Pada Menu Pimpinan



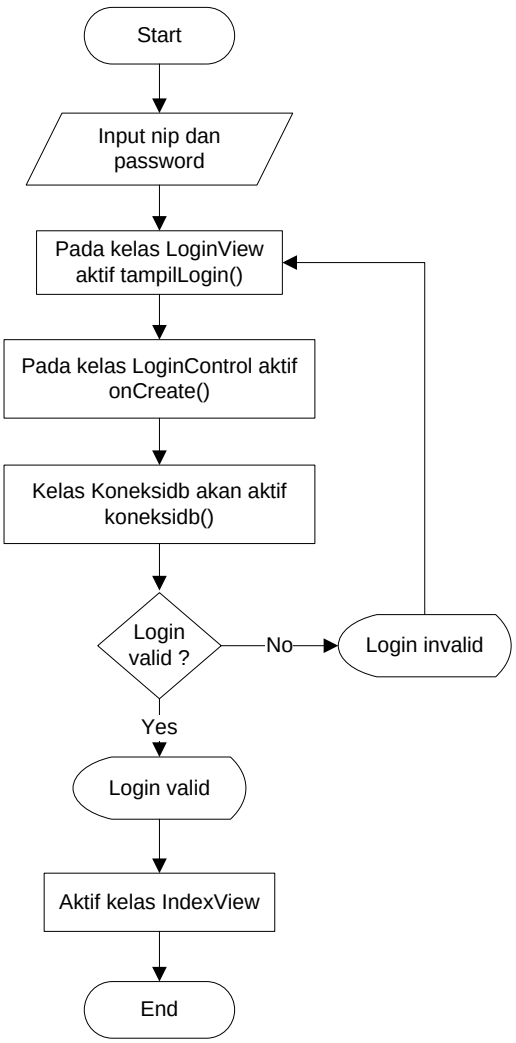
Gambar 3.65 Jaringan Semantik Pada Menu Pegawai

3.2.7 Perancangan *Method*

Perancangan *method* merupakan perancangan yang dilakukan untuk menetapkan detail algoritma yang akan dinyatakan ke dalam suatu perangkat lunak. Perancangan *method* digambarkan dengan menggunakan *flowchart.*

1. Perancangan *Method Login*

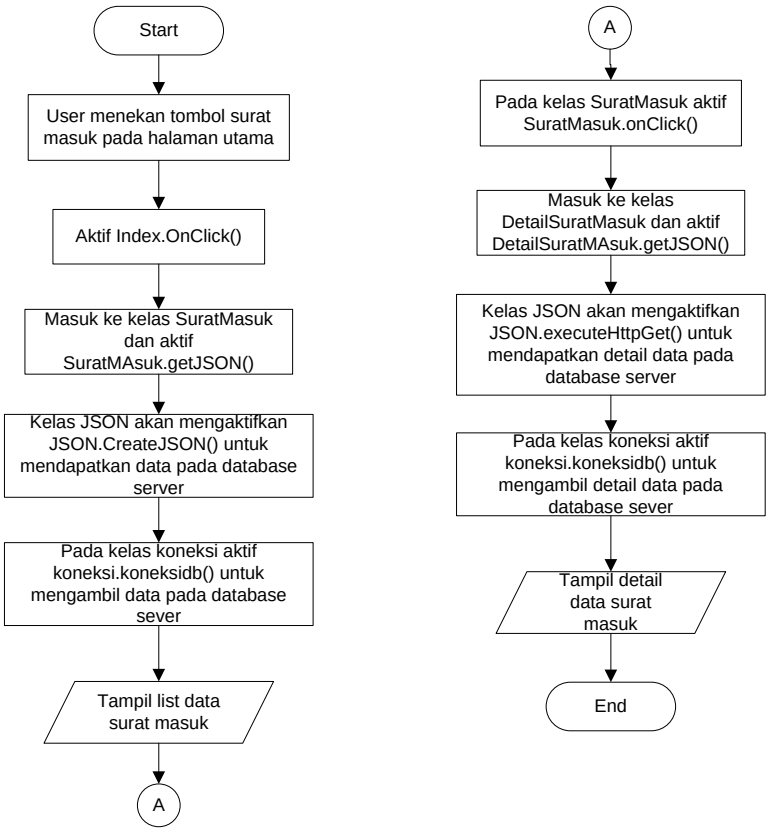
Method login mengimplementasikan *interface login* dan digunakan untuk masuk kedalam aplikasi. *Method* ini langsung dijalankan saat masuk ke kelas login, untuk lebih jelasnya dapat dilihat pada **Error! Reference source not found.**3.66.



