**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Pada setiap instansi atau kelembagaan yang resmi dan juga mempunyai struktur organisasi yang jelas, dapat dipastikan bahwa surat menyurat dilakukan oleh instansi atau lembaga untuk bertukar informasi ke lembaga lain agar terkesan santun dan formal, bahkan informasi yang disampaikan akan dapat cepat ditindaklanjuti. Setiap surat yang dikirim maupun diterima biasanya dicatat dalam sebuah sistem administrasi pembukuan yang merupakan hal yang wajib dilakukan mengingat jika suatu saat surat yang dikirim atau diterima dapat dipertanggungjawabkan sesuai dengan tujuan dan penerimanya. Dalam prakteknya dilapangan surat menyurat membutuhkan beberapa komponen administrasi fisik berupa buku catatan dan cetak fisik lembat disposisi dan lain sebagainya untuk menunjang pengarsipan dan tindak lanjut surat yang masuk atau keluar, sehingga membuat administrasi dan pencatatan dapat lebih riskan terjadinya hilangnya data surat masuk maupun keluar oleh karna bencana maupun hilang karna tidak disengaja. Oleh karna itu dirasa diperlukanya sebuah sistem manajemen surat menyurat yang terkomputerisasi dan terpusat dalam sebuah server dimana dapat diakses dan di-*backup* agar menjaga data tersebut dari hal hal yang tidak diinginkan.

Pada beberapa instansi atau lembaga telah menerapkan sistem surat menyurat yang dapat dilihat pada penelitian sebelumnya yaitu Sistem Informasi Administrasi Surat Berbasis Web Pada Kantor Camat Gianyar oleh I Kadek Ariana [1], dimana sistem ini merupakan system informasi berbasis web dan didukung dengan *sms gateway*. Pada pembuatan sistem informasi ini, konsep dan perancangannya dilakukan dengan pengumpulan data dari proses administrasi surat pada Kantor Camat Gianyar. tetapi sistem tersebut hanya bisa digunakan oleh satu lembaga karna dibangun sendiri sehingga masih belum bisa digunakan secara umum. Dengan hadirnya *cloud computing* dan konsep *software as a service* sistem manajemen surat menyurat dapat digunakan oleh semua lembaga atau perusahaan sesuai dengan kebutuhan dan mudah dalam penggunaan baik instalasi maupun perawatanya. Hal ini dapat dilihat dari penelitian berjudul Implementasi *Software as a Service* Pada Sistem Layanan Klinik Kesehatan oleh I Gusti Putu Arlina Ayuningsih [2]. Dimana penerapan *software as a service cloud computing* dapat memudahkan klinik kesehatan dalam pengadaan sistem layanan klinik kesehatan yang mudah dan efisien dalam penerapanya.

*Cloud computing* merupakan metode komputasi yang memanfaatkan teknologi internet dalam menyediakan layanan yang dapat diakses dimanapun, serta akses kedalam jaringan yang terkonfigurasi seperti penyimpanan, server, dan perangkat lunak yang sesuai dengan permintaan pengguna, serta fleksibel dan dapat diatur sesuai dengan kebutuhan [3]. *Software as a Service* merupakan salah satu dari tiga jenis layanan *Cloud Computing* yang diantaranya yaitu, *Platform as a Service* dan *Infrasructure as a Service* dimana Software as a Service merupakan sebuah layanan aplikasi atau perangkat lunak yang terdistribusi oleh provider pihak ke tiga yang disediakan untuk konsumen melalui jaringan internet berbasiskan *Cloud Computing* [3].

Konsep *software as a service* membuat implementasi sebuah aplikasi menjadi lebih sederhana, ketika oranisasi ataupun individu membutuhkan sebuah aplikasi, maka hanya perlu mengakses aplikasi tersebut pada jaringan internet dan tentu saja dapat digunakan dimana saja. Pemanfaatan teknologi internet membuat penerapan aplikasi berbasis *cloud computing* dengan konsep *software as a service* menjadi sangat menguntungkan karena pengguna sistem tidak perlu berinteraksi langsung dengan penyedia layanan dan juga sangat fleksibel dalam menyesuaikan dengan kebutuhan pengguna.Dengan uraian tersebut maka penulis akan melakukan penelitian dan melakukan perekayasaan dengan mengangkat judulImplementasi Software As A Service *Cloud Computing* Sistem Manajemen Surat Menyurat yang diharapkan dapat membantu mengurangi beban sumberdaya baik komputer maupun manusia dalam mengimpelemntasikan sebuah sistem manajemen surat menyurat sehingga dapat digunakan di semua instansi atau lembaga.

* 1. **Rumusan Masalah**

Berdasarkan latar belakang yang diuraikan di bagian 1.1 rumusan masalah dalam pengambilan topik skripsi ini adalah Bagaimana membuat sebuah sistem surat menyurat dan dapat digunakan oleh semua instansi atau lembaga dengan berbasiskan *Cloud Computing* dan menggunakan konsep *Software as a Service*.

* 1. **Tujuan Penelitian**

Dalam sebuah penelitian terdapat tujuan yang dapat diuraikan. Adapun tujuan dalam penelitian ini adalah menghasilkan aplikasi surat menyurat yang dapat digunakan oleh semua instansi atau lembaga dengan berbasiskan *Cloud Computing* dan menggunakan konsep *Software as a Service*.

* 1. **Manfaat Penelitian**

Adapun manfaat yang didapat dalam penelitian ini dimana manfaat tersebut dibagi menjadi 2 yaitu manfaat praktis dan juga manfaat akademis.

* + 1. **Manfaat Praktis**

1. Mempermudah dalam proses pencatatan administrasi surat masuk maupun surat keluar pada instansi atau lembaga.
2. Mempermudah penggunaan sistem, karena sistem yang berupa layanan sehingga dapat digunakan oleh berbagai perusahaan, lembaga, atau instansi mulai dari skala kecil karena mudah untuk diimplementasikan.
   * 1. **Manfaat Akademis**
3. Penelitian yang dibuat dapat memberi kontribusi keilmuan pada bidang pengembangan *cloud computing* dan menjadi refrensi untuk mengembangkan sistem dengan konsep layanan atau s*oftware as a service.*
4. Penelitian ini juga dapat menjadi refrensi bagi penelitian selanjutnya yang mengangkat tema *cloud computing.*
   1. **Ruang Lingkup Penelitian**

Untuk membatasi pembahasan dan memberi pemahaman pada topik yang diambil pada penelitian ini tentang batasan pembahasan, maka dijabarkan ruang lingkup penelitian agar memberikan pemikirian yang sesuai dengan tujuan penelitian dan tidak menimbulkan kesalahan persepsi tentang penelitian ini. Berikut adalah ruang lingkup penelitian :

1. Sistem surat menyurat dapat memanajemen surat masuk, disposisi surat dan surat keluar.
2. Sistem surat menyurat menggunakan konsep *software as a service cloud computing* yang dirancang agar dapat digunakan disetiap instansi atau lembaga.
3. Sistem surat menyurat dijalankan di *web browser* (*mozila*, *google* *chrome*, *safari*) baik di *desktop*, *tablet*, atau *smartphone* dengan berbagai sistem operasi.
4. Terdapat perbedaan layanan yang tersedia terkait dengan kebutuhan instansi atau lembaga.
5. Dalam perbedaan layanan tersebut juga terdapat perbedaan harga yang harus dibayarkan dalam sebuah layanan. Adapun paket layanan yang dimaksud dibedakan menjadi empat yaitu Paket *Free*, Paket *Basic*, Paket *Premium*, dan Paket *Unlimited*.
6. Sistem pembayaran yang dilakukan adalah hanya sebatas agar sistem dapat digunakan sebagaimana paket layanan yang telah dipilih oleh *user*.
7. Sistem dapat melakukan *backup* *data* surat masuk, surat keluar, dan disposisi.
8. Dalam paket layanan sistem tersebut dibedakan juga level akses yang diberikan.
9. Secara garis besar level akses yang diberikan adalah:
   1. Level Administrator Sistem.

Pada level administrator sistem berfungsi untuk memanajemen sistem secara keseluruhan. Fungsi administrator sistem yaitu mengaktivasi *user*, memanipulasi layanan yang disediakan.

* 1. Level *User* Administrator Perusahaan/Lembaga

Level akses *user* administratir perusahaan/lembaga berfungsi untuk mengelola sistem yang digunakan oleh satu perusahaan dan tidak akan bisa mengelola data perusahaan lainya. Adapun fungsi lainya yaitu: registrasi perusahaan, mendaftarkan *user* kepala perusahaan/lembaga, mendaftarkan *user* bagian atau operator, memanipulasi data seluruh *user*, memonitoring data surat, meningkatkan dan menurunkan kemampuan sistem, melakukan pembayaran, dan memanipulasi data tentang perusahaan.

* 1. Level User Kepala Perusahaan/Lembaga

Pada level akses kepala perusahaan ini mempunya fungsi untuk memvalidasi *user* yang ditambahkan oleh *user* Administrator Perusahaan/lembaga, kemudian membaca dan mendisposisi surat masuk, memanipulasi data tindak lanjut disposisi,dan membaca data surat keluar.

* 1. Level User Bagian atau Operator.

Level user bagian atau operator berfungsi untuk memanipulasi data surat keluar dan surat masuk, serta dapat melihat data surat yang telah didisposisi oleh *user* kepala lembaga.

1. Sistem dirancang dengan bahasa pemrograman *PHP*, dengan desain layout menggunakan *CSS*, serta basis data *MySQL*.
2. Pada penelitian ini tidak membahas tentang kemanan data.

**1.6. Metode Penelitian**

Pada penelitian ini dilakukan beberapa metode dalam memecahkan masalah yang disimpulkan pada rumusan masalah yang akan dijabarkan dalam beberapa sub bab.

**1.6.1 Metode Pengumpulan Data**

Dalam pembuatan skripsi ini digunakan dua metode pengumpulan data yang bertujuan untuk menghimpun data yang akan menunjang adanya penelitian ini. Adapun metode pengumpulan data yang dilakukan yaitu:

1. Metode Studi Literatur : Studi Literatur diadakan bertujuan untuk mencari referensi tentang teori-teori yang mendukung adanya penelitian, agar dapat menjadi acuan yang sesuai dengan topik skripsi yang diambil. Dimana dalam studi literatur ini mencari topik tentang *cloud computing* dan surat menyurat.
2. Metode Observasi : Metode observasi dilakukan untuk mencari data surat masuk atau keluar sebagai sampel data dimana dari data tersebut akan didapatkan sebuah informasi untuk menganalisis kebutuhan sistem yang selanjutnya akan diimplementasikan sebagai fitur fitur didalam sistem yang akan dibuat.

**1.6.2 Analisis Kebutuhan**

Pada proses pertama adalah analisis kebutuhan sistem dimana penulis menganalisis kebutuhan selama perancangan. Kebutuhan yang dimaksud adalah sistem *cloud computing* seperti apa yang dibutuhkan oleh *user*.

**1.6.3 Desain Sistem**

Pada tahap kedua penulis mendesain sistem yang akan diaplikasikan menggunakan beberapa diagram seperti *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram, Class Diagram* dan Struktur Tabel. Dalam tahapan ini dirancang juga antar muka dari sistem *cloud computing* surat menyurat.

**1.6.4 Penulisan Kode Program**

Setelah mengetahui desain sistem, penulis kemudian masuk ketahap penulisan kode program. Bahasa pemrograman yang digunakan adalah bahasa pemrograman *PHP*, *JavaScript* dimana bahasa pemrograman tersebut berbasiskan web. Dan didukung dengan sistem *DBMS* (Basis data *Management Sistem*) *MySQL*.

**1.6.5 Pengujian Program**

Pengujian sistem dilakukan apabila tahapan implementasi telah selesai dilakukan. Pengujian sistem menggunakan pendekatan *Blackbox Testing* (pengujian kotak hitam). Hal ini dilakukan untuk mengetahui apakah aplikasi ini memenuhi fungsionalitas sistem yang ingin capai.

**BAB II**

**TINJAUAN PUSTAKA**

**2.1. Penelitian Terdahulu**

Agar proses pengkajian yang dilakukan dalam penelitian ini tidak jauh melenceng dari permasalahan utama diperlukanlah data-data pendukung sebagai perbandingan dalam membahas fokus permasalahan dari penelitian ini. Adapun data-data pendukung, yang merupakan hasil dari jurnal-jurnal penelitian terdahulu dibidang pengembangan Sistem Surat Menyurat dan *Software as a Service*, yang dianggap relevan sebagai data-data pendukung.

Pada penelitian yang dilakukan oleh I Kadek Ariana Sistem Informasi Administrasi Surat Berbasis Web Pada Kantor Camat Gianyar [1], dimana didapatkan kesimpulan pada penelitian tersebut adalah sistem yang dibuat telah dapat membantu administrasi surat mulai dari pencatatan, kemudian pencarian data surat masuk, disposisi dan surat keluar. Penerapannya yang menggunakan aplikasi berbasis web menjadikan sebuah kemudahan dalam pengimplementasianya. Tetapi terdapat masalah dalam perawatan dan pembaharuan sistem sehingga diperlukanya biaya yang lebih untuk menunjang kedua hal tersebut. Sehingga permasalahan tersebut dapat dipecahkan pada penelitan yang dilakukan oleh I Gusti Putu Arlina Ayuningsih yang berjudul Implementasi *Software as a Service* Pada Sistem Layanan Klinik Kesehatan [2]. Dalam pembahasanya implementasi *software as a service* mengambil studi kasus klinik kesehatan dimana aplikasi yang dihasilkan dapat digunakan diberbagai klinik kesehatan, sehingga dapat membantu petugas klinik atau tim medis memiliki sebuah layanan kesehatan. Dengan implementasi *software as a service* memberikan biaya yang relatif murah. Hal tersebut dikarenakan tidak adanya proses pengembangan sistem yang panjang. Begitupun juga perawatan dan pembaharuan applikasinya dilakukan oleh penyedia layanan sehingga dari segi biaya dapat ditekan dan aplikasi yang dijadikan layanan dapat dikatakan sangat praktis baik dalam implementasi dan juga penggunaannya, sehingga pengguna dapat menentukan sebagaimana kebutuhan akan layanan klinik tersebut.

Dari penjabaran diatas penelitian yang telah dilakukan masih terdapat kekurangan dalam pengembangan dan pemeliharaanya dikarenakan biaya yang diperlukan akan sangat besar dan tentu saja memerlukan sumber daya yang baik dalam mengoperasikan sistem. Sementara penelitian yang akan di lakukan adalah implementasi konsep *Software as a Service Cloud Computing* dapat memberikan dampak positif bagi perusahaan atau lembaga yang mengimplementasikannya, mulai dari berkurangnnya biaya dalam pengembangan dan perawatan serta pembaharuan aplikasi yang digunakan.

**2.2. Surat Menyurat**

Surat adalah alat komunikasi tertulis yang berasal dari suatu pihak dan ditujukan kepada pihak lain untuk menyampaikan warta [4,5] Sedangkan menurut Wirladiharjo surat adalah setiap tulisan berisikan pernyataan dari penulisnya, yang dibuat dengan tujuan menyampaikan informasi kepada pihak lain [4,6]. Sedangkan pengertian surat menurut Silmi surat adalah sehelai kertas atau lebih yang digunakan untuk mengadakan komunikasi secara tertulis [4,7].

Dari penjelasan diatas maka dapat ditarik kesimpulan bahwa surat adalah alat komunikasi yang berisi informasi baik tertulis maupun bergambar yang hendak disampaikan kepada pihak lain yang bersangkutan dan memiliki kelebihan tersendiri dalam hal kerahasiaan, keefektifan dan ekonomis. Jadi dari beberapa pengertian di atas dapat disimpulkan bahwa surat adalah alat komunikasi yang secara tertulis untuk menyampaikan informasi.

**2.2.1. Fungsi Surat**

Sebagai sarana dalam penyampaian pesan secara tertulis, surat berperan dalam mencapai tujuan suatu instansi atau organisasi dalam menjalin kerjasama antar organisasi. Menurut Barthos, surat memiliki fungsi sebagai berikut:

* 1. Wakil dari pengirim atau penulis
  2. Bahan pembukti
  3. Pedoman dalam mengambil tindakan lebih lanjut
  4. Alat pengukur kegiatan organisasi
  5. Sarana memperpendek jarak (fungsi abstrak)

Dari uraian diatas mengenai fungsi surat, dapat disimpulkan bahwa surat dapat digunakan sebagai bukti yang paling akurat bila terjadi suatu permasalahan dengan pihak lain [5].

* + 1. **Surat Masuk**

Surat masuk adalah surat yang diterima dari perusahaan atau instansi lain kepada pihak yang bersangkutan. Surat masuk merupakan saran komunikasi tertulis yang diterima dari instansi atau perorangan. Dapat pula diartikan, surat masuk adalah semua jenis surat yang diterima dari instansi lain maupun perorangan, baik yang diterima melalui pos maupun yang diterima dari kurir dengan mempergunakan buku pengiriman [6].

* + 1. **Prosedur Penanganan Surat Masuk**

Prosedur penanganan surat masuk pada suatu organisasi secara umum meliputi aktivitas-aktivitas sebagai berikut:

* 1. Penerimaan Surat Masuk, Mengumpulkan surat masuk dan meneliti alamat surat yang diterima.
  2. Pensortiran Surat, Memisahkan surat ke dalam surat dinas (biasa, penting, atau rahasia ) maupun surat pribadi, lalu menyortir berdasarkan tujuannya.
  3. Pembukaan Surat, membaca surat dinas ( biasa, penting, atau rahasia ), memeriksa lampiran kemudian membubuhkan sampel pengagendaan.
  4. Pengangendaan Surat Masuk, mencatat ke dalam buku agenda atribut atribut pada surat masuk tersebut.
  5. Pengklasifikasian Surat, mengumpulkan bedasarkan jenis dan tingkatan kepentingan surat.

Pendistribusian Surat, menyampaikan surat ke tujuan tersebut, kemdian mngklasifikasikan surat surat kepada pimpinan dengan menyusun dari yang paling penting sampai yang kurang penting. Dan yang terakhir adalah menentukan pejabat mana yang harus menangani surat, dan menyertakan surat disposisi [4].

* + 1. **Surat Keluar**

Surat keluar yaitu surat yang dikirim dari pihak baik instansi, organisasi atau perusahaan yang berisi tentang suatu informasi atau data baik itu perintah, pemberitahuan maupun informasi lainnya. Surat keluar adalah surat yang sudah lengkap bertanggal, bernomor, berstempel dan telah ditandatangani oleh pejabat yang berwenang yang dibuat oleh suatu instansi atau lembaga lain [6].

**2.4. *Cloud Computing***

Komputasi awan (cloud computing; cloud) adalah sebuah bentuk layanan yang membuka peluang untuk menggunakan komputasi agar dapat hadir dimanapun, memberikan kenyamanan, akses jaringan sesuai permintaan (on-demand) ke lokasi sumber daya komputasi yang terkonfigurasi (misalnya, jaringan, server, penyimpanan, aplikasi, dan layanan), yang dapat dengan cepat dijalankan dan digunakan, dengan upaya pengelolaan minimal atau dengan menggunakan penyedia jasa layanan. Bentuk komputasi ini terdiri dari lima karakteristik utama, tiga model layanan, dan empat model penyebaran [3].

**2.4.1. Karakteristik *Cloud Computing***

Adapun Karakteristik Cloud Computing menurut NIST (National Institute of Standards and Technology) yang telah dipublikasi yaitu:

1. On-demand self-service. Konsumen dapat menentukan sendiri kemampuan komputasi yang diinginkan, seperti waktu penggunaan server dan penyimpanan pada jaringan, tanpa harus berinteraksi langsung (antar manusia) dengan pemberi jasa layanan.
2. Broad network access. Kemampuan tersebut tersedia pada jaringan dan diakses melalui mekanisme standar yang dapat digunakan dengan semua jenis platform klien, thin or thick  (misalnya, ponsel, tablet, laptop, maupun workstation).
3. Resource pooling. Sumber daya komputasi dari penyedia layanan terkumpul untuk melayani banyak konsumen, menggunakan model multi-tenant, dengan sumber daya yang berbeda-beda, baik fisik maupun virtual, yang secara dinamis dapat dijalankan dan dipindahkan sesuai dengan permintaan konsumen. Ada nuansa kemandirian lokasi yang umumnya tidak dialami pelanggan, yang biasanya tidak dapat mengetahui atau dapat mengendalikan lokasi, secara tepat, dari sumber daya daya yang disediakan, namun mungkin dapat menentukan lokasi pada tingkat abstraksi yang lebih tinggi (misalnya, negara, negara atau *data center*). Contoh sumber daya termasuk penyimpanan, pemrosesan, memori, dan bandwidth jaringan.
4. Rapid elasticity. Kemampuan dapat secara elastis ditentukan dan diluncurkan, dalam beberapa kasus secara otomatis, pada skala yang cepat berkembang, naik maupun turun, sepadan dengan permintaan. Untuk konsumen, kemampuan yang tersedia untuk *provisioning* menjadi tidak terbatas dan dapat disesuaikan dalam secara kuantitas setiap saat.
5. Measured service. Sistem *cloud* secara otomatis mengontrol dan mengoptimalisasi penggunaan sumber daya, memanfaatkan kemampuan pengukuran (biasanya secara *pay-per-use* atau *charge-per-use*)  pada tiap tingkat abstraksi yang sesuai dengan jenis layanan (misalnya: penyimpanan, pemrosesan, *bandwidth*, dan akun pengguna aktif). Penggunaan sumber daya dapat dipantau, dikendalikan dan dilaporkan, menciptakan keterbukaan kepada baik penyedia jasa maupun pengguna pada layanan yang dilakukan [3].

**2.4.2. Jenis Layanan Cloud Computing**

Pada Cloud Computing terdapat juga perbedaan layanan yang diberikan berikut adalah jenis-jenis layanan Cloud Computing menurut NIST (National Institute of Standards and Technology), yaitu:

1. *Software as a Service (SaaS).* Kemampuan yang diberikan kepada konsumen untuk menggunakan aplikasi milik pemberi layanan yang bekerja pada infrastruktur *cloud*\*. Aplikasi-aplikasi tersebut dapat diakses dari apapun perangkat klien, melalui baik *thin client interface* seperti *web-browser* (contohnya: *web-based email*) atau tampilan antarmuka program. Konsumen tidak mengatur atau mengendalikan infrastruktur *cloud*, termasuk jaringan, servers, sistem operasi, penyimpanan, atau bahkan kemampuan aplikasi individual, dengan kemungkinan pengecualian pada penggunaan terbatas, aplikasi tertentu untuk mengatur konfigurasi.
2. *Platform as a Service (PaaS)*. Kemampuan diberikan kepada konsumen untuk meluncurkan kedalam infrastruktur *cloud* aplikasi-aplikasi yang dibuat atau dibutuhkan mereka, menggunakan bahasa pemerograman, liberaries, layanan, dan alat bantu yang disediakan oleh penyedia jasa layanan.  Konsumen tidak mengatur atau mengendalikan infrastruktur cloud, termasuk jaringan, servers, sistem operasi maupun peyimpanan, namun memiliki kendali atas aplikasi yang dimasukkannya.
3. *Infrastructure as a Service (IaaS).* Kemampuan diberikan pada konsumen untuk menentukan pengelolaan, penyimpanan, networks dan sumber komputansi fundamental lainnya, dimana konsumen dapat meluncurkan dan menjalankan sebarang perangkat lunak, yang mungkin dapat termasuk sistem operasi dan aplikasi. Konsumen tidak mengatur atau mengendalikan infrastruktur *cloud* namun memiliki kendali atas sistem operasi, penyimpanan dan aplikasi yang diluncurkan. Dan mungkin kendali terbatas atas komponen *networking* tertentu [3].

**2.4.3. Jenis Penyebaran *Cloud Computing***

1. *Private cloud.* Infrastruktur *cloud* yang ditetapkan secara eksklusif untuk digunakan oleh satu organisasi yang membawahi beberapa konsumen (misalnya: unit bisnis). Dapat dimiliki, diatur dan dikendalikan oleh organisasi, pihak ketiga, atau paduan antar keduanya, dan dapat berada pada atau diluar lokasi.
2. *Community cloud.* Infrastruktur *cloud* yang ditetapkan secara eksklusif untuk digunakan oleh komunitas konsumen dari organisasi yang memiliki kepedulian yang sama (misalnya: misi, kebutuhan keamanan, kebijakan dan  pertimbangan kepatuhan). Mungkin dimiliki, diatur dan dijalankan oleh satu atau lebih organisasi dalam komunitas, pihak ketiga, atau paduan antar keduanya, dan dapat berada pada atau diluar lokasi.
3. *Public cloud.* Infrastruktur *cloud* yang ditetapkan untuk bebas digunakan oleh masyarakat luas. Mungkin dimiliki, diatur, dan dijalankan oleh organisasi bisnis, akademis atau pemerintahan, atau kombinasi diantaranya. Berada pada lokasi penyedia layanan *cloud*.
4. *Hybrid cloud*. Infrastruktur *cloud* yang merupakan komposisi dari dua atau lebih  infrastruktur *cloud* yang berbeda (*private, community,* atau *public*), yang masing-masing tetap berdiri sendiri sebagai entitasnya, namun terikat dengan standar atau kepemilikan teknologi yang sama, yang memungkinan portabilitas data dan aplikasi (seperti: *cloud bursting* untuk penyeimbang beban antar *cloud*) [3].

**2.5. *PHP***

*PHP* merupakan singkatan dari *PHP Hypertext Preprocessor* yang merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan proses di server. Hasilnya yang dikirimkan ke klien, tempat pemakai menggunakan *browser.* Secara khusus, *PHP* dirancang untuk membentuk aplikasi web dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, Anda bisa menampilkan isi basis data ke halaman web. Pada prinsipnya *PHP* mempunyai fungsi yang sama dengan skrip-skrip seperti *ASP* (*Active Server Page*), *Cold Fusion*, ataupun *Perl*. Namun, perlu diketahui bahwa *PHP* sebenarnya bisa dipakai secara *command line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser* [8].

**2.6. *MySQL***

*MySQL* adalah salah satu jenis basis data *server* yang sangat terkenal. Kepopulerannya disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses *database*. Selain itu, ia bersifat *Open Source* (Anda tidak perlu membayar untuk menggunakannya) pada pelbagai platform (kecuali untuk jenis Enterprise, yang bersifat komersial).

*MySQL* termasuk jenis *RBDMS* (*Relational Database Management System*). Itulah sebabnya, istilah seperti tabel, baris, dan kolom digunakan pada *MySQL*. Pada *MySQL*, sebuah *database* mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom [8].

* 1. ***Unified Modeling Language***

UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek”. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami, pada pengertian lain juga *UML* merupakan Metodologi kolaborasi antara metoda-metoda *Booch*, *OMT* (*Object Modeling Technique*), serta OOSE (*Object Oriented Software Enggineering*) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisis dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (*OOP*) [9]. Beberapa literature menyebutkan bahwa *UML* menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misanya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi”. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis [10].

* + 1. ***Use Case Diagram***

Diagram ini bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna [10].

Tabel 2.1 Tabel Simbol *Use Case Digaram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *Actor* | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

* + 1. ***Activity Diagram***

Diagram ini bersifat dinamis. Diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi – fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek [10].

Tabel 2.2 Simbol *Activity Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| **1** |  | *Actifity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| **2** |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| **3** |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| **4** |  | *Actifity Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| **5** |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

* + 1. ***Sequence Diagram***

Diagram ini bersifat dinamis. *Diagram sequence* merupakan diagram interaksi yang menekankan pada pengiriman pesan (message) dalam suatu waktu tertentu [10].

Tabel 2.3 Simbol *Sequence Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *LifeLine* | Objek *entity*, antarmuka yang saling berinteraksi. |
| 2 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |
| 3 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |

* + 1. ***Class Diagram***

Diagram kelas bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka, kolaborasi serta relasi [10].

Tabel 2.4 Simbol *Class Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Generalization* | Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 2 |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | *Collaboration* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 5 |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |

**BAB III**

**ANALISIS DAN DESAIN SISTEM**

* 1. **Analisis Kebutuhan**

Analisis kebutuhan merupakan suatu tahapan untuk menganalisa semua kebutuhan yang diperlukan dalam mengembangkan sistem dengan berdasarkan masalah yang dipaparkan dalaram rumusan masalah. Analisis kebutuhan ini mencakup kebutuhan secara fungsional dan non fungsional. Sistem ini dibuat dengan tujuan untuk menghasilkan aplikasi sistem surat menyurat yang dapat mencatat administrasi surat menyurat yang dapat diimplementasikan pada setiap perusahaan dengan menggunakan konsep *Cloud Computing Software As A Service*. *User* dibagi menjadi 4 tipe *user*, yang pertama adalah admin sistem, kemudian terdapat juga admin perusahaan, *user* kepala perusahaan dan *user* bagian atau operator yang ada pada perusahaan tersebut. Sistem yang dirancang agar *user* dapat menentukan sendiri kemampuan sistem tanpa harus berinteraksi langsung dengan pemberi jasa layanan. Sistem yang akan dibuat nantinya dapat dijalankan di browser, dan dapat terus diakses kapanpun dan dimana saja. sistem dapat mengelola data administrasi surat menyurat seperti surat masuk, surat keluar, dan disposisi yang dapat dimanipulasi oleh *user* sesuai dengan keinginan dari *user* tersebut. Sistem nantinya juga dapat memilah *user* sesuai dengan Layanan yang dijalankan dan dapat membatasi fungsi fungsi sistem seuai dengan paket yang dibayarkan. Sistem nantinya juga dapat monitor atau dipantau serta dilaporkan.

* + 1. **Analisis Kebutuhan Fungsional**

Analisis kebutuhan funsional merupakan analisa kebutuhan-kebutuhan funsional dari perangkat lunak yang akan dibuat, dalam arti lain kebutuhan fungsional merupakan fungsi-fungsi atau fitur-fitur yang terdapat pada perangkat lunak yang akan dibuat. Pada sistem yang dibuat terdapat 5 macam kebutuhan fungsional, yaitu kebutuhan *user* dengan level yang berbeda, pengelolaan *user* bagian oleh user perusahaan, pencatatan administrasi surat menyurat yang dilakukan oleh setiap *user*, mencetak disposisi dan penyimpanan ke dalam basis data. Adpun kebutuhan fungsional diatas dapat dijabarkan sebagai berikut.

* + 1. Kebutuhan *user* dengan level berbeda

Sistem dapat membedakan jenis *user* pada sistem agar nantinya *user* dapat bekerja sesuai dengan fungsi dan tingkatanya. Setiap tingkatan *user* akan dapat menggunakan fitur sistem yang berbeda-beda sesuai dengan fungsi dari *user* itu sendiri. Terdapat 4 tingkat fungsi *user*, yaitu admin sistem, admin lembaga, ketua atau pimpinan lembaga, dan operator lembaga. Admin sistem dapat melakukan aktivasi layanan, manipulasi layanan, dan melihat pembayaran yang dilakukan oleh *user*. Admin lembaga atau perusahaan berfungsi untuk mendaftarkan perusahaan, hingga menambahkan *user* kepala atau pimpinan lembaga, dan juga mendaftarkan *user* operator lembaga, hingga *memonitoring* sumber daya sistem yang dibatasi oleh paket layanan, kemudian *user* kepala atau pimpinan lembaga dapat mengijinkan penamhan operator, mendisposisi surat masuk, membaca surat keluar, dan memanipulasi data keterangan tindak lanjut disposisi, lalu *user* operator lembaga berfungsi untuk memanipulasi administrasi surat masuk, dan surat keluar, kemudan mendapatkan disposisi dari pimpinan.

* + 1. Pencatatan administrasi surat menyurat.

Sistem dapat menyimpan data administrasi surat menyurat seperti surat masuk, surat keluar, dan disposisi surat dalam satu lembaga.

* + 1. Memanipulasi Layanan Sistem

*User* dapat meningkatkan atau menurunkan status penggunaan layanan sistem dengan fleksibel sesuai dengan kebutuhan perusahaan tanpa harus bertemu langsung dengan penyedia layanan.

* + 1. *Monitoring Data*

*User* dapat memonitoring jumlah penggunaan baris data yang dibatasi oleh sistem berdasarkan paket layanan yang dipilih agar dapat mengetahui informasi tentang penggunaan sistem.

* + 1. Melakukan *Backup Data*

*User* dapat mengunduh data yang sudah disimpan di basis datake direktori *user*, sehingga data yang disimpan dapat dikembalikan lagi.

* + 1. **Analisis Kebutuhan *Non* Fungsional**

Analisis kebutuhan non fungsional ini merupakan analisis kebutuhan yang menunjang dalam pengembangan dan penggunaan perangkat lunak yang akan dibuat. Hal ini diperlukan untuk menunjang agar maksimalnya pengembangan dan penggunaan perangkat lunak dalam penelitian ini. Kebutuhan pengembangan adalah hal-hal yang dibutuhkan dalam mengembangkan prangkat lunak. Berikut adalah hal-hal yang dibutuhkan dalam mengmbangkan prangkat lunak.

1. Unit Komputer

Dalam pengembangan prangkat lunak dibutuhkan perangkat keras yang memadai dengan spesifikasi yaitu Prosessor 1.4 GHz, RAM 2 GB, dan kartu grafis yang cukup agar dapat memenuhi standar aplikasi pendukung.

1. Sistem Operasi

Sistem Operasi yang digunakan adalah *Windows 10 Pro* agar mendukung kinerja aplikasi terbaru seperti *Adobe Photoshop CC* dan lain sebagainya.

1. Text Editor

Untuk penulisan kode program *PHP* dan *CSS* dibutuhkan sebuah text editor yang telah mendukung kode program tersebut sehingga dapat mempermudah penulisan kode program. Dalam hal ini digunakan text editor *Sublim Text Editor* versi 3.

1. Server Lokal

Pengembangan prangkat lunak pada penelitian ini digunakan *Apache* sebagai *Web Server* yang menjalankan *PHP* dan *MySQL* sebagai server basis data yang dijadikan dalam satu paket yaitu aplikasi *XAMPP*.

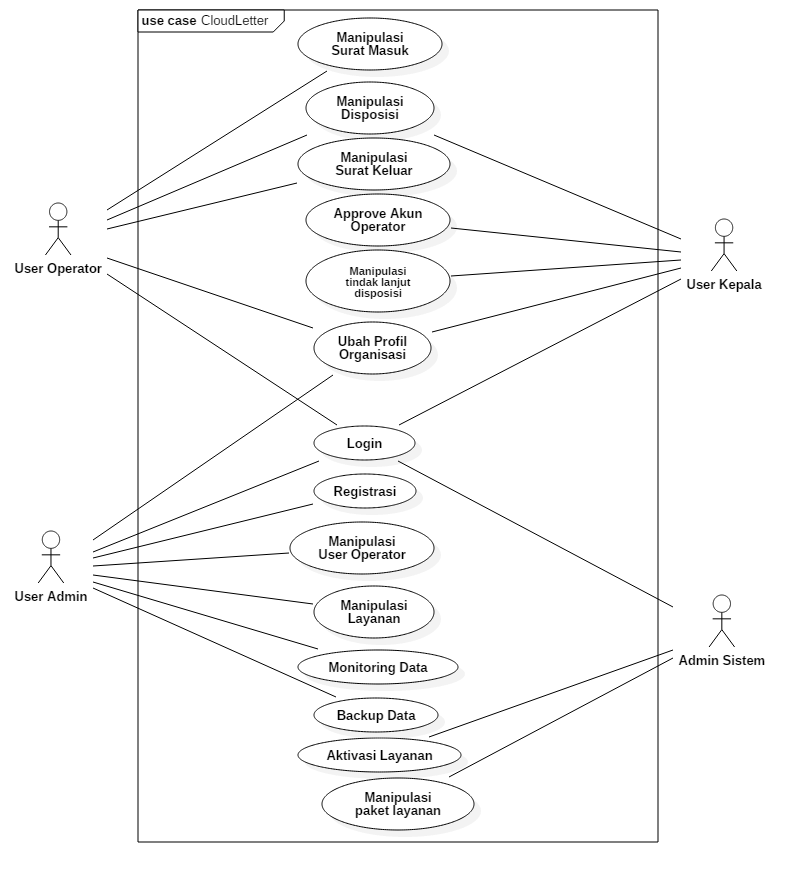
1. Aplikasi Pengolah Grafis

Dalam mendesain *user interface* dibutuhkan desain grafis yang baik oleh karena itu *Adobe Photoshop CC* digunakan untuk mengolah grafis yang mendukung desain *user interface*.

* 1. **Analisis Desain Sistem**

Pada tahapan ini dilakukan perancangan sistem berdasarkan atas hasil analisis kebutuhan yang telah dibahas sebelumnya pada bagian 3.1. analisis sistem dirancang untuk mengetahui proses bisnis dari perangkat lunak yang dibuat dengan menggunakan pemodelan berorientasi objek. Oleh karena itu untuk menggambarkan proses bisnis tersebut digambarkan menggunakan *UML* (*Undifined Modeling Language*) yang terdiri dari *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *Sequence Diagram*, serta dibuat juga Konseptual Basis data dan Struktur Tabel.

* + 1. ***Use Case* Diagram**

****Pada Gambar 3. 1 *Use Case Diagram* SaaS *Surat Menyurat* di atas menjelaskan tentang hubungan antara objek pengguna dan sistem dimana terdapat 4 aktor yang berhubungan langsung dengan sistem. Adapun ke empat aktor tersebut adalah admin sistem, *user* admin untuk admin perusahaan, *user* kepala untuk kepala perusahaan atau lembaga, dan *user* operator untuk operator sistem pada perusahaan atau lembaga. Dari diagram tersebut digambarkan fungsi masing masing *user*.

Gambar 3. 1 Use Case Diagram SaaS Surat Menyurat

1. *Use Case* Level Tinggi
   1. *Use Case* Login

*Use Case* ini menjelaskan tentang proses *login* pada sistem, berikut adalah *High Level Use Case* dari *Use Case* *Login* pada Tabel 3.1.

Tabel 3.1 *Use Case* Level Tinggi *Login*

|  |  |
| --- | --- |
| **Nama *Use Case*** | Login |
| **Aktor** | Admin Sistem, *User* Admin, *User* Kepala, *User* Operator |
| **Deskripsi** | Pengguna sistem dapat melakukan validasi menggunakan *username* dan *password* yang sudah ditentukan sebelumnya, sistem yang menangani adalah sistem Login. |

* 1. *Use Case* Memanipulasi Surat Masuk

*Use Case* ini menjelaskan tentang proses pencatatan Administrasi Surat Masuk, berikut adalah *High Level Use Case* dari *Use Case* Surat Masuk pada Tabel 3.2.

Tabel 3.2 *Use Case* Level TinggiSurat Masuk

|  |  |
| --- | --- |
| **Nama *Use Case*** | Memanipulasi Surat Masuk |
| **Aktor** | *User* Operator, *User* Kepala |
| **Deskripsi** | Operator melakukan input, ubah, dan hapus data administrasi surat masuk. |

* 1. *Use Case* Memanipulasi Disposisi

*Use Case* ini menjelaskan tentang proses disposisi surat masuk ke bagian yang berwenang dengan tinak lanjut yang berbeda. berikut adalah *High Level Use Case* dari *Use Case* Disposisi pada Tabel 3.3.

Tabel 3.3 *Use Case* Level Tinggi ManipulasiDisposisi

|  |  |
| --- | --- |
| **Nama *Use Case*** | Disposisi |
| **Aktor** | *User* Kepala, *User* Operator |
| **Deskripsi** | Kepala perusahaan mendisposisikan surat masuk yang telah dicatat ke dalam sistem oleh operator kemudian menentukan tujuan dan tindak lanjut dari surat masuk tersebut. |

* 1. *Use Case* Manipulasi Surat Keluar

*Use Case* ini menjelaskan tentang proses pencatatan administrasi surat surat keluar. berikut adalah *High Level Use Case* dari *Use Case* Surat Keluar pada Tabel 3.4.

Tabel 3.4 *Use Case* Level Surat Keluar

|  |  |
| --- | --- |
| **Nama *Use Case*** | Surat Keluar |
| **Aktor** | *User* Operator, *User* Kepala |
| **Deskripsi** | Operator mencatat dan memanipuasi data administrasi surat keluar. |

* 1. *Use Case* UbahProfil Organisasi

*Use Case* ini menjelaskan tentang proses *view* data profil organisasi. berikut adalah *High Level Use Case* dari *Use Case* Profil organisasi pada Tabel 3.5.

Tabel 3.5 *Use Case* Level TinggiUbah Profil Organisasi

|  |  |
| --- | --- |
| **Nama *Use Case*** | Profil Organisasi |
| **Aktor** | *User* Admin, *User* Operator, *User* Kepala |
| **Deskripsi** | Pengguna Sistem dapat melihat data dan mengubah profil organisasi. |

* 1. *Use Case* Registrasi

*Use Case* ini menjelaskan tentang proses pendaftaran data perusahaan dan registrasi akun admin pada lembaga/perusahaan yang didaftarkan tersebut serta akun ketua lembaga/perusahaan. berikut adalah *High Level Use Case* dari *Use Case* *Login* Registrasi Tabel 3.6.

Tabel 3.6 *Use Case* Level TinggiRegistrasi

|  |  |
| --- | --- |
| **Nama *Use Case*** | Registrasi |
| **Aktor** | *User* Admin |
| **Deskripsi** | *User* Admin mendaftarkan data organisasi dan registrasi akun admin pada lembaga atau organisasi yang didaftarkan tersebut serta akun ketua lembaga/perusahaan. |

* 1. *Use Case* Manipulasi Data Operator

*Use Case* ini menjelaskan tentang proses pendaftaran data akun operator. berikut adalah *High Level Use Case* dari *Use Case* Manipulasi Data Operator pada Tabel 3.7.

Tabel 3.7 *Use Case* Level TinggiManipulasi Data Operator

|  |  |
| --- | --- |
| **Nama *Use Case*** | Manipulasi Data Operator |
| **Aktor** | *User* Admin, *User* Kepala |
| **Deskripsi** | *User* admin mendaftarkan akun operator dan dapat juga memanipulasi akun operator dalam satu perusahaan. Pendaftaran akun operator juga harus melalui verifikasi kepala perusahaan/ lembaga. |

* 1. *Use Case* Memanipulasi Layanan

*Use Case* ini menjelaskan tentang proses pemilihan paket, dan pembayaran layanan yang dilakukan oleh user. berikut adalah *High Level Use Case* dari *Use Case* Manipulasi Layanan pada Tabel 3.8

Tabel 3.8 *Use Case* Level TinggiMemanipulasi Layanan

|  |  |
| --- | --- |
| **Nama *Use Case*** | Memanipulasi Layanan |
| **Aktor** | *User* Admin |
| **Deskripsi** | *User* Admin melakukan pemilihan paket layanan kemudan melakukan pembayaran yang nantinya akan diaktivasi layanan tersebut oleh admin sistem. |

* 1. *Use Case Data Monitoring*

*Use Case* ini menjelaskan tentang proses *view* berapa jumlah data atau sumber daya sistem yang telah digunakan oleh pengguna sistem sesuai dengan limit layanan yang digunakan. berikut adalah *High Level Use Case* dari *Use Case* *Data Monitoring* pada Tabel 3.9

Tabel 3.9 *Use Case* Level Tinggi *Data Monitoring*

|  |  |
| --- | --- |
| **Nama *Use Case*** | *Data Monitoring* |
| **Aktor** | *User* Admin |
| **Deskripsi** | Admin perusahaan atau lembaga melakukan pengamatan kepada sumber daya sistem yang telah digunakan sesuai dengan batasan penggunaan dari paket layanan yang dijalankan. |

* 1. *Use Case Backup Data*

*Use Case* ini menjelaskan tentang proses *backup data* agenda surat. berikut adalah *High Level Use Case* dari *Use Case* *Backup Data* pada Tabel 3.10.

Tabel 3.10 *Use Case* Level Tinggi *Backup Data.*

|  |  |
| --- | --- |
| **Nama *Use Case*** | *Backup Data* |
| **Aktor** | *User* Admin |
| **Deskripsi** | Admin perusahaan melakukan *backup data* administrasi surat. |

* 1. *Use Case* Level Tinggi Aktivasi Layanan

*Use Case* ini menjelaskan tentang proses Aktivasi Layanan. berikut adalah *High Level Use Case* dari *Use Case* Aktivasi Layanan pada Tabel 3.11.

Tabel 3.11 *Use Case* Level TinggiAktivasi Layanan

|  |  |
| --- | --- |
| **Nama *Use Case*** | Aktivasi Layanan |
| **Aktor** | Admin Sistem |
| **Deskripsi** | Admin sistem menampilkan data pembayaran yang dilakukan oleh pengguna, kemudian melakukan aktivasi paket layanan yang telah dibayarkan. |

* 1. *Use Case* Level Tinggi *Approve* *User* Operator

*Use Case* ini menjelaskan tentang proses *Approve* Akun Operator. berikut adalah *High Level Use Case* dari *Use Case* *Approve* Akun Operator pada Tabel 3.12.

Tabel 3.12 *Use Case* Level Tinggi *Approve* *Akun* Operator

|  |  |
| --- | --- |
| **Nama *Use Case*** | *Approve* Akun Operator |
| **Aktor** | *User* Kepala |
| **Deskripsi** | *User* kepala menyetujui akun operator yang didaftarkan oleh *user* admin. |

* 1. *Use Case* Level Tinggi Manipulasi Tindak Lanjut Disposisi

*Use Case* ini menjelaskan tentang proses Manipulasi Tindak Lanjut Disposisi. berikut adalah *High Level Use Case* dari *Use Case* Manipulasi Tindak Lanjut Disposisi Tabel 3.13.

Tabel 3.13 *Use Case* Level TinggiManipulasi Tindak Lanjut Disposisi

|  |  |
| --- | --- |
| **Nama *Use Case*** | Manipulasi Tindak Lanjut Disposisi |
| **Aktor** | *User* Kepala |
| **Deskripsi** | *User* kepala melakukan manipulasi keterangan tindak lanjut disposisi surat masuk. |

* 1. *Use Case* Level Tinggi Manipulasi Paket Layanan

*Use Case* ini menjelaskan tentang proses Manipulasi paket layanan. berikut adalah *High Level Use Case* dari *Use Case* Manipulasi Tindak Lanjut Disposisi Tabel 3.14.

Tabel 3.14 *Use Case* Level TinggiManipulasi Paket Layanan

|  |  |
| --- | --- |
| **Nama *Use Case*** | Manipulasi paket layanan |
| **Aktor** | Admin Sistem |
| **Deskripsi** | Admin sistem melakukan manipulasi terhadap paket layanan yang disediakan oleh sistem. |

1. *Use Case* Level Perluasan
2. *Use Case* Level Perluasan Login

Berdasarkan *Use Case Level* Tinggi Login pada Tabel 3.1 diatas dapat di jabarkan kembali melalui *Use Case* Level Perluasan Login. Penjabaran *Use Case* Level Perluasan *Login* dijelaskan pada Tabel 3.15.

Tabel 3.15 *Use Case* Level PerluasanLogin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* Login | |
| **Aktor**  **Aktor Tambahan** | | | | Admin sistem, User admin, User kepala, User Operator.  - | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1. | Use case dimulai saat admin/operator mengakses halama login | | | 2. | Sistem menampilkan form *username* dan *password*. |
| 3. | Pengguna Sistem memasukan *username* dan *password* pada form yang telah disediakan. | | | 4 | Sistem memvalidasi *username* dan *password* yang dimasukan lalu mencocokan ke basisdata. |
| . |  | | | 5 | Sistem memeriksa status akun apakah sudah sudah atau belum aktif. |
|  |  | | | 6 | Sistem memeriksa status level akun. |
|  |  | | | 7 | Sistem memeriksa aktivasi layanan akun |
|  |  | | | 8 | kemudian sistem mengarahkan ke halaman yang sesuai dengan level akun. |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **4a** | Jika *Username* dan atau *Password* yang dimasukan tidak benar atau tidak cocok maka akan muncul pesan error : *password* atau *username* salah. | | |
| **5a** | Jika status akun adalah tidak aktif maka muncul pesan error : akun tidak aktif. | | |
| **8a** | Jika aktivasi layanan adalah *false* maka sistem akan cek ketersediaan akun *user* Kepala, jika *user* kepala tidak ada maka *redirect* ke halaman daftar akun *user* Kepala | | |

1. *Use Case* Level Perluasan Manipulasi Surat Masuk

Berdasarkan *Use Case* level tinggimanipulasi surat masuk sistem pada Tabel 3.2 dapat di jabarkan kembali melalui *Use Case* level perluasan manipulasi surat masuk. Penjabaran *Use Case* level perluasan manipulasi surat masuk dijelaskan pada Tabel 3.16.

Tabel 3.16 *Use Case* Level Perluasan Manipulasi Surat Masuk

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* Manipulasi Surat Masuk | |
| **Aktor**  **Aktor Tambahan** | | | | *User* Operator, *User* Kepala  - | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1. | *Use case* dimulai saat *User* Operator / Kepala mengakses halaman surat masuk. | | | 2. | Sistem menampilkan tabel dengan konten data surat masuk yang telah diinputkan oleh *user*.  Sistem menampilkan tombol tambah surat, ubah surat, hapus surat. |
| 3. | *User* Operator mengakses *input* surat masuk. | | | 4. | Sistem mengarahkah kehalaman *tambah* data pengirim. |
| 5 | *User* Operator memasukan data pengirim. | | | 6 | Sistem memvalidasi kemudian menyimpan data pengirim. |
|  |  | | | 7 | Sistem mengarahkan ke *form* *input* surat masuk. |
| 8 | *User* Operator memasukan data surat masuk. | | | 9 | Sistem memvalidasi kemudan menyimpan data surat masuk. |
|  |  | | | 10 | Sistem mengarahkan kembali *user* ke halaman surat masuk. |
| 11 | *User* Operator mengakses ubah surat masuk | | | 12 | Sistem mengarahkan ke halaman *form* ubah surat masuk dengan membawa atribut id surat masuk yg akan di ubah. |
| 13 | *User* Operator melakukan ubah data surat masuk. | | | 14 | Sistem memvalidasi kemudian menyimpan data ubah surat masuk kedalam basis data. |
|  |  | | | 15 | Sistem mengarahkan kembali *user* ke halaman surat masuk. |
| 16 | *User* Operator mengakses hapus data surat masuk. | | | 17 | Sistem memunculkan dialog : apakah yakin ingin menghapus data surat, tampilkan *button* “ya” atau “tidak” |
| 18 | *User* Operator menekan tombol “ya” | | | 19 | Sistem menghapus data surat yang dipilih oleh user pada basis data. |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **2a** | Jika *user* adalah *user* kepala maka munculkan tombol disposisi surat. | | |
| **6a** | Jika data pengirim tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada *form* pengirim. | | |
| **9a** | Jika data surat masuk tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada *form* pengirim. | | |
| **14a** | Jika data ubah surat masuk tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada form pengirim. | | |
| **19a** | Jika user menekan tombol “tidak” maka data tidak akan dihapus pada basis data. | | |

1. *Use Case* Level Perluasan Manipulasi Disposisi

Berdasarkan *Use Case* level tinggi manipulasi disposisi pada Tabel 3.3 diatas dapat di jabarkan kembali melalui *Use Case* level perluasan manipulasi disposisi. Penjabaran *Use Case* Level Perluasan Manipulasi Disposisi dijelaskan pada Tabel 3.17.

Tabel 3.17 *Use Case* Level Perluasan Manipulasi Disposisi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* Disposisi | |
| **Aktor**  **Aktor Tambahan** | | | | *User* Kepala  - | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1. | *Use case* dimulai saat user kepala mengakses data surat masuk lalu menekan tombol ‘disposisi’ | | | 2. | Sistem menampilkan data detal surat dan form disposisi yang berisi tindak lajut, tujuan, tanggal dan *note*. |
| 3. | *User* kepala mengisikan data disposisi, kemudian klik submit | | | 4. | Sistem memvalidasi data disposisi surat masuk kemudian menyimpan data surat masuk kedalam basis data. |
|  |  | | | 5. | Sistem mengarahkan kehalaman surat masuk |
|  |  | | | 6. | Sistem memuncul pesan “data disposisi berhasil disimpan”. |
| 7. | Pengguna Sistem telah selesai melakukan disposisi surat masuk. | | |  |  |
| 8 | *User* kepala mengakses halaman disposisi | | | 9 | Sistem mengarahkan ke halaman disposisi memunculkan konten surat yang sudah didisposisi.  Sistem memunculkan tombol ubah, dan hapus data disposisi. |
| 10 | *User* kepala memilih tombol ubah disposisi | | | 11 | Sistem mengarahkan ke halaman *form* ubah disposisi dengan membawa atribut id surat masuk yg akan di ubah. |
| 12 | *User* kepala mengubah data disposisi. | | | 13 | Sistem memvalidasi kemudian menyimpan data ubah disposisi kedalam basis data. |
|  |  | | | 14 | Sistem mengarahkan kembali user ke halaman surat masuk. |
| 15 | *User* kepala memilih tombol hapus disposisi | | | 16 | Sistem memunculkan dialog : apakah yakin ingin menghapus data surat, tampilkan *button* ya atau tidak |
| 17 | *User* kepala menekan tombol “ya” | | | 18 | Sistem menghapus data surat yang dipilih oleh *user* pada basis data. |
| 18 | *User* kepala telah selesai memanipulasi data disposisi | | |  |  |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **4a** | Jika data disposisi surat masuk tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada form disposisi. | | |
| **13a** | Jika data ubah disposisi surat masuk tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada form disposisi. | | |
| **18a** | Jika user menekan tombol “tidak” maka data tidak akan dihapus pada basis data. | | |

1. *Use* *Case* Level Perluasan Manipulasi Surat Keluar

` Berdasarkan *Use Case* Level Tinggi Surat Keluar pada Tabel 3.4 diatas dapat di jabarkan kembali melalui *Use Case* Level Perluasan Manipulasi Surat Keluar. Penjabaran *Use Case* Level Perluasan Manipulasi Surat Keluar dijelaskan pada Tabel 3.18.

Tabel 3.18 *Use Case* Level Perluasan Surat Keluar

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* Manipulasi Surat Keluar | |
| **Aktor**  **Aktor Tambahan** | | | | *Use* Operator, User Kepala  - | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1. | *Use case* dimulai saat User Operator / Kepala mengakses halaman surat keluar. | | | 2. | Sistem menampilkan tabel dengan konten data surat keluar yang telah diinputkan oleh *user*.  Sistem menampilkan tombol tambah, ubah, hapus surat keluar. |
| 3. | *User* operator memilih input surat keluar. | | | 4. | Sistem mengarahkan ke halaman tambah surat masuk. |
| 5 | *User* operator menginput data surat masuk. | | | 6 | Sistem memvalidasi kemudian menyimpan data surat keluar. |
|  |  | | | 7 | Sistem mengarahkan kembali user ke halaman surat keluar. |
| 8 | *User* operator memilih ubah surat masuk | | | 9 | Sistem mengarahkan ke halaman form ubah surat keluar dengan membawa atribut id surat keluar yg akan di ubah. |
| 10 | *User* operator melakukan ubah surat keluar. | | | 11 | Sistem memvalidasi kemudian menyimpan data ubah surat keluar kedalam basis data. |
|  |  | | | 12 | Sistem mengarahkan kembali *user* ke halaman surat keluar. |
| 13 | *User* operator memilih hapus surat masuk. | | | 14 | Sistem memunculkan dialog : apakah yakin ingin menghapus data surat, tampilkan *button* “ya” atau “tidak” |
| 15 | *User* Operator menekan tombol “ya”. | | | 16 | Sistem menghapus data surat yang dipilih oleh user pada basis data. |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **2a** | Jika *user* adalah user kepala maka munculkan tombol detail surat keluar. | | |
| **6a** | Jika data surat masuk tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada form pengirim. | | |
| **11a** | Jika data ubah surat masuk tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada form pengirim. | | |
| **16a** | Jika user menekan tombol “tidak” maka data tidak akan dihapus pada basis data. | | |

1. *Use Case* Level Perluasan Ubah Profil Organisasi

Berdasarkan *Use Case* Level Tinggi Profil Organisasi pada Tabel 3.5 dapat dijabarkan kembali melalui *Use Case* Level Ubah Profil Organisasi. Penjabaran *Use Case* Level Perluasan Ubah Profil Perusahaan dijelaskan pada Tabel 3.19.

Tabel 3.19 *Use Case* Level Perluasan Ubah Profil Organisasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* Ubah Profil Organisasi | |
| **Aktor**  **Aktor Tambahan** | | | | *User* Operator, User Kepala, User Admin  - | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1 | *Use Case* dimulai ketika *user* admin / operator / kepala mengakses halaman profil organisasi | | | 2 | Sistem menampilkan data perusahaan.  Sistem menampilkan tombol ubah profil organisasi. |
| 3 | *User* Operator / Admin / Kepala mengakses form ‘ubah data organisasi. | | | 4 | Sistem menampilkan form ubah data perusahaan sesuai dengan data yang sudah dibawa dari atribut id organisasi. |
| 5 | *User* Operator mengubah data perusahaan kemudian klik tombol simpan | | | 6 | Sistem memvalidasi ubah data perusahaan kemudian menyimpan ubah data perusahaan kedalam basis data. |
|  |  | | | 7 | Sistem mengarahkan kehalaman Profil Perusahaan. |
|  |  | | | 8 | Sistem memuncul pesan “data surat berhasil diubah”. |
| 9 | Pengguna Sistem telah selesai melakukan *input* data Ubah Profil organisasi. | | |  |  |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **6a** | Jika data ubah perusahaan tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada form ubah surat masuk. | | |
| **6b** | Jika data ubah perusahaan tidak dapat tersimpan pada basis data maka munculkan pesan eror : “data gagal tersimpan pada basis data”. | | |

1. *Use Case* Level Perluasan Registrasi

Berdasarkan *Use Case* Level Tinggi Registrasi pada Tabel 3.6 dapat di jabarkan kembali melalui *Use Case* Level Perluasan Registrasi. Penjabaran *Use Case* Level Perluasan Registrasi dijelaskan pada Tabel 3.20.

Tabel 3.20 *Use Case* Level Perluasan Registrasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* Registrasi | |
| **Aktor** | | | | *User* Admin | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1. | *Use case* dimulai saat *User* Admin mengakses menu registrasi | | | 2. | Sistem menampilkan form input data perusahaan. |
| 3. | *User* admin mengisikan data-data organisasi sesuai dengan form yang ditampilkan kemudian klik simpan | | | 4. | Sistem memvalidasi data kemudian menyimpan data surat masuk kedalam basis data. |
|  |  | | | 5. | Sistem mengarahkan ke halaman Registrasi akun *User* level admin |
|  |  | | | 6. | Sistem menampilkan form input data akun *User* Admin. |
| 7. | *User* Admin mengisikan data- data akun *username*, *password*, dan *email*. | | | 8. | Sistem memvalidasi data akun *user* admin kemudian menyimpan data surat masuk kedalam basis data. |
|  |  | | | 9. | Sistem mengarahkan ke halaman registrasi akun kepala organisasi. |
|  |  | | | 10 | Sistem menampilkan form registrasi akun kepala organisasi |
| 11 | *User* Admin mengisikan data-data akun kepala. *Username*, *password*, dan *email*. | | | 12 | Sistem memvalidasi data akun *user* kepala kemudian menyimpan data surat masuk kedalam basis data. |
|  |  | | | 13 | Sistem mengarahkan ke *dashboard* *user* admin. |
| 14 | *User* Admin telah berhasil melakukan registrasi sistem. | | |  |  |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **4a** | Jika data organisasi tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada form input data organisasi. | | |
| **4b** | Jika data organisasi tidak dapat tersimpan pada basis data maka munculkan pesan eror : “data gagal tersimpan pada basis data”. | | |
| **8a** | Jika data akun *User* admin tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada form input data akun admin. | | |
| **8b** | Jika data akun *User* admin tidak dapat tersimpan pada basis data maka munculkan pesan eror : “data gagal tersimpan pada basis data”. | | |
| **12a** | Jika data akun *User* kepala perusahaan tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada form input data akun kepala. | | |
| **12b** | Jika data akun kepala perusahaan tidak dapat tersimpan pada basis data maka munculkan pesan eror : “data gagal tersimpan pada basis data”. | | |

1. *Use Case* Level Perluasan Memanipulas *User* Operator

Berdasarkan *Use Case* Level Tinggi Manipulasi User Operator pada Tabel 3.7 diatas dapat di jabarkan kembali melalui *Use Case* Level Manipulasi User Operator Perusahaan. Pada *Use Case* Level Perluasan Manipulasi User Operator dijelaskan pada Tabel 3.21.

Tabel 3.21 *Use Case* Level Perluasan Manipulasi *User* Operator

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* Manipulasi *User* Operator | |
| **Aktor**  **Aktor Tambahan** | | | | *User* Admin  - | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1. | *Use case* dimulai saat *User* Admin / Kepala mengakses halaman data *User* Operator. | | | 2. | Sistem menampilkan tabel dengan konten data *user* operator yang telah diinputkan oleh *user*.  Sistem menampilkan tombol tambah, ubah, dan hapus data *User* Operator. |
| 3. | *User* Operator / Kepala memilih tambah *user* operator | | | 4. | Sistem menampilkan form pendaftaran *user* operator. |
| 5 | *User* admin memasukan data Akun operator pada form input data Operator. *User* klik simpan | | | 6 | Sistem memvalidasi data akun operator kemudian menyimpan data akun masuk kedalam basis data. |
| 7 |  | | | 8 | Sistem mengarahkan kehalaman Tampil Data akun operator. |
| 9 | *User* Admin mengakses form ubah data Akun Operator. | | | 10 | Sistem menampilkan form ubah data akun operator sesuai dengan data yang sudah dibawa dari atribut id akun. |
| 11 | *User* Admin mengubah data akun operator kemudian klik tombol simpan | | | 12 | Sistem memvalidasi ubah akun operator kemudian menyimpan ubah data akun kedalam basis data. |
|  |  | | | 13 | Sistem mengarahkan kehalaman Tampil data akun *user* operator. |
| 14 | *User* admin memilih tombol hapus akun operator. | | | 15 | Sistem memunculkan dialog : apakah yakin ingin menghapus akun operator, tampilkan *button* “ya” atau “tidak” |
| 16 | *User* Operator menekan tombol “ya”. | | | 17 | Sistem menghapus data surat yang dipilih oleh *user* pada basis data. |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **6a** | Jika data akun operator tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada form pengirim. | | |
| **12a** | Jika data ubah akun operator tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada form pengirim. | | |
| **17a** | Jika user menekan tombol “tidak” maka data tidak akan dihapus pada basis data. | | |

1. *Use Case* Level Perluasan ManipulasiLayanan

Berdasarkan *Use Case* Level Tinggi ManipulasiLayanan pada Tabel 3.8 dapat dijabarkan kembali melalui *Use Case* Level Perluasan ManipulasiLayanan. *Use Case* Level Perluasan ManipulasiLayanan dijelaskan pada Tabel 3.22.

Tabel 3.22 *Use Case* Level Perluasan ManipulasiLayanan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* *Upgrade* Layanan | |
| **Aktor**  **Aktor Tambahan** | | | | *User* Admin, Admin Sistem  - | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1. | *Use case* dimulai saat *User* Admin mengakses menu Tingkatkan layanan Sistem. | | | 2. | Sistem menampilkan informasi harga dan paket Layanan Sistem yang tersedia. |
| 3. | *User* Admin memilih paket layanan yang diingin kan | | | 4. | Sistem mengarahkan ke halaman pilih lama langganan.  Sistem memunculkan form pilih lama langganan. |
| 5. | *User* Admin memilih lama langganan. lalu Klik submit | | | 6 | Sistem memunculkan detail pembayaran |
| 7. | *User* Admin memeriksa kembali detail pembayaran dan kemudian klik simpan pembayaran | | | 8 | Sistem memvalidasi data pembayaran kemudian menyimpan ubah data akun kedalam basis data. |
|  |  | | | 9 | Sistem mengarahkan ke tampilan info pembayaran. |
|  |  | | | 10 | Sistem menampilkan pesan “ data pembayaran telah disimpan “. |
| 11 | *User* Admin telah berhasil melakukan pembayaran paket layanan. | | |  |  |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **8a** | Jika data pembayaran tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada tampilan detail pembayaran. | | |
| **8b** | Jika data pembayaran tidak dapat tersimpan pada basis data maka munculkan pesan eror : “data gagal tersimpan pada basis data”. | | |

1. *Use Case* Level Perluasan *Monitoring Data*

Berdasarkan *Use Case* Level Tinggi *Monitoring Data* pada Tabel 3.9 diatas dapat di jabarkan kembali melalui *Use Case* Level Perluasan *Monitoring Data*. Penjabaran *Use Case* Level Perluasan *Monitoring Data* dijelaskan pada Tabel 3.23.

Tabel 3.23 *Use Case* Level Perluasan *Monitoring Data*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* *Monitoring Data* | |
| **Aktor**  **Aktor Tambahan** | | | | *User* Admin  - | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1. | *Use case* dimulai saat *User* Admin mengakses halaman data *Monitoring Data*. | | | 2. | Sistem menampilkan informasi penggunaan sumber daya sistem sesuai dengan layanan yang digunakan dalam bentuk *progress bar*. |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **-** | - | | |

1. *Use Case* Level Perluasan *Backup Data*

Berdasarkan *Use Case* Level Tinggi *Backup Data* pada Tabel 3.10 diatas dapat di jabarkan kembali melalui *Use Case* Level Perluasan *Backup Data*. *Use Case* Level Perluasan *Backup Data* dijelaskan pada Tabel 3.24.

Tabel 3.24 *Use Case* Level Perluasan *Backup Data*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* *Backup Data* | |
| **Aktor**  **Aktor Tambahan** | | | | *User* Admin  - | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1. | *Use case* dimulai saat User Admin mengakses halaman data *Backup Data*. | | | 2. | Sistem menampilkan halaman *backup data* dengan tombol eksport surat masuk, surat keluar, dan disposisi |
| 3. | *User* Admin menekan tombol eksport surat masuk | | | 4 | Sistem mengolah data surat masuk pada basis data lalu disimpan dan di unduh ke direktori *user*. |
| 5 | *User* admin menekan tombol eksport surat keluar | | | 6 | Sistem mengolah data surat keluar pada basis data lalu disimpan dan di unduh ke direktori *user*. |
| 6 | *User* admin menekan tombol eksport disposisi | | | 7 | Sistem mengolah data disposisi pada basis data lalu disimpan dan di unduh ke direktori *user*. |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **-** | - | | |

1. *Use Case* Level Perluasan Aktivasi Layanan

Berdasarkan *Use Case* Level Tinggi Aktivasi Layanan pada Tabel 3.10 diatas dapat di jabarkan kembali melalui *Use Case* Level Perluasan Aktivasi Layanan. Penjabaran *Use Case* Level Perluasan Aktivasi Layanan dijelaskan pada Tabel 3.25.

Tabel 3.25 *Use Case* Level Perluasan Aktivasi Layanan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* Aktivasi Layanan | |
| **Aktor**  **Aktor Tambahan** | | | | Admin Sistem  - | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1. | *Use case* dimulai saat admin sistem mengakses halaman data pembayaran | | | 2. | Sistem menampilkan tabel dengan konten data pembayaran yang belum teraktivasi. Sistem menampilkan tombol detail pembayaran |
| 3 | Admin sistem menekan tombol aktivasi layanan | | | 4 | Sistem merubah status layanan dari seluruh user dalam satu perusahaan yang melakukan pembayaran sesuai dengan paket layanan yang dipilih. |
|  |  | | | 5 | Sistem mengarahkan kembali ke halaman data pembayaran |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **-** | - | | |

1. *Use Case* Level Perluasan *Approve* *User* Operator

Berdasarkan *Use Case* Level Tinggi *Approve* User Operator pada Tabel 3.12 diatas dapat di jabarkan kembali melalui *Use Case* Level Perluasan Approve User Operator. Penjabaran *Use Case* Level Perluasan Approve User Operator dijelaskan pada Tabel 3.26.

Tabel 3.26 *Use Case* Level Perluasan *Approve* *User* Operator

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* Approve *User* Operator | |
| **Aktor**  **Aktor Tambahan** | | | | *User* Kepala  - | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1 | *User* kepala mengakses halaman manipulasi user. | | | 2 | Sistem menampilkan data *user* operator yang belum mendapat persetujuan oleh user kepala.  Sistem menampilkan tombol terima, tolak akun operator. |
| 3 | *User* kepala memilih tombol terima atau tolak | | | 4 | Sistem mengubah status akun *user* operator menjadi aktif. |
|  |  | | | 5 | Sistem mengarahkan ke halaman manipulasi *user*. |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **4a** | Jika *user* kepala memilih tombol tolak, maka sistem akan mengubah status akun *user* operator menjadi ditolak. | | |

1. *Use Case* Level Perluasan Manipulasi Tindak Lanjut Disposisi

Berdasarkan *Use Case* Level Tinggi Tindak Lanjut pada Tabel 3.13 diatas dapat di jabarkan kembali melalui *Use Case* Level Perluasan Tindak lanjut Disposisi. Penjabaran *Use Case* Level Perluasan Manipulasi Tindak lanjut Disposisi dijelaskan pada Tabel 3.27.

Tabel 3.27 *Use Case* Level Perluasan Manipulasi Tindak lanjut Disposisi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* Tindak Lanjut Disposisi | |
| **Aktor**  **Aktor Tambahan** | | | | *User* Kepala  - | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1. | *User* kepala menekan tombol input tindak lanjut, untuk menambah data keterangan tindak lanjut disposisi surat masuk. | | | 2. | Sistem memvalidasi ubah data disposisi surat masuk kemudian menyimpan data surat masuk kedalam basis data. |
| 3 | *User* mengisi dan menyimpan data keterangan tindak lanjut disposisi surat masuk | | | 4 | Sistem memvalidasi data keterangan tindak lanjut disposisi surat masuk kemudian menyimpan data surat masuk kedalam basis data. |
|  |  | | | 5 | Sistem mengarahkan kehalaman data keterangan tindak lanjut disposisi surat masuk |
|  |  | | | 6 | Sistem memuncul pesan “data tindak lanjut disposisi berhasil disimpan”. |
| 7 | *User* kepala menekan tombol hapus. | | | 8 | Sistem memunculkan pesan dialog. |
| 8 | *User* kepala menekan tombol dialog | | | 10 | Sistem menghapus tindak lanjut disposisi |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **4a** | Jika data tindak lanjut disposisi surat masuk tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada form input tindak lanjut disposisi. | | |
| **4b** | Jika data tindak lanjut disposisi surat masuk tidak dapat disimpan pada basis data maka munculkan pesan eror : “data gagal disimpan pada basis data”. | | |

1. *Use Case* Level Perluasan Manipulasi layanan paket

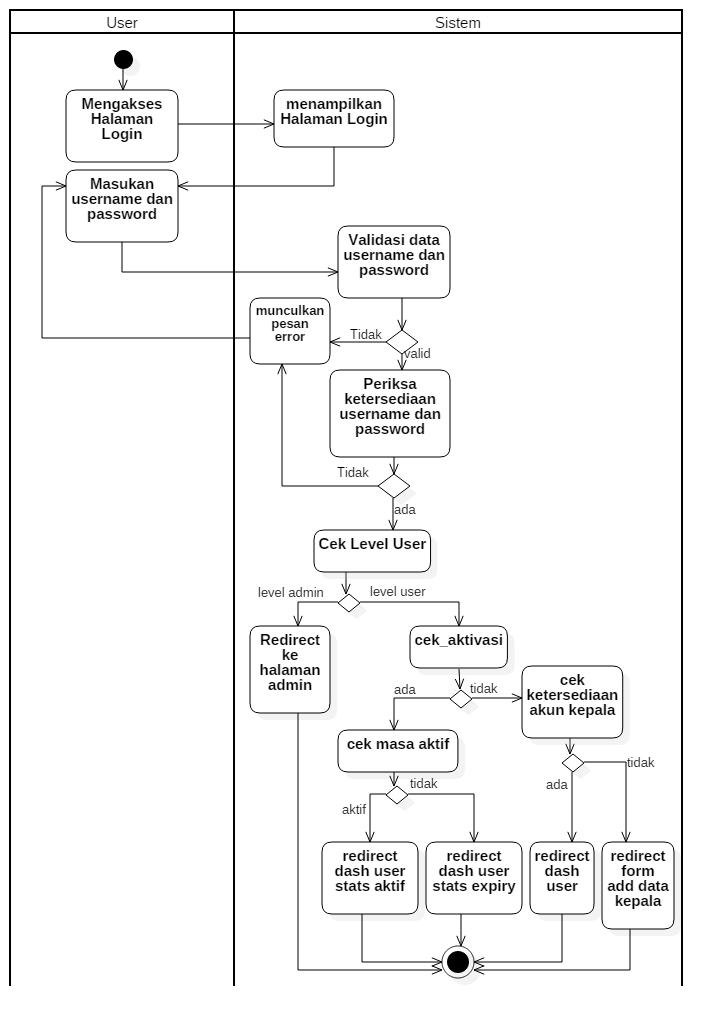
Berdasarkan *Use Case* Level Tinggi Manipulasi layanan pada Tabel 3.14 diatas dapat di jabarkan kembali melalui *Use Case* Level Perluasan Tindak lanjut Disposisi. Penjabaran *Use Case* Level Perluasan Tindak lanjut Disposisi dijelaskan pada Tabel 3.28.

Tabel 3.28 *Use Case* Level Perluasan Manipulasi Paket Layanan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nama *Use Case*** | | | | *Use Case* manipulasi layanan paket | |
| **Aktor**  **Aktor Tambahan** | | | | Admin sistem  - | |
| **Rangkaian Kejadian Khusus** | | | | | |
| Aktor | | | | Sistem | |
| 1. | Admin sistem mengakses halaman layanan | | | 2. | sistem menampilkan data paket layanan  sistem menampilkan tombol tambah, ubah, hapus paket layanan |
| 3 | Admin sistem memilih tombol Tambah layanan | | | 4 | Sistem menampilkan form tambah layanan |
| 5 | Admin sistem menginput data layanan | | | 6 | Sistem memvalidasi kemudian menyimpan data kedalam basis data. |
| 7 | Admin sistem memilih tombol ubah layanan | | | 8 | Sistem menampilkan form ubah paket layanan yang berisikan data paket layanan yang dipilih |
| 9 | Admin sistem mengubah data paket layanan | | | 10 | Sistem memvalidasi data ubah paket layanan. Sistem menyimpan kedalama basis data. |
| **Rangkaian Kejadian Alternatif** | | | | | |
| **Nomor** | | **4a** | Jika data paket layanan tidak valid maka akan muncul pesan peringatan “ data tidak valid ” pada form input paket layanan. | | |
| **4b** | Jika data ubah paket layanan valid maka maka akan muncul pesan peringatan “ data tidak valid ” pada form ubah paket layanan. | | |

* + 1. **Activity Diagram**
    2. *Activity Diagram Login*

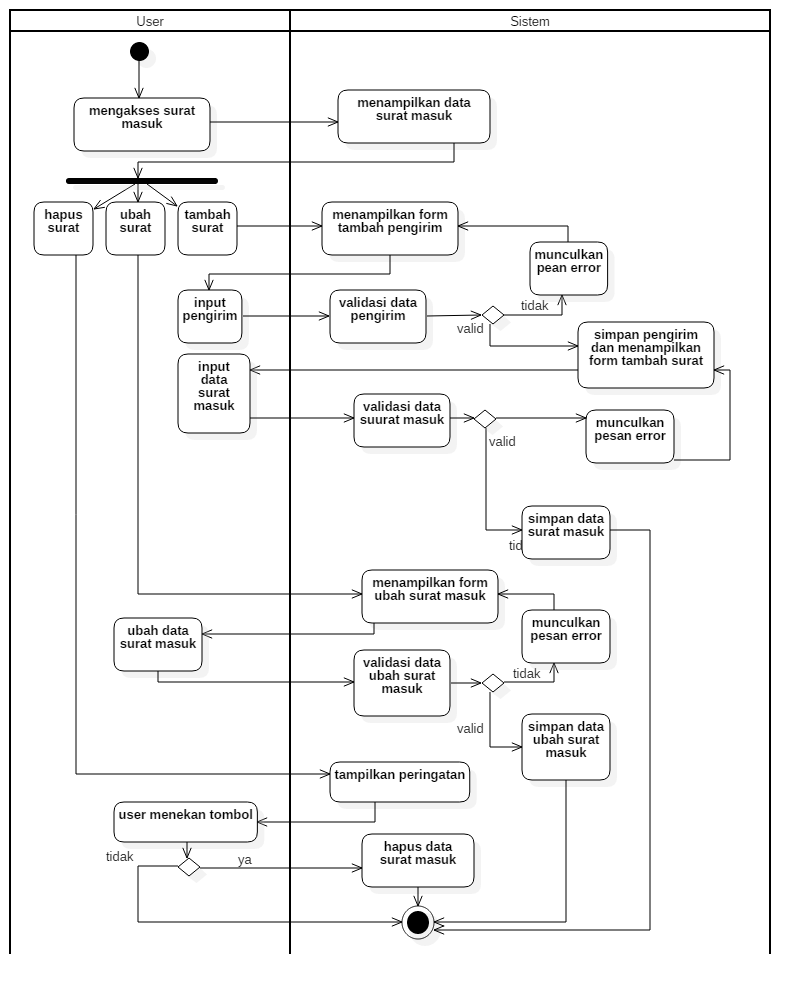
Pada *activity diagram* yang ditunjukan pada Gambar 3.2 ditampilkan proses login, dimana admin dan pengguna digambarkan memasukan *username* dan *password* kemudian sistem akan memvalidasi *username* dan *password* tersebut apakah form sudah terisi atau belum, jika iya sistem akan mencocokan ke basis data dan kemudian mengarahkan ke halaman sesuai dengan level pengguna yang telah ditentukan. Digambarkan juga sistem yang melakukan cek aktivasi pengguna.



Gambar 3.2 Activity Diagram Login

* + 1. *Activity Diagram* Manipulasi Data Surat Masuk

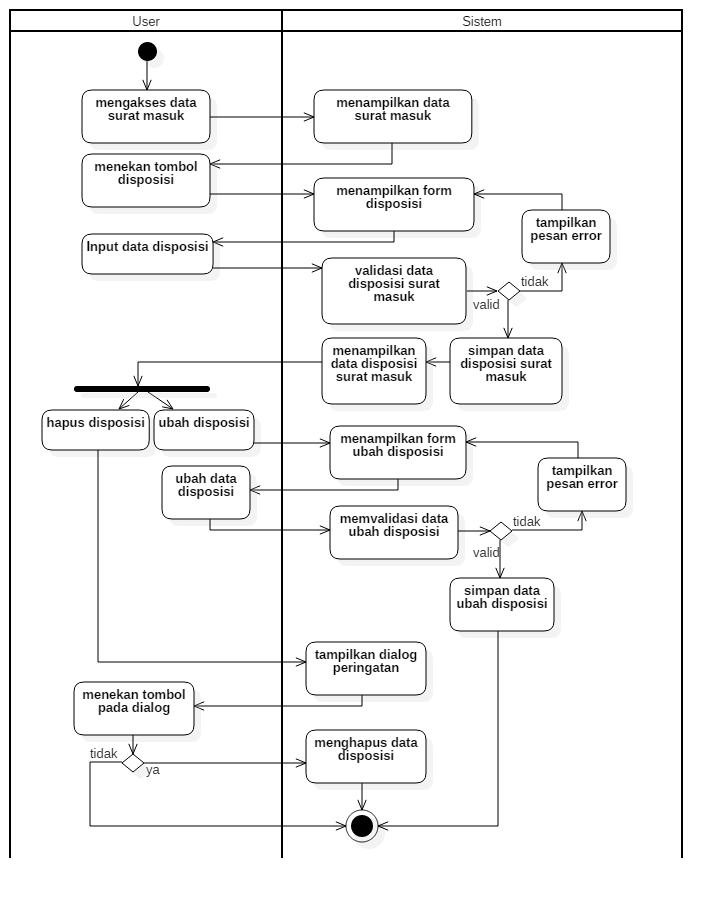
Pada *activity diagram* yang ditunjukan pada Gambar 3.3 Activity Diagram *Manipulasi Surat Masuk* merupakan activity *diagram* proses manipulasi data surat masuk dimana pada *activity diagram* tersebut menjelaskan tentang aktivitas manipulasi surat masuk.



Gambar 3.3 Activity Diagram Manipulasi Surat Masuk

* + 1. *Activity Diagram* Manipulasi Data Disposisi

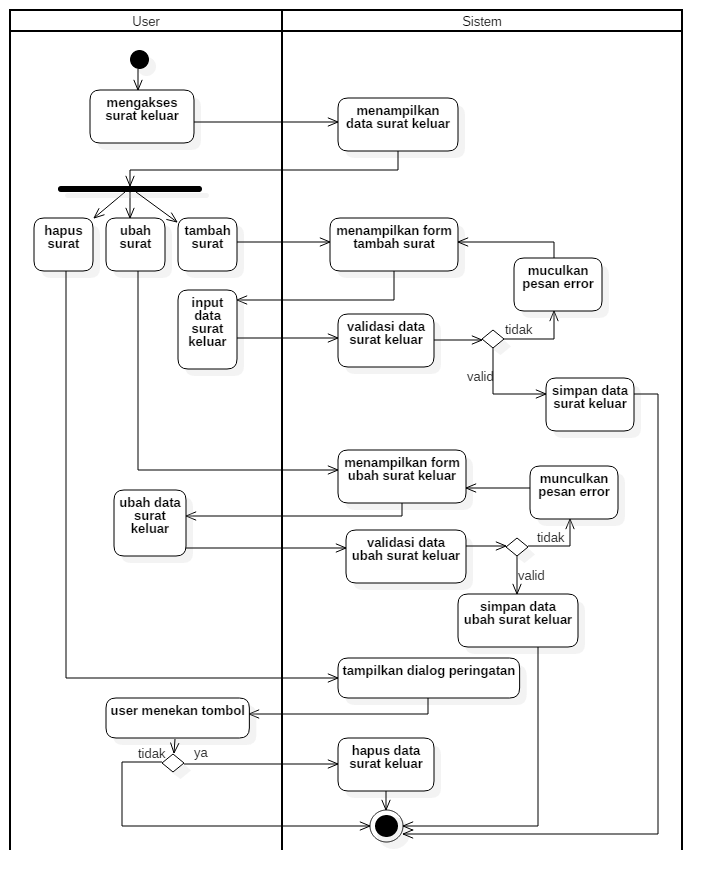
Pada *activity diagram* yang ditunjukan pada Gambar 3.4 merupakan activity *diagram* proses manipulasi data disposisi dimana pada *activity diagram* tersebut menjelaskan tentang aktivitas manipulasi disposisi yang dilakukan *user* admin.



Gambar 3.4 Activity Diagram Manipulasi Disposisi.

* + 1. *Activity Diagram* Manipulasi Surat Keluar

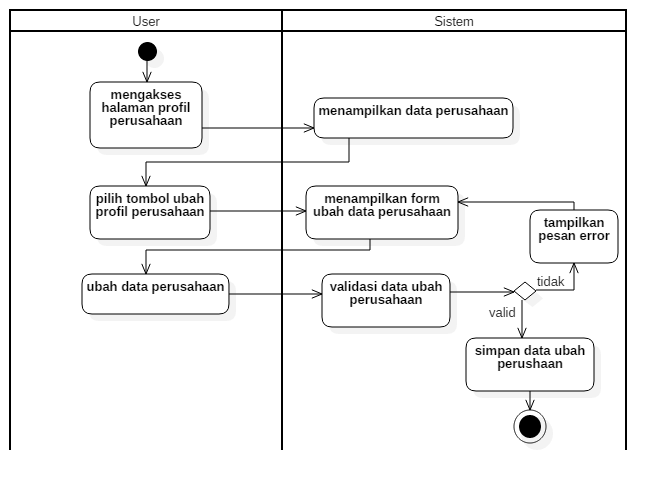
Pada *activity diagram* yang ditunjukan pada Gambar 3.5 merupakan activity *diagram* proses manipulasi data surat keluar dimana pada *activity diagram* tersebut menjelaskan tentang aktivitas manipulasi surat keluar yang dilakukan *user* operator.



Gambar 3.5 Activity Diagram Manipulasi Surat Keluar

1. *Activity Diagram* Ubah Profil Organisasi

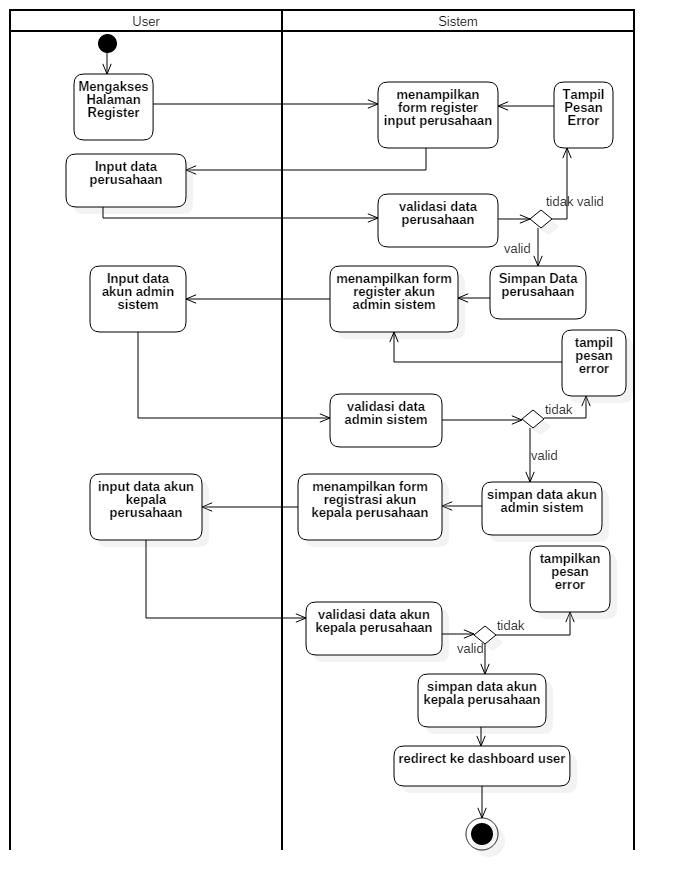
Pada *Activity Diagram* yang ditunjukan pada Gambar 3.6 merupakan activity *diagram* proses ubah profil organisasi dimana pada diagram tersbut menjelaskan tentang aktivitas *user* yang melakukan ubah profil perusahaan.



Gambar 3.6 Activity Diagram Ubah Profil Organisasi

1. *Activity Diagram* Proses Registrasi Organisasi

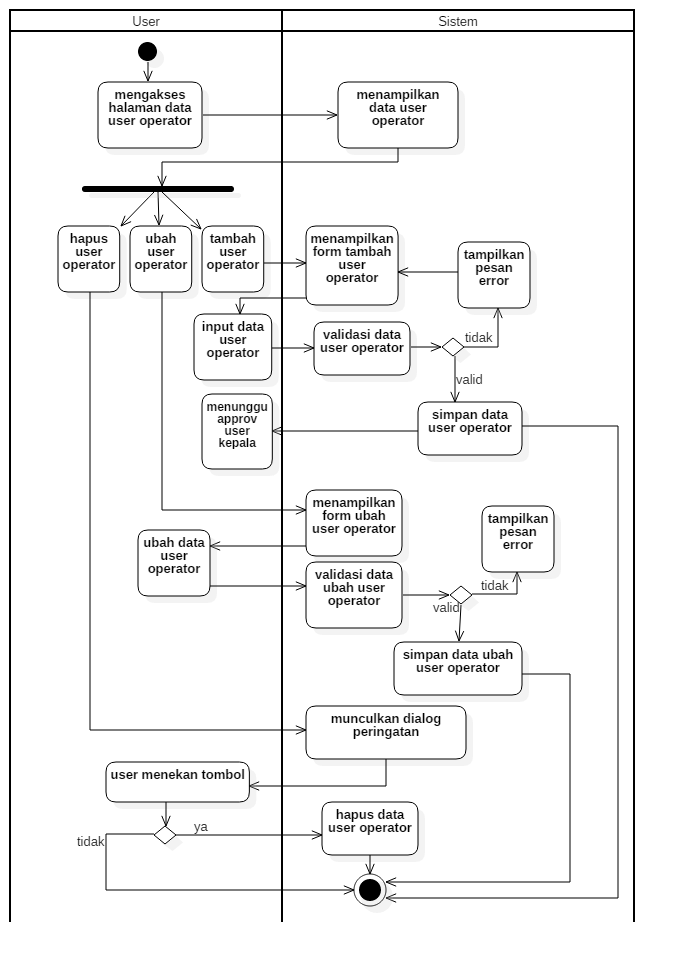
Pada *activity diagram* yang ditunjukan pada Gambar 3.7 merupakan activity *diagram* proses registrasi organisasi dimana pada diagram tersebut menjelaskan tentang aktivitas user yang melakukan regristrasi ke sistem yang akan digunakan. Proses tersebut meliputi proses registrasi organisasi, registrasi akun *user* admin, registrasi akun *user* kepala perusahaan.



Gambar 3.7 Activity Diagram Registrasi

1. *Activity Diagram* Manipulasi Data Operator

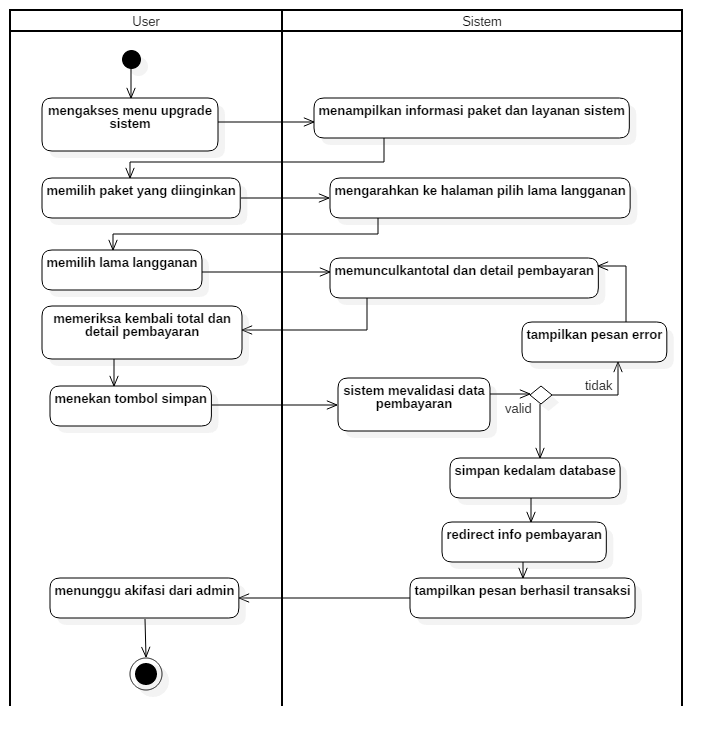
Pada *activity diagram* yang ditunjukan pada Gambar 3.8 merupakan activity *diagram* proses Manipulasi data operator dimana pada diagram tersebut menjelaskan tentang aktivitas user yang melakukan tambah data operator, ubah data operator, dan hapus data operator.



Gambar 3.8 Activity Diagram Manipulasi Operator

1. *Activity Diagram* Manipulasi Layanan

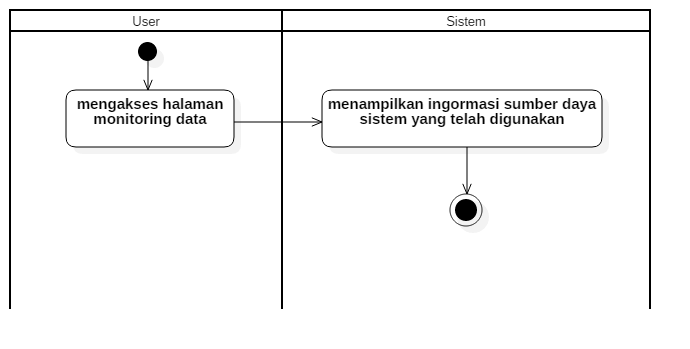
Pada *activity diagram* yang ditunjukan pada Gambar 3.9 merupakan activity *diagram* proses manipulasi layanan dimana pada diagram tersebut menjelaskan tentang aktivitas *user* yang melakukan meningkatkan langganan sistem dimana *user* melakukan pembayaran sesuai dengan paket layanan yang dipilih.



Gambar 3.9 Activity Diagram Manipulasi Layanan

1. *Activity Diagram* *Monitoring Data*

Pada *activity diagram* yang ditunjukan pada Gambar 3.10 merupakan activity *diagram* proses *monitoring* *data* dimana pada diagram tersebut menjelaskan tentang aktivitas *user* melakukan *monitoring* *data* agar mengetahui seberapa banyak *user* telah menggunakan suber daya sistem sesuai dengan paket layanan yang dimana *user* telah berlangganan paket layanan tersebut.



Gambar 3. 10 Activity Diagram Monitoring Data

1. *Activity Diagram Backup Data*

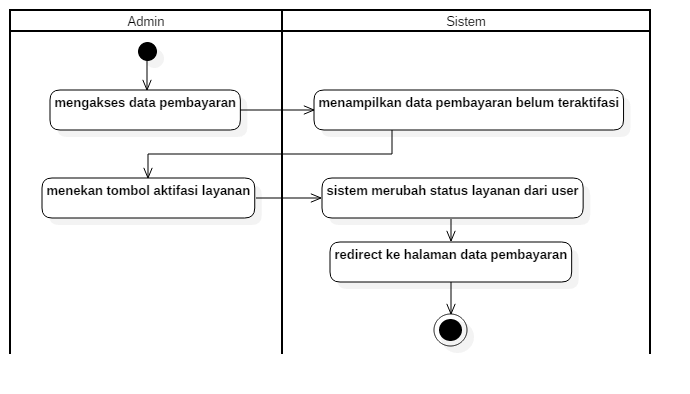
Pada *activity diagram* yang ditunjukan pada Gambar 3.11 merupakan activity *diagram* proses *backup* *data* dimana pada diagram tersebut menjelaskan tentang aktivitas *user* yang melakukan *backup* *data* surat masuk, surat keluar, dan disposisi dengan bermaksud menyiman data ketika hendak menurunkan fungsi user atau telah habis masa berlangganan paket layanan.



Gambar 3.11 Activity Diagram Backup Data

1. *Activity Diagram* Aktivasi Layanan

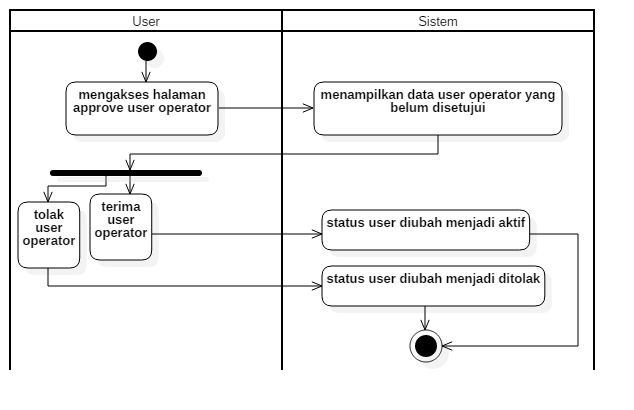
Pada *activity diagram* yang ditunjukan pada Gambar 3.12 merupakan activity *diagram* proses aktivasi layanan pada perusahaan dimana pada diagram tersebut menjelaskan tentang aktivitas *administrator* sistem yang melakukan aktivasi paket layanan yang telah dibayarkan oleh *user*.



Gambar 3.12 Activity Diagram Aktivasi Layanan

1. *Activity Diagram* *Approve* Akun Operator

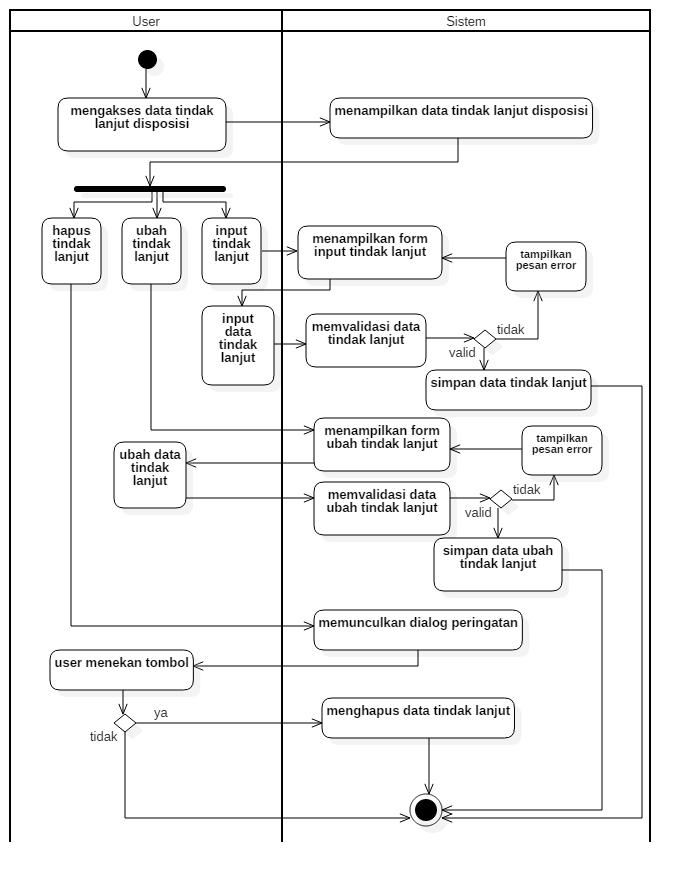
Pada *activity diagram* yang ditunjukan pada Gambar 3.13 merupakan activity *diagram* proses *approve* akun operator dimana pada diagram tersebut menjelaskan tentang aktivitas *user* kepala yang melakukan *approve* atau persetujuan dalam menambahkan akun operator yang akan dibuat. Approve akun operator berkaitan dengan manipulasi akun operator yang dilakukan oleh user admin dimana saat *user* admin menambahkan akun operator maka diperlukan persetujuan oleh *user* kepala lembaga atau perusahaan.



Gambar 3.13 Activity Diagram Approve Akun Operator

1. *Activity Diagram* Manipulasi Tindak Lanjut Disposisi

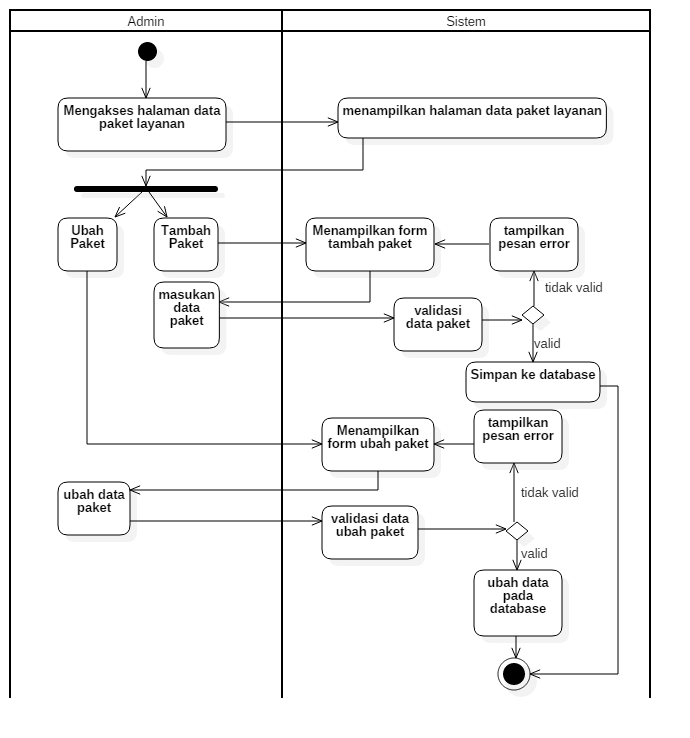
Pada *activity diagram* yang ditunjukan pada Gambar 3.14 merupakan activity *diagram* proses manipulasi tindak lanjut disposisi dimana pada diagram tersebut menjelaskan tentang aktivitas *user* kepala yang melakukan input, dan hapus data tindak lanjut disposisi surat masuk. Adanya manipulasi tindak lanjut agar tindak lanjut surat masuk lebih dinamis sesuai dengan organisasi yang menggunakan sistem.