Gambar 3.66 ***Flowchart Method Login***

2. Perancangan *Method* Pengolahan Surat Masuk

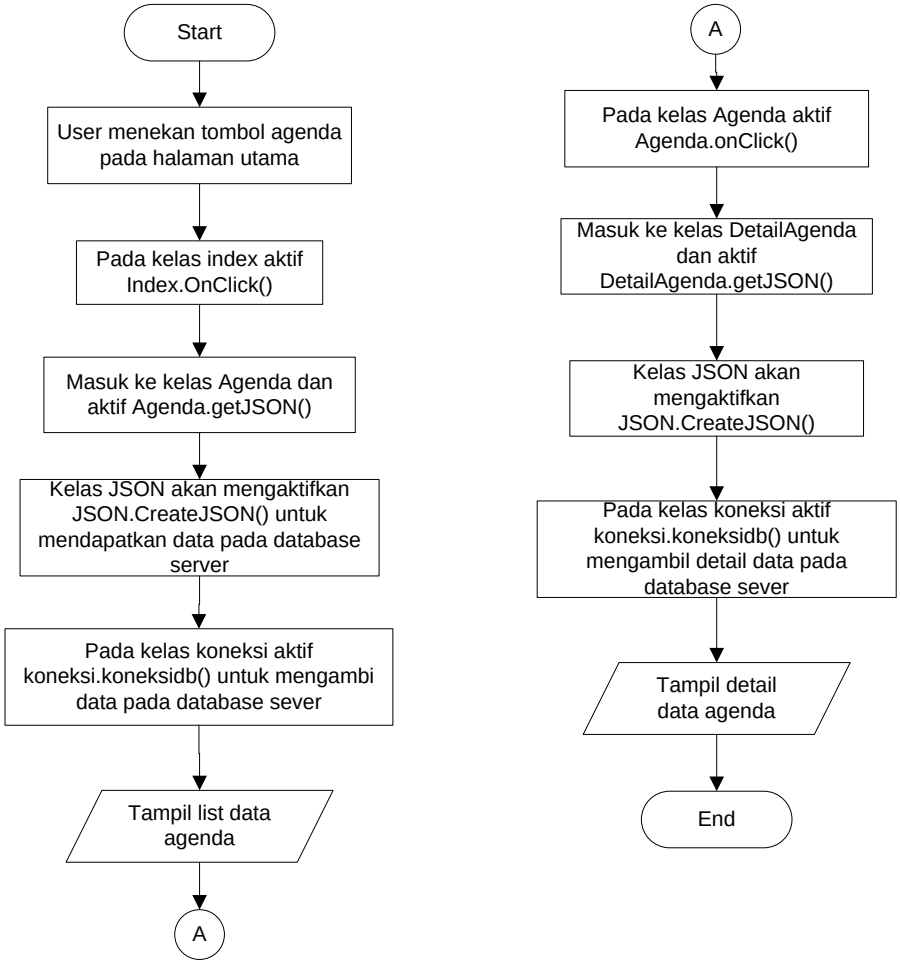
Method ini dijalankan untuk melakukan pengolahan data pada kelas surat masuk. Perancangan *method* ini dapat dilihat pada gambar 3.67.



Gambar 3.67 *Flowchart Method* Surat Masuk

3. Perancangan *Method* Lihat Agenda

Method ini dijalankan untuk pengolahan data pada kelas agenda. Perancangan *method* ini dapat dilihat pada gambar 3.68.



Gambar 3.68 *Flowchart Method* lihat Agenda