Gambar 3.14 Activity Diagram Manipulasi Tindak Lanjut Disposisi

1. *Activity Diagram* Manipulasi Paket Layanan

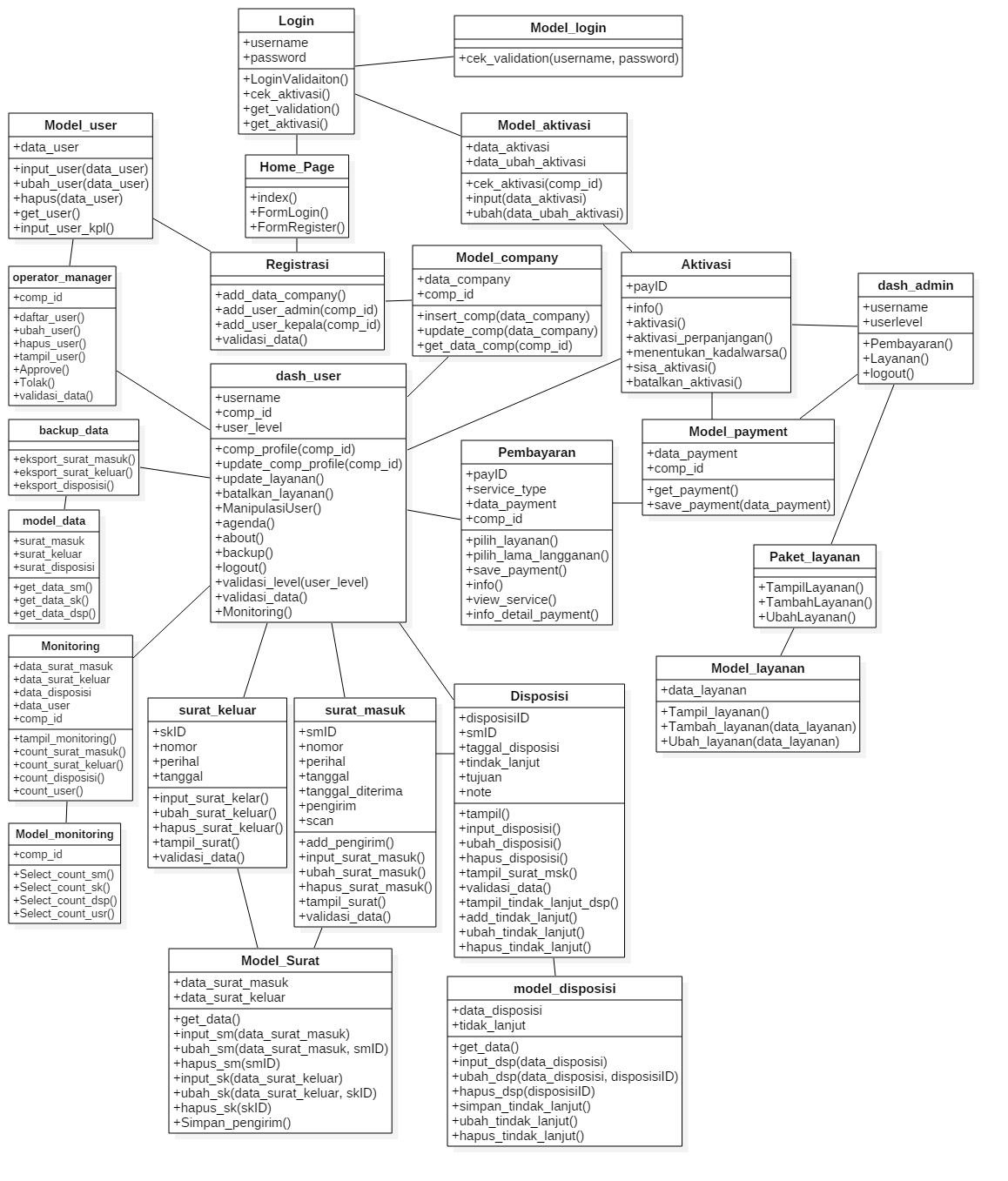
Pada *activity diagram* yang ditunjukan pada Gambar 3.15 merupakan activity *diagram* proses manipulasi paket layanan dimana pada gambar tersebut menjelaskan tentang aktivitas admin yang melakukan manipulasi paket layanan dengan bertujuan agar paket layanan lebih dinamis.



Gambar 3. 15 Activity Diagram Manipulasi Paket Layanan

* + 1. ***Class Diagram***

*Class* *Diagram* adalah diagram yang menjelaskan bagian hubungan antara *class* yang ada pada sistem SaaS Surat Menyurat dimana digambarkan dimana didalamnya terdapat atribut, dan *method* yang dimiliki suatu *class*. pada sistem saas manajemen surat menyurat terdapat 24 *class* yang masing-masing *class* menmpunyai hubungan antar *class* lainya*.* Adapun class diagram yang dimaksud dijelaskan pada Gambar 3. 16.

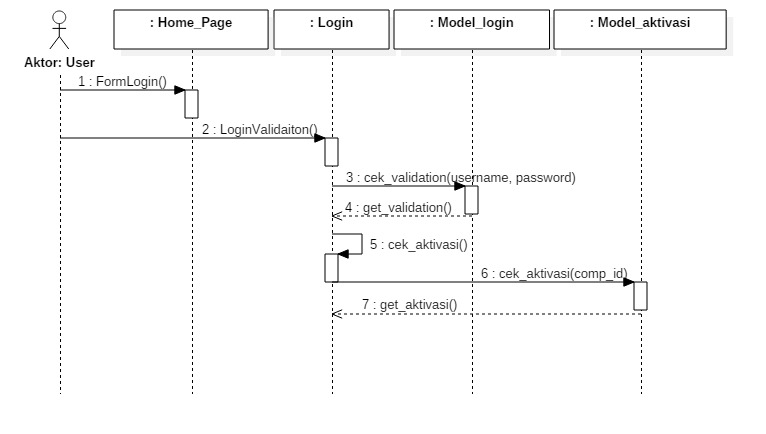


Gambar 3. 16 Class Diagram SaaS Surat Menyurat

* + 1. ***Sequence Diagram***

1. *Sequence Diagram* Login

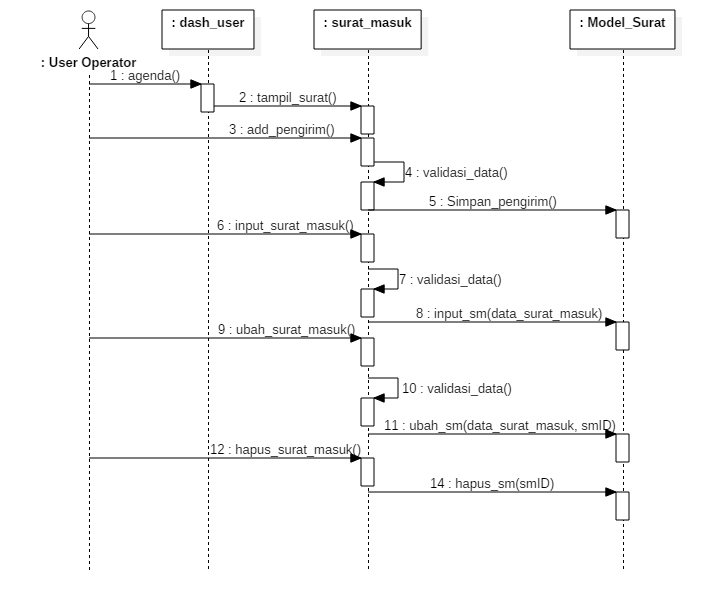
Porses login dapat digambarkan dalam *sequence* *diagram* yang menggambarkan interaksi pesan yang dikirim melalui satu *class* menuju *class* yang lain. Pada *sequence* *diagram login,* pesan tersebut meliputi LoginValidation() yang berfungsi untuk memvalidasi *username* dan *password* dari masukan *user*, kemudian *method* cek\_validation() merupakan fungsi untuk memeriksa ketersediaan *username* dan *password* dari masukan *user* dengan data yang telah tersimpan pada basis data. *method* cek\_aktifasi() merupakan fungsi untuk memeriksa aktivasi paket layanan pengguna. Berikut adalah *sequence* diagram login yang ditunjukan pada Gambar 3.*17*.



Gambar 3.17 Sequence Diagram Login

1. *Sequence Diagram* Manipulasi Surat Masuk

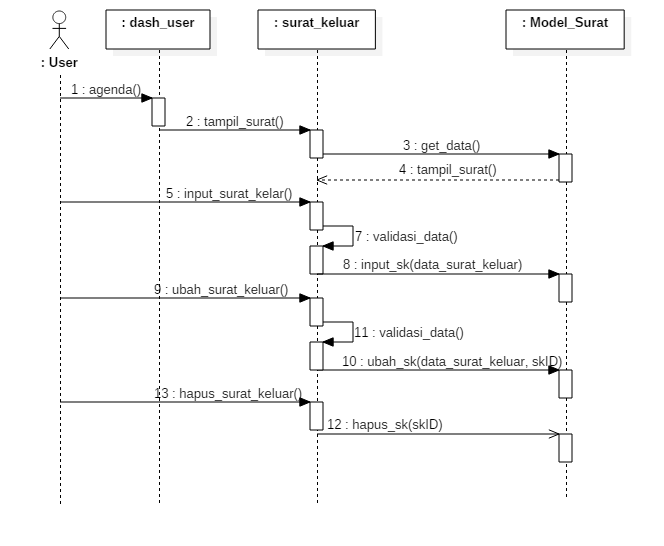
Adapun proses manipulasi data surat masuk digambarkan dengan *sequence* *diagram* yang ditunjukan pada Gambar 3.18 dimana pada diagram tersebut menjelaskan tentang interaksi antar *class* dalam proses manipulasi data surat dengan *method* add\_pengirim() untuk menambah data pengirim, input\_surat\_masuk() untuk menyimpan data surat masuk, *method* ubah\_surat\_masuk() berfungsi untuk mengubah atribut surat masuk dan *method* hapus\_surat\_masuk() berfungsi untuk menghapus data surat masuk.



Gambar 3.18 Sequence Diagram Manipulasi Surat Masuk

1. *Sequence Diagram* Manipulasi Surat Keluar

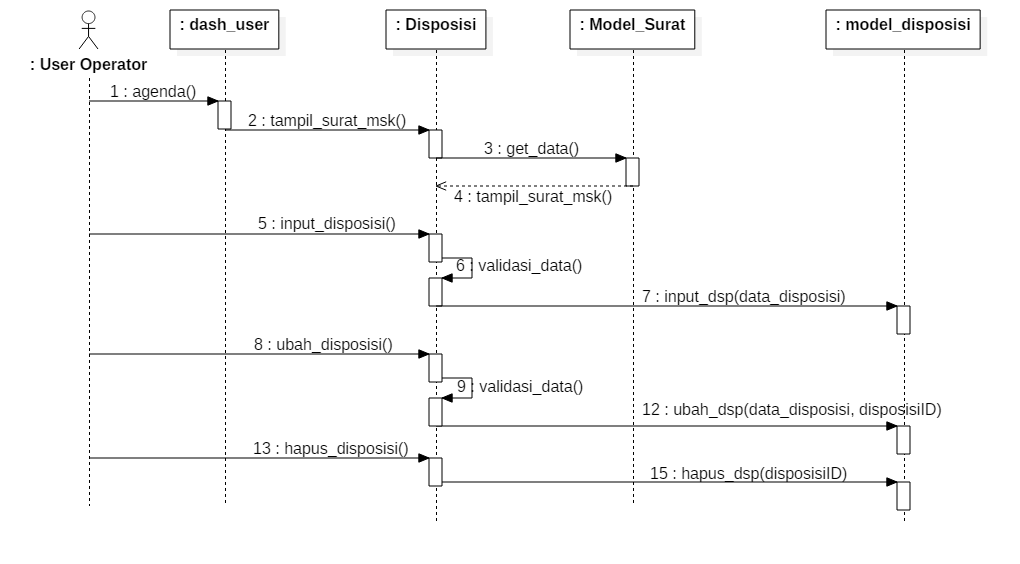
Proses manipulasi surat keluar digambarkan dengan *sequence* diagram yang ditunjukan pada Gambar 3.19 dimana pada diagram tersebut menjelaskan tentang interaksi antar *class* dalam proses manipulasi data surat dengan *method* input\_surat\_keluar() untuk menyimpan data surat keluar, *method* ubah\_surat\_keluar() berfungsi untuk mengubah atribut surat keluar dan *method* hapus\_surat\_keluar() berfungsi untuk menghapus data surat keluar.



Gambar 3.19 Sequence Diagram Manipulasi Surat Keluar

1. *Sequence Diagram* Manipulasi Disposisi

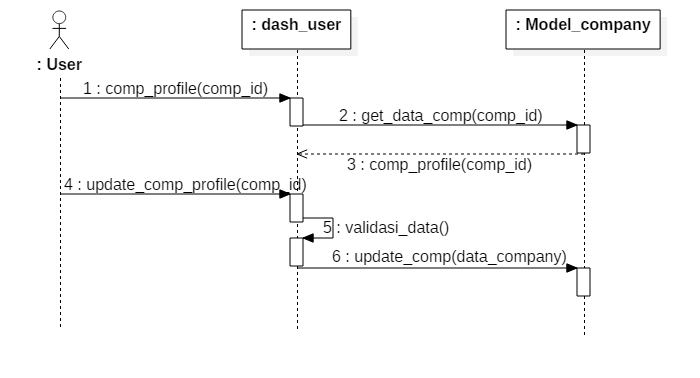
Proses manipulasi disposisi surat masuk digambarkan dengan *sequence* diagram yang ditunjukan pada Gambar 3.20 dimana pada diagram tersebut menjelaskan tentang interaksi antar *class* dalam proses manipulasi data disposisi surat masuk dengan *method* insert\_disposisi() untuk menyimpan data disposisi, *method* ubah\_disposisi() berfungsi untuk mengubah atribut disposisi dan *method* hapus\_disposisi() berfungsi untuk menghapus data disposisi.



Gambar 3.20 Sequence Diagram Disposisi Surat Masuk

1. *Sequence Diagram* Ubah Profil Organisasi

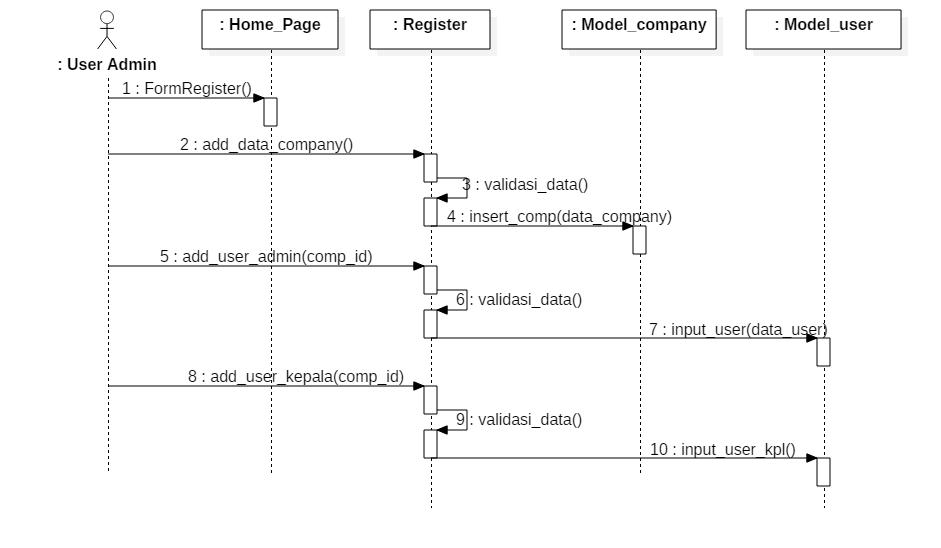
Proses ubah profil organisasi digambarkan dengan *sequence* diagram yang ditunjukan pada Gambar 3. 21Sequence Diagram *Ubah Profil*  dimana pada diagram tersebut menjelaskan tentang proses ubah profil organisasi yang digambarkan dengan interaksi antara kelas dan *method* yang digunakan. Salah satunya adalah update\_comp\_profile() merupakan fungsi untuk menyimpan data ubah profil organisasi.



Gambar 3. 21 Sequence Diagram Ubah Profil Organisasi

1. *Sequence Diagram* Registrasi

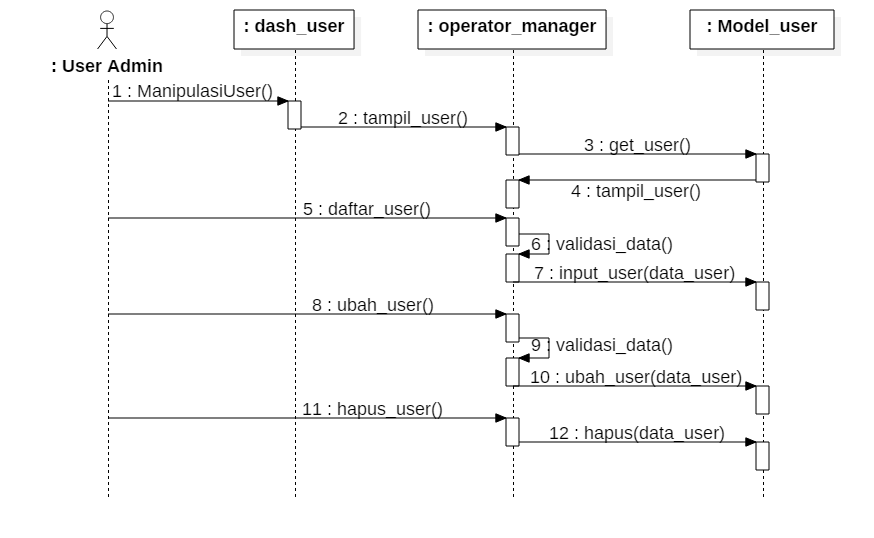
Adapun proses registrasi organisasi digambarkan dengan *sequence* diagram yang ditunjukan pada Gambar 3. 22 dimana pada diagram tersebut menjelaskan tentang proses registrasi yang terdapat *method* add\_data\_company() untuk menambah data organisasi, kemudian add\_user\_admin() untuk menyimpan data akun admin dan add\_user\_kelapa() untuk menyimpan data akun kepala organisasi.



Gambar 3. 22 Sequence Diagram Registrasi

1. *Sequence Diagram* Manipulasi Akun Operator

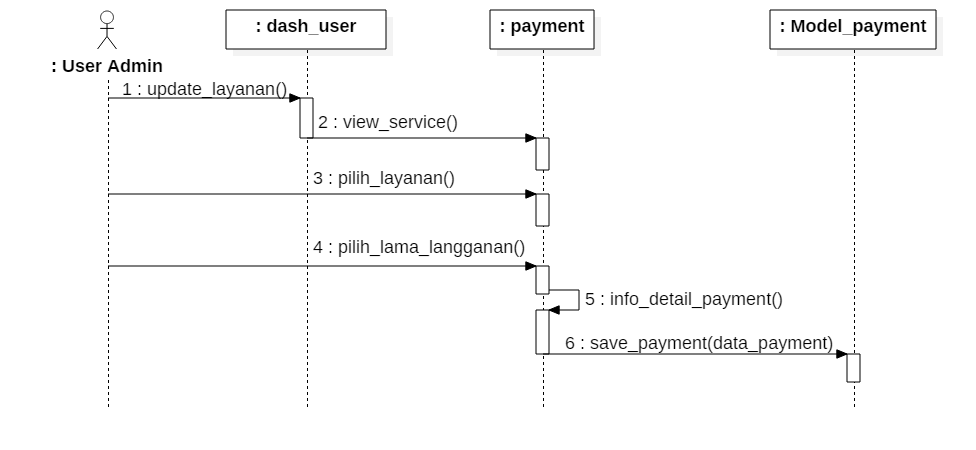
Adapun proses manipulasi akun operator pada perusahaan digambarkan dengan *sequence* diagram yang ditunjukan pada Gambar 3. 23 dimana pada diagram tersebut menjelaskan tentang interaksi antar *class* dalam proses manipulasi data akun operator dengan *method* daftar\_user() untuk menyimpan data akun operator, *method* ubah\_user() berfungsi untuk mengubah atribut akun operator dan *method* hapus\_user() berfungsi untuk menghapus data akun user.



Gambar 3. 23 Sequence Diagram Manipulasi Operator

1. *Sequence Diagram* Manipulasi Layanan

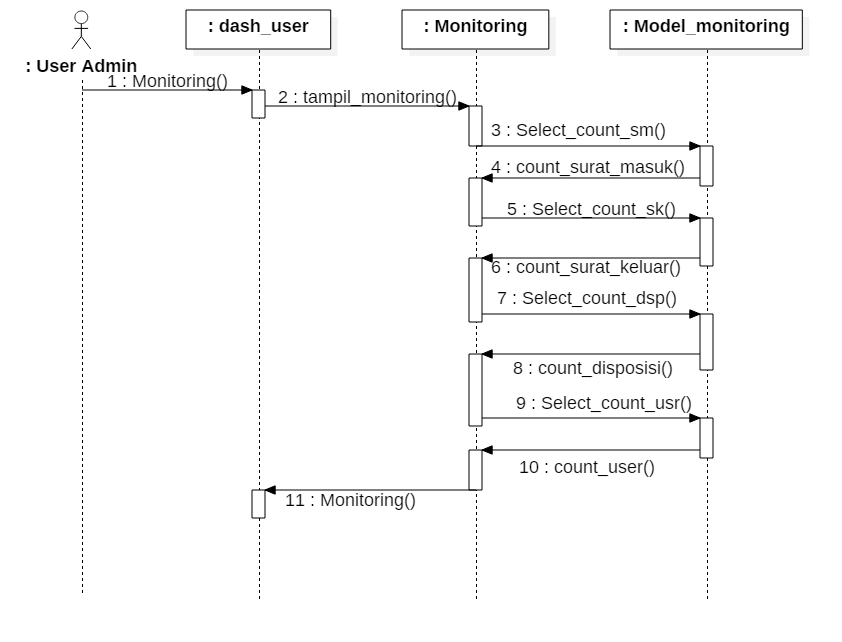
Proses manipulasi paket layanan user digambarkan dengan *sequence* *diagram* yang ditunjukan pada Gambar 3. 24 dimana pada diagram tersebut menjelaskan tentang interaksi antar *class* dimana dalam proses manipulasi layanan dimana terdapat *method* pilih\_layanan() untuk menampilkan pilihan layanan, pilih\_lama\_langganan() untuk menampilkan form pilih lama langganan.



Gambar 3. 24 Sequence Diagram Manipulasi Layanan

1. *Sequence Diagram Monitoring Data*

Proses menampilkan jumlah batas data sesuai dengan paket layanan pada sistem yang telah digunakan digambarkan dengan *sequence* *diagram* yang ditunjukan pada Gambar 3. *25* dimana pada diagram tersebut menjelaskan tentang menampikan informasi sumber daya sistem yang telah digunakan oleh user yang digambarkan dengan interaksi antara *class* pada data monitoring pada *method* tampil\_monitoring() yang berguna untuk menampilkan *monitoring data*.



Gambar 3. 25 Sequence Diagram Data Monitoring

1. *Sequence Diagram Backup Data*

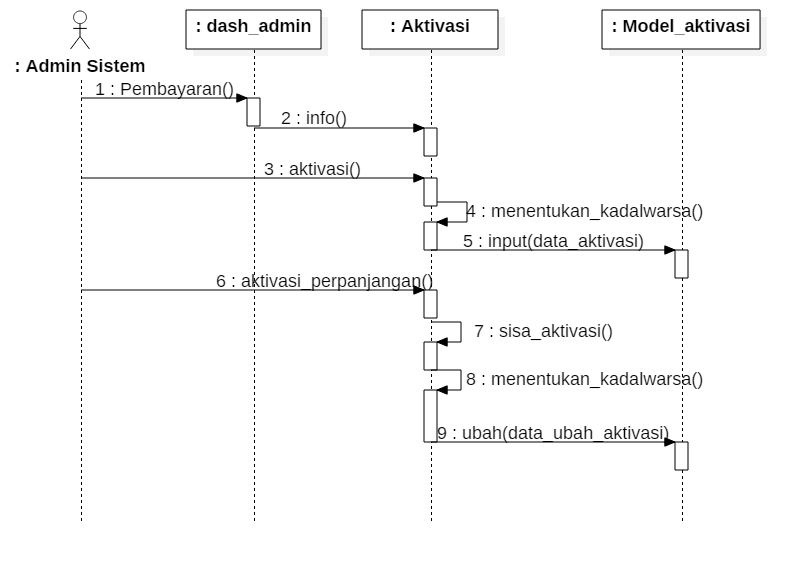
Proses *backup data* digambarkan dengan *sequence* *diagram* yang ditunjukan pada Gambar 3. *26* dimana pada diagram tersebut menjelaskan tentang proses pencadangan data oleh user admin yang digambarkan dengan interaksi antara *class* dimana dalam proses *backup data* dimana terdapat *method* eksport\_surat\_masuk() untuk unduh data surat masuk, eksport\_surat\_keluar() untuk unduh data surat keluar dan eksport\_disposisi() untuk mengunduh data disposisi.



Gambar 3. 26 Sequence Diagram Backup

1. *Sequence Diagram* Aktivasi Layanan

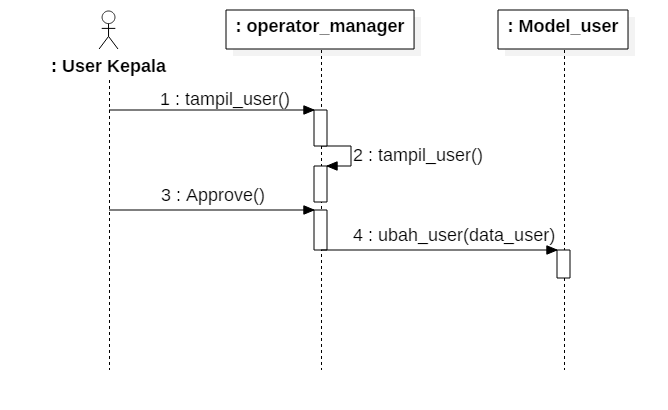
Proses aktivasi layanan digambarkan dengan *sequence* *diagram* yang ditunjukan pada Gambar 3. 27 SequenceDiagram *aktivasi layanan* pada diagram tersebut menjelaskan proses aktivasi layanan yang telah dibayarkan oleh *user*. Proses aktivasi dilakukan oleh admin sistem yang digambarkan dengan interaksi antara *class* dimana terdapat *method* aktivasi() untuk mengubah status *user* sesuai dengan paket yang telah dipilih.



Gambar 3. 27 Sequence Diagram aktivasi layanan

1. *Sequence Diagram* *Approve* Akun Operator

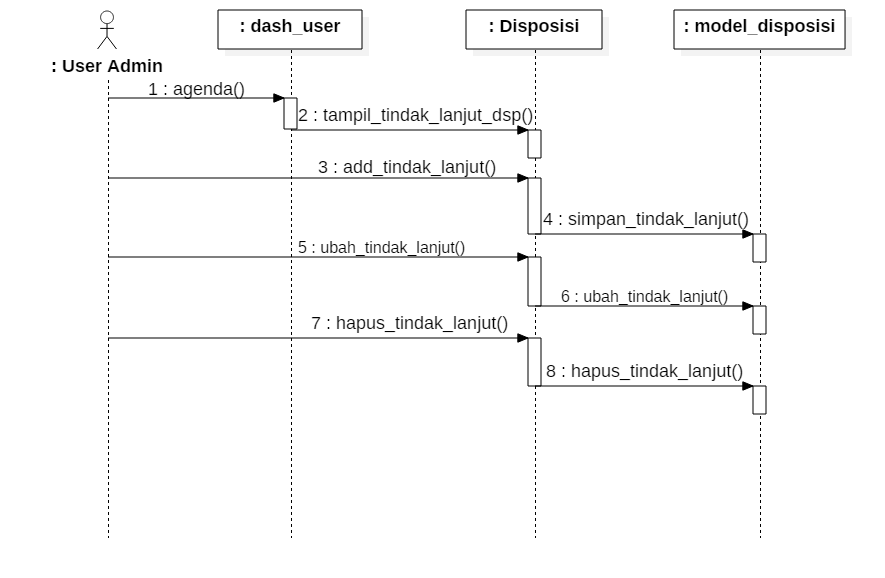
Proses *approve* akun operator digambarkan dengan *sequence* *diagram* yang ditunjukan pada Gambar 3. 28 pada diagram tersebut menjelaskan tentang proses *approve* akun operator yang dilakukan oleh akun kepala sesuai dengan pengajuan yang dilakukan oleh akun *user* operator. Adapun method approve() berfungsi untuk merubah status user operator menjadi aktif.



Gambar 3. 28 Sequence Diagram Approve Akun Operator

1. *Activity Diagram* Manipulasi Tindak Lanjut Disposisi

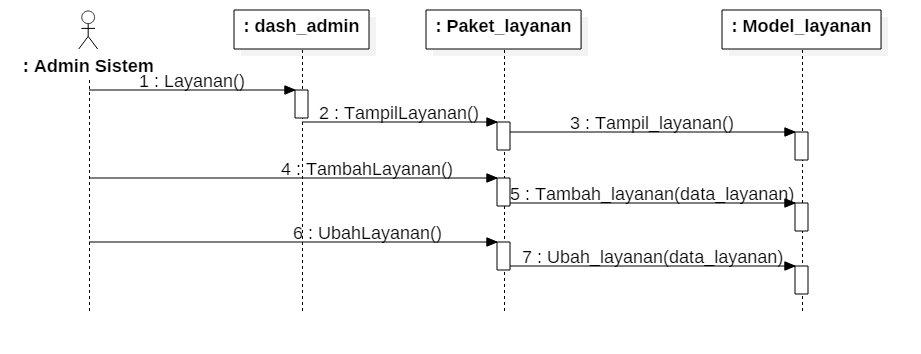
Proses manipulasi tindak lanjut disposisi digambarkan dengan *sequence* *diagram* yang ditunjukan pada Gambar 3. 29 dimana pada diagram tersebut menjelaskan tentang interaksi antar *class* dalam proses manipulasi data tindak lanjut disposisi dengan *method* add\_tindak\_lanjut() untuk menyimpan data tindak lanjut, *method* ubah\_tindak\_lanjut() berfungsi untuk mengubah atribut tindak lanjut disposisi dan *method* hapus\_tindak\_lanjut() berfungsi untuk menghapus data tindak lanjut disposisi.



Gambar 3. 29 Sequence Diagram manipulasi tindak lanjut disposisi

1. Sequence Diagram Manipulasi Paket Layanan

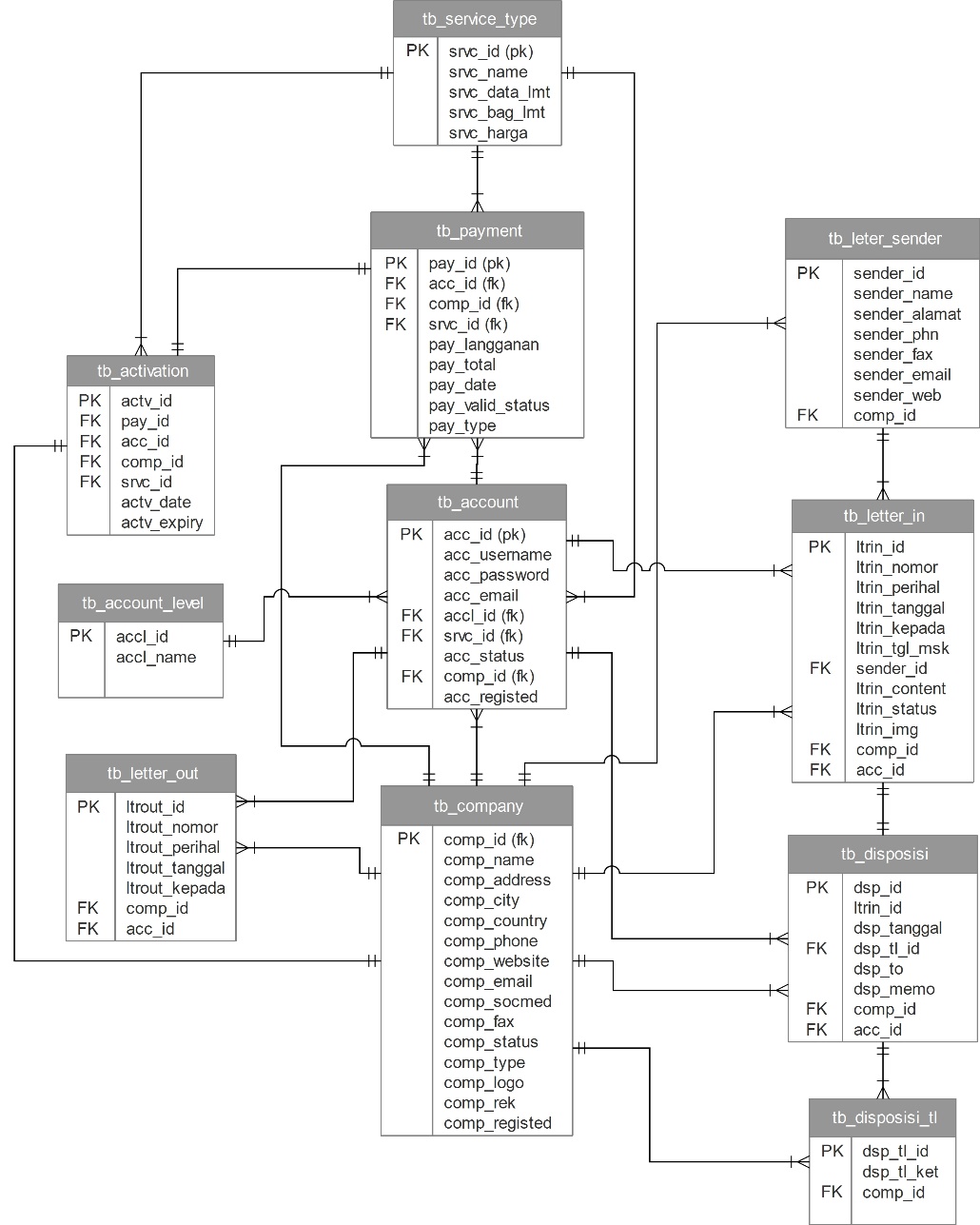
Proses manipulasi paket layanan digambarkan *dengan* *sequence* diagram yang ditunjukan pada Gambar 3. 30 dimana pada diagram tersebut menjelaskan tentang interaksi antar *class* dimana terdapat *method* TambahLayanan() yang berfungsi untuk menambah paket layanan dan UbahLayanan() merupakan fungsi untuk mengubah atribut paket layanan.



Gambar 3. 30 Sequence Diagram Manipulasi Paket Layanan

* + 1. **Basis Data Konseptual**

Basis data konseptual pada Gambar 3.31 merupakan gambaran relasi konseptual antar tabel dari sistem surat menyurat dimana terdapat 11 (sebelas) buah tabel diantaranya, tb\_service\_type, tb\_payment, tb\_account, tb\_company, tb\_letter\_sender, tb\_letter\_in, tb\_letter\_disposisi, tb\_letter\_out, tb\_letter\_disposisi\_tl, tb\_account\_level, dan tb\_activation. Setiap tabel pada basis data konseptual memiliki hubungan antar tabel.

****

Gambar 3. 31 Basis Data Konseptual

Tabel pada basis data konseptual yang ditunjukan pada Gambar 3. *31* memiliki relasi yaitu antara tb\_account dengan tb\_service\_type berelasi *one to many,* kemudan tb\_account berelasi *many to one* ke tb\_account\_level, Begitu juga tb\_accout berelasi dengan tb\_letter\_in, tb\_letter\_out, tb\_disposisi dengan *one to many,* dan yang terakhir untuk tb\_account berelasi *one to one* dengan tb\_activation. Sementara itu tb\_letter\_sender, tb\_letter\_in, tb\_disposisi, tb\_disposisi\_tl, tb\_letter\_out, berelasi dengan tb\_company sebagai *many to one*. Sementara tb\_aktivasi berelasi dengan tb\_company, tb\_service\_type, tb\_account, tb\_pembayaran berelasi sebagai *one to many*.

* + 1. **Struktur Tabel**

Struktur tabel digunakan untuk memudahkan dalam merancang program yang akan dibuat. Dalam struktur file terdapat nama *field* beserta tipe data dan jenis *key*. Terdapat 11 tabel yang dapat dijabarkan field name, dan tipe datanya. Struktur tabel yang terdapat pada basis data pada Sistem *Cloud Computing* Surat Menyurat adalah sebagai berikut.

* + - 1. Tb\_account

Struktur tabel *tb\_account* yang diuraikan pada Tabel 3.29 merupakan tabel yang menyimpan data *account* atau *user* pada sistem. Dalam tabel ini berisi atribut-atribut yang lebih detail dijabarkan pada Table 3.29.

Tabel 3.29 Struktur Tabel tb\_account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field** | **Tipe Data** | **Size** | **Keterangan** |
| 1 | acc\_id | Int | 9 | *Primary Key* |
| 2 | acc\_usrn | Char | 12 |  |
| 3 | acc\_pswd | Char | 12 |  |
| 4 | acc\_email | Varchar | 50 |  |
| 5 | accl\_id | Int | 9 | *Foreign Key* |
| 6 | srvc\_id | Int | 9 | *Foreign Key* |
| 7 | comp\_id | Int | 9 | *Foreign Key* |
| 8 | acc\_status | Int | 1 |  |
| 9 | acc\_registed | Datetime |  |  |

* + - 1. Tb\_company

Struktur tabel *tb\_company* yang diuraikan pada Tabel 3.30 merupakan tabel yang menyimpan data perusahaan pada sistem. Dalam tabel ini berisi atribut-atribut yang lebih detail dijabarkan pada Table 3.30.

Tabel 3.30 Struktur Tabel tb\_company

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field** | **Tipe Data** | **Size** | **Keterangan** |
| 1 | comp\_id | Int | 9 | *Primary Key* |
| 2 | comp\_name | Text | - |  |
| 3 | comp\_address | Text | - |  |
| 4 | comp\_city | Varchar | 50 |  |
| 5 | comp\_country | Varchar | 60 |  |
| 6 | comp\_phone | Varchar | 10 |  |
| 7 | comp\_website | Varchar | 25 |  |
| 8 | comp\_email | Varchar | 25 |  |
| 9 | comp\_fax | Varchar | 25 |  |
| 10 | comp\_status | Varchar | 25 |  |
| 11 | comp\_type | Varchar | 25 |  |
| 12 | comp\_logo | Text |  |  |
| 13 | comp\_reknum | Varchar | 16 |  |

* + - 1. Tb\_account\_level

Struktur tabel *tb\_account\_level* yang diuraikan pada Tabel 3.31 merupakan tabel yang menyimpan data level akses akun pada sistem. Dalam tabel ini berisi atribut-atribut yaitu accl\_id, dana accl\_name.

Tabel 3.31 Struktur Tabel tb\_account\_level

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field** | **Tipe Data** | **Size** | **Keterangan** |
| 1 | accl\_id | Int | 9 | *Primary Key* |
| 2 | accl\_name | Varchar | 20 |  |

* + - 1. Tb\_service\_type

Struktur tabel *tb\_service\_type* yang diuraikan pada Tabel 3.32 merupakan tabel yang menyimpan data paket layanan pada sistem. Dalam tabel ini berisi atribut-atribut yaitu srvc\_id, srvc\_name, srvc\_data\_lmt, srvc\_bag\_lmt, dan srvc\_price.

Tabel 3.32 Struktur Tabel tb\_service\_type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field** | **Tipe Data** | **Size** | **Keterangan** |
| 1 | srvc\_id | Int | 9 | *Primary Key* |
| 2 | srvc\_name | Varchar | 20 |  |
| 3 | srvc\_data\_lmt | Int | 5 |  |
| 4 | srvc\_bag\_lmt | Int | 5 |  |
| 5 | srvc\_price | Int | 5 |  |

* + - 1. Tb\_letter\_sender

Struktur tabel tb\_ letter\_sender yang diuraikan pada Tabel 3.33 merupakan tabel yang menyimpan data pengirim surat masuk pada sistem. Dalam tabel ini berisi atribut-atribut yang lebih detail dijelaskan pada Table 3.33.

Tabel 3.33 Struktur Tabel tb\_ letter\_sender

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field** | **Tipe Data** | **Size** | **Keterangan** |
| 1 | sender\_id | Int | 9 | *Primary Key* |
| 2 | sender\_name | Text | - |  |
| 3 | sender\_address | Varchar | 25 |  |
| 4 | sender\_phn | Varchar | 25 |  |
| 5 | sender\_fax | Varchar | 25 |  |
| 6 | sender\_email | Varchar | 25 |  |
| 7 | sender\_web | Varchar | 25 |  |
| 8 | comp\_id | Int | 9 | *Foreign Key* |

* + - 1. Tb\_letter\_in

Struktur tabel tb \_letter\_in yang diuraikan pada Tabel 3.34 merupakan tabel yang menyimpan data surat masuk pada sistem. Dalam tabel ini berisi atribut-atribut yang lebih detail dijelaskan pada Table 3.34.

Tabel 3.34 Struktur Tabel tb\_letter\_in

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field** | **Tipe Data** | **Size** | **Keterangan** |
| 1 | ltrin\_id | Int | 9 | *Primary Key* |
| 2 | ltrin\_nomor | Text | - |  |
| 3 | ltrin\_perihal | Text | - |  |
| 4 | ltrin\_tanggal | Date | - |  |
| 5 | ltrin\_tujuan | Varchar | 50 |  |
| 6 | ltrin\_tgl\_msk | Date | - |  |
| 7 | sender\_id | Int | 9 | *Foreign Key* |
| 8 | ltrin\_content | Text | - |  |
| 9 | ltrin\_status | Boolean | - |  |
| 10 | ltrin\_img | Text | - |  |
| 11 | acc\_id | Int | 9 | *Foreign Key* |
| 12 | comp\_id | Int | 9 | *Foreign Key* |

* + - 1. Tb\_letter\_out

Struktur tabel tb \_surat\_keluar yang diuraikan pada Tabel 3.35 merupakan tabel yang menyimpan data surat keluar pada sistem. Dalam tabel ini berisi atribut-atribut yang lebih detail dijelaskan pada Table 3.35.

Tabel 3.35 Struktur Tabel tb\_letter\_out

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field** | **Tipe Data** | **Size** | **Keterangan** |
| 1 | ltrout\_id | Int | 9 | *Primary Key* |
| 2 | acc\_id | Int | 9 | *Foreign Key* |
| 3 | comp\_id | Int | 9 | *Foreign Key* |
| 4 | ltrout\_nomor | Text | - |  |
| 5 | ltrout\_perihal | Text | - |  |
| 6 | ltrout\_tanggal | Date | - |  |
| 7 | ltrout\_kepada | Varchar | 50 |  |

* + - 1. Tb\_disposisi

Struktur tabel tb \_disposisi yang diuraikan pada Tabel 3.36 merupakan tabel yang menyimpan data disposisi surat masuk pada sistem. Dalam tabel ini berisi atribut-atribut yang lebih detail dijelaskan pada Table 3.36.

Tabel 3.36 Struktur Tabel tb\_disposisi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field** | **Tipe Data** | **Size** | **Keterangan** |
| 1 | dsp\_id | Int | 9 | *Primary Key* |
| 2 | acc\_id | Int | 9 | *Foreign Key* |
| 3 | comp\_id | Int | 9 | *Foreign Key* |
| 4 | sm\_id | Int | 9 | *Foreign Key* |
| 5 | dsp\_tanggal | Date | - |  |
| 6 | dsp\_tl\_id | Int | 9 | *Foreign Key* |
| 7 | dsp\_to | Int | 9 | *Foreign Key* |
| 8 | dsp\_memo | Text | - |  |

* + - 1. Tb\_disposisi\_tl

Struktur tabel tb \_disposisi\_tl yang diuraikan pada Tabel 3.37 merupakan tabel yang menyimpan data tindak lanjut disposisi surat masuk pada sistem. Dalam tabel ini berisi atribut-atribut yang lebih detail dijelaskan pada Table 3.37.

Tabel 3.37 Struktur Tabel tb\_disposisi\_tl

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field** | **Tipe Data** | **Size** | **Keterangan** |
| 1 | dsp\_tl\_id | Int | 9 | *Primary Key* |
| 2 | dsp\_ket | Text | - |  |
| 3 | comp\_id | Int | 9 | *Foreign Key* |

* + - 1. Tb\_payment

Struktur tabel tb \_pembayaran yang diuraikan pada Tabel 3.38 merupakan tabel yang menyimpan data pembayaran aktivasi paket layanan pada sistem. Dalam tabel ini berisi atribut-atribut yang lebih detail dijelaskan pada Table 3.38.

Tabel 3.38 Struktur Tabel tb\_payment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field** | **Tipe Data** | **Size** | **Keterangan** |
| 1 | pay\_id | Int | 9 | *Primary Key* |
| 2 | comp\_id | Int | 9 | *Foreign Key* |
| 3 | acc\_id | Int | 9 | *Foreign Key* |
| 4 | srvc\_id | Int | 9 | *Foreign Key* |
| 5 | pay\_langganan | Int | 5 |  |
| 6 | pay\_total | Int | 5 |  |
| 7 | pay\_date | Datetime | - |  |
| 8 | pay\_status | Int | 1 |  |
| 9 | pay\_type | Int | 1 |  |

* + - 1. Tb\_activation

Struktur tabel tb \_aktivasi yang diuraikan pada Tabel 3.39 merupakan tabel yang menyimpan data pembayaran aktivasi paket layanan pada sistem. Dalam tabel ini berisi atribut-atribut yang lebih detail dijelaskan pada Table 3.39.

Tabel 3.39 Struktur Tabel tb\_activation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field** | **Tipe Data** | **Size** | **Keterangan** |
| 1 | Actv\_id | Int | 9 | *Primary Key* |
| 2 | Comp\_id | Int | 9 | *Foreign Key* |
| 3 | Acc\_id | Int | 9 | *Foreign Key* |
| 4 | Srvc\_id | Int | 9 | *Foreign Key* |
| 5 | Pay\_id | Int | 9 | *Foreign Key* |
| 6 | Actv\_date | Date | - |  |
| 7 | Actv\_expiry | Date | - |  |

* 1. **Perancangan Desain Antar Muka (*interface*)**

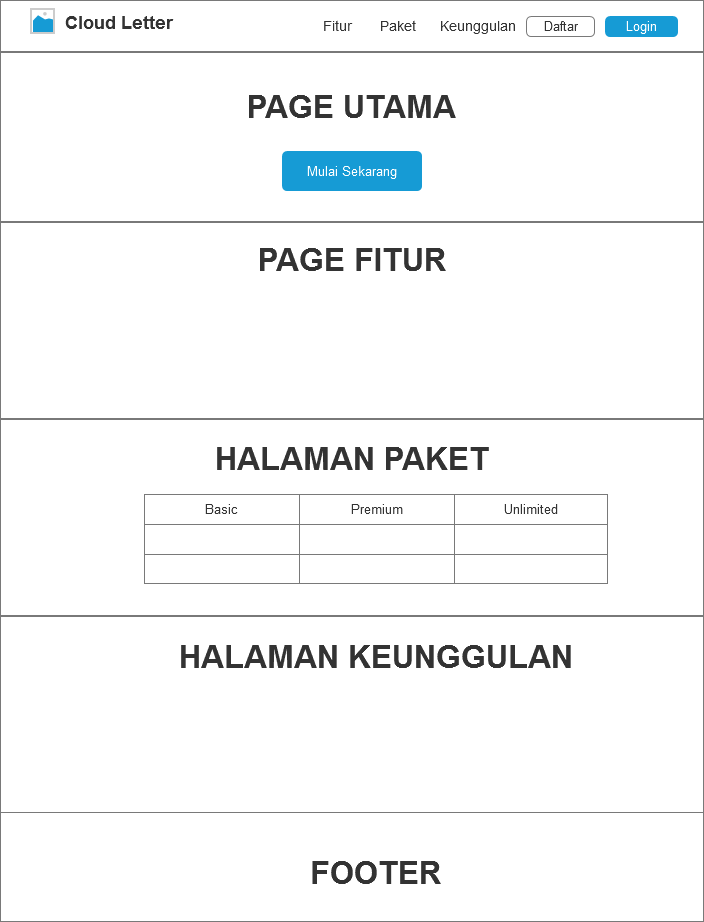
Halaman *front end* merupakan halaman dari situs web yang muncul pertama kali pada browser ketika pengunjung membuka halaman web. Halaman ini dapat diakses oleh pengunjung tanpa harus melakukan *login* terlebih dahulu. Pada halaman ini terdapat informasi sederhana yang terbagi dalam beberapa halaman. Berikut ini rancangan halaman *front end* pada perekayasaan sistem *cloud computing* surat menyurat.

* + 1. **Rancangan Tampilan Depan (*Front End*)**

Halaman ini digunakan untuk mengelola menampilkan fungsi-fungsi awal yang digunakan untuk memulai mengakses sistem yang telah dibuat. Berikut rancangan halaman *front end* pada perekayasaan sistem *cloud computing* surat menyurat.

1. Rancangan Halaman Beranda (*home)*

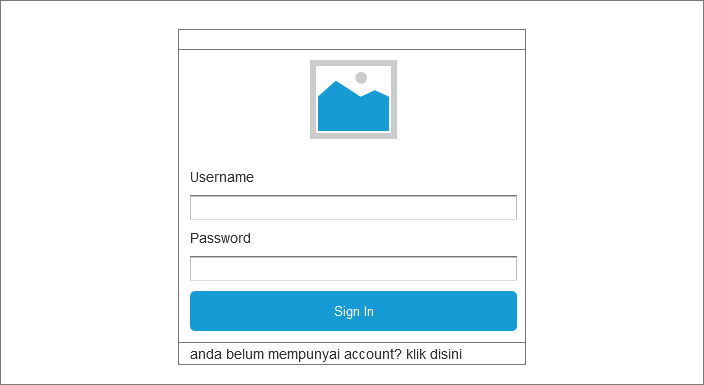
Halaman *home* yang ditujukan pada Gambar 3. 32 merupakan halaman awal ketika mengakses alamat situs web. Pada desain halaman ini akan menyediakan informasi–informasi kepada pengunjung terkait sistem *cloud computing* surat menyurat dimana pada halaman depan dibagi menjadi empat bagian.



Gambar 3. 32 Rancangan Halaman Beranda

1. Rancanga Halaman Login

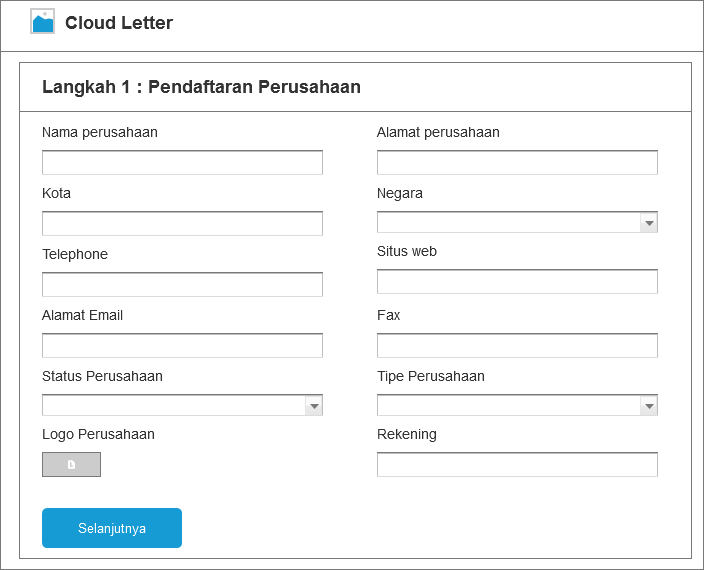
Pada desain halaman *login* yang ditunjukan pada gambar Gambar 3. 33 *user* yang sudah terdaftar dapat melakukan *login* dengan memasukan *username* dan *password* pada kolom yang disediakan.



Gambar 3. 33 Rancangan Halaman Login

1. Rancangan Halaman Registrasi Organisasi

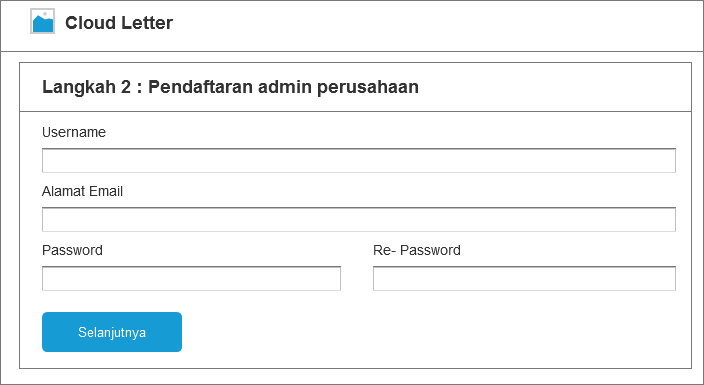
Halaman registrasi organisasi yang ditunjukan pada Gambar 3. 34 digunakan untuk registrasi perusahaan atau pendaftaran organisasi untuk didapatkan data organisasi sebagai langkah pertama dalam registrasi sistem *cloud computing* surat menyurat.



Gambar 3. 34 Rancangan Halaman Daftar Organisasi

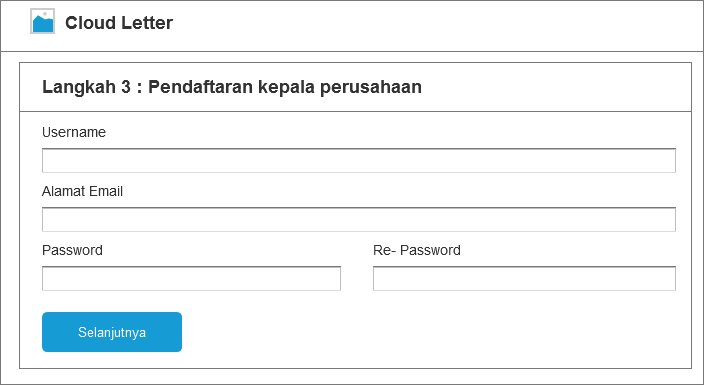
1. Rancangan Halaman Registrasi *User* Admin Dan *User* Kepala

Halaman registrasi akun *user* admin yang ditunjukan pada Gambar 3. 35 digunakan untuk registrasi admin organisasi untuk didapatkan data akun admin perusahaan sebagai langkah kedua dalam registrasi sistem *cloud computing* surat menyurat.



Gambar 3. 35 Rancangan Halaman Daftar User Admin

Halaman registrasi *user* kepala yang ditunjukan pada Gambar 3. *36* digunakan untuk registrasi kepala organisasi untuk didapatkan data akun kepala perusahaan sebagai langkah terakhir dalam registrasi sistem *cloud computing* surat menyurat.



Gambar 3. 36 Rancangan Halaman Daftar User Kepala

* + 1. **Rancangan Tampilan Antar Muka Admin**

Halaman ini digunakan *admin* untuk mengelola website atau digunakan untuk mengakses sistem yang telah dibuat. Berikut rancangan halaman *admin* pada perekayasaan sistem *cloud computing* surat menyurat.

1. Rancangan *Dashboard Admin* Sistem

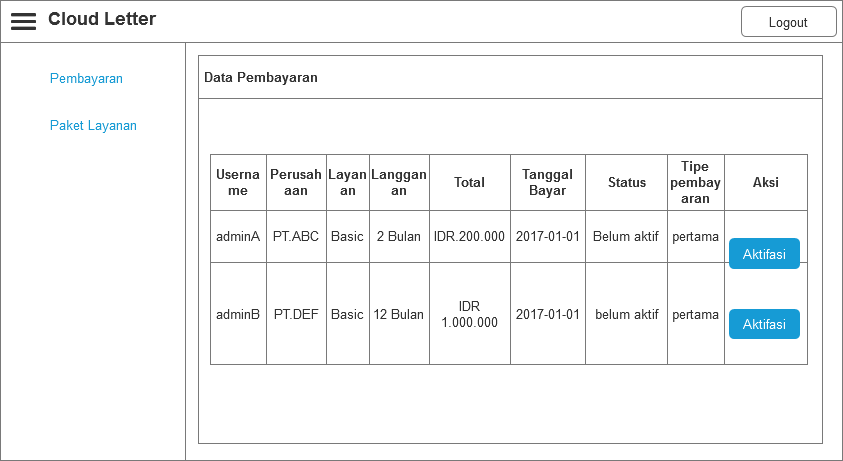
Pada desain halaman *dashboard* *admin* yang ditunjukan pada Gambar 3. *37* *admin* akan dapat melihat menu side nav untuk mengarahkan ke menu selanjutnya.



Gambar 3. 37 Rancangan Halaman Dashboard Admin

1. Rancangan Halaman Pembayaran Dan Perpanjangan

Pada desain halaman pembayaran, admin akan dapat melihat data pembayaran dengan status pembayaran pertama oleh *user* yang telah melakukan pembayaran. Admin dapat melakukan aktifasi paket layanan yang dibayarkan oleh *user* agar dapat menggunakan paket layanan yang telah dipilih.



Gambar 3. 38 Rancangan Halaman Aktivasi Pembayaran

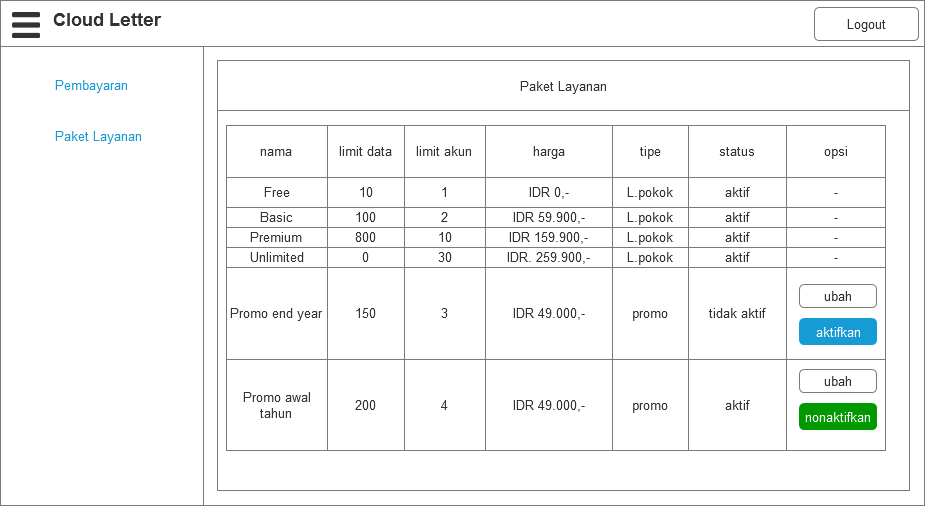
Pada desain halaman pembayaran perpanjangan layanan yang ditunjukan pada Gambar 3. 39 admin akan dapat melihat data pembayaran perpanjangan dengan status pembayaran perpanjang oleh *user*. Admin dapat melakukan aktifasi paket layanan yang dibayarkan oleh *user* agar dapat menggunakan paket layanan yang telah dipilih.



Gambar 3. 39 Rancangan Halaman Aktivasi Perpanjangan Layanan

1. Rancangan Halaman Data Paket Layanan

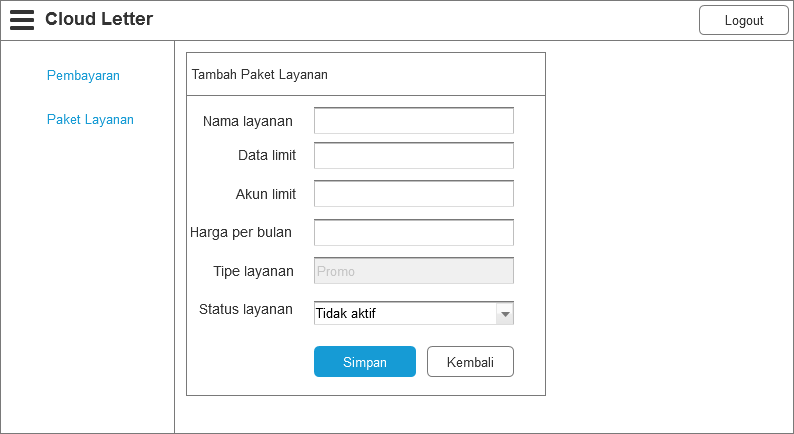
Pada desain halaman paket layanan yang ditujukan pada Gambar 3. 40 admin akan dapat mengelola data paket yang disediakan oleh sistem seperti mengubah dan menambah paket layanan sesuai dengan kebutuhan sistem. Tidak adanya *button* hapus layanan agar tidak berprngaruh pada layanan yang telah aktif.



Gambar 3. 40 Rancangan Halaman Data Paket Layanan

1. Rancangan Halaman Tambah Paket Layanan

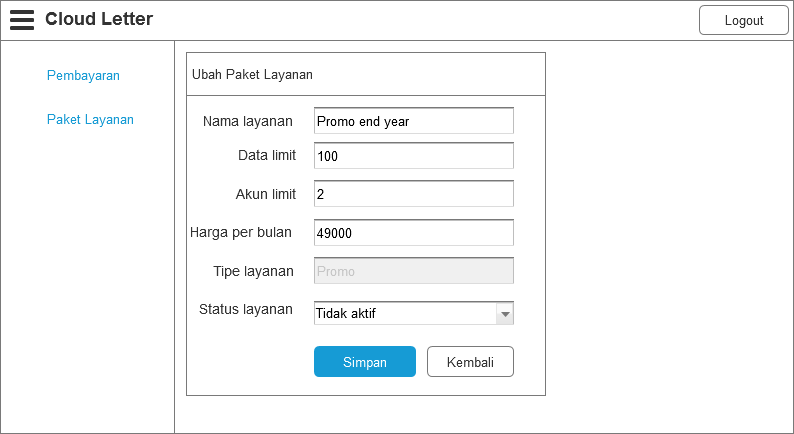
Pada desain halaman tambah paket layanan yang ditujukan pada Gambar 3. 41 admin *sistem* dapat menambah data paket yang disediakan untuk *user* oleh sistem agar paket layanan yang disediakan lebih dimanis.



Gambar 3. 41 Rancangan Halaman Tambah Paket Layanan

1. Rancangan Halaman Ubah Paket Layanan

Pada desain halaman ubah paket layanan yang ditujukan pada Gambar 3. 42 *admin sistem* dapat mengubah data paket yang disediakan sistem *cloud computing* surat menyurat sesuai dengan kebutuhan penyediaan paket layanan agar paket layanan yang disediakan lebih dinamis.



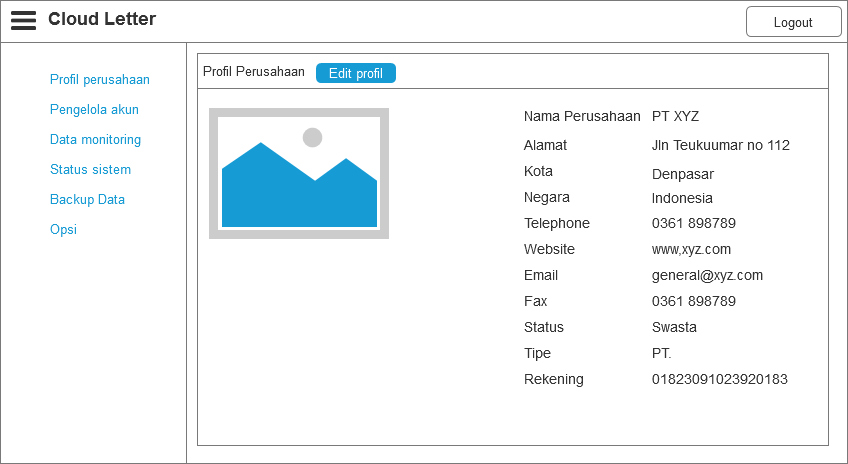
Gambar 3. 42 Rancangan Halaman Ubah Paket Layanan

* + 1. **Rancangan Tampilan Antar Muka *User* Admin**

Halaman ini digunakan *user* admin untuk mengelola sistem yang digunakan pada suatu perusahaan. Berikut rancangan halaman *admin* pada perekayasaan sistem *cloud computing* surat menyurat.

1. Rancangan Halaman Profil Organisasi

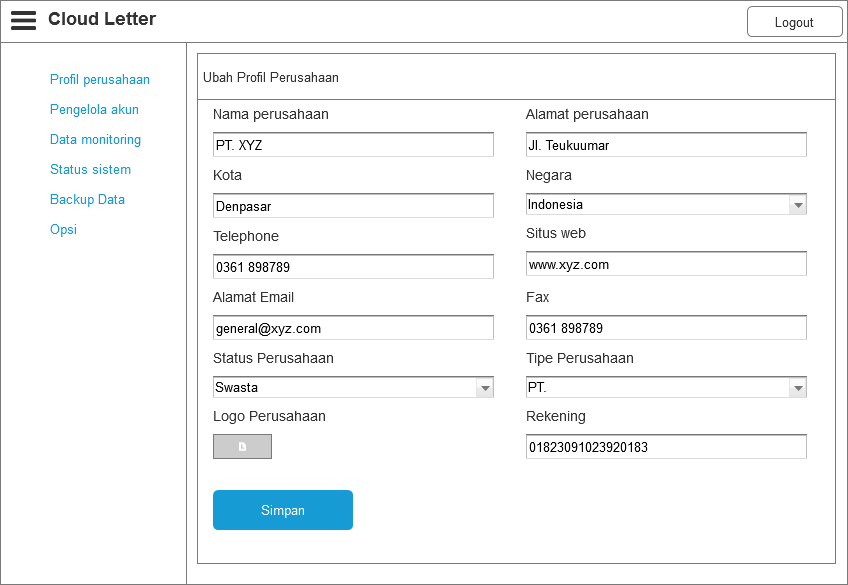
Pada desain halaman profil organisasi yang ditujukan pada Gambar 3. 43 *user* admin*, user* kepala *dan user* operator dapat melihat informasi perusahaan yang telah tersimpan pada sistem *cloud computing* surat menyurat.



Gambar 3. 43 Rancangan Halaman Profil Organisasi

1. Rancangan Halaman Ubah Profil Organisasi

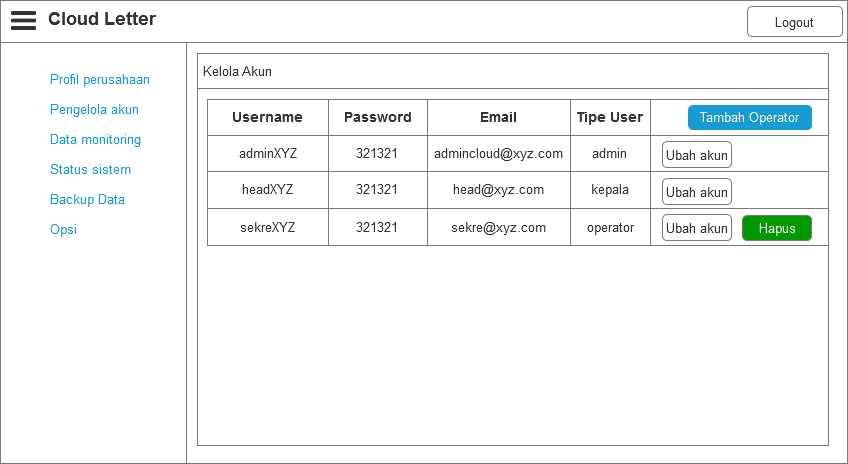
Pada desain halaman ubah profil organisasi yang ditujukan pada Gambar 3. 44 dibawah user admin*, user* kepala *dan user* operator dapat mengubah informasi perusahaan yang telah tersimpan pada sistem *cloud computing* surat menyurat.



Gambar 3. 44 Rancangan Halaman Ubah Profil Organisasi

1. Rancangan Halaman Kelola Akun Operator

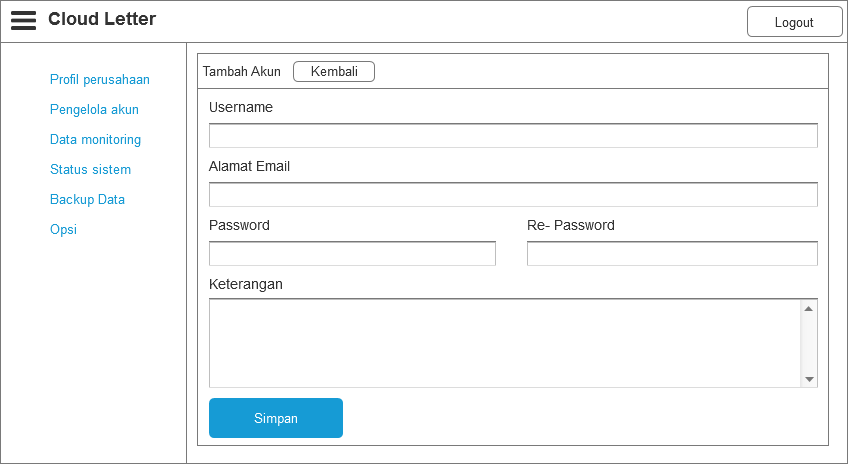
Pada desain halaman kelola akun operator yang ditujukan pada Gambar 3. 45 dibawah *user* admin dapat memanipulasi data akun yang telah didaftarkan tersimpan pada sistem.



Gambar 3. 45 Rancangan Halaman Kelola Akun Operator

1. Rancangan Halaman Form Tambah Akun Operator

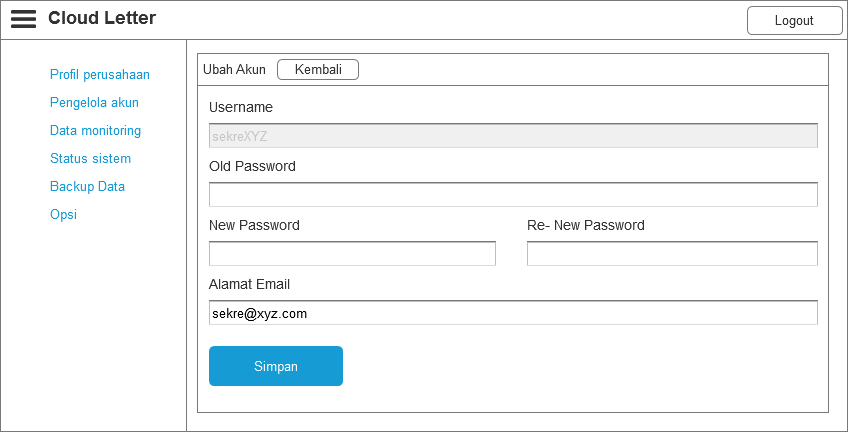
Pada desain halaman tambah akun operator yang ditujukan pada Gambar 3. 46 dibawah *user* admin dapat menambah atau mengusulkan akun operator pada sistem *cloud computing* surat menyurat.



Gambar 3. 46 Rancangan Halaman Form Tambah Akun

1. Rancangan Halaman Form Ubah Akun Operator

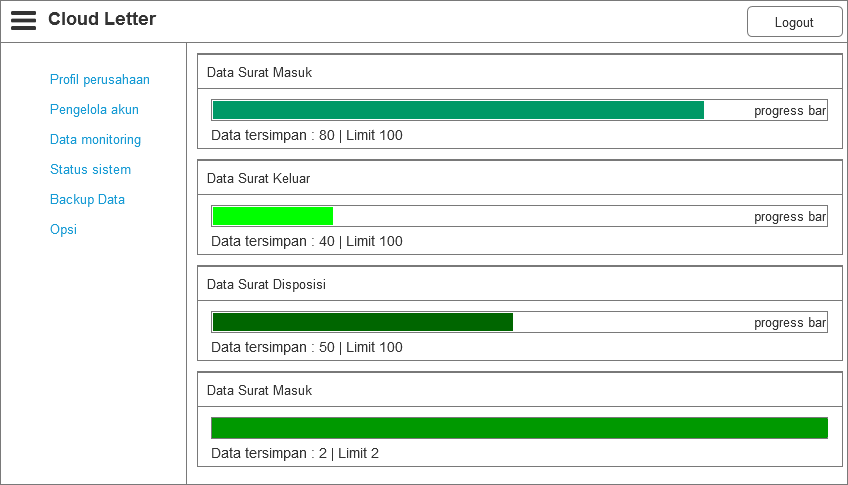
Pada desain halaman ubah akun operator yang ditujukan pada Gambar 3. 47 dibawah *user* admin dapat mengubah data akun yang telah diaktifkan oleh user kepala dan tersimpan pada sistem *cloud computing* surat menyurat.



Gambar 3. 47 Rancangan Halaman Ubah Akun

1. Rancangan Halaman Data *Monitoring*

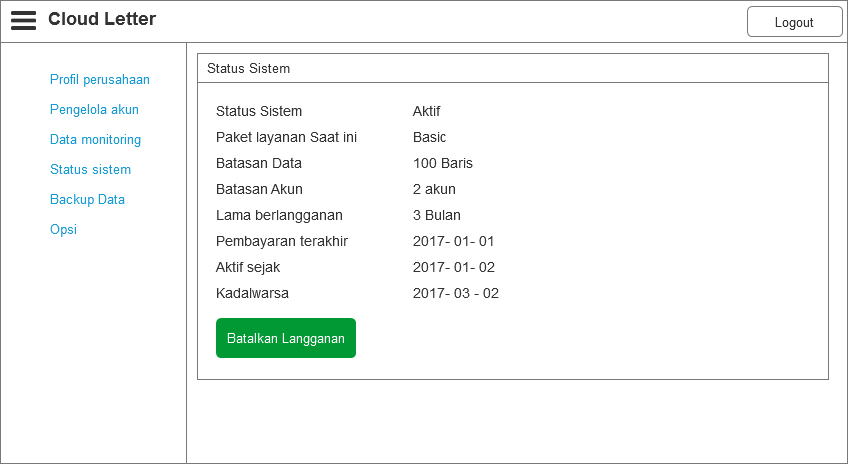
Pada desain halaman data monitoring yang ditujukan pada Gambar 3. 48, *user* admin dapat memantau atau melihat jumlah sumber daya layanan yang telah digunakan. Tampilan data monitoring dibagi menjadi 4 panel yaitu surat masuk, surat keluar, surat disposisi, dan user.



Gambar 3. 48 Rancangan Halaman Data Monitoring

1. Rancangan Halaman Status Sistem

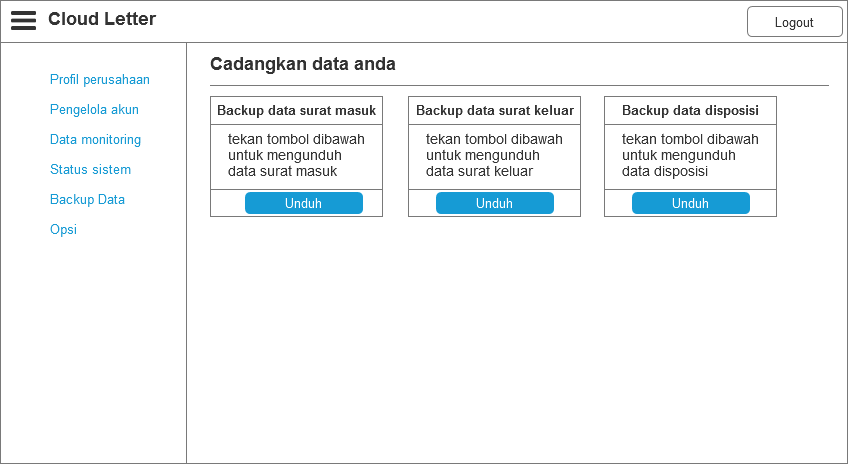
Rancangan halaman status sistem yang ditunjukan pada Gambar 3. 49 menampilkan informasi tentang status sistem, mulai dari status paket layanan *user* yang telah aktif, tanggal aktivasi dan batas waktu langganan paket layanan.



Gambar 3. 49 Rancangan Halaman Status Sistem

1. Rancangan Halaman *Backup Data*

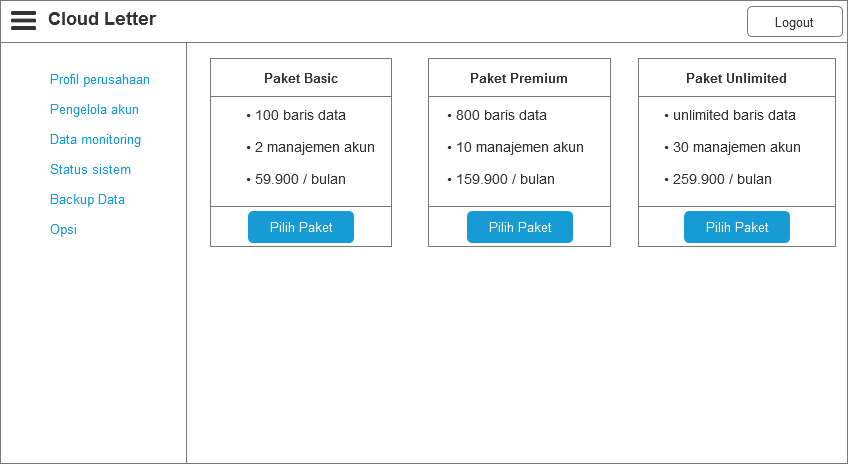
Tampilan rancangan halaman backup data dapat dilihat pada Gambar 3. 50 menampilkan panel yang masing masingnya terdapat tombol unduh disetiap data surat yang masing masing memiliki fungsi yang berbeda.



Gambar 3. 50 Rancangan Halaman Backup Data

1. Rancangan Halaman Manipulasi Layanan Pengguna

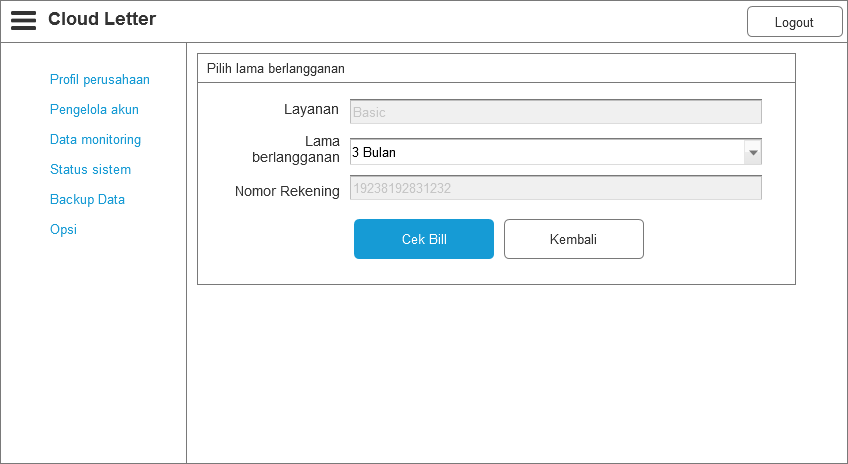
Pada halaman manipulasi layanan pengguna terdapat panel-panel yang menunjukan paket layanan yang tersedia, tampilan yang dapat dilihat pada Gambar 3. 51 juga terdapat button untuk memilih paket yang diinginkan.



Gambar 3. 51 Rancangan Halaman Manipulasi Layanan

1. Rancangan Halaman Form Lama Berlangganan

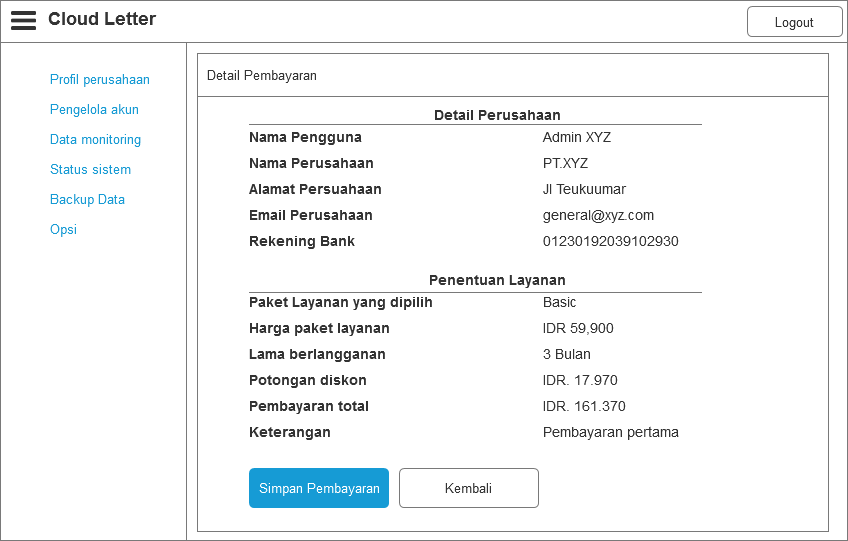
Rancangan halaman form lama berlangganan yang ditunjukan pada Gambar 3. 52 menampilkan jumlah bulan langganan sebagai salah satu proses pembayaran. User dapat memilih lama langganan pada *combobox.*



Gambar 3. 52 Rancangan Halaman Pilih Lama Langganan

1. Rancangan Halaman Detail Pembayaran

Rancangan halaman detail pembayaran yang ditunjukan pada Gambar 3. 53 menampilkan detail pembayaran mulai dari informasi perusahaan dan penentuan layanan juga jumlah nominal yang harus dibayarkan.



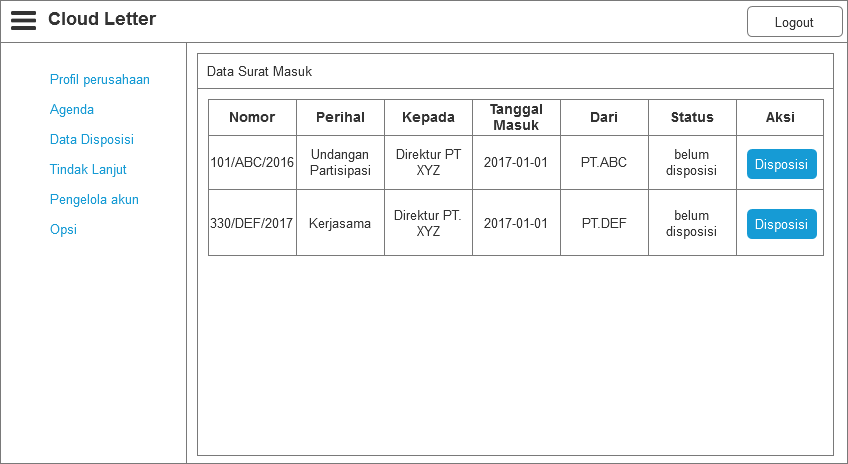
Gambar 3. 53 Rancangan Halaman Detail Pembayaran

* + 1. **Rancangan Tampilan Antar Muka *User* Kepala**

Halaman ini digunakan *user* kepala untuk mengelola sistem yang digunakan pada suatu perusahaan. Berikut rancangan halaman *user* kepala pada perekayasaan sistem *cloud computing* surat menyurat.

1. Rancangan Halaman Surat Masuk *User* Kepala

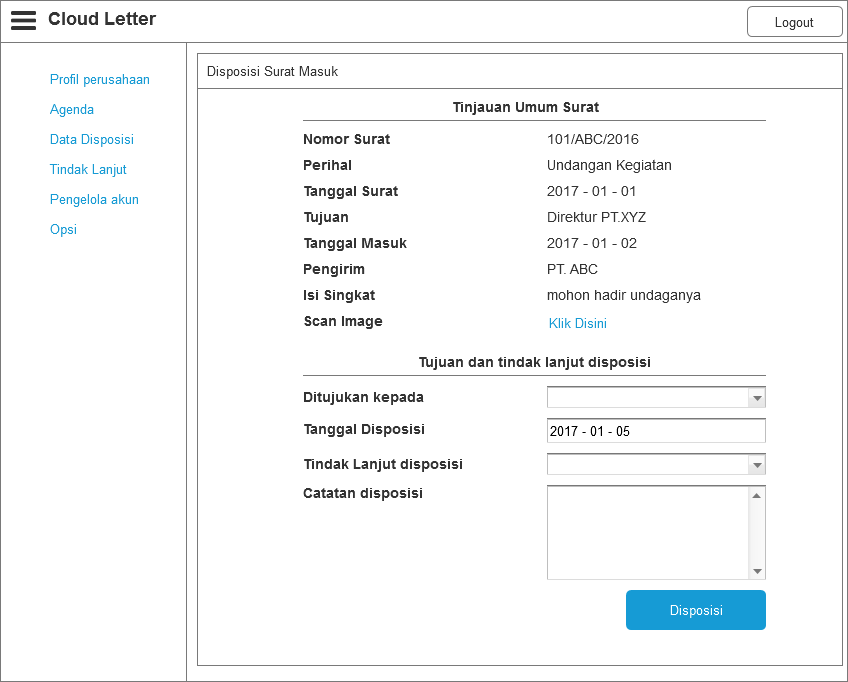
Pada rancangan halaman surat masuk yang ditunjukan pada Gambar 3. 54 menampilkan data surat masuk kepada user kepala yang tampilkan ke dalam bentuk tabel. Terdapat juga sebuah tombol untuk menuju halaman disposisi.

****

Gambar 3. 54 Rancangan Halaman Data Surat Masuk User Kepala

1. Rancangan Halaman Form Disposisi

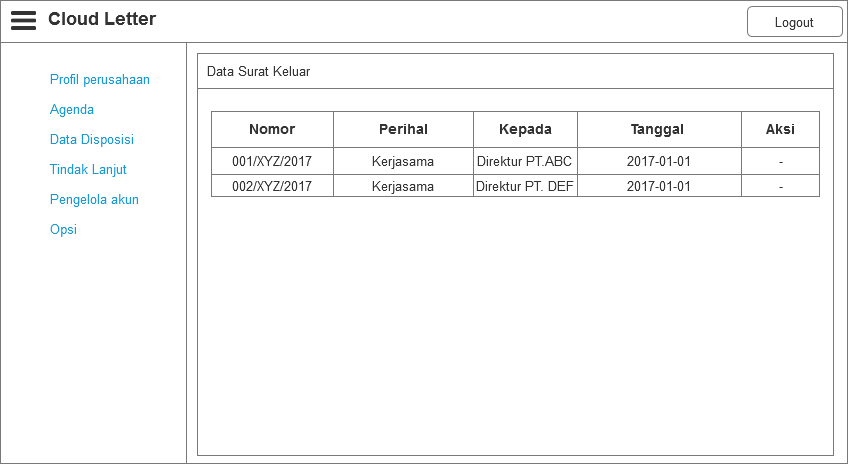
Halaman form disposisi yang ditunjukan pada Gambar 3. 55 menampilkan data detail surat masuk dan form disposisi, dimana user kepala dapat melakukan disposisi surat masuk ke *user* operator yang dituju.

****

Gambar 3. 55 Rancangan Halaman Disposisi Surat Masuk

1. Rancangan Halaman Surat Keluar

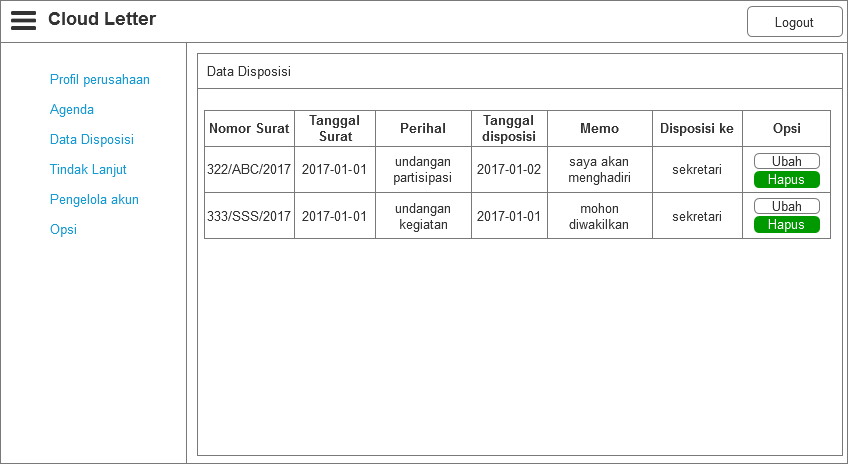
Pada Gambar 3. 56 ditunjukan tapilan rancangan halaman surat keluar *user* kepala yang dimana data surat keluar ditampilkan dalam bentuk tabel.

****

Gambar 3. 56 Rancangan Halaman Surat Keluar

1. Rancangan Halaman Data Disposisi

Halaman data disposisi yang ditunjukan pada Gambar 3. 57 menampilkan data disposisi surat masuk oleh *user* kepala dimana data ditampilkan dalam format tabel.

****

Gambar 3. 57 Rancangan Halaman Data Disposisi

1. Rancangan Halaman Ubah Data Disposisi

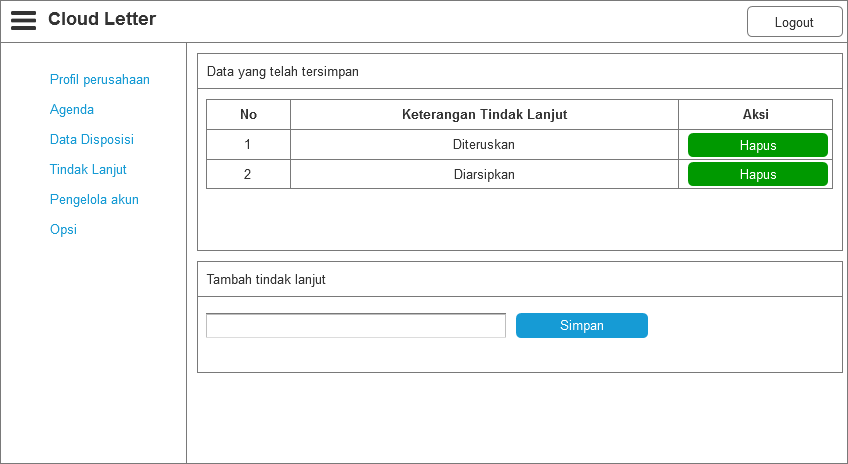
Halaman data disposisi yang ditunjukan pada Gambar 3. 58 menampilkan data detail disposisi surat masuk beserta form ubah data disposisi.

****

Gambar 3. 58 Rancangan Halaman Ubah Disposisi

1. Rancangan Halaman Tindak Lanjut

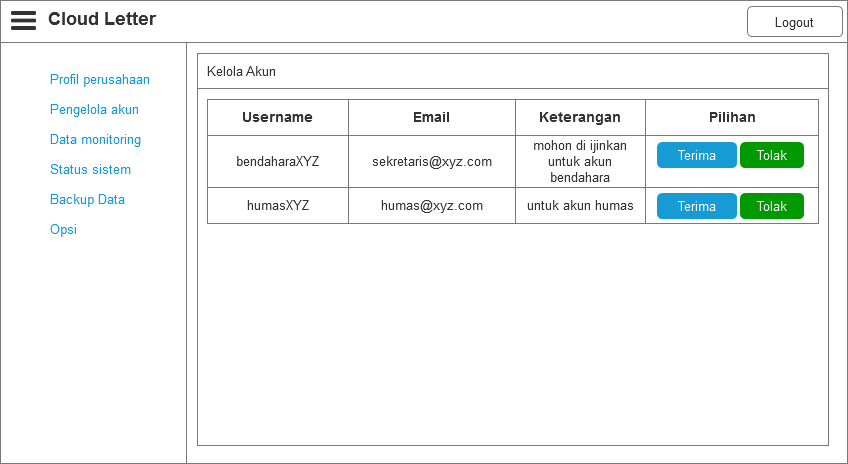
Halaman tindak lanjut disposisi yang ditunjukan pada Gambar 3. 59 menampilkan data tindak lanjut diposisi beserta form tambah keterangan tindak lanjut disposisi.

****

Gambar 3. 59 Rancangan Halaman Tindak Lanjut

1. Rancangan Halaman *Approve* Akun Operator

Pada rancangan halaman *approve* akun operator pada Gambar 3. 60 menampilkan data akun operator yang diajukan oleh *user* admin kepada *user* kepala agar dapat disetujui atau ditolak.

****

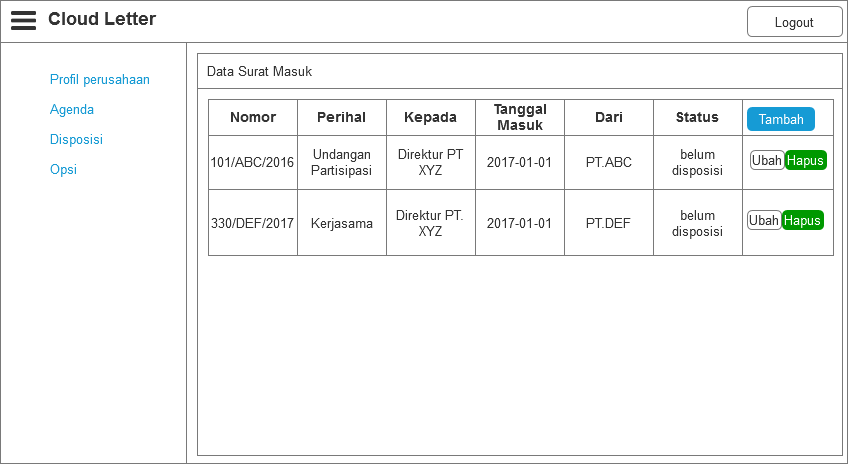
Gambar 3. 60 Rancangan Halaman Approve Akun Operator

* + 1. **Rancangan Tampilan Antar Muka *User* Operator**

Halaman ini digunakan *user* kepala untuk mengelola sistem yang digunakan pada suatu perusahaan. Berikut rancangan halaman *user* kepala pada perekayasaan sistem *cloud computing* surat menyurat.

1. Rancangan Halaman Surat Masuk *User* Operator

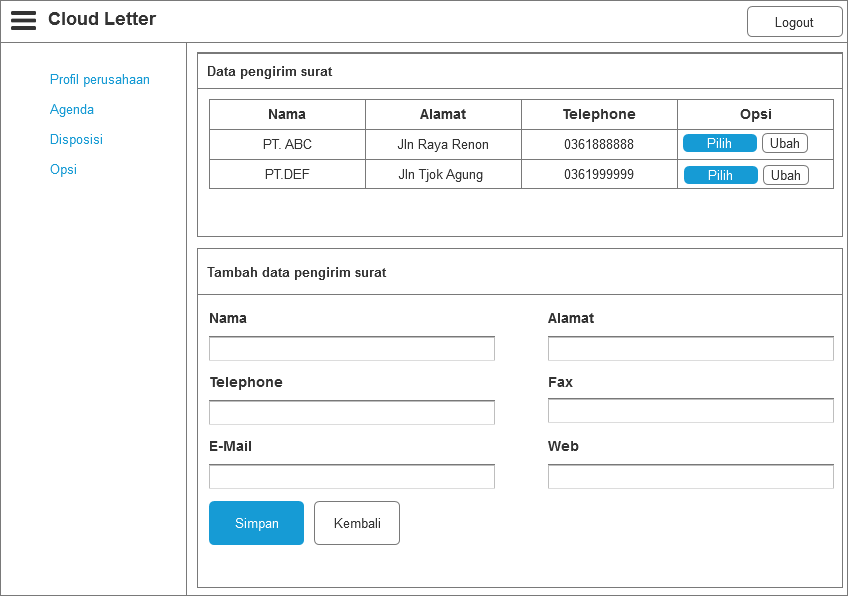
Pada rancangan halaman surat masuk yang ditunjukan pada Gambar 3. 61 menampilkan data surat masuk tampilkan ke dalam bentuk tabel. Terdapat juga sebuah tombol untuk tambah, ubah, dan hapus surat masuk.



Gambar 3. 61 Rancangan Halaman Surat Masuk User Operator

1. Rancangan Halaman Tambah Data Pengirim

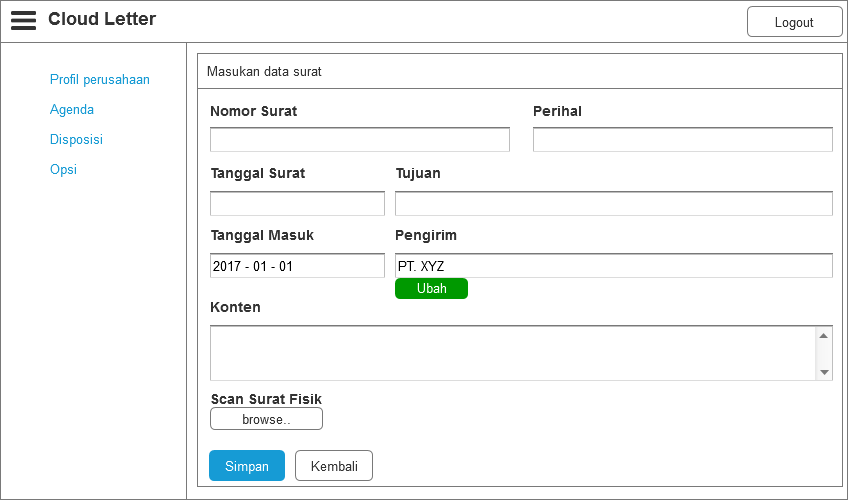
Pada rancangan halaman tambah data pengirim yang ditunjukan pada Gambar 3. 62 menampilkan data pengirim yang tampilkan ke dalam bentuk tabel. Terdapat juga sebuah form tambah pengirim untuk menambah data pengirim.



Gambar 3. 62 Rancangan Halaman Tambah Data Pengirim

1. Rancangan Halaman Tambah Data Surat Masuk

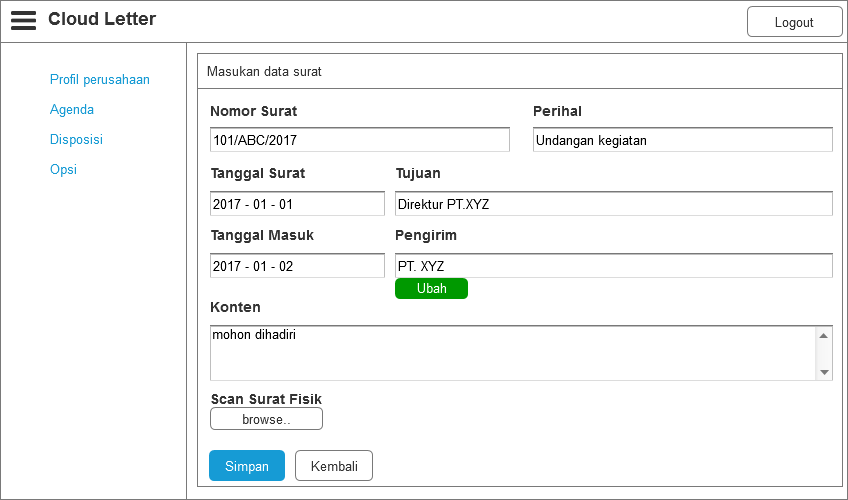
Pada Gambar 3. 63 ditunjukan rancangan tampilan form *input* surat masuk dimana pada form tersebut terdapat beberapa *field* yang harus diinputkan.



Gambar 3. 63 Rancangan Halaman Tambah Surat Masuk

1. Rancangan Halaman Ubah Data Surat Masuk

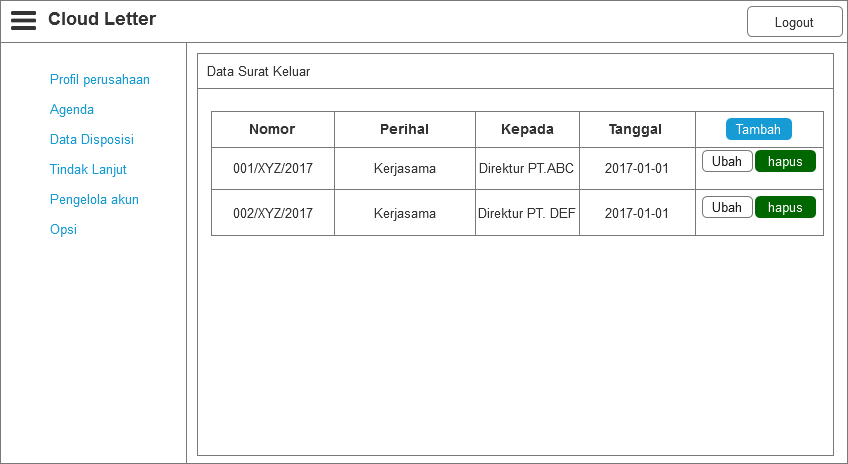
Pada Gambar 3. 64 ditunjukan rancangan tampilan form ubah surat masuk dimana pada form tersebut terdapat beberapa *field* yang sudah terisi sesuai dengan data surat yang ingin diubah.



Gambar 3. 64 Rancangan Halaman Ubah Data Surat Masuk

1. Rancangan Halaman Data Surat Keluar

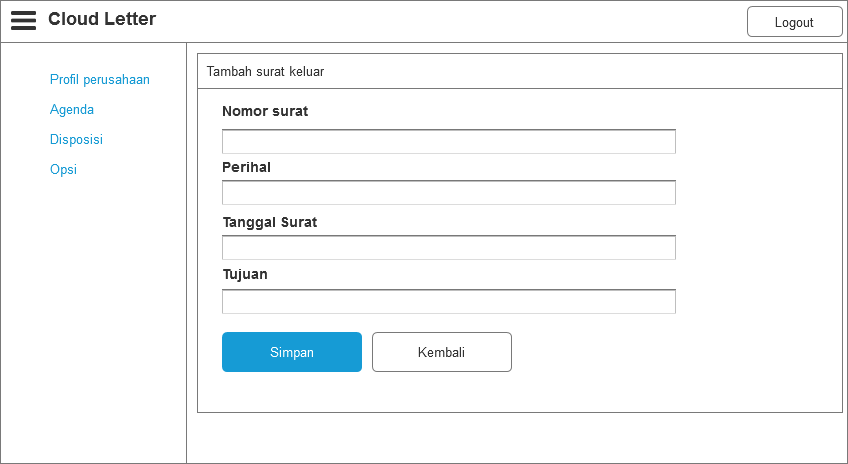
Pada Gambar 3. 65 ditunjukan tapilan rancangan halaman surat keluar yang dimana data surat keluar ditampilkan dalam bentuk tabel.



Gambar 3. 65 Rancangan Halaman Data Surat Keluar

1. Rancangan Halaman Tambah Data Surat Keluar

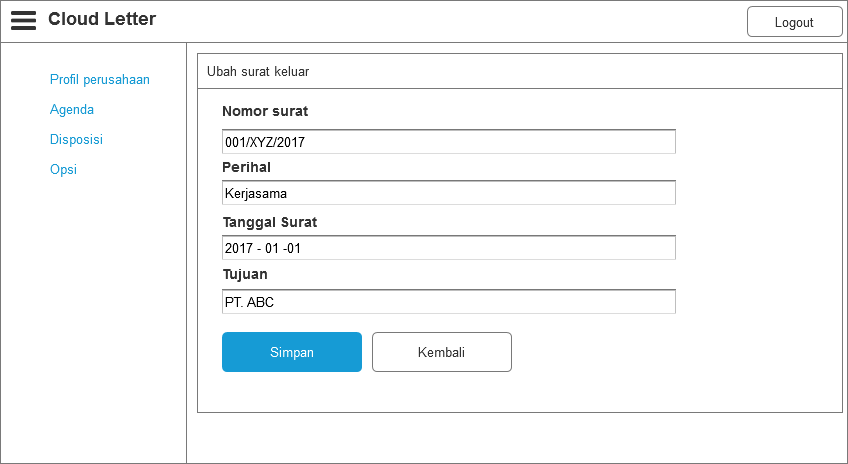
Pada Gambar 3. 66 ditunjukan tapilan rancangan halaman form tambah surat keluar yang didalamnya terdapat text field yang harus ditambahkan.



Gambar 3. 66 Rancangan Halaman Tambah Data Surat Keluar

1. Rancangan Halaman Ubah Data Surat Keluar

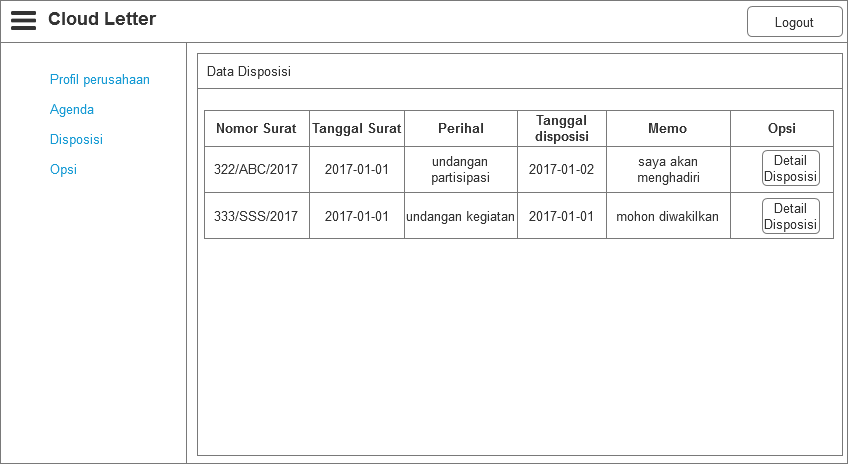
Pada Gambar 3. 67 ditunjukan tapilan rancangan halaman uba surat keluar yang dimana terdapat form yang secara langsung berisikan data surat yang hendak diubah.



Gambar 3. 67 Rancangan Halaman Ubah Data Surat Keluar

1. Rancangan Halaman Data Disposisi *User*

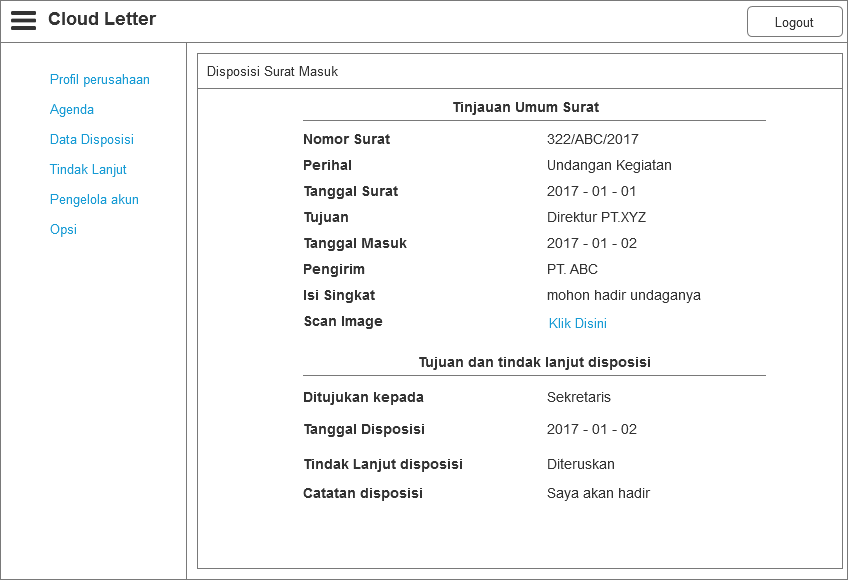
Halaman data disposisi yang ditunjukan pada Gambar 3. 68 menampilkan data disposisi surat masuk oleh *user* kepala dimana data ditampilkan dalam format tabel.



Gambar 3. 68 Rancangan Halaman Data Disposisi User

1. Rancangan Halaman Detail Disposisi

Pada halaman detail disposisi akan menampilkan data detail surat masuk dan juga detail disposisi. Rancangan halaman detail disposisi ditunjukan pada Gambar 3. 69.



Gambar 3. 69 Rancangan Halaman Detail Disposisi

**BAB IV**

**IMPLEMENTASI**

* 1. **Spesifikasi Perangkat Keras**

Spesifikasi *hardware* (perangkat keras) yang dapat digunakan dalam merancang dan mengimplementasikan sistem untuk pengguna(*user*) adalah.

1. *Processor* : *Intel Core i3*
2. *Memory* : 2 GB
3. *Hardisk* : 500 GB
4. *VGA* : *NVIDIA GeForce* 1GB
5. *Input Device* : *Keyboard* dan *Mouse*

Konfigurasi *hardware* dengan spesifikasi yang lebih tinggi dari spesifikasi di atas akan menghasilkan kinerja yang lebih optimal.

* 1. **Implementasi Sistem**

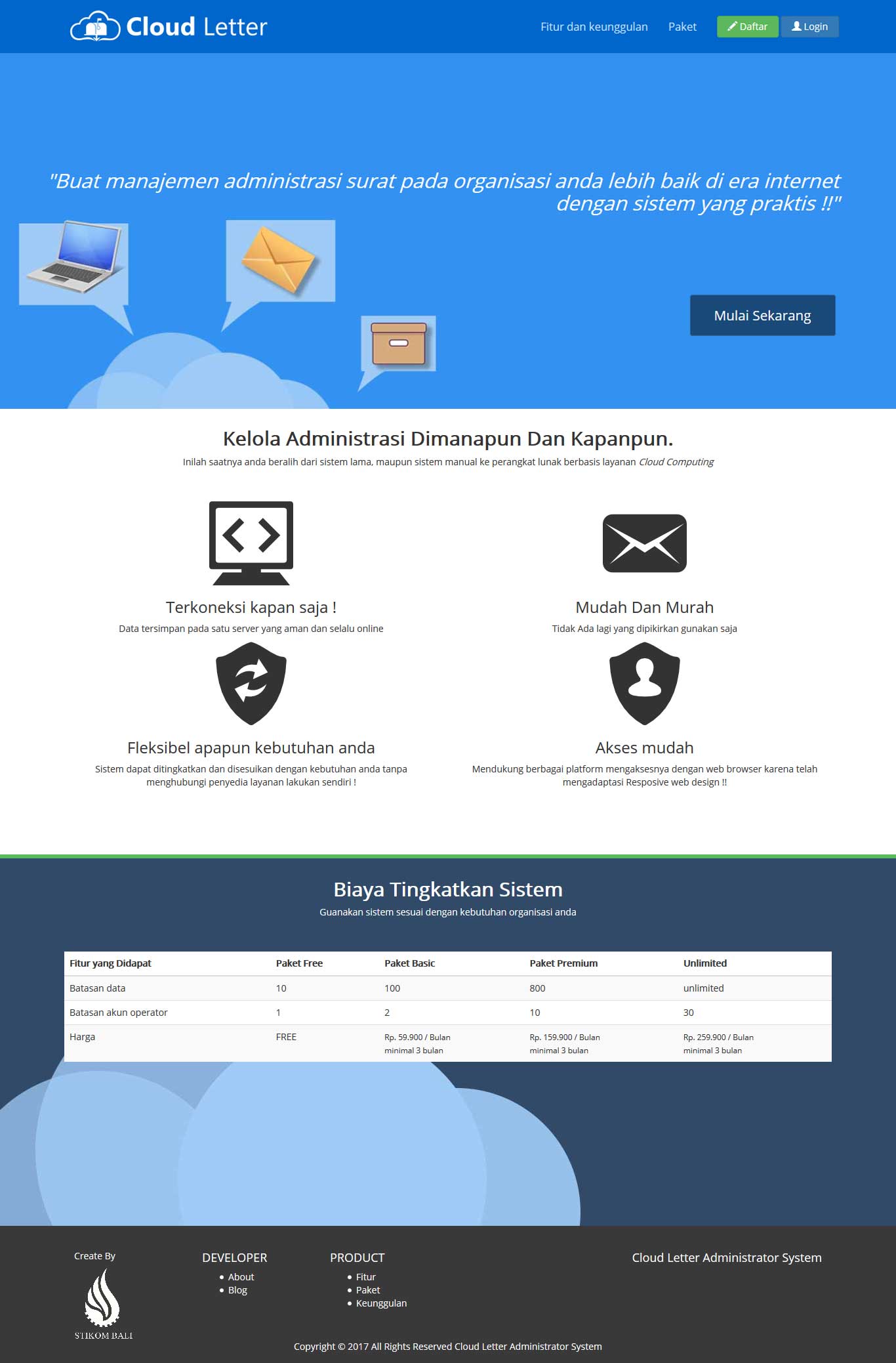
Setelah melakukan analisis terhadap kebutuhan dan kemudian dari permasalahan tersebut dilakukan perancangan sistem, selanjutnya perancangan tersebut diimplementasikan ke dalam kode program sehingga dihasilkanya sistem manajemen surat menyurat dengan berbasiskan *Software As A Service Cloud Computing*. Berikut adalah *screenshot interface* hasil dari implementasi terhadap sistem.

* + 1. **Tampilan Depan (*Front End*)**

Halaman ini digunakan untuk mengelola menampilkan fungsi-fungsi awal yang digunakan untuk memulai mengakses sistem yang telah dibuat. Berikut rancangan halaman *front end* pada perekayasaan sistem.

1. Halaman Beranda (*home)*

Halaman *home* yang ditujukan pada Gambar 4.1 merupakan halaman awal ketika mengakses alamat situs web. Pada halaman ini akan menyediakan informasi–informasi kepada pengunjung terkait sistem *cloud computing* surat menyurat dimana pada halaman depan dibagi menjadi empat bagian.



Gambar 4.1 Halaman beranda (*home)*

Potongan kode program pada halaman beranda (home) dapat dilihat pada Gambar 4.2.

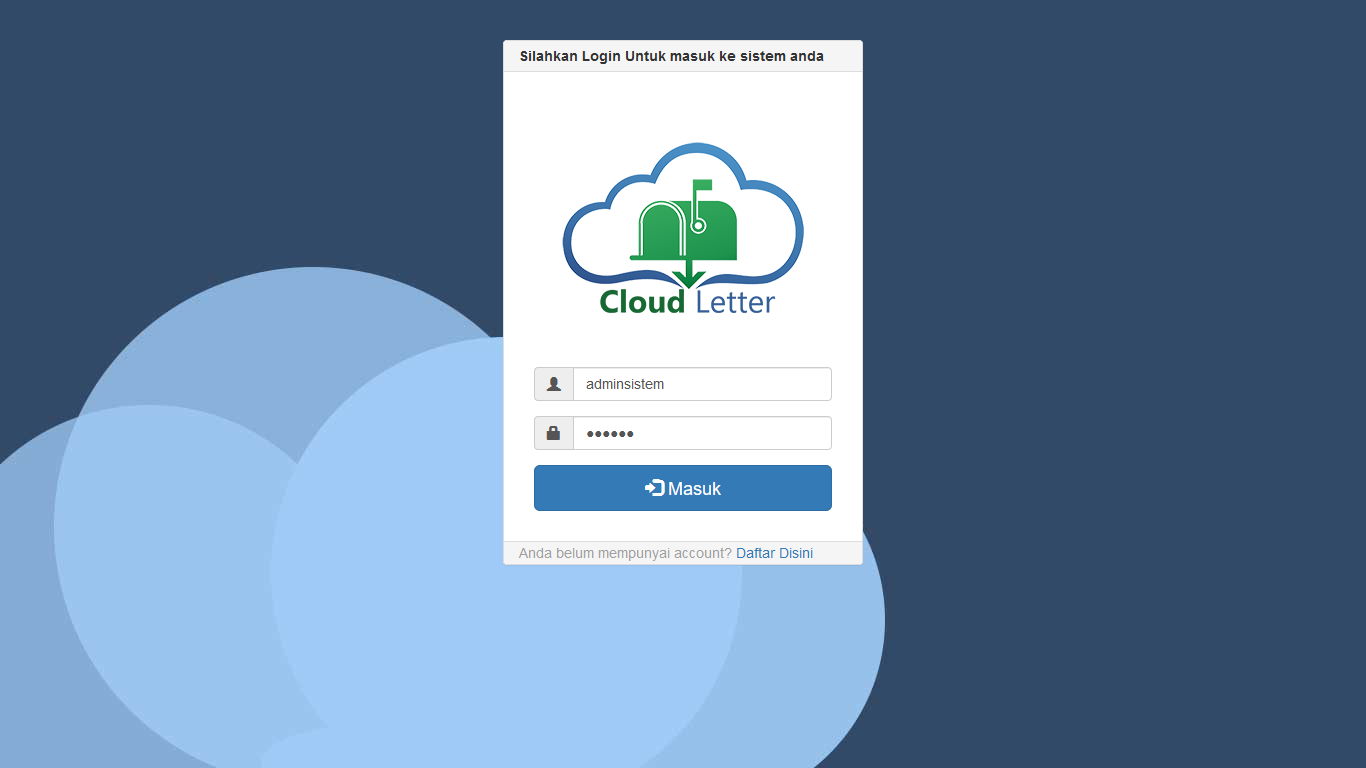
|  |
| --- |
| class Welcome extends CI\_Controller {  public function index()  {  $data[‘service’] = $this->model\_service->select\_all();  $this->load->view(‘home\_page',$data);  }  } |

Gambar 4.2 Potongan kode program beranda *(home)*

Potongan kode program pada Gambar 4.2 merupakan kode program yang berfungsi untuk menampilkan halaman *home* *page*.

1. Halaman Login

Pada halaman loginyang ditunjukan pada Gambar 4.3, *user* yang sudah terdaftar dapat melakukan logindengan memasukan *username* dan *password* pada kolom yang telah disediakan sebagai autentifikasi agar dapat menggunakan sistem.



Gambar 4.3 Halaman Login

Potongan kode program pada form login dapat dilihat pada Gambar 4.4.

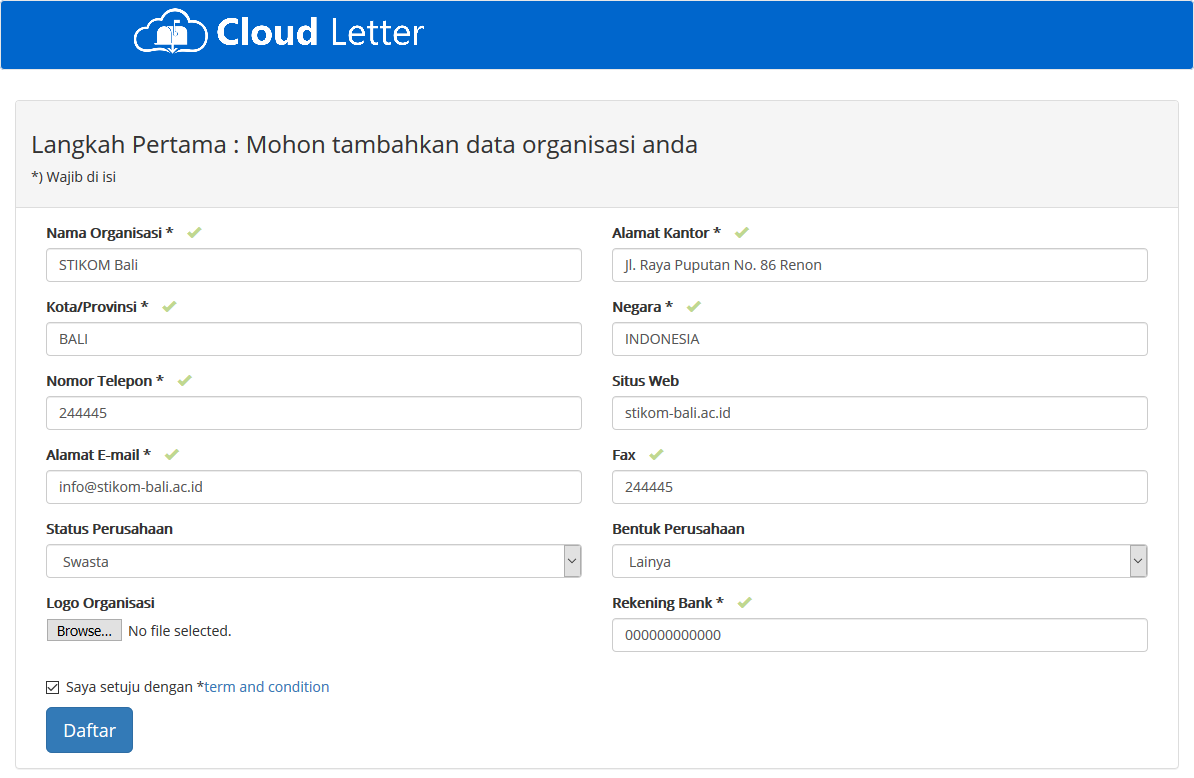
|  |
| --- |
| public function login(){  $this->load->model('model\_users');  $valid\_user = $this->model\_users->check\_credential();  if ($valid\_user == FALSE) {  $this->session->set\_flashdata('error','Wrong Username or Password');  redirect('login');  }else{  switch ($valid\_user->accl\_id) {  case 1 : redirect('dashboard/admin');  break;  case 2 : $this->activation\_check();  break;  case 3 : redirect('dashboard/user');  break;  case 4 : redirect('dashboard/user');  break;  default: break;  }} |

Gambar 4.4 Potongan kode program login

Potongan kode program pada Gambar 4.4 merupakan kode program yang berfungsi untuk menampilkan mencocokan username dan password yang dimasukan oleh user ke basis data kemudian mengarahkan ke halaman masing-masing user.

1. Halaman Registrasi Organisasi

Halaman registrasi organisasi yang ditunjukan pada Gambar 4.5 digunakan untuk registrasi organisasi atau pendaftaran perusahaan untuk didapatkan data perusahaan sebagai langkah pertama dalam registrasi sistem *cloud computing* surat menyurat.



Gambar 4.5 Registrasi organisasi

Potongan kode program pada halaman registrasi perusahaan dapat dilihat pada Gambar 4.6.

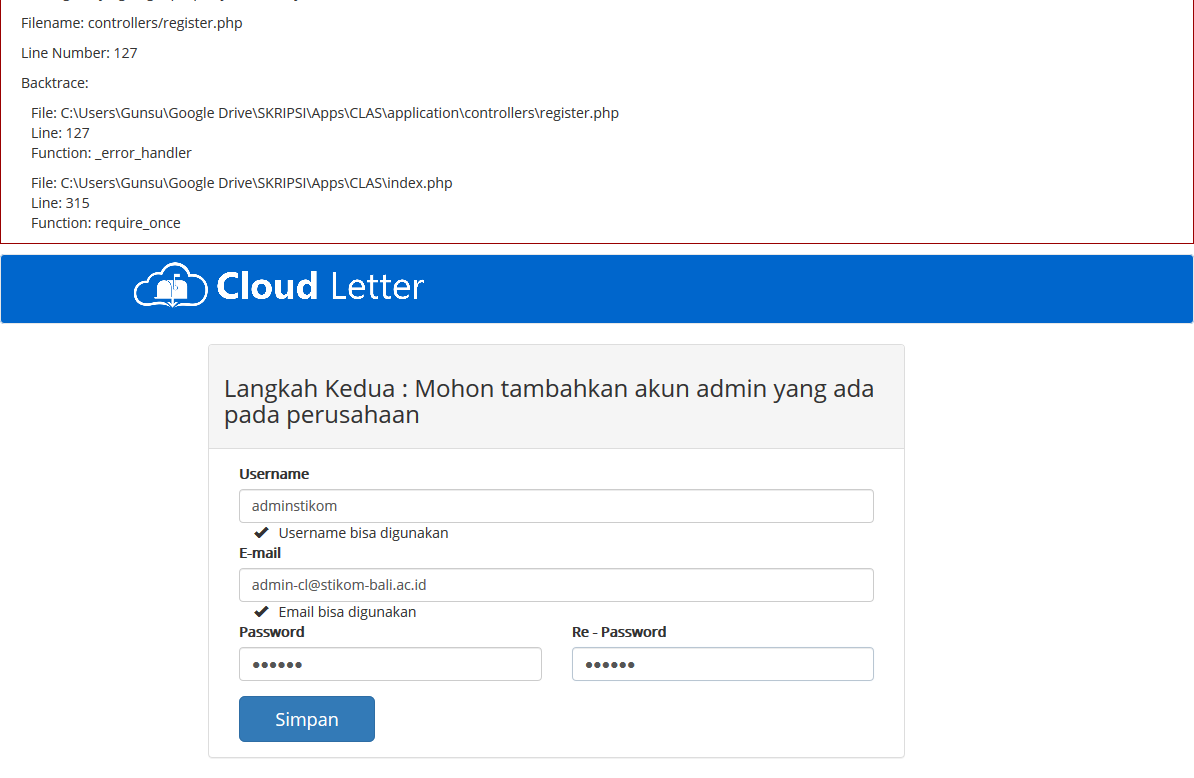
|  |
| --- |
| public function add\_company\_data(){  $data\_comp = array(  'comp\_name' => set\_value('cname'),'comp\_address' => set\_value('caddress'),'comp\_city' => set\_value('ccity'),'comp\_country' => set\_value('ccountry'), 'comp\_phone' => set\_value('cphone'),'comp\_website' => set\_value('cweb'), 'comp\_email' => set\_value('cemail'), 'comp\_fax' => set\_value('cfax'),'comp\_status' => set\_value('cstatus'),'comp\_type' => set\_value('ctype'),'comp\_reknum' => set\_value('crek'),$this->model\_users->input\_company($data\_comp);  $this->session->set\_userdata('reg\_code', $rand\_code);  redirect('register/add\_admin\_acc'); |

Gambar 4.6 Potongan kode program registrasi organisasi

Potongan kode program pada Gambar 4.6 merupakan kode program yang berfungsi untuk menyimpan data registrasi yang dimasukan oleh user ke Basis Data kemudian mengarahkan ke halaman registrasi admin.

1. Halaman Registrasi *User* Admin

Halaman registrasi perusahaan yang ditunjukan pada Gambar 4.7 digunakan untuk registrasi admin perusahaan untuk didapatkan data akun admin perusahaan sebagai langkah kedua dalam registrasi sistem *cloud computing* surat menyurat.



Gambar 4.7 Halaman registrasi akun *user* admin

Potongan kode program pada halaman registrasi user admin dapat dilihat pada Gambar 4.8.

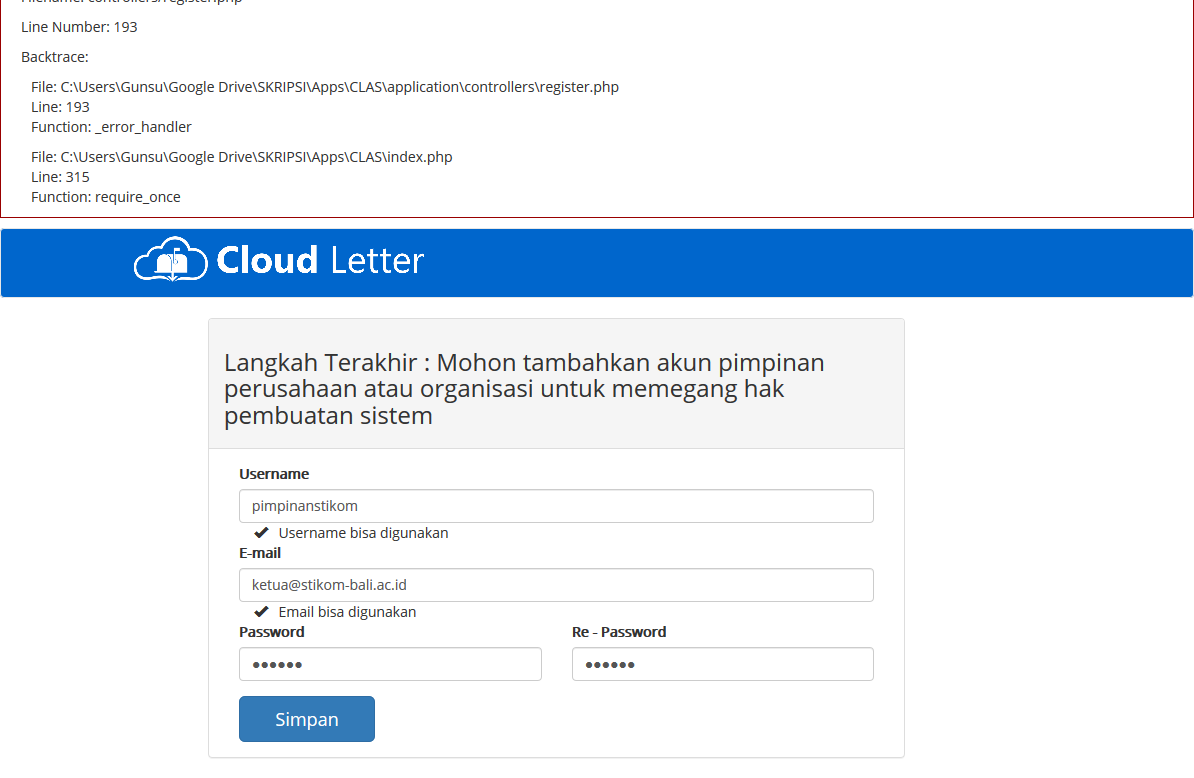
|  |
| --- |
| public function add\_admin\_acc(){  $data\_register = array(  'acc\_usrn' => $username, 'acc\_pwd' => $password, 'acc\_email' => $email, 'accl\_id' => $user\_lvl, 'srvc\_id' => 1, 'comp\_id' => $comp\_id, 'acc\_status' => 1);  $this->model\_users->input\_register($data\_register);  redirect('register/add\_head\_account');  } |

Gambar 4.8 Potongan kode program registrasi *user* admin

Potongan kode program pada Gambar 4.8 merupakan kode program yang berfungsi untuk menyimpan data registrasi admin yang dimasukan oleh *user* ke basis data kemudian mengarahkan ke halaman registrasi user kepala.

1. Halaman Registrasi *User* Kepala

Halaman registrasi perusahaan yang ditunjukan pada Gambar 4.9 digunakan untuk registrasi kepala perusahaan untuk didapatkan data akun kepala perusahaan sebagai langkah terakhir dalam registrasi sistem *cloud computing* surat menyurat.



Gambar 4.9 Halaman Registrasi *user* kepala.

Potongan kode program pada halaman registrasi user kepala dapat dilihat pada Gambar 4.10.

|  |
| --- |
| public function add\_head\_acc(){  $data\_register = array(  'acc\_usrn' => $username, 'acc\_pwd' => $password, 'acc\_email' => $email,  'accl\_id' => $user\_lvl, 'srvc\_id' => 1, 'comp\_id' => $comp\_id, 'acc\_status' => 1);  $this->model\_users->input\_register($data\_register); redirect('dashboard/user'); |

Gambar 4.10 Potongan kode program registrasi *user* kepala

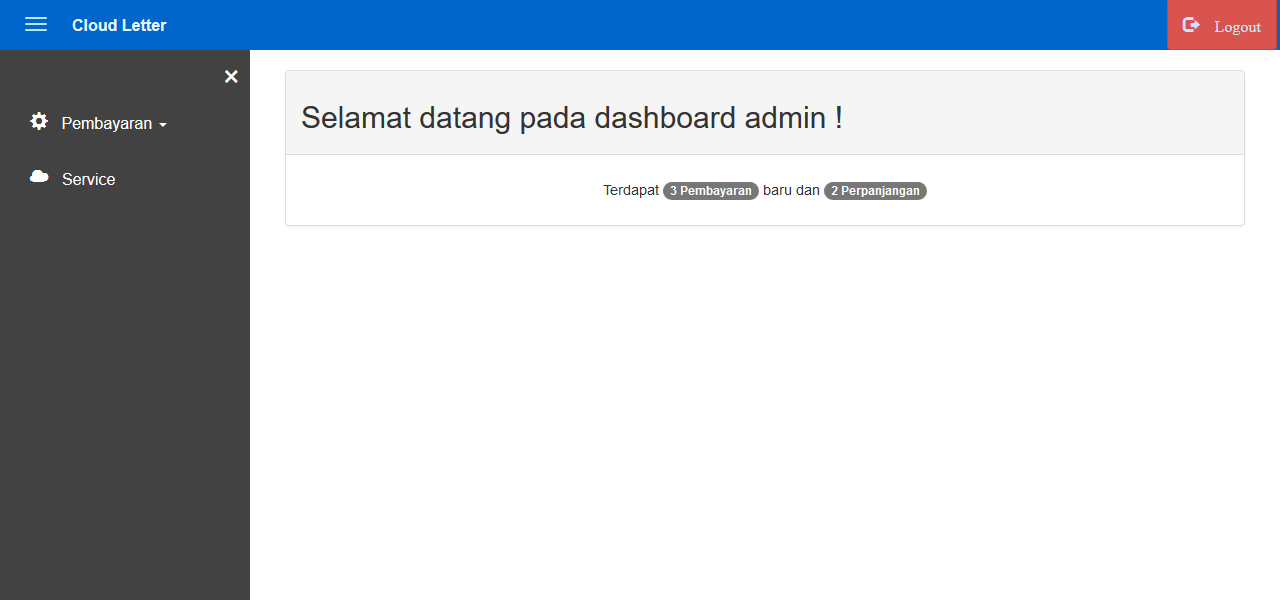
Potongan kode program pada Gambar 4.10 merupakan kode program yang berfungsi untuk menyimpan data registrasi *user* kepala yang dimasukan oleh user ke basis data kemudian mengarahkan ke tampilan antar muka admin.

* + 1. **Tampilan Antar Muka Admin**

Halaman ini digunakan *admin* sistem untuk mengelola aktivasi layanan atau digunakan untuk memanipulasi paket layanan dan mengubah paket layanan yang telah dibuat. Berikut halaman admin pada sistem *cloud computing* surat menyurat.

1. Halaman *Dashboard* Admin Sistem

Pada halaman *dashboard* *admin* yang ditunjukan pada Gambar 4.11, admin akan dapat melihat menu *side nav* untuk mengarahkan ke menu selanjutnya.



Gambar 4.11 *Dashboard* admin sistem

Potongan kode program pada halaman dashboard admin sistem dapat dilihat pada Gambar 4.12.

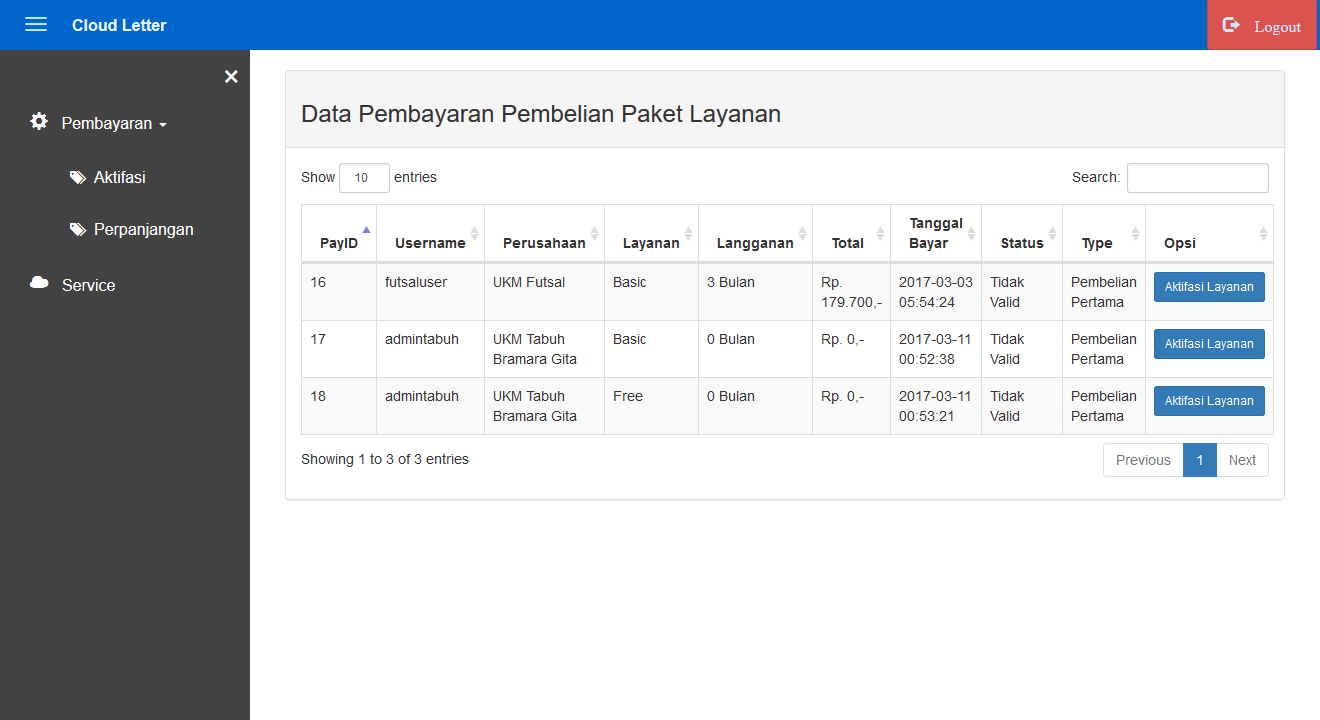
|  |
| --- |
| public function admin()  {  $this->load->view('backend/dash\_admin');  } |

Gambar 4.12 Potongan kode program *dashboard* admin

Potongan kode program pada Gambar 4.12 merupakan kode program yang berfungsi untuk menampilkan *dashboard* admin sistem.

1. Halaman Aktifasi dan Perpanjangan

Pada halaman aktifasi, *admin* akan dapat melihat data pembayaran dengan status pembayaran pertama oleh *user* yang telah melakukan pembayaran. Admin dapat melakukan aktifasi paket layanan yang dibayarkan oleh *user* agar dapat menggunakan paket layanan.



Gambar 4.13 Halaman pembayaran paket layanan.

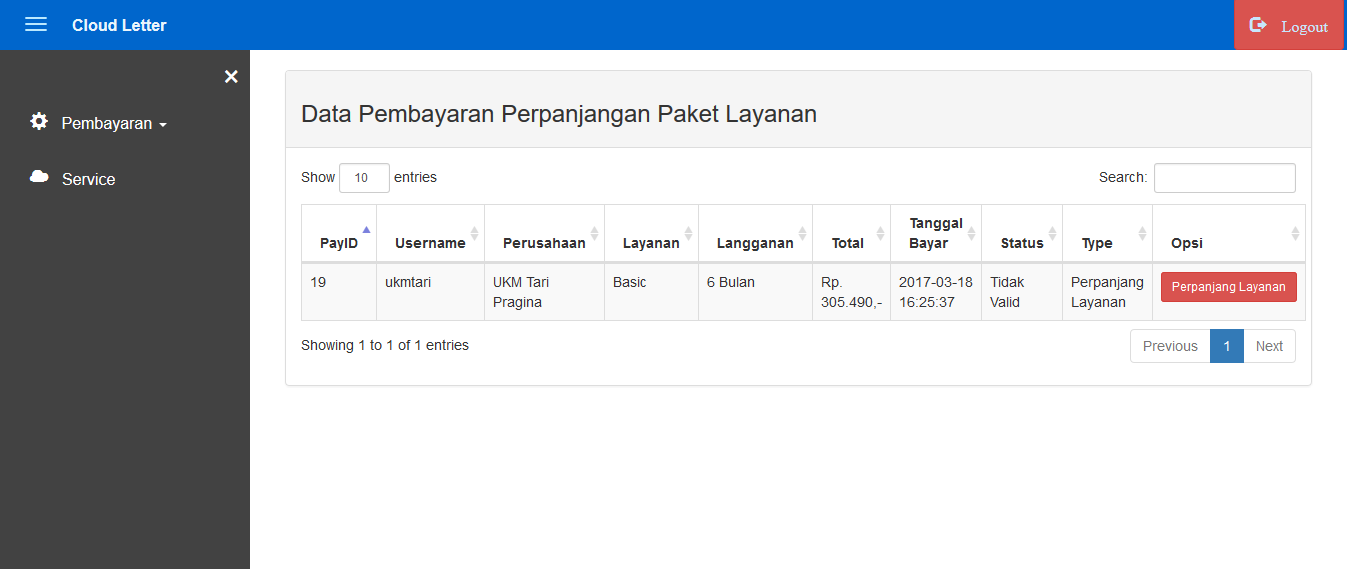
Potongan kode program pada halaman pembayaran paket layanan dapat dilihat pada Gambar 4.14.

|  |
| --- |
| public function view\_payment(){  $comp\_id = $this->session->userdata('comp\_id');  $aktifasi = $this->model\_activation->select\_by\_comp($comp\_id);  $data['payment'] = $this->model\_payment->select\_all();  $this->load->view('backend/dash\_payment\_view', $data);  } |

Gambar 4.14 Potongan kode program halaman pembayaran paket

Potongan kode program pada Gambar 4.14 merupakan kode program yang berfungsi untuk menampilkan data paket layanan yang belum diaktifkan oleh admin.

Pada halaman pembayaran perpanjangan layanan yang ditunjukan pada 4.15 *admin* akan dapat melihat data pembayaran perpanjangan dengan status pembayaran perpanjang oleh user yang telah melakukan pembayaran. Admin dapat melakukan aktifasi paket layanan yang dibayarkan oleh user agar dapat menggunakan paket layanan.



Gambar 4.15 Halaman perpanjangan layanan.

Potongan kode program pada halaman perpanjangan layanan dapat dilihat pada Gambar 4.16.

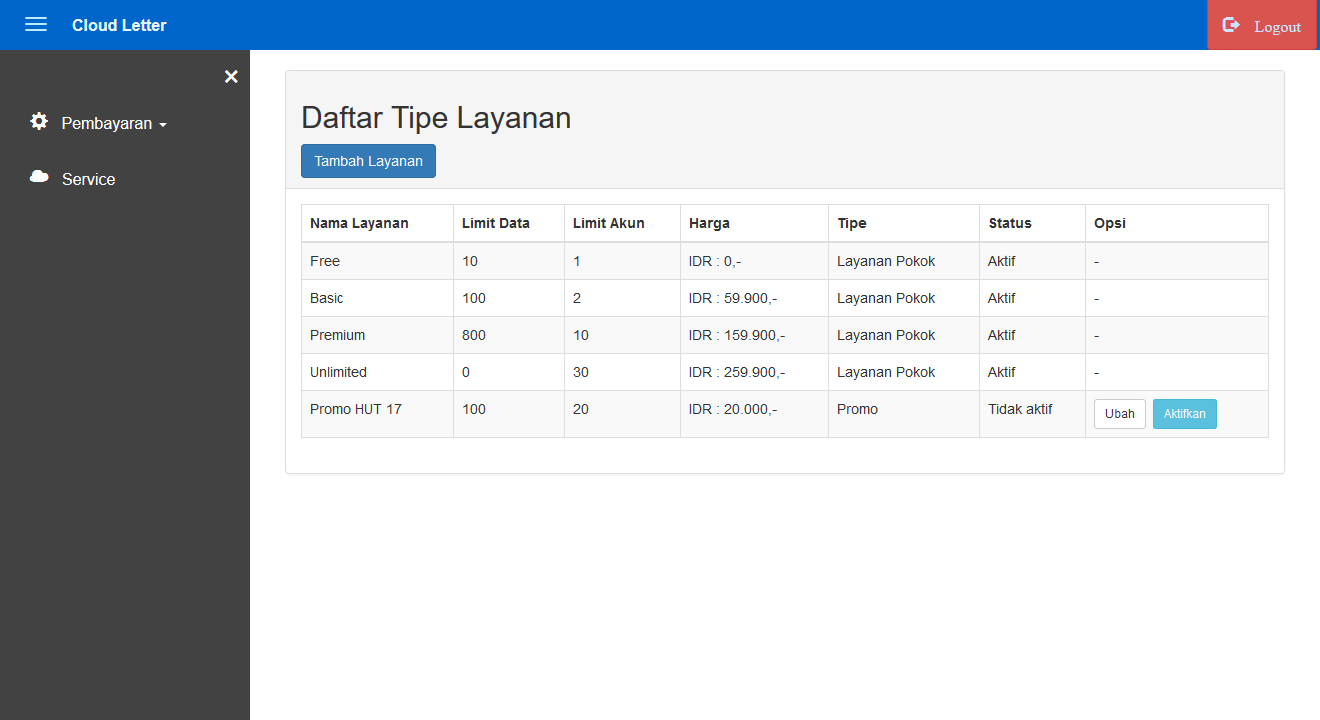
|  |
| --- |
| public function continue\_subscrb(){  $comp\_id = $this->session->userdata('comp\_id');  $aktifasi = $this->model\_activation->select\_by\_comp($comp\_id);  $data['payment\_langganan'] = $this->model\_payment->select\_sort\_by\_type();  $this->load->view('backend/dash\_payment\_subscrb', $data);  } |

Gambar 4.16 Potongan kode program halaman perpanjangan layanan

Potongan kode program pada Gambar 4.16 merupakan kode program yang berfungsi untuk menampilkan data perpanjangan layanan yang belum diaktifasi oleh admin sistem.

1. Halaman Data Paket Layanan

Pada halaman paket layanan yang ditujukan pada Gambar 4.17 *admin* akan dapat mengelola data paket yang disediakan untuk sistem *cloud computing* surat menyurat. *Admin* dapat mengubah dan menambah paket layanan sesuai dengan kebutuhan sistem.



Gambar 4.17 Halaman data paket layanan

Potongan kode program pada halaman data paket layanan dapat dilihat pada Gambar 4.18.

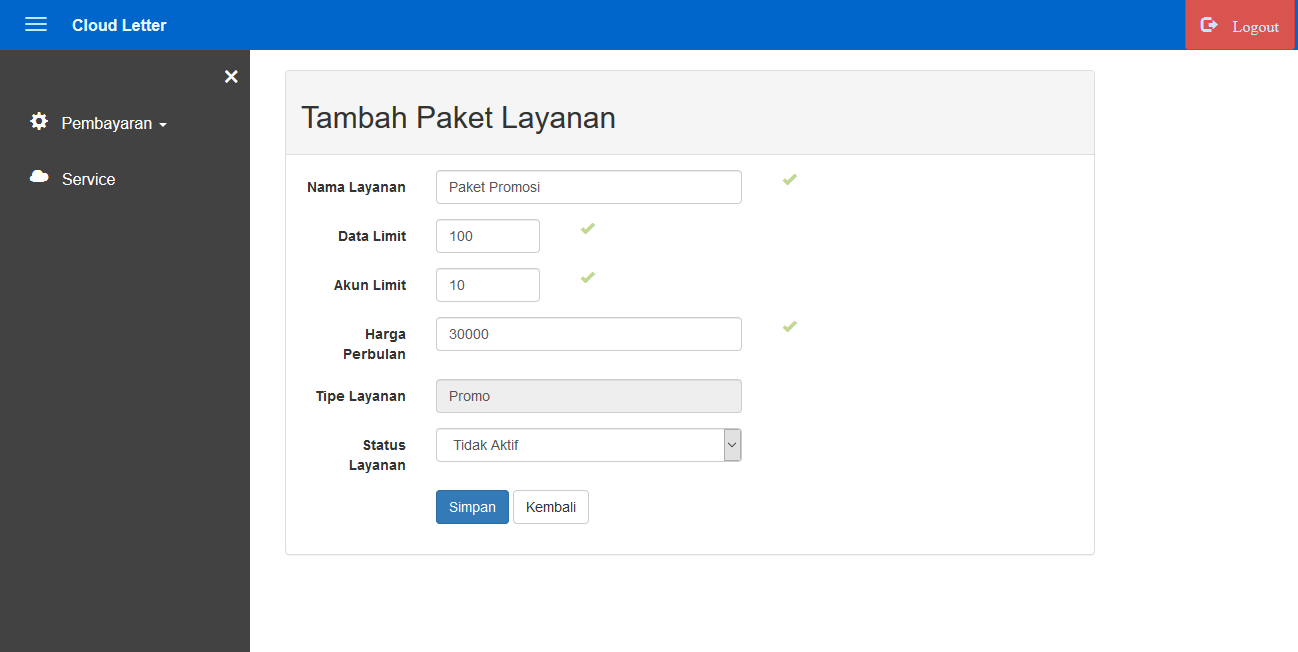
|  |
| --- |
| public function service(){  $data['service'] = $this->model\_service->select\_all();  $this->load->view('backend/dash\_data\_service',$data);  } |

Gambar 4.18 Potongan kode program halaman data paket layanan

Potongan kode program pada Gambar 4.18 merupakan kode program yang berfungsi untuk menampilkan data paket layanan.

1. Halaman Tambah Paket Layanan

Pada halaman tambah paket layanan yang ditujukan pada Gambar 4.19, *admin sistem* dapat menambah data paket yang disediakan untuk sistem *cloud computing* surat menyurat.



Gambar 4.19 Halaman Tambah Paket Layanan

Potongan kode program pada halaman tambah paket layanan dapat dilihat pada Gambar 4.20.

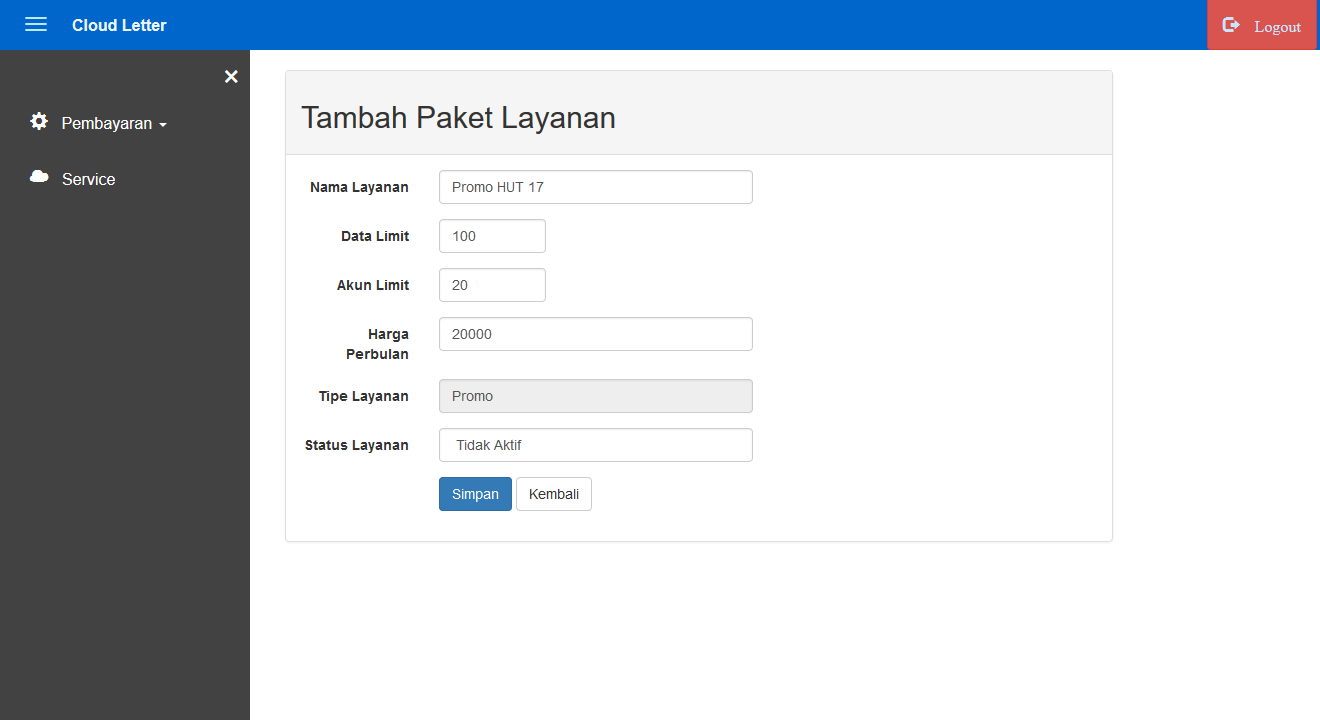
|  |
| --- |
| public function service(){  $data\_layanan = array(  'srvc\_name' => set\_value('name'), 'srvc\_data\_lmt' => ‘set\_value('data\_lmt'),'srvc\_bag\_lmt' => set\_value('bag\_lmt'), 'srvc\_price' => set\_value('price'), 'srvc\_promo' => set\_value('promo'), 'srvc\_status' => set\_value('status') );  $simpan = $this->model\_service->insert\_service($data\_layanan);  if($simpan == FALSE){  $this->session->set\_flashdata('error','Gagal disimpan');  redirect('dashboard/service/add\_service');  }else{  $this->session->set\_flashdata('info','Berhasil disimpan');  redirect('dashboard/admin/service'); }} |

Gambar 4.20 Potongan kode program tambah paket layanan

Potongan kode program pada Gambar 4.20 merupakan kode program yang berfungsi untuk menyimpan data paket layanan kedalam basis data.

1. Halaman Ubah Paket Layanan

Pada halaman ubah paket layanan yang ditujukan pada Gambar 4.21 *admin sistem* dapat mengubah data paket yang disediakan untuk sistem *cloud computing* surat menyurat sesuai dengan kebutuhan layanan.



Gambar 4.21 Halaman ubah paket layanan

Potongan kode program pada halaman ubah paket layanan dapat dilihat pada Gambar 4.22.

|  |
| --- |
| public function edit\_service(){  $data\_layanan = array(  'srvc\_name' => set\_value('name'), 'srvc\_data\_lmt' => set\_value('data\_lmt'),  'srvc\_bag\_lmt' => set\_value('bag\_lmt'), 'srvc\_price' => set\_value('price'), 'srvc\_promo' => set\_value('promo'), 'srvc\_status' => set\_value('status') );  $edit = $this->model\_service->update\_service($data\_layanan, $srvc\_id);  if($edit == FALSE){  $this->session->set\_flashdata('error','Gagal disimpan');  redirect('dashboard/service/edit\_service');  }else{  $this->session->set\_flashdata('info','Berhasil disimpan');  redirect('dashboard/admin/service'); }} |

Gambar 4.22 Potongan kode program ubah paket layanan

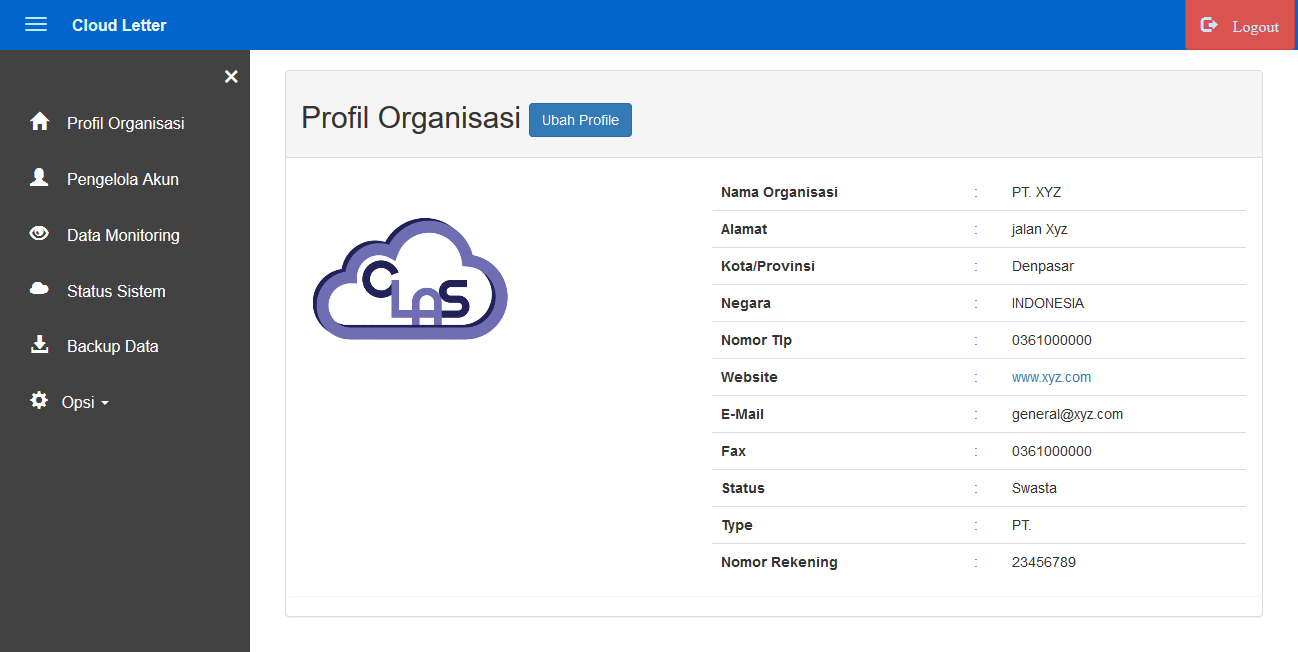
Potongan kode program pada Gambar 4.22 merupakan kode program yang berfungsi untuk menyimpan data ubah paket layanan yang disediakan oleh admin user kedalam basis data.

* + 1. **Tampilan Antar Muka *User* Admin**

Halaman ini digunakan *user* admin untuk mengelola sistem yang digunakan pada suatu perusahaan. Berikut halaman *admin* pada perekayasaan sistem *cloud computing* surat menyurat.

1. Halaman Profil Organisasi

Halaman profil perusahaan yang ditujukan pada Gambar 4.23 merupakan halaman dimana *user* admin*, user* kepala *dan user operator* dapat melihat informasi perusahaan yang telah tersimpan pada sistem *cloud computing* surat menyurat.



Gambar 4.23 Halaman profil organisasi

Potongan kode program pada halaman profil organisasi dapat dilihat pada Gambar 4.24.

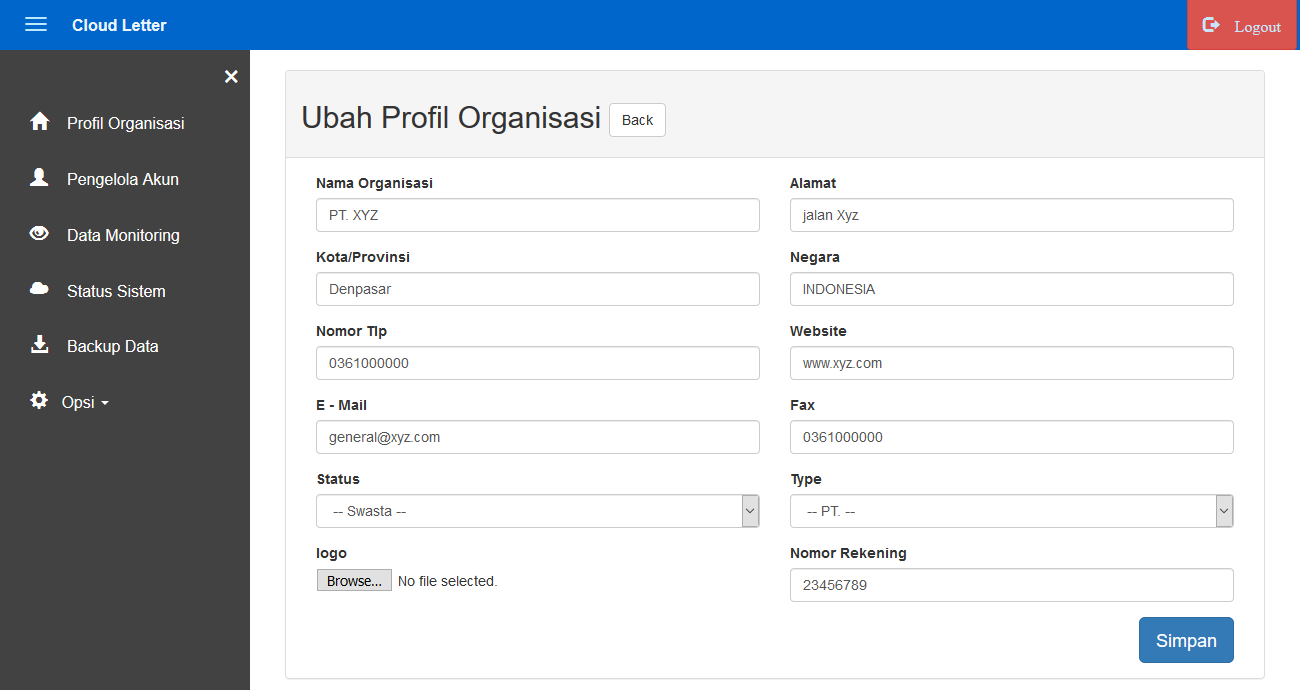
|  |
| --- |
| public function profile(){  $username = $this->session->userdata('username');  $user\_id = $this->model\_users->select\_username($username);  $comp\_id = $user\_id->comp\_id;  $data['company'] = $this->model\_company->select\_id($comp\_id);  $this->load->view('backend/dash\_company\_profile',$data);  } |

Gambar 4.24 Potongan kode program profil organisasi

Potongan kode program pada Gambar 4.24 merupakan kode program yang berfungsi untuk menampilkan data profil organisasi yang sebelumnya telah didaftarkan oleh *user* admin.

1. Halaman Ubah Profil Organisasi

Halaman ubah profil perusahaan yang ditujukan pada Gambar 4.25 merupakan form ubah organisasi dimana *user* admin*, user* kepala *dan user* operator dapat mengubah informasi organisasi yang telah tersimpan pada sistem *cloud computing* surat menyurat.



Gambar 4.25 Halaman ubah profil organisasi

Potongan kode program pada halaman ubah profil organisasi dapat dilihat pada Gambar 4.26.

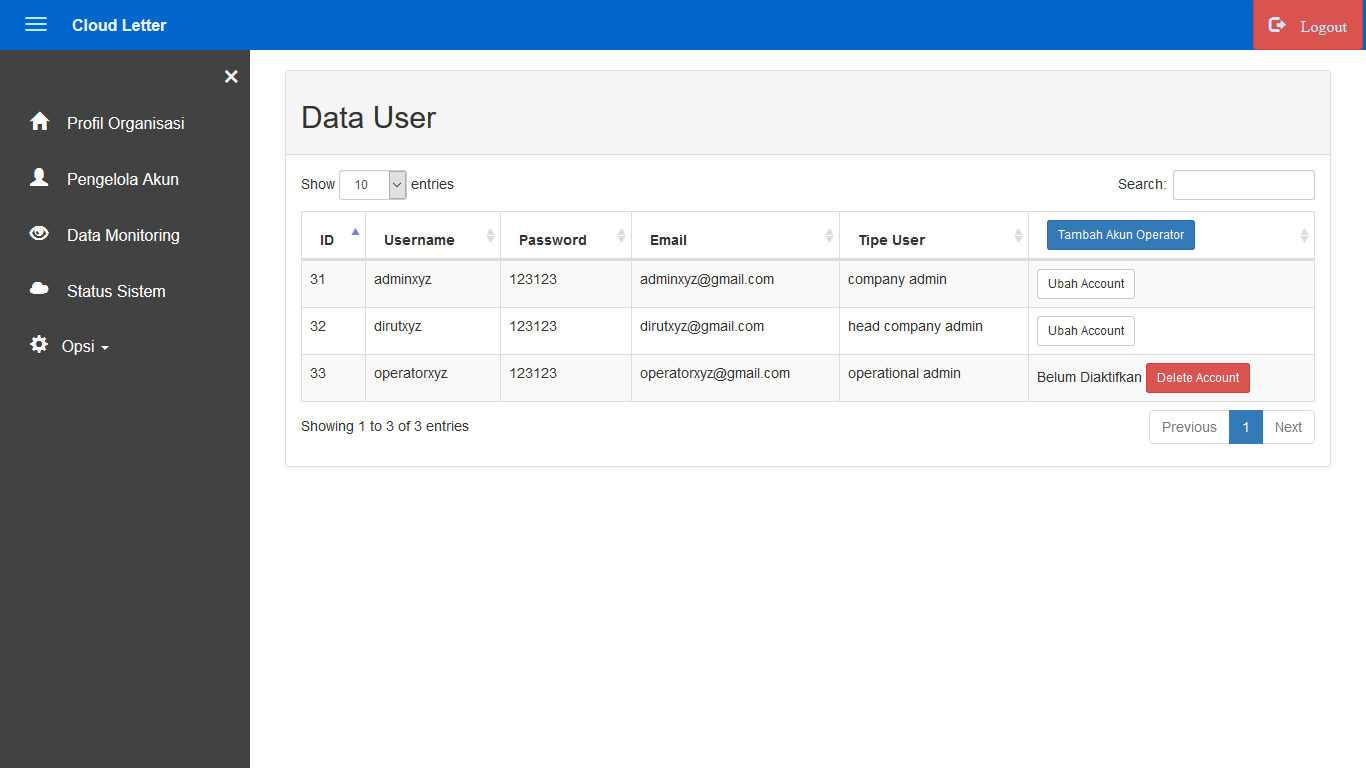
|  |
| --- |
| public function ubah\_profile(){  $data\_comp = array( 'comp\_name' => set\_value('cname'), 'comp\_address' => set\_value('caddress'),'comp\_city' => set\_value('ccity'),'comp\_country' => set\_value('ccountry'),'comp\_phone'=> set\_value('cphone'),'comp\_website' => set\_value('cweb'),'comp\_email'=> set\_value('cemail'),'comp\_fax' => set\_value('cfax'),'comp\_status' => set\_value('cstatus'),'comp\_type'=> set\_value('ctype'), );  $update = $this->model\_company->update($comp\_id , $data\_comp);}} |

Gambar 4.26 Potongan kode program ubah profil organisasi

Potongan kode program pada Gambar 4.26 merupakan kode program yang berfungsi untuk menyimpan data ubah profil organisasi kemudian menyimpan ke basisdata.

1. Halaman Kelola Akun Operator

Halaman kelola akun operator yang ditujukan pada Gambar 4.27 dibawah merupakan halaman dimana *user* admin dapat memanipulasi data akun user operator baik menambah atau mengubah dan menghapus akun yang telah didaftarkan tersimpan.



Gambar 4.27 Halaman kelola akun operator

Potongan kode program pada halaman kelola akun operator dapat dilihat pada Gambar 4.28.

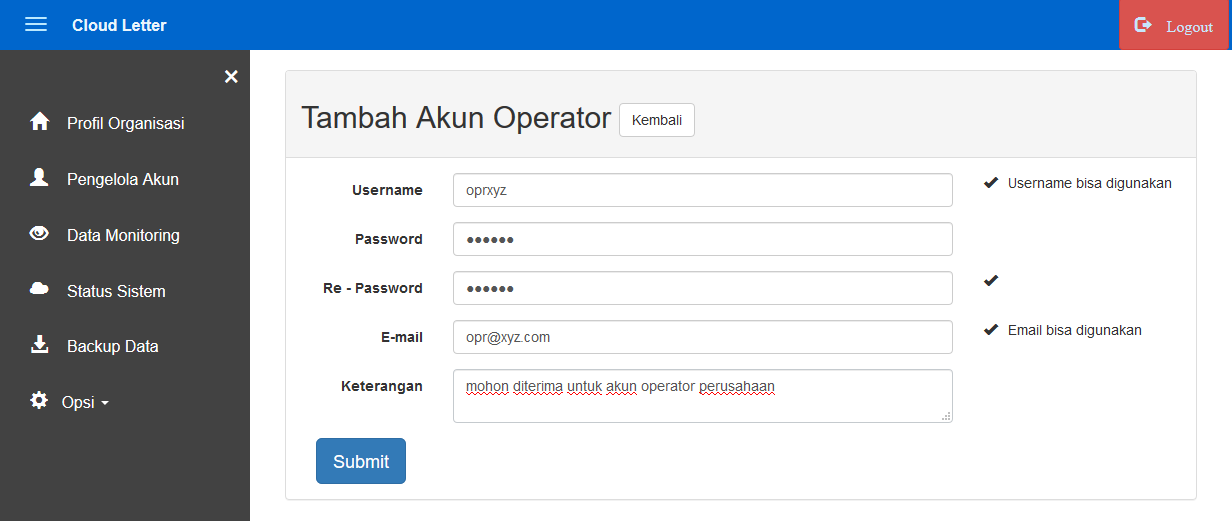
|  |
| --- |
| public function account\_manager(){  $username = $this->session->userdata('username');  $comp\_id = $this->session->userdata('comp\_id');  $data['user'] = $this->model\_setting->select\_all($comp\_id);  $data['user\_induk'] = $this->model\_users->select\_username($username);  $srvc\_id = $data['user\_induk']->srvc\_id;  $data['service'] = $this->model\_service->select\_id($srvc\_id);  $data['count\_user\_data'] = $this->model\_users->count\_data($comp\_id);  $this->load->view('backend/dash\_account\_manager',$data);} |

Gambar 4.28 Potongan kode program kelola akun operator

Potongan kode program pada Gambar 4.28 merupakan kode program yang berfungsi untuk menampilkan data operator yang ada pada basis data.

1. Halaman Form Tambah Akun Operator

Halaman tambah akun operator yang ditujukan pada Gambar 4.29 dibawah merupakan form yang ditujukan kepada *user* admin agar dapat menambah akun operator untuk dapat disetujui oleh *user* kepala.



Gambar 4.29 Halaman Tambah akun operator

Potongan kode program pada halaman tambah akun operator dapat dilihat pada Gambar 4.30.

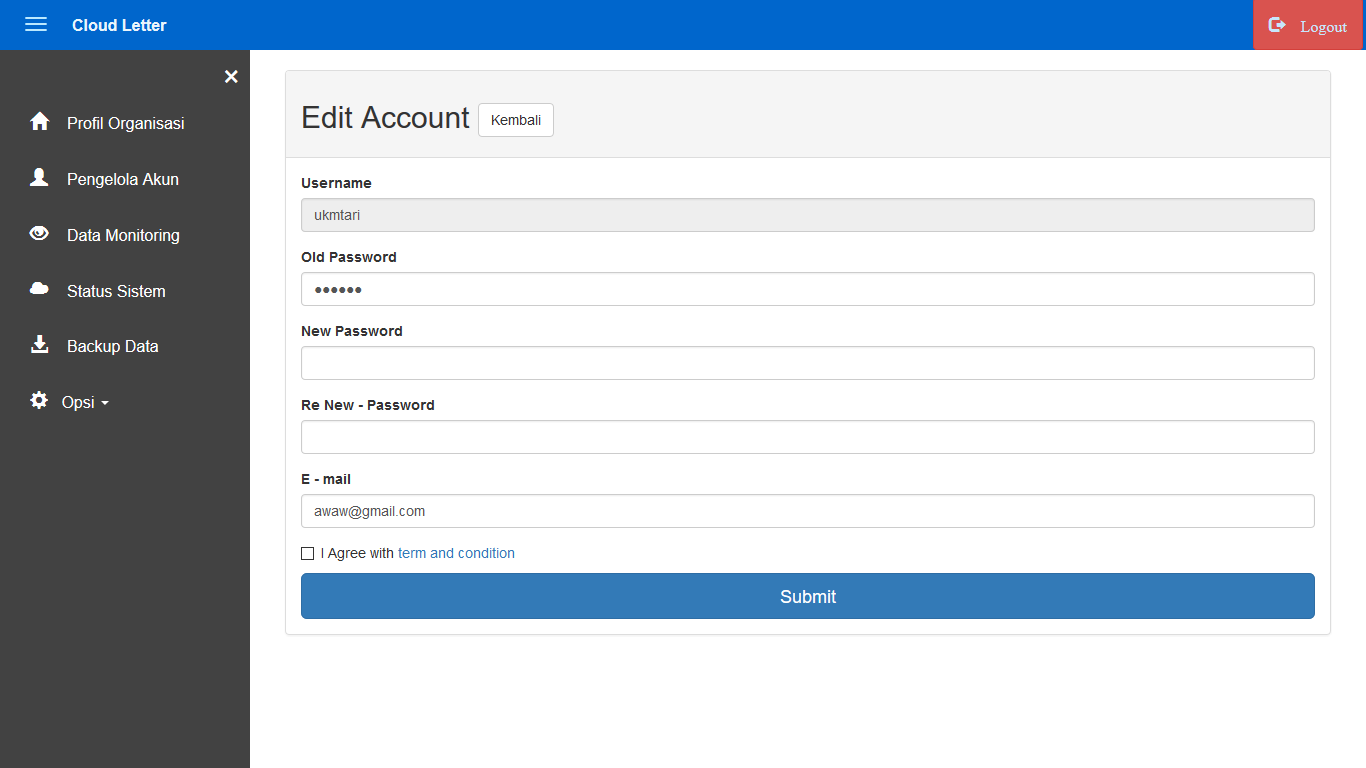
|  |
| --- |
| public function add\_account\_oprator(){  $data\_register = array( 'acc\_usrn' => $username, 'acc\_pwd' => $password, 'acc\_email' => $email, 'accl\_id' => $user\_lvl, 'srvc\_id' => $service\_id, 'comp\_id' => $comp\_id, 'acc\_status' => 0, 'acc\_descrip' => $keterangan);  $this->model\_users->input\_register($data\_register);  $this->session->set\_userdata('username\_bag', $username);redirect('dashboard/account\_manager/);  } |

Gambar 4.30 Potongan kode program tambah akun operator

Potongan kode program pada Gambar 4.30 merupakan kode program yang berfungsi untuk tambah data akun operator kedalam basis data.

1. Halaman Form Ubah Akun Operator

Halaman ubah akun operator yang ditujukan pada Gambar 3. *47* dibawah merupakan form yang ditujukan untuk user *admin* dapat mengubah data akun yang telah diaktifkan oleh user kepala lalu kemudian disimpan padabasis data.



Gambar 4.31 Halaman ubah akun operator

Potongan kode program pada halaman ubah akun operator dapat dilihat pada Gambar 4.32.

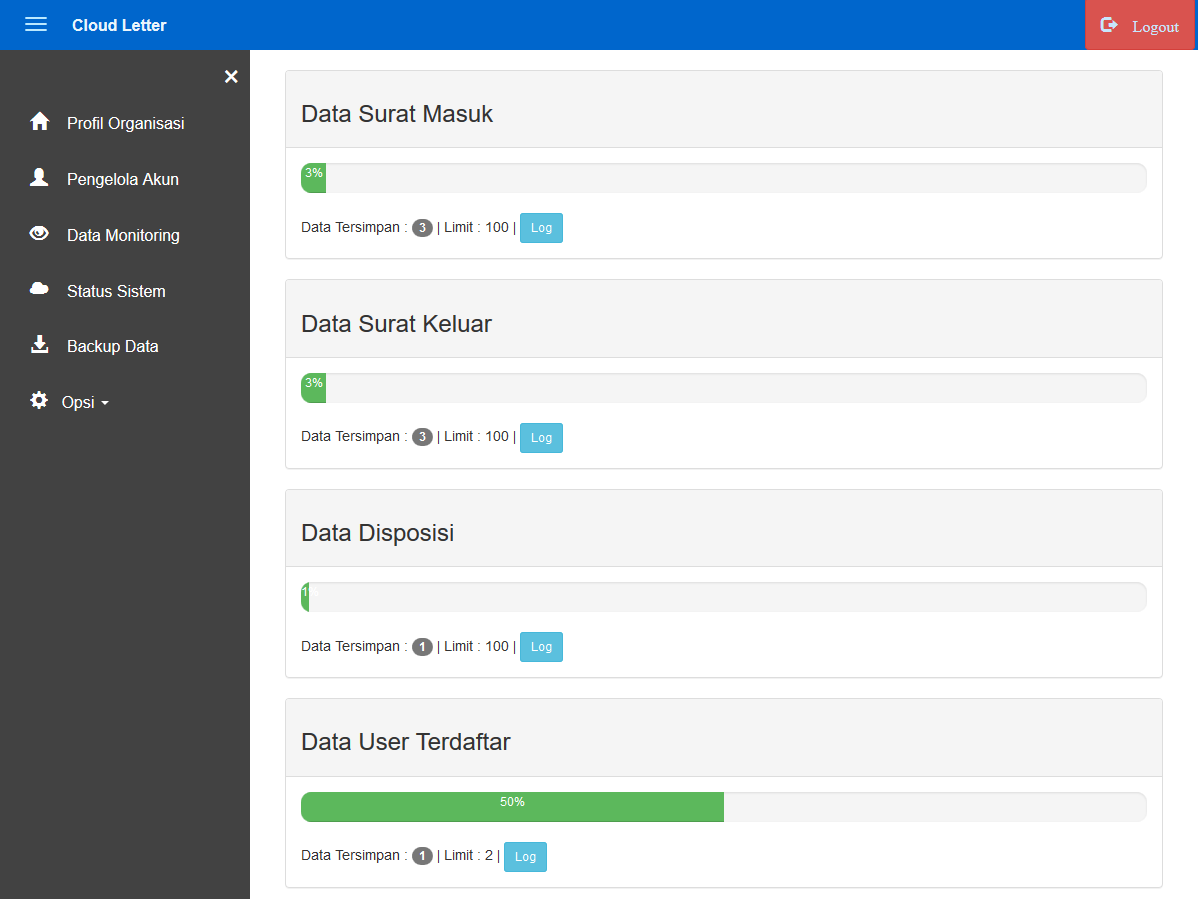
|  |
| --- |
| public function add\_account\_oprator(){  $data\_update = array('acc\_usrn' => $username, ‘acc\_pwd' => $password, 'acc\_email' => $email, 'accl\_id' => $user\_lvl, 'srvc\_id' => $service\_id, 'comp\_id' => $comp\_id, 'acc\_status' => 0, 'acc\_descrip' => $keterangan );  $this->model\_users->update\_account($data\_update, $acc\_id);  $this->session->set\_userdata('username\_bag', $username);  redirect('dashboard/account\_manager/);  } |

Gambar 4.32 Potongan kode program ubah akun operator

Potongan kode program pada Gambar 4.32 merupakan kode program yang berfungsi untuk menyimpan data ubah akun operator yang kemudian disimpan kedalam basis data.

1. Halaman *Data* *Monitoring*

Halaman *data* *monitoring* yang ditujukan pada Gambar 4.33 dibawah merupakan halaman dimana user *admin* dapat memantau atau melihat jumlah sumber daya layanan yang telah digunakan. Tampilan data monitoring dibagi menjadi 4 panel yaitu surat masuk, surat keluar, surat disposisi, dan user.



Gambar 4.33 Halaman data *monitoring*

Potongan kode program pada data monitoring dapat dilihat pada Gambar 4.34.

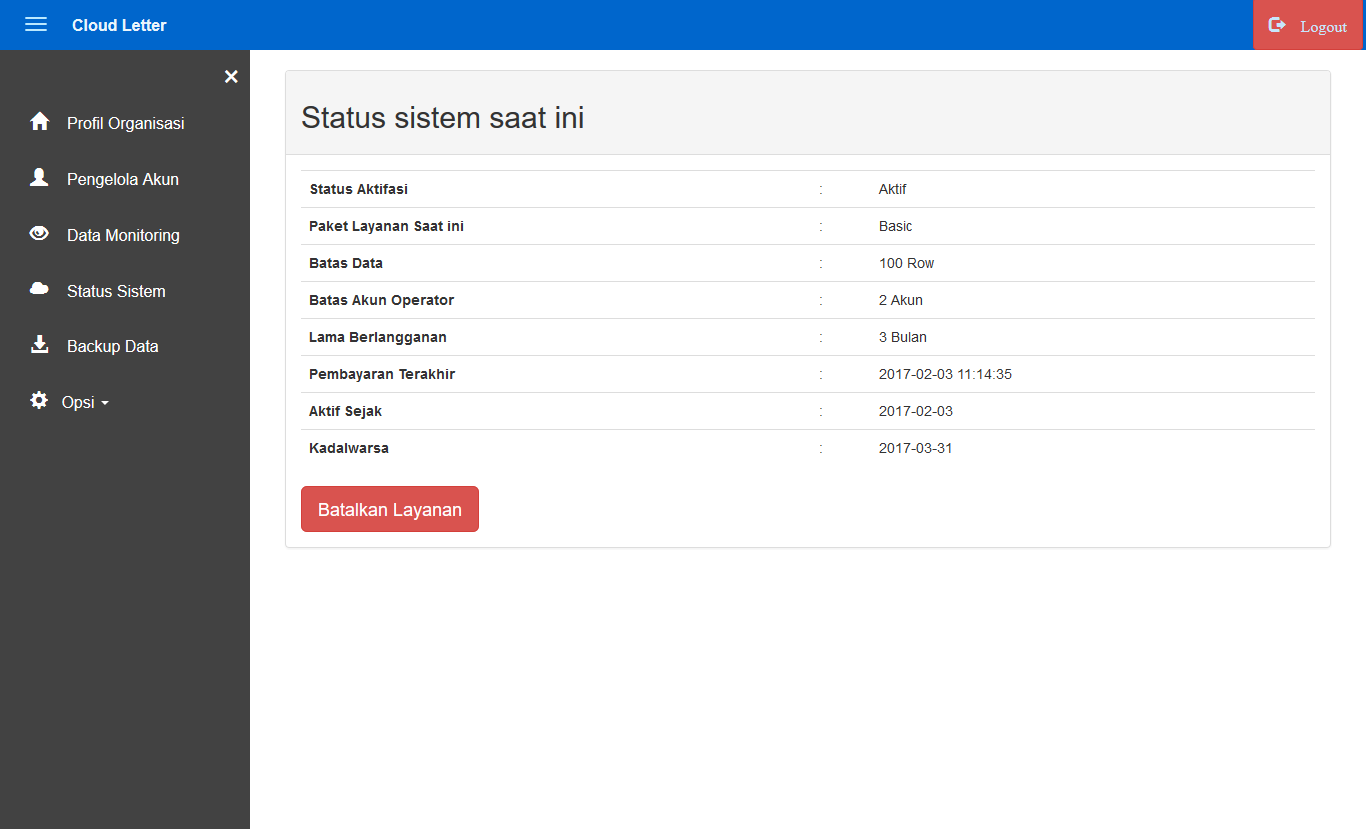
|  |
| --- |
| public function monitoring(){  $data['count\_surat\_msk'] = $this->count\_surat\_msk($comp\_id, $count\_ letter\_in, $srvc\_id);  $data['count\_surat\_out'] = $this->count\_surat\_out($comp\_id, $count\_ letter\_out, $srvc\_id);  $data['count\_disposisi'] = $this->count\_disposisi($comp\_id, $count\_ disposisi, $srvc\_id);  $data['count\_user'] = $this->count\_user($comp\_id, $count\_ user, $srvc\_id);  $data['surat\_masuk\_tersimpan'] = $count\_data\_letter\_in;  $data['surat\_keluar\_tersimpan'] = $count\_data\_letter\_out;  $data['disposisi\_tersimpan'] = $count\_data\_disposisi;  $data['user\_terdaftar']= $count\_data\_user;  $data['limit\_data']= $this->model\_monitoring->select\_by\_id($srvc\_id);  $this->load->view('backend/dash\_data\_monitor', $data);  } |

Gambar 4.34 Potongan kode program data *monitoring*

Potongan kode program pada Gambar 4.34 merupakan kode program yang berfungsi untuk menampilkan data *monitoring*, pada kode diatas terdapat fungsi hitung surat masuk, keluar, disposisi dan akun yang sudah terdaftar.

1. Halaman Status Sistem

Halaman status sistem yang ditunjukan pada Gambar 4.35 menampilkan informasi tentang status sistem, mulai dari paket apa yang terdaftar hingga kapan habisnya aktifasi



Gambar 4.35 Halaman status sistem

Potongan kode program pada status sistem dapat dilihat pada Gambar 4.36.

|  |
| --- |
| public function status\_sistem(){  $comp\_id = $this->session->userdata('comp\_id');  $data['aktivasi'] = $this->model\_activation->select\_join\_all\_data($comp\_id);  $data['comp\_id'] = $comp\_id;  $this->load->view('backend/dash\_service\_info', $data);  } |

Gambar 4.36 Potongan kode program status sistem

Potongan kode program pada Gambar 4.36 merupakan kode program yang berfungsi untuk menampilkan status sistem, terdapat fungsi untuk menggabungkan semua data yang berkaitan dengan layanan yang sedang aktif.

1. Halaman *Backup Data*

Tampilan halaman backup data dapat dilihat pada Gambar 4.37 menampilkan panel yang masing masingnya terdapat tombol unduh disetiap data surat yang masing masing memiliki fungsi yang berbeda.



Gambar 4.37 Halaman *backup data*

Potongan kode program pada backup data dapat dilihat pada Gambar 4.38.

|  |
| --- |
| public function backup\_data(){  $this->load->view('backend/dash\_backup');  } |

Gambar 4.38 Potongan kode program *backup data*

Potongan kode program pada Gambar 4.28 merupakan kode program yang berfungsi untuk menampilkan halaman *backup data.*

1. Halaman Manipulasi Layanan Pengguna

Pada halaman manipulasi layanan pengguna terdapat panel-panel yang menunjukan paket layanan yang tersedia, tampilan yang dapat dilihat pada Gambar 4.39 juga terdapat button untuk memilih paket yang diinginkan.



Gambar 4.39 Halaman manipulasi layanan pengguna

Potongan kode program pada halaman manipulasi layanan pengguna dapat dilihat pada Gambar 4.40.

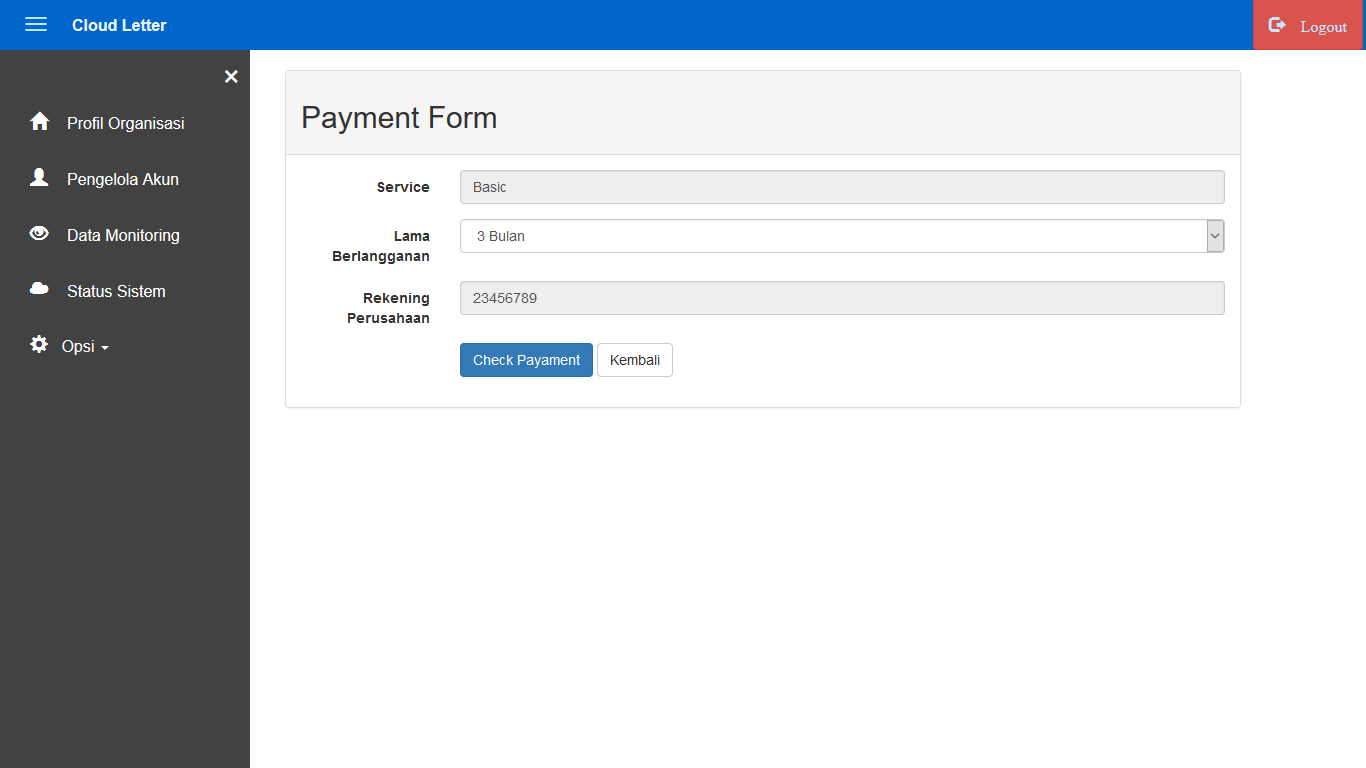
|  |
| --- |
| public function upgrade(){  $username = $this->session->userdata('username');  $data['service'] = $this->model\_service->select\_by\_status();  $data['account'] = $this->model\_users->select\_username($username);  $data['company'] = $this->model\_company->select\_id($data['account']->comp\_id);  $this->load->view('backend/dash\_upgrade\_sistem', $data);} |

Gambar 4.40 Potongan kode program manipulasi layanan pengguna

Potongan kode program pada Gambar 4.40 merupakan kode program yang berfungsi untuk menampilkan data paket layanan yang sedang aktif agar pengguna dapat memilih lalu kemudian berlangganan.

1. Halaman Form Lama Berlangganan

Halaman form lama berlangganan yang ditunjukan pada Gambar 4.41 menampilkan jumlah bulan langganan sebagai salah satu proses pembayaran. *user* dapat memilih lama langganan pada *combobox.*



Gambar 4.41 Halaman form lama berlangganan

Potongan kode program pada halaman form lama berlangganan pengguna dapat dilihat pada Gambar 4.42.

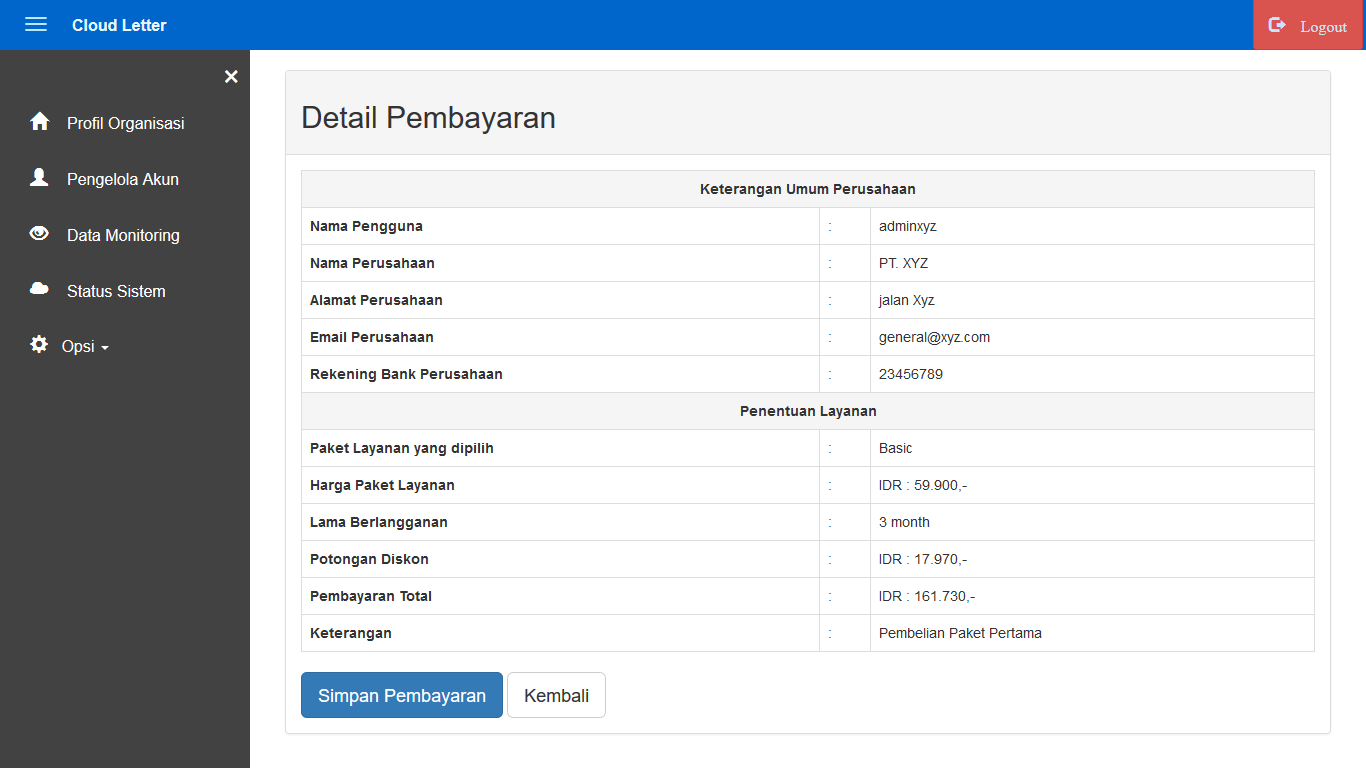
|  |
| --- |
| public function subscript(){  $username = $this->session->userdata('username');  $comp\_id = $this->session->userdata('comp\_id');  $data['account'] = $this->model\_users->select\_username($username);  $data['company'] = $this->model\_company->select\_id($comp\_id);  $data['service'] = $this->model\_service->select\_id($srvc\_id);  $this->load->view('backend/dash\_form\_payment',$data);  } |

Gambar 4.42 Potongan kode program form lama berlangganan

Potongan kode program pada Gambar 4.42 merupakan kode program yang berfungsi untuk menampilkan form lama berlangganan yang kemudian akan dihitung agar mendapatkan detail pembayaran.

1. Halaman Detail Pembayaran

Halaman detail pembayaran yang ditunjukan pada Gambar 4.43 menampilkan detail pembayaran mulai dari informasi perusahaan dan penentuan layanan juga jumlah nominal yang harus dibayarkan*.*



Gambar 4.43 Halaman detail pembayaran

Potongan kode program pada halaman detail pembayaran dapat dilihat pada Gambar 4.44.

|  |
| --- |
| public function detail\_payment(){  $service\_id = set\_value('service');  $acc\_id = set\_value('user\_id');  $comp\_id = $this->session->userdata('comp\_id');  $data['service'] = $this->model\_service->select\_id($service\_id);  $data['account'] = $this->model\_users->select\_id($acc\_id);  $data['jenis\_pembayaran'] = $this->cek\_jenis\_pembayaran($comp\_id);  $data['company'] = $this->model\_company->select\_id($comp\_id);  $harga = $data['service']->srvc\_price;  $data['bulan'] = set\_value('month');  $data['harga\_awal'] = $harga \* $data['bulan'];  $data['potongan'] = $this->cek\_potongan($data['bulan'], $data['harga\_awal']);  $data['total\_bayar'] = $data['harga\_awal'] - $data['potongan'];    $this->load->view('backend/dash\_payment\_detail', $data);  } |

Gambar 4.44 Potongan kode program detail pembayaran

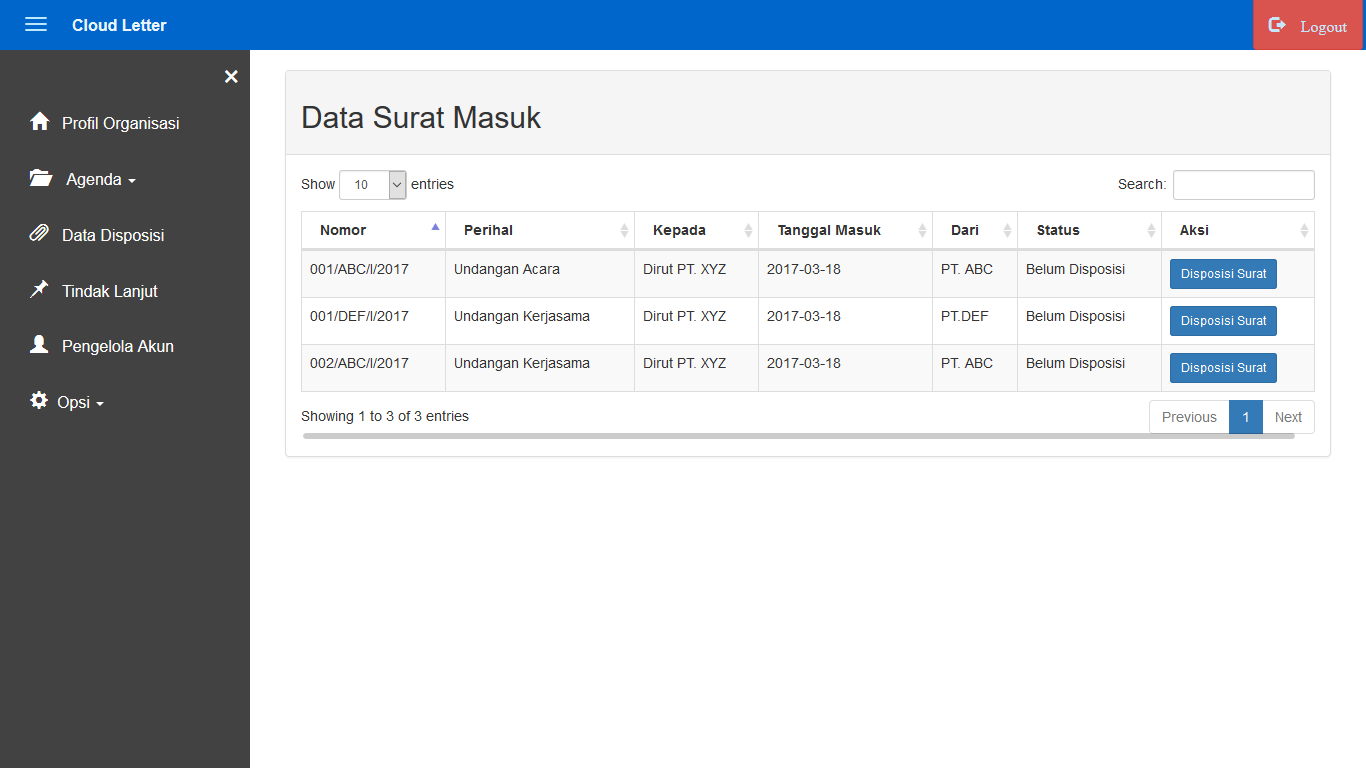
Potongan kode program pada Gambar 4.44 merupakan kode program yang berfungsi untuk menampilkan data detail pembayaran yang telah dihitung berdasarkan lama berlangganan dan diskon yang didapat kemudian ditampilkan pada halaman detail pembayaran.

* + 1. **Tampilan Antar Muka *User* Kepala**

Halaman ini digunakan *user* kepala untuk mengelola data disposisi surat dan autorisasi penambahan akun operator yang digunakan pada suatu perusahaan. Berikut halaman *user* kepala pada perekayasaan sistem *cloud computing* surat menyurat.

1. Halaman Surat Masuk *User* Kepala

Pada halaman surat masuk yang ditunjukan pada Gambar 4.45 menampilkan data surat masuk kepada *user* kepala yang tampilkan ke dalam bentuk tabel. Terdapat juga sebuah tombol untuk menuju halaman disposisi.

****

Gambar 4.45 Halaman surat masuk *user* kepala

Potongan kode program pada halaman surat masuk user kepala dapat dilihat pada Gambar 4.46.

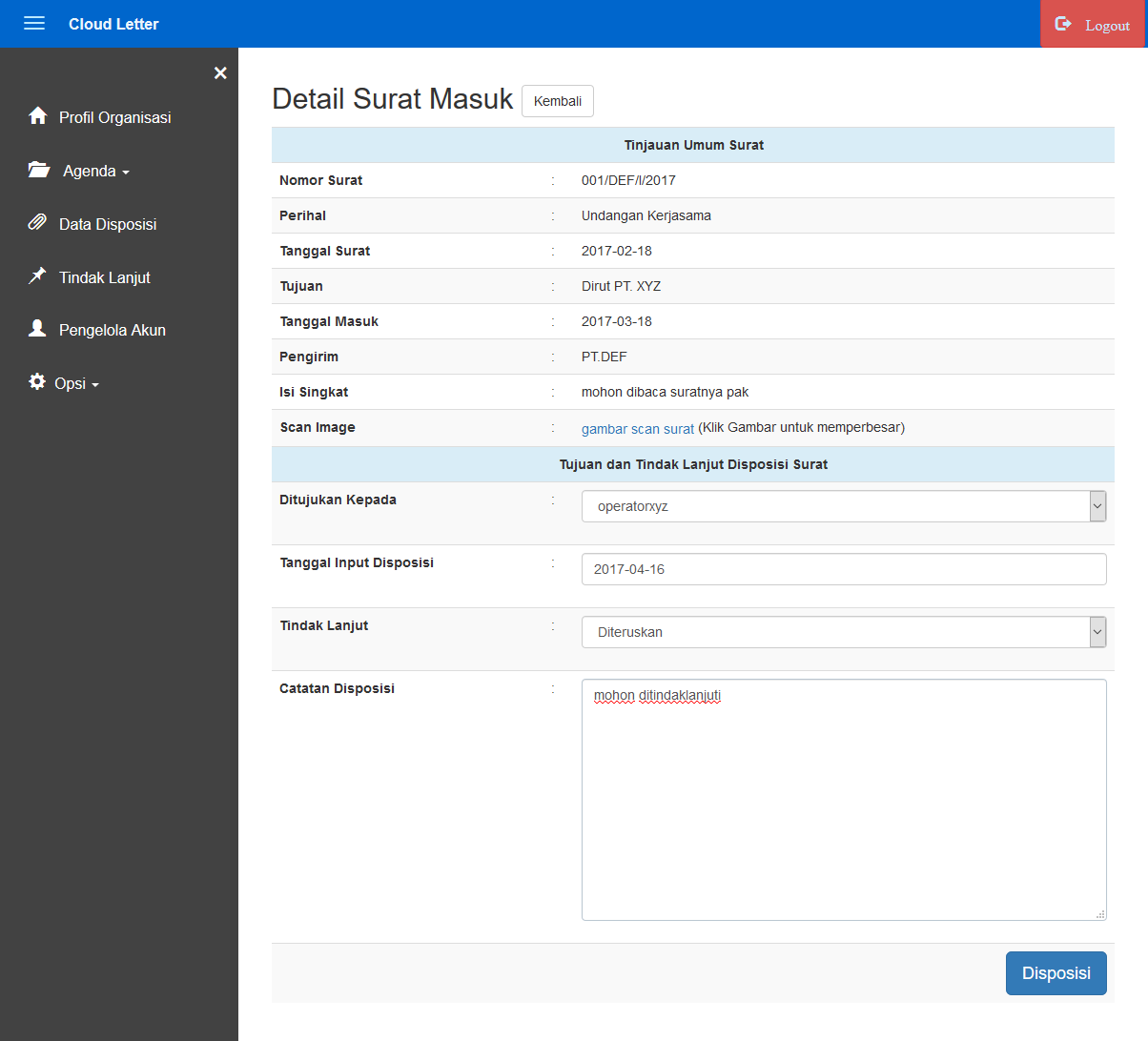
|  |
| --- |
| public function letter\_in(){  if($user\_level == 3){  $comp\_id = $this->session->userdata('comp\_id');  $data['surat'] = $this->model\_surat->select\_all\_bycompid\_status($comp\_id);  $this->load->view("backend/dash\_letter\_in\_info",$data);  } } |

Gambar 4.46 Potongan kode program surat masuk *user* kepala

Potongan kode program pada Gambar 4.46 merupakan kode program yang berfungsi untuk menampilkan data surat masuk user kepala berdasarkan masing masing perusahaan.

1. Halaman Form Disposisi

Halaman form disposisi yang ditunjukan pada Gambar 4.47 menampilkan data detail surat masuk dan form disposisi, dimana *user* kepala dapat melakukan disposisi surat masuk ke *user* operator yang dituju.



Gambar 4.47 Halaman disposisi surat masuk

Potongan kode program pada halaman disposisi surat masuk dapat dilihat pada Gambar 4.48.

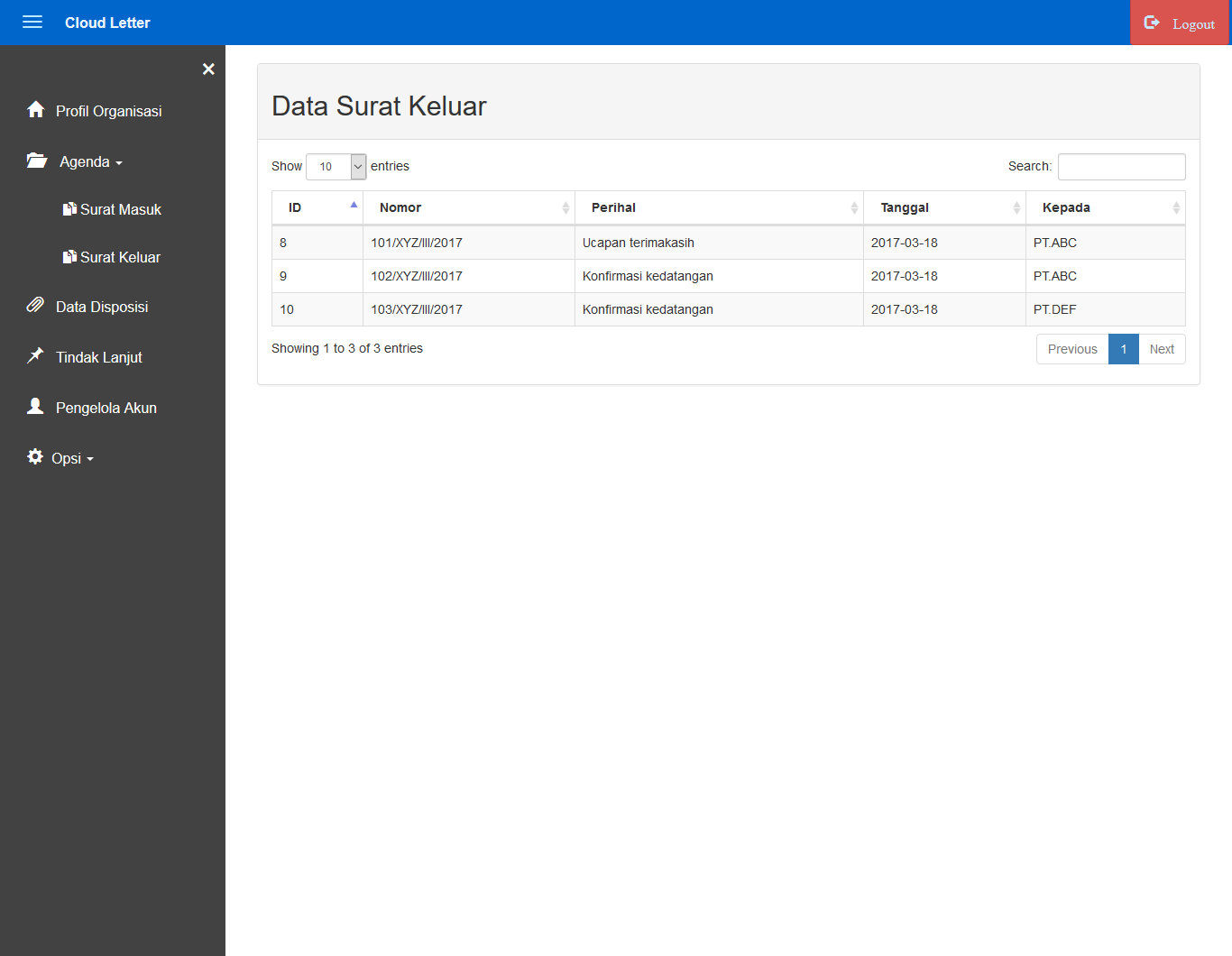
|  |
| --- |
| public function letter\_in\_dsp(){  $data\_disposisi = array( 'ltrin\_id'=> $ltrin\_id, 'dsp\_tanggal' => set\_value('tgl'), 'dsp\_tl\_id' => set\_value('tindak\_lanjut'), 'dsp\_to' => set\_value('ditujukan'), 'dsp\_memo’ => set\_value('catatan'), 'acc\_id => set\_value('acc\_id'), 'comp\_id => $comp\_id); $data\_ubah\_status = array('ltrin\_status' => 1);  $simpan = $this->model\_surat->insert\_dsp($data\_disposisi);  redirect('dashboard/agenda/letter\_in');}} |

Gambar 4.48 Potongan kode program disposisi surat masuk

Potongan kode program pada Gambar 4.48 merupakan kode program yang berfungsi untuk menyimpan data disposisi surat masuk yang dilakukan oleh user kepala kedalam basis data.

1. Halaman Surat Keluar

Pada Gambar 4.49 ditunjukan tapilan halaman surat keluar *user* kepala yang dimana data surat keluar ditampilkan dalam bentuk tabel.



Gambar 4.49 Halaman surat keluar

Potongan kode program pada halaman surat keluar dapat dilihat pada Gambar 4.50.

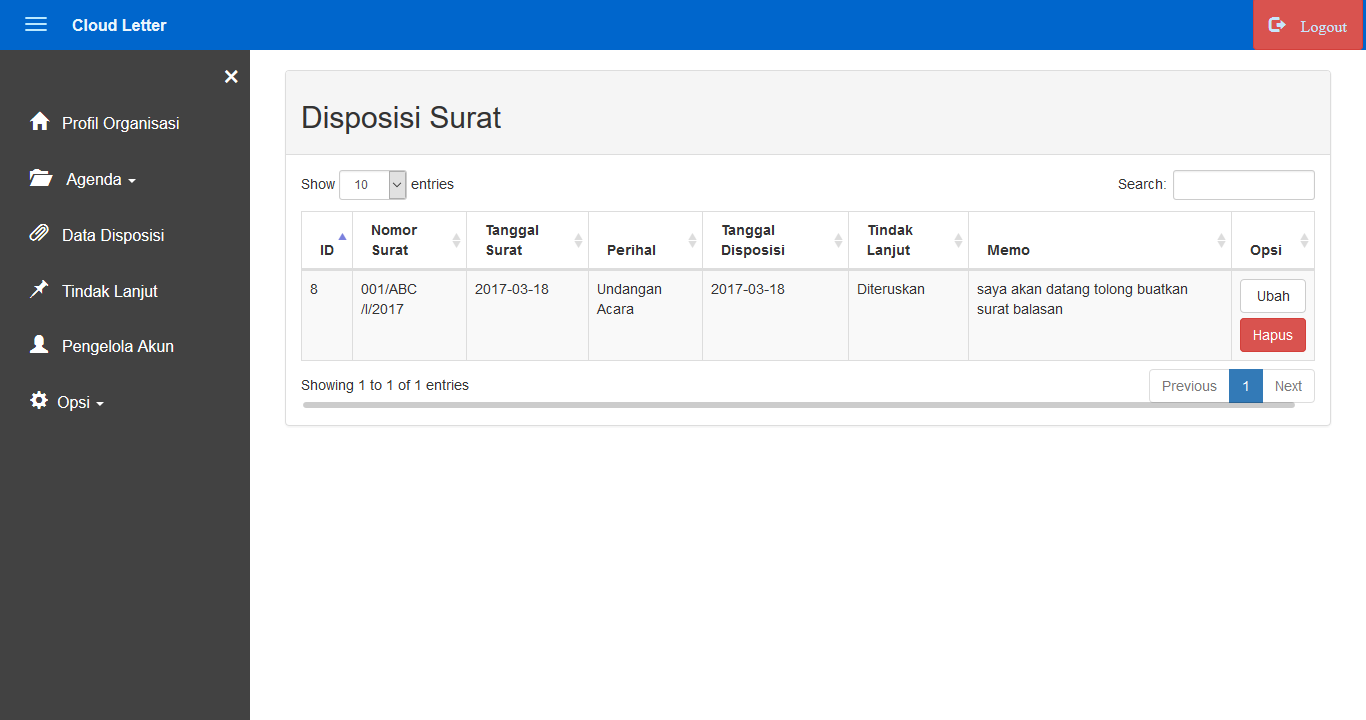
|  |
| --- |
| public function letter\_in(){  $comp\_id = $this->session->userdata('comp\_id');  $user\_level = $this->session->userdata('level');  $data['level\_user'] = $user\_level;  $data['surat\_keluar'] = $this->model\_surat->select\_all\_ltrout($comp\_id);  $this->load->view('backend/dash\_letter\_out\_info', $data);  } |

Gambar 4.50 Potongan kode program surat keluar

Potongan kode program pada Gambar 4.50 merupakan kode program yang berfungsi untuk menampilkan data surat keluar.

1. Halaman data disposisi

Halaman data disposisi yang ditunjukan pada Gambar 4.51 menampilkan data disposisi surat masuk oleh *user* kepala dimana data ditampilkan dalam format tabel.



Gambar 4.51 Halaman data disposisi

Potongan kode program pada halaman data disposisi dapat dilihat pada Gambar 4.52.

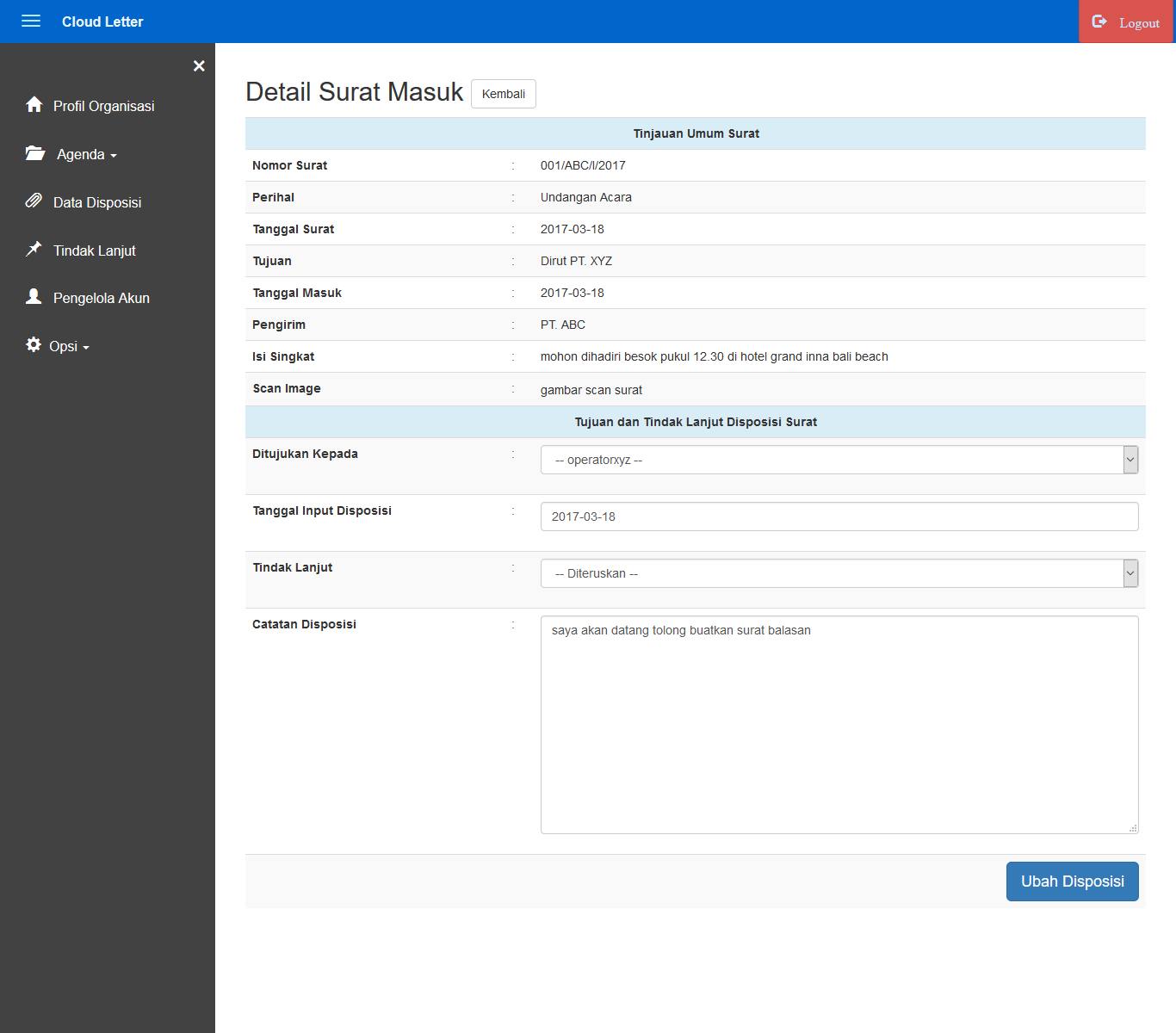
|  |
| --- |
| public function disposisi\_letter(){  if($user\_level == 3){  $data['disposisi'] = $this->model\_surat->select\_all\_dsp($comp\_id);  }  $this->load->view('backend/dash\_letter\_disposisi\_info', $data);  } |

Gambar 4.52 Potongan kode program data disposisi

Potongan kode program pada Gambar 4.52 merupakan kode program yang berfungsi untuk menampilkan data disposisi oleh *user* kepala.

1. Halaman Ubah Data Disposisi

Halaman data disposisi yang ditunjukan pada Gambar 4.53 menampilkan data detail disposisi surat masuk beserta form ubah data disposisi surat masuk.

****

Gambar 4.53 Halaman ubah disposisi surat masuk

Potongan kode program pada halaman ubah data disposisi dapat dilihat pada Gambar 4.54.

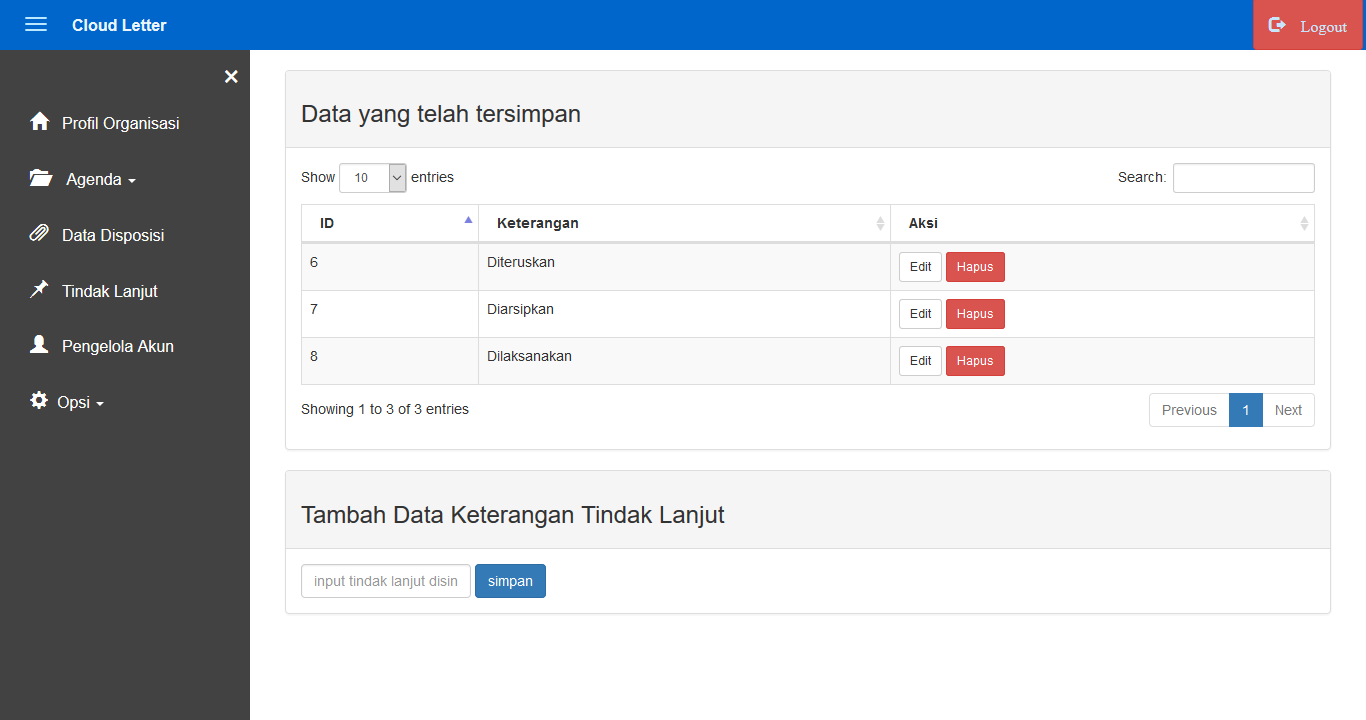
|  |
| --- |
| public function edit\_dsp(){  $data\_disposisi = array( 'dsp\_tanggal' => set\_value('tgl'),'dsp\_tl\_id'=> set\_value('tindak\_lanjut'), 'dsp\_to'=> set\_value('ditujukan'), 'dsp\_memo=> set\_value('catatan'),);  $edit = $this->model\_surat->edit\_dsp($data\_disposisi,$ltrin\_id);  redirect('dashboard/agenda/letter\_dsp');  } |

Gambar 4.54 Potongan kode program ubah data disposisi

Potongan kode program pada Gambar 4.54 merupakan kode program yang berfungsi untuk menyimpan data ubah disposisi kedalam basis data.

1. Halaman Manipulasi Tindak Lanjut

Halaman tindak lanjut disposisi yang ditunjukan pada Gambar 4.55 menampilkan data tindak lanjut diposisi beserta form tambah keterangan tindak lanjut disposisi.

****

Gambar 4.55 Halaman manipulasi tindak lanjut

Potongan kode program pada halaman manipulasi tindak lanjut dapat dilihat pada Gambar 4.56.

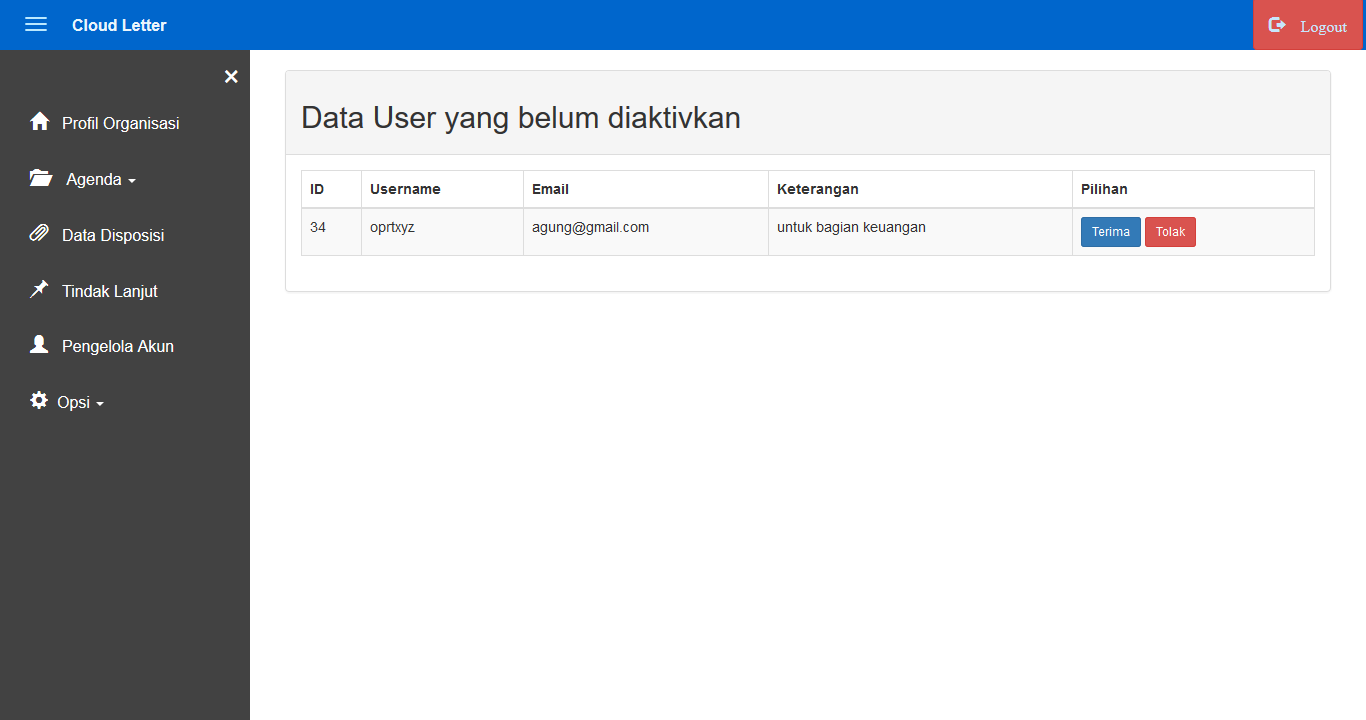
|  |
| --- |
| public function tl\_dsp(){  $keterangan = set\_value('keterangan');  $data\_keterangan = array( 'dsp\_tl\_ket' => $keterangan, 'comp\_id' => $comp\_id );  $simpan = $this->model\_surat->insert\_tl($data\_keterangan);  redirect('dashboard/agenda/tl\_dsp');}  } |

Gambar 4.56 Potongan kode program manipulasi tindak lanjut

Potongan kode program pada Gambar 4.56 merupakan kode program yang berfungsi untuk menyimpan data tindak lanjut disposisi kedalam basis data.

1. Halaman *Approve* Akun Operator

Pada halaman *approve* akun operator pada Gambar 4.57 menampilkan data akun operator yang diajukan oleh *user* admin kepada *user* kepala agar dapat disetujui atau ditolak.



Gambar 4.57 Halaman *approve* akun operator

Potongan kode program pada halaman approve akun operator dapat dilihat pada Gambar 4.58

|  |
| --- |
| public function approve\_acc(){  $comp\_id = $this->session->userdata('comp\_id');  $data['user'] = $this->model\_users->select\_by\_userstatus($comp\_id);  $this->load->view('backend/dash\_approve\_account', $data);}  } |

Gambar 4.58 Potongan kode program *approve* akun operator

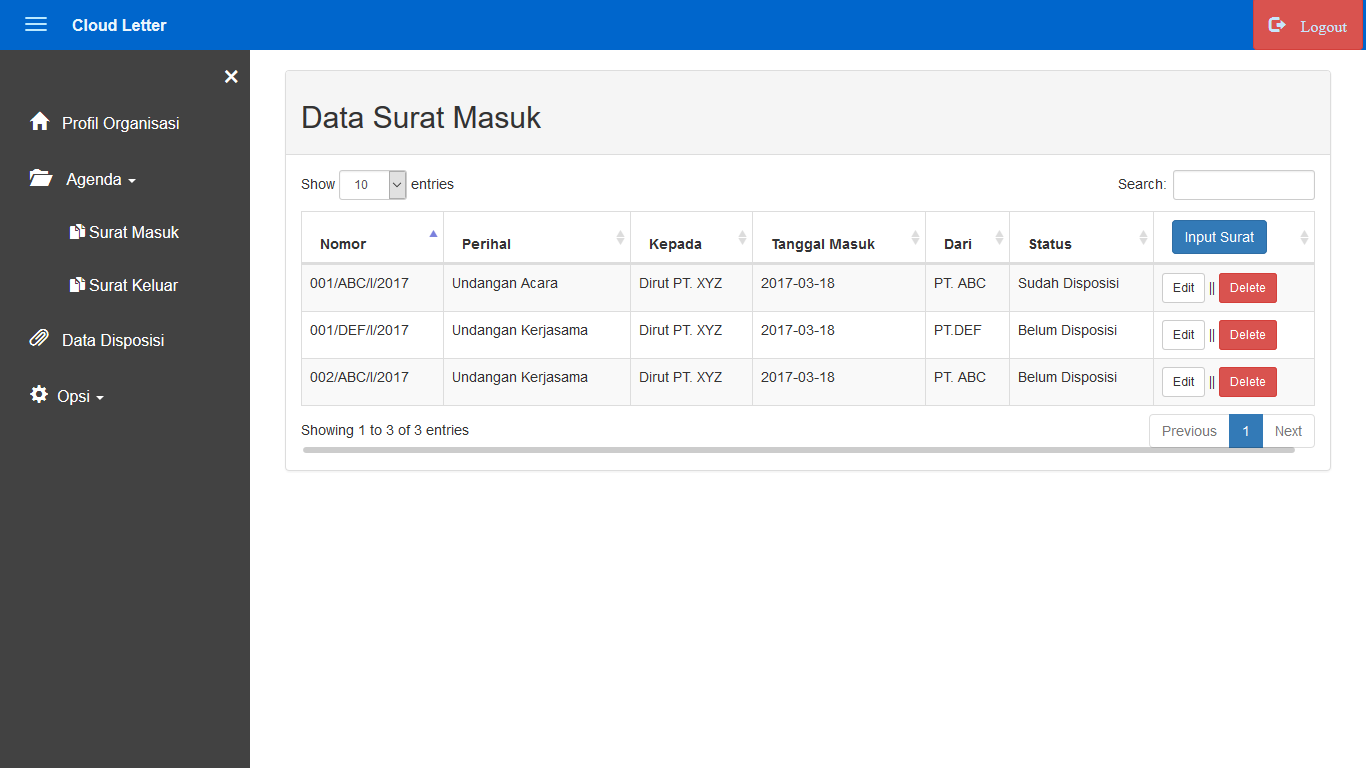
Potongan kode program pada Gambar 4.58 merupakan kode program yang berfungsi untuk menampilkan data akun operator yang belum diaktifkan oleh user kepala.

* + 1. **Tampilan Antar Muka *User* Operator**

Halaman ini digunakan *user* operator untuk mengelola data yang berkaitan dengan surat menyurat pada suatu perusahaan. Berikut halaman *user* operator pada perekayasaan sistem *cloud computing* surat menyurat.

1. Halaman Surat Masuk *User* Operator

Pada halaman surat masuk yang ditunjukan pada Gambar 4.59 menampilkan data surat masuk tampilkan ke dalam bentuk tabel. Terdapat juga sebuah tombol untuk tambah, ubah, dan hapus surat masuk.



Gambar 4.59 Halaman surat masuk *user* operator

Potongan kode program pada halaman surat masuk user operator dapat dilihat pada Gambar 4.60.

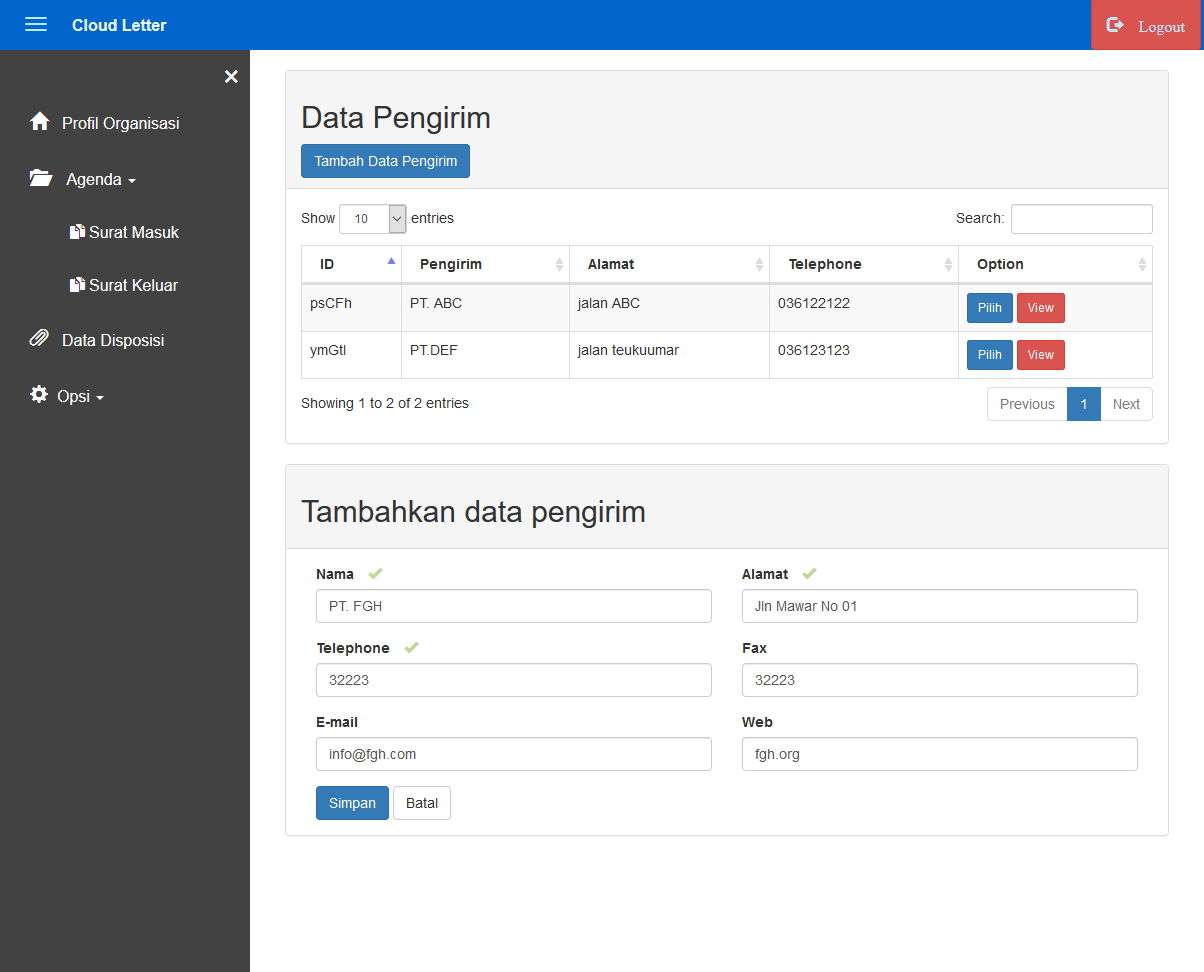
|  |
| --- |
| public function letter\_in(){  $comp\_id = $this->session->userdata('comp\_id');  $username = $this->session->userdata('username');  $data\_user = $this->model\_users->select\_username($username);  $data['surat'] = $this->model\_surat->select\_all\_bycompid($comp\_id);  $data['service'] = $this->model\_service->select\_id($data\_user->srvc\_id);  $data['count\_surat'] = $this->model\_monitoring->select\_count\_letter\_in($comp\_id);  $this->load->view("backend/dash\_letter\_in\_info",$data);  } |

Gambar 4.60 Potongan kode program surat masuk *user* operator

Potongan kode program pada Gambar 4.60 merupakan kode program yang berfungsi untuk menampilkan surat masuk *user* operator per masing-masing perusahaan.

1. Halaman Data Pengirim

Pada halaman tambah data pengirim yang ditunjukan pada Gambar 4.61 menampilkan data pengirim yang tampilkan ke dalam bentuk tabel. Terdapat juga sebuah form tambah pengirim untuk menambah data pengirim.



Gambar 4.61 Halaman data pengirim

Potongan kode program pada halaman data pengirim dapat dilihat pada Gambar 4.62.

|  |
| --- |
| public function add\_sender(){  $comp\_id = $this->session->userdata('comp\_id');  $sender\_id = set\_value('id');  $data\_sender = array(  'sender\_id' => $sender\_id, 'sender\_name' => set\_value('nama'), 'sender\_address' => set\_value('alamat'), 'sender\_phn' => set\_value('tlp'), 'sender\_fax' => set\_value('fax'), 'sender\_email' => set\_value('email'), 'sender\_web' => set\_value('web'), 'comp\_id' => $comp\_id );  $this->model\_surat->insert\_sender($data\_sender) ;  redirect('dashboard/agenda/letter\_in\_add/'.$sender\_id);} |

Gambar 4.62 Potongan kode program halaman data pengirim

Potongan kode program pada Gambar 4.62 merupakan kode program yang berfungsi untuk menyimpan data pengirim kemudian mengarahkan ke halaman tambah data surat.

1. Halaman Tambah Data Surat Masuk

Pada Gambar 4.63 ditunjukan tampilan form input surat masuk dimana pada form tersebut terdapat beberapa field yang dapat dimasukan data surat masuk oleh *user* operator.



Gambar 4.63 Halaman Tambah data surat masuk

Potongan kode program pada halaman tambah data surat masuk dapat dilihat pada Gambar 4.64.

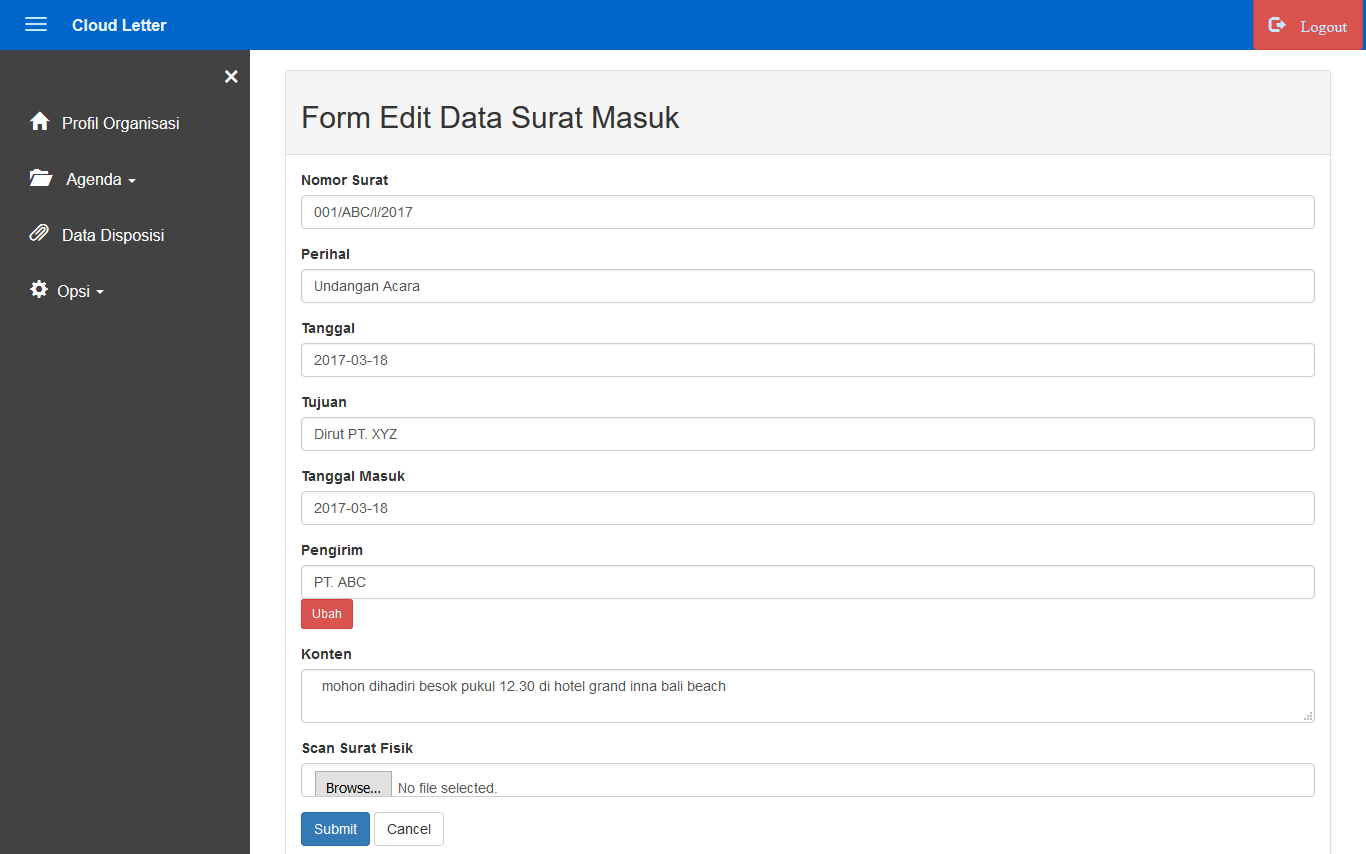
|  |
| --- |
| public function add\_letter\_in(){  $data\_surat = array(  'ltrin\_nomor' => $no\_surat, 'ltrin\_perihal' => $perihal, 'ltrin\_tanggal' => $tgl, 'ltrin\_tujuan' => $tujuan, 'ltrin\_tgl\_msk' => $tgl\_msk, 'sender\_id => $pengirim, 'ltrin\_content' => $konten, 'ltrin\_status'=> $status, 'ltrin\_img' => $filedata['file\_name'], 'acc\_id' => $acc\_id, 'comp\_id'=> $comp\_id );  $add\_letter\_in = $this->model\_surat->letter\_in\_add($data\_surat);  redirect('dashboard/agenda/letter\_in');  } |

Gambar 4.64 Potongan kode program tambah data surat masuk

Potongan kode program pada Gambar 4.64 merupakan kode program yang berfungsi untuk menyimpan data surat masuk kedalam basis data kemudian mengaragkan user operator ke halaman data surat.

1. Halaman Ubah Data Surat Masuk

Pada Gambar 4.65 ditunjukan tampilan form ubah surat masuk dimana pada form tersebut terdapat beberapa field yang sudah terisi sesuai dengan data surat yang ingin diubah.



Gambar 4.65 Halaman ubah data surat masuk

Potongan kode program pada halaman ubah data surat masuk dapat dilihat pada Gambar 4.66.

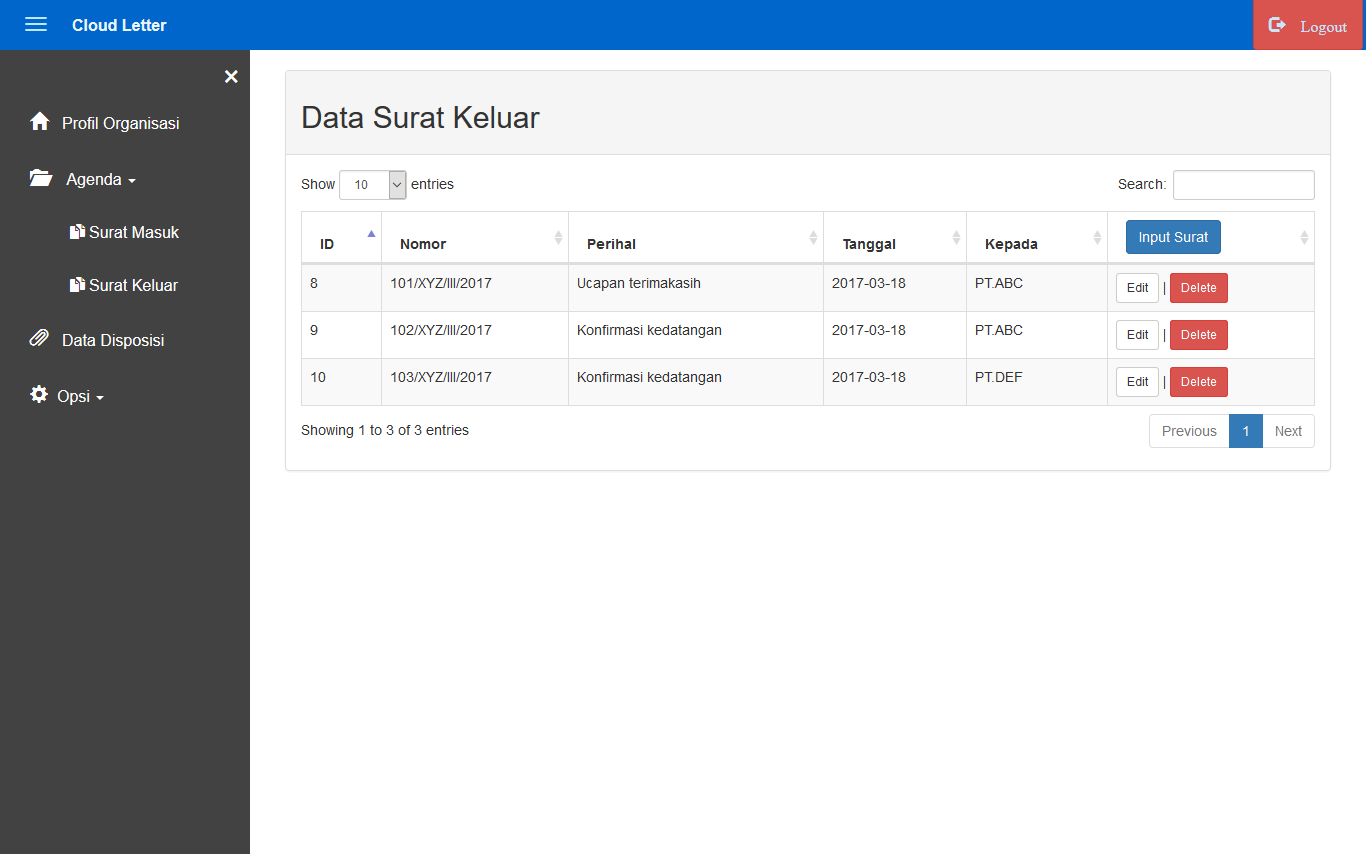
|  |
| --- |
| public function letter\_in\_edit(){  $data\_surat = array(  'ltrin\_nomor' => $no\_surat,'ltrin\_perihal' => $perihal, 'ltrin\_tanggal' => $tgl, 'ltrin\_tujuan' => $tujuan, 'ltrin\_tgl\_msk' => $tgl\_msk, 'sender\_id'=> $pengirim,'ltrin\_content' => $konten, 'ltrin\_img' => 'ltr\_default.jpg' );  $simpan = $this->model\_surat->letter\_in\_edit($data\_surat, $ltrin\_id);  redirect('dashboard/agenda/letter\_in');  redirect('dashboard/agenda/letter\_in\_edit/'.$ltrin\_id);} |

Gambar 4.66 Potongan kode program ubah data surat masuk

Potongan kode program pada Gambar 4.66 merupakan kode program yang berfungsi untuk menyimpan data ubah surat masuk kedalam basis data kemudian mengaragkan user operator ke halaman data surat.

1. Halaman Data Surat Keluar

Halaman data surat keluar yang ditunjukan pada Gambar 4.67 menampilkan data surat keluar dalam bentuk tabel.



Gambar 4.67 Halaman data surat keluar

Potongan kode program pada halaman data surat keluar dapat dilihat pada Gambar 4.68*.*

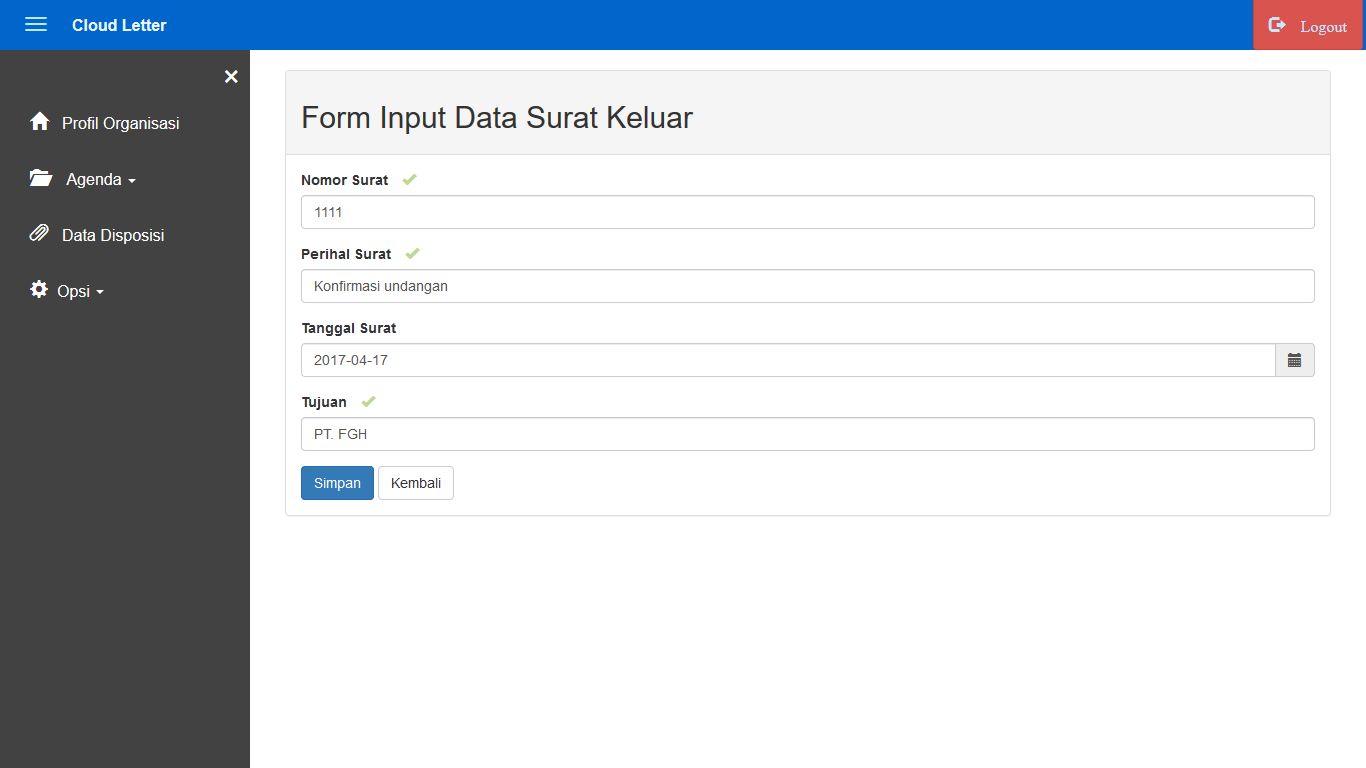
|  |
| --- |
| public function letter\_our (){  $comp\_id = $this->session->userdata('comp\_id');  $user\_level = $this->session->userdata('level');  $data['level\_user'] = $user\_level;  $data['surat\_keluar'] = $this->model\_surat->select\_all\_ltrout($comp\_id);  $this->load->view('backend/dash\_letter\_out\_info', $data);} |

Gambar 4.68 Potongan kode program halaman data surat keluar

Potongan kode program pada Gambar 4.69 merupakan kode program yang berfungsi untuk menampilkan data surat keluar per masing-masing perusahaan.

1. Halaman Tambah Data Surat Keluar

Pada Gambar 4.69 ditunjukan tapilan halaman form tambah surat keluar yang didalamnya terdapat *text field* yang dapat ditambahkan oleh *user* operator.



Gambar 4.69 Halaman input surat keluar

Potongan kode program pada halaman tambah data surat keluar dapat dilihat pada Gambar 4.70.

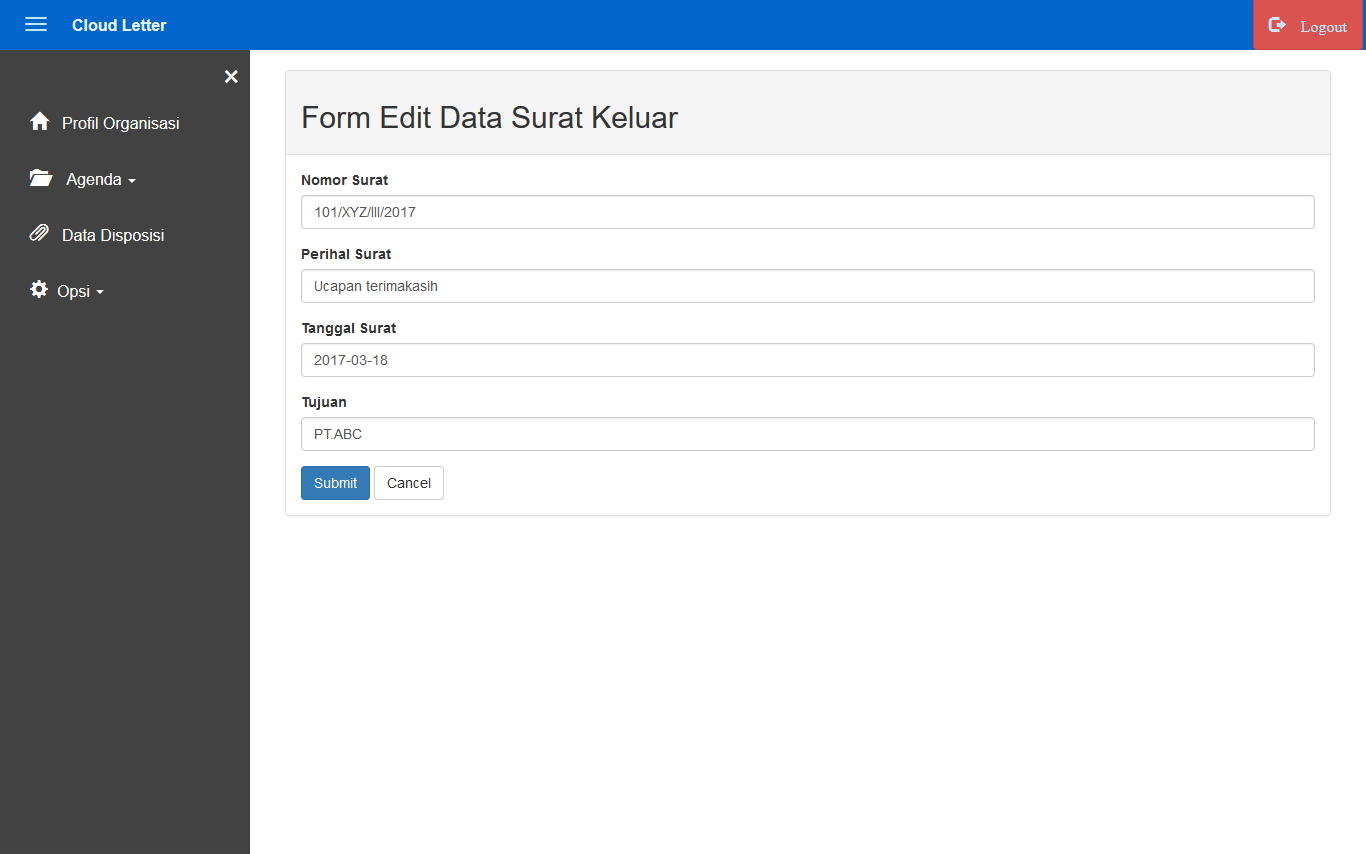
|  |
| --- |
| public function letter\_out\_add (){  $data\_surat = array(  'ltrout\_nomor' => $no\_surat, 'ltrout\_perihal' => $perihal, 'ltrout\_tanggal' => $tanggal, 'ltrout\_kepada' => $tujuan, 'acc\_id' => $user->acc\_id, 'comp\_id' => $comp\_id );  $simpan = $this->model\_surat->letter\_out\_add($data\_surat);  $this->session->set\_flashdata('info','berhasil memasukan data');  redirect('dashboard/agenda/letter\_out');  } |

Gambar 4.70 Potongan kode program tambah data surat keluar

Potongan kode program pada Gambar 4.70 merupakan kode program yang berfungsi untuk menyimpan data surat keluar kedalam basis data kemudian mengaragkan user operator ke halaman data surat keluar.

1. Halaman Ubah Data Surat Keluar

Pada Gambar 4.71 ditunjukan tapilan halaman ubah data surat keluar yang dimana terdapat form yang secara langsung berisikan data surat yang hendak diubah.



Gambar 4.71 Halaman ubah data surat keluar

Potongan kode program pada halaman ubah data surat keluar dapat dilihat pada Gambar 4.72.

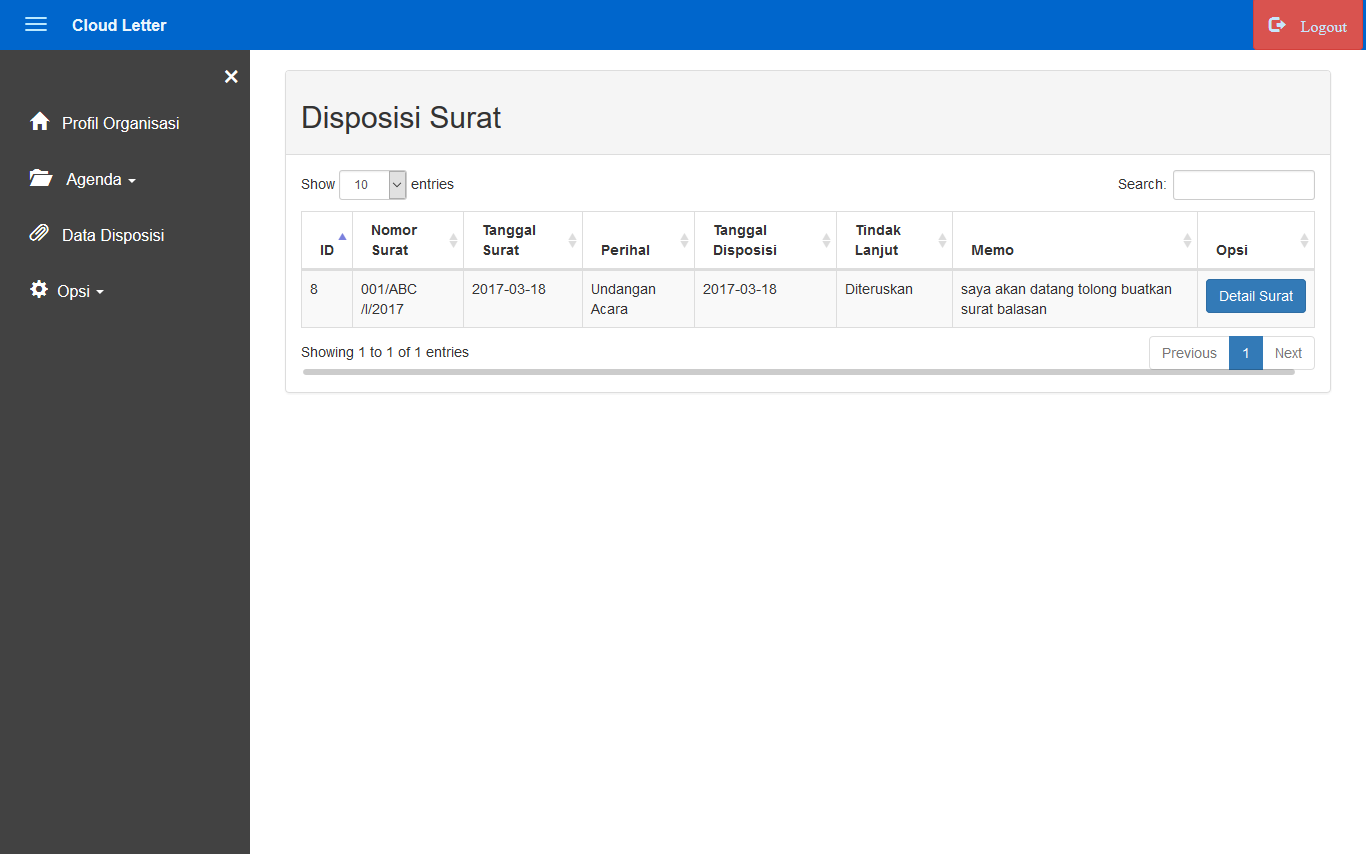
|  |
| --- |
| public function letter\_out\_edit ($ltrout\_id){  $data\_surat = array(  'ltrout\_nomor' => $no\_surat, 'ltrout\_perihal' => $perihal, 'ltrout\_tanggal' => $tanggal, 'ltrout\_kepada' => $tujuan );  $edit = $this->model\_surat->letter\_out\_edit($data\_surat, $ltrout\_id);  $this->session->set\_flashdata('info','berhasil menyimpan data');  redirect('dashboard/agenda/letter\_out');  } |

Gambar 4.72 Potongan kode program ubah data surat keluar

Potongan kode program pada Gambar 4.72 merupakan kode program yang berfungsi untuk menyimpan data ubah surat keluar kedalam basis data kemudian mengaragkan *user* operator ke halaman data surat keluar.

1. Halaman Data Disposisi *User* Operator

Halaman data disposisi yang ditunjukan pada Gambar 4.73 menampilkan data disposisi surat masuk oleh user kepala dimana data ditampilkan dalam format tabel.



Gambar 4.73 Halaman data disposisi surat masuk

Potongan kode program pada halaman data disposisi surat masuk dapat dilihat pada Gambar 4.74.

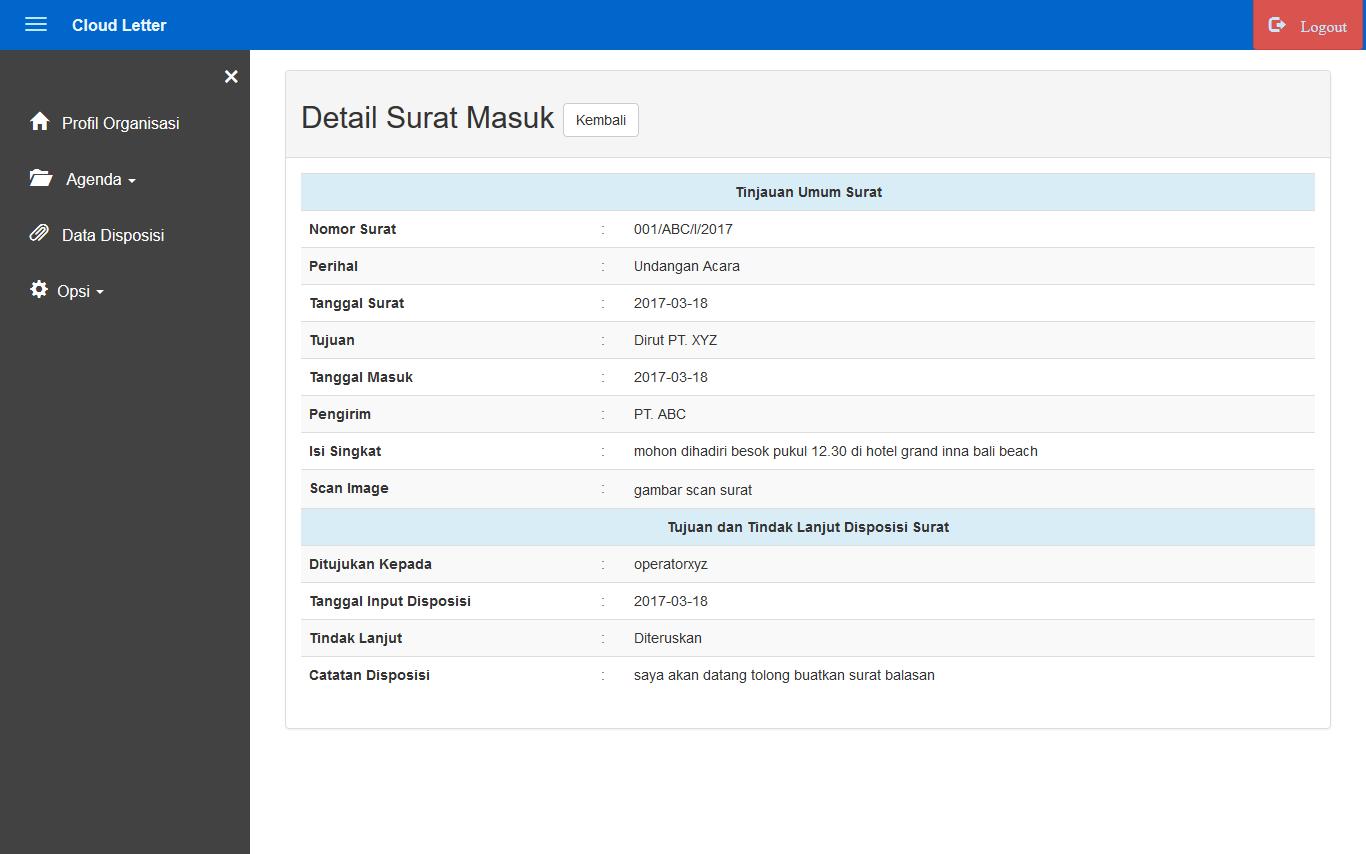
|  |
| --- |
| public function letter\_dsp(){  $username = $this->session->userdata('username');  $user = $this->model\_users->select\_username($username);  $data['disposisi'] = $this->model\_surat->select\_dsp\_bag($user->acc\_id, $comp\_id);} |

Gambar 4.74 Potongan kode program data disposisi surat masuk

Potongan kode program pada Gambar 4.74 merupakan kode program yang berfungsi untuk menampilkan data disposisi yang ditujukan kepada *user* operator untuk ditindaklanjuti.

1. Halaman Detail Disposisi *User* Operator

Pada halaman detail disposisi akan menampilkan data detail surat masuk dan juga detail disposisi. halaman detail disposisi ditunjukan pada Gambar 4.75.

****

Gambar 4.75 Halaman detail disposisi *user* operator

Potongan kode program pada halaman detail disposisi user operator dapat dilihat pada Gambar 4.76.

|  |
| --- |
| public function letter\_dsp(){  $username = $this->session->userdata('username');  $user = $this->model\_users->select\_username($username);  $data['disposisi'] = $this->model\_surat->select\_dsp\_bag($user->acc\_id, $comp\_id);} |

Gambar 4.76 Potongan kode program detail disposisi user operator

Potongan kode program pada Gambar 4.76 merupakan kode program yang berfungsi untuk menampilkan data disposisi yang ditujukan kepada user operator untuk ditindaklanjuti.

* 1. **Pengujian Sistem**

Pengujian sistem dilakukan dengan tujuan untuk pengecekan terhadap keseluruhan fungsional sistem. Pengecekan dilakukan terhadap alur sistem, input sistem dan juga output sistem. Pada penelitian ini menggunakan *blackbox testing* sebagai metode untuk pengujian sistem. Berikut adalah pengujian terkait sistem manajemen surat menyurat *cloud computing* dengan metode *blackbox testing*.

**4.3.1 Pengujian *Blackbox Testing***

1. *Blackbox Testing* pada *Form* *Login*

Pada tabel 4.1 merupakan pengujian pada *form login user* pada sistem ini untuk melakukan verifikasi *username* dan *password*. Setelah berhasil login, maka sistem akan menampilkan halaman *dashboard* masing-masing user. Tabel ini merupakan pengujian pada *login* untuk mengetahui bahwa pada halaman sesuai dengan harapan.

Tabel 4.1 *Blackbox Testing Form Login*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* menginputkan *username* dan password kemudian menekan tombol *login* | *Username* : xyz (benar)  *Password* :  xyz (benar) | Sistem menerima akses login dan menampil-kan halaman *dashboard* sistem | Sistem memvalidasi *username* dan *password* kemudian mengarah-kan ke halaman *dashboard* sistem | Sistem dapat mevalidasi input dari *User* berupa *username* dan *password* lalu mengarahkan user ke halaman *dashboard* |
| 2 | *User* menginputkan *username* dan *password* yang salah kemudian menekan tombol login | *Username* : xyz (benar)  *Password* : 1234 (salah) | Sistem menolak akses *login* dan akan menampilkan pesan kesalahan “ username atau password tidak sesuai” | Sistem menolak input *username* dan *password* ketika tidak sesuai dengan data pada basis data | Sistem dapat menolak akses ketika input dari *User* berupa *username* dan *password* salah |
| 3 | *User* menginputkan *username* dan *password* yang Kosong. | *Username* : (kosong)  *Password* : (kosong) | Sistem menolak akses *login* dan akan menampilkan pesan kesalahan “username dan password diperlukan” | Sistem menolak input *username* dan *password* ketika data input kosong | Sistem dapat menolak akses ketika input dari *User* berupa *username* dan *password* kosong |

1. *Blcakbox Testing* pada *Form* Registrasi Organisasi

Pengujian yang ditunjukan tabel 4.2 merupakan pengujian pada *form* registrasi organisasi yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada form registrasi organisasi.

Tabel 4.2 *Blackbox Testing Form* Registrasi Organisasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* mengisi seluruh *form* kemudian menekan tombol submit | Seluruh *form* terisi dan tombol submit | Sistem menerima registrasi organisasi dan mengarahkan ke halaman registrasi user admin | Sistem menyimpan data registrasi lalu mengarahkan ke halaman registrasi user admin | Sistem telah berhasil menyimpan input dari user pada form registrasi kemudian mengarahkan user ke halaman registrasi user admin. |
| 2 | *User* mengosongkan seluruh *form* kemudian menekan tombol submit | Seluruh *form* kosong dan tombol submit | Sistem menolak registrasi perusahaan dan munculkan pesan error | Sistem menolak input data registrasi ketika form kosong | Sistem telah berhasil menolak input dari user pada form registrasi ketika form kosong. |

1. *Blcakbox Testing* pada *Form* Registrasi *User* Admin

Pada tabel 4.3 ditunjukan pengujian pada *form* registrasi *user* admin yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form* registrasi *user* admin.

Tabel 4.3 *Blackbox Testing Form* Registrasi *User* Admin

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | | **Hasil yang Diharapkan** | | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* mengisi seluruh *form* kemudian menekan tombol submit | Seluruh *form* terisi dan tombol submit | | Sistem menerima registrasi perusahaan dan mengarahkan ke halaman registrasi *user* kepala | | Sistem menyimpan input data registrasi user admin kemudian mengarahkan ke halaman registrasi kepala organisasi | Sistem telah berhasil menyimpan data input registrasi dari user kemudian mengarahkan ke halaman registrasi kepala organisasi |
| 2 | *User* mengosongkan seluruh *form* kemudian menekan tombol submit | | Seluruh *form* kosong dan tombol submit | | Sistem menolak registrasi perusahaan dan munculkan pesan error | Sistem menolak input data registrasi *user* admin Karena form kosong | Sistem telah berhasil menolak data input registrasi dari user ketika form kosong |

1. *Blcakbox Testing* pada *Form* Registrasi *User* kepala

Pada tabel 4.4 merupakan pengujian pada *form* registrasi *user* kepala yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form* registrasi *user* kepala.

Tabel 4.4 *Blackbox Testing Form* Registrasi *User* kepala

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | | **Hasil yang Diharapkan** | | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* mengisi seluruh *form* kemudian menekan tombol *submit* | Seluruh *form* terisi dan tombol *submit* | | Sistem menerima registrasi perusahaan dan mengarahkan ke halaman *dashboard* admin | | Sistem menyimpan input data registrasi *user* kepala kemudian mengarahkan ke halaman *dashboard* admin | Sistem telah berhasil menyimpan data input registrasi dari *user* kemudian mengarahkan ke halaman *dashboard* admin |
| 2 | *User* mengosongkan seluruh *form* kemudian menekan tombol *submit* | | Seluruh *form* kosong dan tombol *submit* | | Sistem menolak registrasi perusahaan dan munculkan pesan *error* | Sistem menolak input data registrasi *user* kepala karena data input pada form kosong | Sistem telah berhasil menolak data input registrasi dari *user* ketika form kosong |

1. *Blcakbox Testing* Pada Menu Aktivasi

Pada tabel 4.5 merupakan pengujian pada menu aktivasi layananpada menu aktivasi layanan yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada menu aktifasi layanan.

Tabel 4.5 *Blackbox Testing* Menu Aktivasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *Admin* sistemmenekan tombol aktivasi layanan pada data layanan yg telah masuk | *Button* aktivasi layanan | Sistem mengaktivasi layanan yang telah dibayar oleh *user* | Sistem mengarahkan kembali ke halaman aktivasi kemudian mengubah status user sesuai dengan paket layanan yang dipilih | Sistem telah berhasil mengubah status user saat admin menekan tombol aktifasi. |

1. *Blcakbox Testing* Pada Menu Perpanjangan

Pada tabel 4.6 merupakan pengujian pada menu perpanjangan layananpada menu aktivasi layanan yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada menu perpanjangan layanan.

Tabel 4.6 *Blackbox Testing* Menu Perpanjangan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *Admin sistem* menekan tombol perpanjang layanan | *Button* aktifasi layanan | Sistem memperpanjang layanan yang telah dibayar oleh pengguna | Sistem mengarahkan kembali ke halaman perpanjang langganan kemudian mengubah status user sesuai layanan | Sistem telah berhasil mengubah status user saat admin menekan tombol perpanjang aktivasi layanan. |

1. *Blcakbox Testing* Pada Menu Layanan

Pada tabel 4.7 merupakan pengujian proses manipulasi layananpada menu layanan yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada menu layanan.

Tabel 4.7 *Blackbox Testing* Menu Layanan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | Adminsistemmenekan tombol tambah layanan | *Button* Tambah layanan | Sistem mengarahkan ke form tambah layanan | Sistem mengarah ke halaman form tambah layanan | Admin mekenan *Button* Tambahlayanan kemudian sistem mengarahkan ke halaman form tambah layanan |
| 2 | Adminsistemmenekan tombol ubah layanan | *Button* ubah layanan | Sistem mengarahkan ke form ubah layanan | Sistem mengarah ke halaman form ubah layanan | Admin mekenan *Button* Ubahlayanan kemudian sistem mengarahkan ke halaman form ubah layanan |
| 3 | Adminsistemmenekan tombol aktif atau nonaktifkan layanan | *Button* aktif  Atau  *Button* non aktifkan layanan | Sistem mengubah status layanan sesuai dengan yang diinginkan | Sistem mengubah status layanan sesuai dengan yang diinginkan | Admin mekenan *Button* aktif  Atau  *Button* non aktifkan layanan kemudian sistem mengubah status layanan |

1. *Blcakbox Testing* Pada *Form* Tambah Layanan

Pada tabel 4.8 merupakan pengujian form tambah layananpada menu tambah layanan yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada menu *form* tambah layanan.

Tabel 4.8 *Blackbox Testing Form* Tambah Layanan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | Admin sistemmenginput data layanan dengan mengisi semua form | Mengisi semluruh form dan menekan  *Button* Simpan layanan | Sistem menyimpan data tambah layanan | Sistem berhasil menyimpan data layanan | Sistem telah mampu menyimpan data layanan saat admin mengisi seluruh form. |
| 2 | Admin sistemmengosongkan form | Mengosongkan semluruh form dan menekan  *Button* Simpan layanan | Sistem menolak data layanan memunculkan pesan error | Sistem berhasil menolak masukan dari admin ketikda *form* kosong. | Sistem telah mampu menolak masukan data kosong saat admin mengosongkan *form* |

1. *Blcakbox Testing* Pada *Form* Ubah Layanan

Pada tabel 4.9 merupakan pengujian form ubah layananpada menu ubah layanan yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada menu *form* ubah layanan.

Tabel 4.9 *Blackbox Testing Form* Ubah Layanan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | Admin sistemmengubah data layanan | Mengubah data pada form dan menekan  *Button* Simpan layanan | Sistem menyimpan data ubah layanan | Sistem berhasil menyimpan data ubah layanan | Sistem telah berhasil menyimpan data ubah layanan ketika admin mengubah data layanan pada *form* |

1. *Blcakbox Testing* pada menu profil organisasi

Pada tabel 4.10 merupakan pengujian menu profil organisasi yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada menu profil organisasi.

Tabel 4.10 *Blackbox Testing* Menu Profil Organisasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* admin menekan tombol ubah organisasi | Button ubah Profile | Mengarahkan ke form ubah Profil organisasi | Sistem mengarahk-an ke halaman form ubah profil organisasi | Sistem telah mampu mengarahkan pengguna saat menekan tombol Ubah Profil |

1. *Blcakbox Testing* Pada *Form* Ubah Profil Organisasi

Pada tabel 4.11 merupakan pengujian form ubah layananpada menu ubah layanan yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form* ubah profil organisasi

Tabel 4.11 *Blackbox Testing Form* Ubah Profil Organisasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* admin mengubah profil organisasi | Form organisasi seluruhnya ter isi dan menekan *Button* Ubah | Sistem dapat menyimpan data ubah profil organisasi | Sistem menyimpan data ubah profil organisasi | Sistem telah berhasil menyimpan data ubah profil organisasi saat *user* admin mengubah data profil organisasi |

1. *Blcakbox Testing* Pada Menu Kelola Akun Operator

Pada tabel 4.12 merupakan pengujian menu kelola akun operator yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada menu kelola akun operator.

Tabel 4.12 *Blackbox Testing* Menu Kelola Akun Operator

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* admin menekan tombol tambah akun | *Button* tambah akun | Sistem mengarahkan ke form tambah akun | Sistem mengarah-kan ke form tambah akun | Sistem telah berhasil mengarahkan ke form tambah akun ketika user admin menekan tombol tambah akun |
| 2 | *User* admin menekan tombol tambah akun dengan data akun sudah ada pada batas layanan | *Button* tambah akun | Sistem menolak mengarahkan ke form tambah akun | Sistem menolak mengarah-kan ke form tambah akun | Sistem telah berhasil menolak mengarahkan ke form tambah akun Karena data akun sudah pada batas layanan. |
| 3 | *User* admin menekan tombol ubah akun | *Button* ubah akun | Sistem mengarahkan ke form ubah akun | Sistem mengarah-kan ke halaman form ubah akun | Sistem telah berhasil mengarahkan ke form ubah akun ketika user admin menekan tombol ubah akun |
| 4 | *User* adminmenekan tombol hapus akun | *Button* hapus akun | Menghapus data akun operator | Sistem menghapus data akun operator | Sistem telah berhasil menghapus data akun ketika user admin menekan tombol hapus akun |
| 5 | *User* kepalamenekan tombol terima | *Button* terima | Mengubah status akun menjadi aktif | Sistem menyimpan data ubah status akun operator | Sistem telah berhasil mengubah data status akun ketika user kepala menekan tombol terima akun |
| 6 | *User* kepalamenekan tombol tolak | *Button* tolak | Mengubah status akun menjadi ditolak | Sistem menyimpan data ubah status akun operator | Sistem telah berhasil mengubah data status akun ketika user kepala menekan tombol tolak akun |

1. *Blcakbox Testing* Pada *Form* Tambah Akun Operator

Pada tabel 4.13 merupakan pengujian form tambah akun operatorpada menu kelola akun operator yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada menu *form* tambah akun operator.

Tabel 4.13 *Blackbox Testing Form* Ubah Akun Operator

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* admin mengisi form tambah akun kemudian menekan tombol simpan | Mengisi semluruh form dan menekan  *Button* Simpan akun | Menyimpan data akun operator | Sistem menyimpan data akun operator | Sistem telah mampu menyimpan data akun operator ketika user admin mengisi dan menekan tombol simpan. |
| 2 | *User* admin mengosongkan form tambah akun kemudian menekan tombol simpan | Mengosongkan semluruh form dan menekan  *Button* Simpan | Sistem menolak data layanan memunculkan pesan error | Sistem menolak masukan data akun operator | Sistem telah mampu meno data akun operator ketika user admin mengisi dan menekan tombol simpan pada form tambah akun |

1. *Blcakbox Testing* pada *Form* Ubah Akun Operator

Pada tabel 4.14 merupakan pengujian form tambah akun operatorpada menu kelola akun operator yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form* ubah akun operator.

Tabel 4.14 *Blackbox Testing Form* Ubah Akun Operator

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* admin mengubah data pada form ubah akun kemudian menekan tombol ubah | Mengubah data akun pada form dan menekan  *Button* Simpan akun | Sistem menyimpan data ubah akun operator | Sistem menyimpan data ubah akun operator | Sistem telah berhasil menyimpan data ubah akun operator saat user admin mengubah dan menekan tombol ubah pada form ubah akun operator |

1. *Blcakbox Testing* Pada Menu Data Monitoring

Pada tabel 4.15 merupakan pengujian pada menu data monitoring yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada data monitoring.

Tabel 4.15 *Blackbox Testing* Menu Data Monitoring

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* admin menekan tombol log | Link *log* | Sistem mengarahkan ke *log* penggunaan data | Sistem mengarahkan ke *log* penggunaan data | Sistem telah mampu menampilkan data *log* penggunaan data ketika user admin mekenan tombol log |

1. *Blcakbox Testing* Pada Menu Status Sistem

Pada tabel 4.16 merupakan pengujian pada menu status sistem yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada menu status sistem.

Tabel 4.16 *Blackbox Testing* Menu Status Sistem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* admin menekan batalkan layanan sistem | *Button* batalkan layanan | Sistem memberikan peringatan, dan menampilkan button backup, ya, dan batal | Sistem memunculkan peringatan, dan menampilkan button backup, ya, dan batal | Sistem telah berhasil memberikan peringatan ketika user admin menkan batalkan layanan |
| 2 | *User* admin menekan tombol backup | *Button* pilihan *backup data* | Sistem mengarahkan ke halaman *backup data* | Sistem mengarahkan ke halaman *backup data* | Sistem telah berhasil mengaraha-kan user admin ke halaman *backup data* menkan batalkan layanan |
| 3 | *User* admin menekan tombol ya pada pilihan | *Button* pilihan ‘ya’ | Sistem menghapus semua data record pada satu perusahaan | Sistem menghapus semua data record pada satu perusahaan | Sistem telah berhasil menghapus record pada satu perusahaan ketika user menekan tombol pada pilihan*.* |

1. *Blcakbox Testing* Pada Menu *Backup* Data

Pada tabel 4.17 merupakan pengujian pada menu *backup* data yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada menu *backup* data.

Tabel 4.17 *Blackbox Testing* Menu *Backup* Data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* admin menekan *button* *backup data* surat masuk | *Button* *backup* data surat masuk | Sistem dapat mengambil data dan menjadikan unduhan berupa file | Sistem menampilkan dialog unduh *file* *backup* surat masuk | Sistem telah berhasil mengambil data surat masuk kemudian dapat diunduh oleh *user* |
| 2 | *User* admin menekan button backup data surat keluar | *Button* *backup* data surat keluar | Sistem dapat mengambil data dan menjadikan unduhan berupa file. | Sistem menampilkan dialog unduh *file* *backup* surat keluar | Sistem telah berhasil mengambil data surat keluar kemudian dapat diunduh oleh *user* |
| 3 | *User* admin menekan button backup data disposisi | *Button* backup data surat disposisi | Sistem dapat mengambil data dan menjadikan unduhan berupa file. | Sistem menampilkan dialog unduh *file* *backup* disposisi | Sistem telah berhasil mengambil data disposisi kemudian dapat diunduh oleh *user* |

1. *Blcakbox Testing* Pada Menu *Upgrade* Layanan Pengguna

Pada tabel 4.18 merupakan pengujian pada menu upgrade layanan pengguna yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada menu *upgrade* layanan pengguna.

Tabel 4.18 *Blackbox Testing* Menu *Upgrade* Layanan Pengguna

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* admin memilih salah satu paket yang tersedia pada tampilan upgrade layanan | *Button* pilih paket | Sistem mengarah-kan pada tampilan form lama langganan | Sistem mengarah-kan pada tampilan form lama langganan | Sistem berhasil mengarahkan user admin ke halaman memilih lama langganan saat menekan tombol upgrade |

1. *Blcakbox Testing* Pada *Form* Lama Berlangganan

Pada tabel 4.19 merupakan pengujian *form* lama berlangganan pada menu *upgrade* layanan pengguna yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form* lama berlangganan.

Tabel 4.19 *Blackbox Testing Form* Lama Berlangganan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* admin memilih lama berlanggan-an kemudian menekan tombol cek *payment* | *Button* cek *payment* | Sistem mengarah-kan ke halaman detail pembayaran | Sistem mengarah-kan ke halaman detail pembayaran | Sistem berhasil mengarahkan *user* admin ke halaman detail pembayaran ketika *user* menekan tombol cek *payment* |

1. *Blcakbox Testing* Pada *Form* Detail Pembayaran

Pada tabel 4.20 merupakan pengujian *form* lama berlangganan pada menu *upgrade* layanan pengguna yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form* detail pembayaran.

Tabel 4.20 *Blackbox Testing Form* Detail Pembayaran

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* admin menekan tombol simpan pembayaran | *Button* simpan pembaya-ran | menyimpan data pembayaran lalu mengarahkan ke halaman info pembayaran | Sistem menyimpan data pembayaran, mengarahkan ke halaman info pembayaran | Sistem telah berhasil menyimpan data pembayaran ketika |

1. *Blcakbox Testing* Pada Menu Surat Masuk

Pada tabel 4.21 merupakan pengujian menu surat masuk pengguna yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada surat masuk.

Tabel 4.21 *Blackbox Testing* Menu Surat Masuk

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* operator menekan tombol *input* surat masuk | *Button Input* surat masuk | Sistem mengarahkan ke halaman tambah pengirim | Sistem mengarahkan ke halaman tambah pengirim | Sistem telah berhasil mengarah-kan ke halaman tambah pengirim saat menekan tombol input surat masuk |
| 2 | *User* operator menekan tombol *input* surat masuk saat jumlah data surat sudah berada batas layanan | *Button Input* surat masuk | Sistem tidak mengarahkan ke halaman tambah pengirim karena data surat masuk telah mencapai batas layanan | Sistem tidak mengarahkan ke halaman tambah pengirim | Sistem telah berhasil tidak mengarah-kan ke halaman tambah pengirim saat menekan tombol input surat masuk karena  data  telah mencapai batas layanan |
| 3 | *User* operator menekan tombol ubah surat masuk | *Button* ubah surat masuk | Sistem mengarahkan ke *form* ubah surat masuk | Sistem mengarahkan ke *form* ubah surat masuk | Sistem telah berhasil untuk mengarah-kan ke *form* ubah surat masuk ketika user menekan tombol ubah surat masuk |
| 4 | *User* operator menekan tombol hapus surat masuk | *Button* hapus surat masuk | Sistem menghapus data surat masuk | Sistem menghapus data surat masuk | Sistem telah mampu menghapus data surat masuk ketika user menekan tombol hapus surat masuk |
| 5 | *User* kepala menekan tombol disposisi surat masuk | *Button* disposisi surat masuk | Sistem mengarahkan ke form disposisi surat masuk | Sistem mengarahkan ke form disposisi surat masuk | Sistem telah mampu mengarah-kan ke halaman form disposisi surat masuk ketika menekan tombol disposisi surat masuk |

1. *Blcakbox Testing* Pada *Form* Tambah Pengirim Surat Masuk

Pada tabel 4.22 merupakan pengujian *form* tambah pengirim surat masuk pada menu surat masuk pengguna yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form* tambah pengirim surat masuk.

Tabel 4.22 *Blackbox Testing Form* Tambah Pengirim Surat Masuk

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* operator mengisi *form* pengirim kemudian menekan tombol simpan | Mengisi semluruh *form* dan menekan  *Button* Simpan | Sistem menyimpan data pengirim ketika menekan tombol simpan | Sistem menyimpan data pengirim | Sistem telah berhasil menyimpan masukan data pengirim ketika *user* mengisi *form* kemudian menekan tombl simpan |
| 2 | *User* admin mengosongkan *form* pengirim menekan tombol simpan | Mengosongkan semluruh *form* dan menekan  *Button* Simpan | Sistem menolak data pengirim ketika data pada form kosong | Sistem menolak data pengirim. | Sistem telah berhasil menolak masukan dari *user* ketika form kosong. |

1. *Blcakbox Testing* Pada *Form* Tambah Surat Masuk

Pada tabel 4.23 merupakan pengujian *form* input surat masuk pada menu surat masuk pengguna yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form* input surat masuk.

Tabel 4.23 *Blackbox Testing Form* Tambah Surat Masuk

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* operator mengisi form surat masuk kemudian menekan tombol simpan | Mengisi semluruh form dan menekan  *Button* Simpan | Sistem menyimpan data surat masuk setelah menekan tombol simpan | Sistem menyimpan data surat masuk | Sistem berhasil menyimpan data surat masuk ketika user mengisi form dan kemudian menekan tombol simpan |
| 2 | *User* admin mengosongkan form surat masuk menekan tombol simpan | Mengosongkan semluruh form dan menekan  *Button* Simpan | Sistem menolak data surat masuk, munculkan pesan error | Sistem menolak data surat masuk, munculkan pesan error | Sistem berhasil menolak data masukan dari user ketika mengosongkan form dan kemudian menekan tombol simpan |

1. *Blcakbox Testing* Pada *Form* Ubah Surat Masuk

Pada tabel 4.24 merupakan *form* ubah surat masuk pada pengujian menu surat masuk pengguna yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form* ubah surat masuk.

Tabel 4.24 *Blackbox Testing Form* Ubah Surat Masuk

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* operator mengubah data pada form ubah surat masuk kemudian menekan tombol simpan | Meng-ubah data pada form dan menekan  *Button* ubah | Sistem menyimpan data ubah surat masuk. | Sistem menyimpan data ubah surat masuk. | Sistem telah berhasil menyimpan data ubah surat masuk ketika user mengubah data pada form dan menekan tombol ubah. |

1. *Blcakbox Testing* Pada *Form* Disposisi Surat Masuk

Pada tabel 4.25 merupakan *form* ubah surat masuk pada pengujian menu surat masuk pengguna yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form* disposisi surat masuk.

Tabel 4.25 *Blackbox Testing Form* Disposisi Surat Masuk

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* kepala mendisposisi surat masuk | Mengisi data pada form dan menekan  *Button* simpan | Sistem menyimpan data disposisi surat masuk. | Sistem menyimpan data disposisi surat masuk. | Sistem telah berhasil menyimpan data disposisi surat masuk ketika *user* mengisi data dan menekan tombol simpan pada form disposisi surat masuk |

1. *Blcakbox Testing* Pada Menu Surat Keluar

Pada tabel 4.26 merupakan pengujian menu surat keluar pengguna yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada surat keluar.

Tabel 4.26 *Blackbox Testing* Menu Surat Keluar

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* operator menekan tombol *input* surat keluar | *Button Input* surat keluar | Sistem mengarah-kan ke form *input* surat keluar | Sistem mengarah-kan ke form *input* surat keluar | Sistem telah berhasil mengarahkan *user* ke halaman form surat keluar ketika menekan tombol input surat keluar |
| 2 | *User* operator menekan tombol input surat keluar saat jumlah data telah sama dengan batas layanan | *Button* Input surat keluar | Sistem tidak mengarah-kan ke form input surat keluar, munculkan pesan error | Sistem tidak mengarah-kan ke form input surat keluar, munculkan pesan error | Sistem telah berhasil untuk tidak mengarahkan *user* ke halaman form surat keluar ketika menekan tombol input surat keluar |
| 3 | *User* operator menekan tombol ubah surat keluar | *Button* ubah surat keluar | Sistem mengarah-kan ke form ubah surat keluar | Sistem mengarah-kan ke form ubah surat keluar | Sistem telah berhasil mengarahkan *user* ke halaman form ubah surat keluar ketika menekan tombol ubah surat  keluar |
| 4 | *User* operator menekan tombol hapus surat keluar | *Button* hapus surat keluar | Sistem menghapus data surat keluar | Sistem menghapus data surat keluar | Sistem telah berhasil menghapus data surat keluar ketika user menekan tombol hapus surat keluar |

1. *Blcakbox Testing* Pada *Form* Tambah Surat Keluar

Pada tabel 4.27 merupakan pengujian *form* input surat keluar pada menu surat masuk pengguna yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form* input surat keluar.

Tabel 4.27 *Blackbox Testing Form* Tambah Surat Keluar

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* operator mengisi form surat keluar kemudian menekan tombol simpan | Mengisi semluruh form dan menekan  *Button* Simpan | Sistem menyimpan data surat keluar. | Sistem menyimpan data surat keluar. | Sistem telah berhasil menyimpan data surat keluar ketika *user* mengisi form dan menekan tombol simpan. |
| 2 | *User* admin mengosongkan form surat keluar menekan tombol simpan | Mengosongkan semluruh form dan menekan  *Button* Simpan | Sistem menolak data surat keluar, munculkan pesan error | Sistem menolak data surat keluar, munculkan pesan error | Sistem telah berhasil menolak data surat keluar ketika *user* mengosongkan form kemudian menekan tombol simpan. |

1. *Blcakbox Testing* Pada *Form* Ubah Surat Keluar

Pada tabel 4.28 merupakan *form* ubah surat keluar pada pengujian menu surat masuk pengguna yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form* ubah surat keluar.

Tabel 4.28 *Blackbox Testing Form* Ubah Surat Keluar

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* operator mengubah data pada form ubah surat keluar kemudian menekan tombol ubah | Meng-ubah data pada form dan menekan  *Button* ubah | Sistem menyimpan data ubah surat keluar. | Sistem menyimpan data ubah surat keluar. | Sistem telah berhasil menyimpan data ubah surat keluar ketika user mengisi form dan menekan tombol ubah surat. |

1. *Blcakbox Testing* Pada Menu Data Disposisi

Pada tabel 4.29 merupakan pengujian menu data disposisi yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada menu data disposisi.

Tabel 4.29 *Blackbox Testing* Menu Data Disposisi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* operator menekan tombol detail disposisi | *Button* detail disposisi | Sistem mengarah-kan ke halaman detail disposisi | Sistem mengarah-kan ke halaman detail disposisi | Sistem berhasil mengarahkan *user* ke halaman detail disposisi ketika user menekan tombol detail disposisi |
| 2 | *User* kepala menekan tombol ubah disposisi | *Button* ubah disposisi | Sistem mengarah-kan ke form ubah disposisi | Sistem mengarah-kan ke form ubah disposisi | Sistem berhasil mengarahkan *user* ke halaman ubah disposisi ketika *user* menekan tombol ubah disposisi |
| 3 | *User* kepala menekan tombol hapus disposisi | *Button* hapus disposisi | Sistem menghapus data disposisi | Sistem menghapus data disposisi | Sistem telah berhasil menghapus data disposisi ketika *user* menekan tombol hapus disposisi |

1. *Blcakbox Testing* Pada *Form* Ubah Disposisi Surat Masuk

Pada tabel 4.30 merupakan *form* ubah disposisi surat masuk pada pengujian menu data disposisi pengguna yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form* ubah disposisi surat masuk.

Tabel 4.30 *Blackbox Testing Form* Ubah Disposisi Surat Masuk

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* operator mengubah data disposisi kemudian menekan tombol ubah disposisi | Meng-ubah data pada *form* dan menekan  *Button* ubah disposisi | Sistem menyimpan data ubah disposisi. | Sistem me-nyimpan data ubah disposisi. | Sistem telah berhasil menyimpan data ubah disposisi ketika user merubah data pada form kemudian menekan tombol ubah disposisi |

1. *Blcakbox Testing* Pada Menu Data Tindak Lanjut

Pada tabel 4.31 merupakan pengujian menu data tindak lanjut yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada menu data tindak lanjut.

Tabel 4.31 *Blackbox Testing* Menu Data Tindak Lanjut

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* kepala menekan tombol tambah tindak lanjut | *Button* tambah tindak lanjut | Sistem mengarah-kan ke form tambah tindak lanjut | Sistem mengarah-kan ke form tambah tindak lanjut | Sistem telah berhasil mengarahkan *user* ke halaman form tambah tindak lanjut ketika *user* menekan tombol tambah tindak lanjut. |
| 2 | *User* kepala menekan tombol ubah tindak lanjut | *Button* ubah tindak lanut | Sistem mengarah-kan ke form ubah tindak lanjut | Sistem mengarah-kan ke form ubah tindak lanjut | Sistem telah berhasil mengarahkan *user* ke halaman form ubah tindak lanjut ketika *user* menekan tombol ubah tindak lanjut. |
| 3 | *User* kepala mengisi form tindak lanjut kemudian menekan *tombol* simpan | Mengisi data tindak lanjut dan menekan *button* simpan | Sistem menyimpan data tindak lanjut. | Sistem men-yimpan data tindak lanjut. | Sistem telah berhasil menyimpan data tindak lanjut ketika *user* mengisi form tindak lanjut lalu menekan tombol simpan |

1. *Blcakbox Testing* Pada *Form* Ubah Tindak Lanjut.

Pada tabel 4.32 merupakan *form* ubah tindak lanjut pada pengujian menu data menu data tindak lanjut yang bertujuan untuk mengetahui keluaran dari sistem serta hasil pengujian pada *form ubah tindak lanjut*.

Tabel 4.32 *Blackbox Testing Form* Ubah Tindak Lanjut

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Sekenario Pengujian** | **Test Case** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | *User* operator mengubah data tindak lanju dan menekan tombol ubah | Meng-ubah data pada *form* dan menekan  *Button* ubah | Sistem menyimpan data tindak lanjut. | Sistem me-nyimpan data tindak lanjut. | Sistem telah berhasil mengubah data tindak lanjut saat user mengubah data pada form kemudian menekan tombol ubah tindak lanjut |

**BAB V**

**KESIMPULAN DAN SARAN**

* 1. **Kesimpulan**

Telah dilakukannya penelitian mengenai Implementasi *Software as a Service Cloud Computing* Sistem Manajemen Surat Menyurat dimana didapatkan kesimpulan pada penelitian ini, yaitu:

1. Dihasilkanya aplikasi surat menyurat yang dapat digunakan oleh semua instansi atau lembaga dengan berbasiskan *Cloud Computing* dan dengan mengimplementasikan konsep *Software as a Service*.
2. Sistem yang telah dibuat dapat dikatakan sebagai *Cloud Computing* karena telah memenuhi 5 karakteristik dari *Cloud Computing* menurut NIST (*National Institute of Standards and Technology*) yaitu, sistem ini berbasiskan web yang memanfaatkan berbagai sumber daya server yang terpusat (*Resource pooling*), sistem dapat diakses menggunakan berbagai macam perangkat (*Broad network access­*), terdapat informasi batas jumlah baris data dan seberapa banyak *user* telah memasukan data kedalam sistem (*Measured service*), kemudian paket layanan yang dapat dengan mudah ditingkatkan dan dibatalkan sesuai dengan kebutuhan pengguna (*Rapid elasticity*) dan pengguna tidak perlu berinteraksi langsung dengan penyedia layanan dalam meningkatkan atau menurunkan layanan (*On-demand self-service*).
3. Berdasarkan pengujian sistem menggunakan metode *blackbox testing*, maka didapatkan hasil dari pengujian tersebut bahwa aplikasi yang dibuat telah sesuai dengan fungsionalitas sistem.
   1. **Saran**

Sistem yang dihasilkan pada penelitian ini masih jauh dari kata sempurna Karena didalamnya masih terdapat kekurangan yang dapat diperbaiki dan dikembangkan untuk menghasilkan sistem yang lebih baik, berikut adalah beberapa hal yang dapat dilakukan untuk pengembangan sistem.

1. Menambah fitur-fitur dalam manajemen surat menyurat agar dapat lebih mudah digunakan dan lebih fleksibel untuk diimplementasikan.
2. Sistem *cloud* ini dapat dikembangkan menjadi sistem *cloud computing* berbasis *Infrastructure as a Service* (IAAS) untuk memberikan layanan penyimpanan data yang lebih optimal dan lebih fleksibel.
3. Aplikasi ini dapat dikembangkan pada platform yang lain misalnya pada mobile untuk lebih mempermudah mengkases data surat agar lebih cepat.

**DAFTAR PUSTAKA**

1. I Kadek Ariana, Gde Sastrawangsa, Agus Purwanto. Sistem Informasi Administrasi Surat Berbasis Web Pada Kantor Camat Gianyar. JOSINFO. 2015; Vol. 01, No. 01.
2. I Gusti Putu Arlina Ayuningsih, Dian Pramana, Dadang Hermawan. Implementasi Software As A Service Pada Sistem Layanan Klinik Kesehatan. JOSINFO. 2015; Vol. 01, No. 01.
3. National Institute Of Standards and Technology. Special Publication 800-145. *The NIST Definition of Cloud Computing.* Computer Security Division Information Technology Laboratory NIST Gaithersburg. U.S Department of Commerse. 2011.
4. Yoga Kurniawan, Sholiq, Feby Artwodini Muqtadiroh. Rancang Bangun Perangkat Lunak untuk Workflow Pengelolaan Surat Menyurat Dinas Bagian Surat Masuk di Kabupaten Buton Utara. 2012; JURNAL TEKNIK ITS Vol.1: A272.
5. B. Barthos, Manajemen Kearsipan*.* Jakarta: Bumi Aksara, 2003.
6. H. M. Wirladiharjo, Pedoman Administrasi Umum. Jakarta: Balai Pustaka. 1991.
7. S.M Silmi, Panduan Menulis Surat Lengkap. Yogyakarta: Absolit. 2002.
8. Kadir, Abdul. Dasar Pemrograman Web Dinamis Menggunakan PHP. Edisi Revisi. Yogyakarta: Penerbit Andi. 2008.
9. Nugroho.Adi. Rekayasa Perangkat Lunak Menggunakan UML & Java. Yogyakarta: Andi Offset. 2009.
10. Herlawati,Widodo. Menggunakan UML. Bandung:Informatika. 2011
11. Prastuti Sulistyorini, Pemodelan Visual dengan Menggunakan UML dan Rational Rose, Jurnal Teknologi Informasi DINAMIK.Volume XIV, No.1, Januari 2009.23-29