

BAB III

ANALISIS

3.1 Deskripsi Sistem Yang Berjalan

Kebutuhan informasi dalam menjalani kehidupan sehari-hari merupakan sesuatu yang penting untuk menunjang berbagai jenis objek pekerjaan. Seseorang akan lebih maju, sukses dan kuat dalam menghadapi persaingan jika mempunyai pengetahuan yang luas serta *up to date*. Dengan informasi kita akan mendapatkan inspirasi tidak hanya dalam konteks pekerjaan saja namun juga dalam bidang pendidikan bahkan hiburan.

Untuk melayani kebutuhan akan informasi tersebut telah banyak media baik cetak (koran dan majalah) ataupun elektronik (televisi dan radio) yang memenuhi kebutuhan tersebut. Tidak hanya itu saja berdasarkan kemajuan ilmu pengetahuan dan teknologi, sekarang telah lahir media jaringan virtual yang juga ikut serta mengambil bagian dalam menyediakan layanan informasi yaitu internet. Dengan lahirnya media penyampaian informasi ini seseorang dapat menjelajah dan mendapat informasi lebih luas, mudah dan efektif jika dibandingkan dengan kedua media penyedia layanan informasi lainnya. Melalui jaringan internet pula seseorang dapat melakukan promosi usaha, jasa, hiburan bahkan menyumbangkan ilmu pengetahuan yang mereka miliki untuk dapat dikonsumsi oleh publik dengan membuat suatu *website*. Pada saat ini penyampaian informasi melalui *web* sebagian besar hanya menyampaikan informasi yang mereka miliki dalam bentuk *teks*, gambar atau sebatas animasi.

Salah satu teknologi pengiriman data dalam *web* yaitu dengan melakukan proses *download*. Tetapi teknik ini tidak tepat dilakukan oleh penyedia layanan selain tidak efektif, karena seseorang/*user* harus menunggu terlebih dahulu dalam waktu yang cukup lama hingga *file* tersebut selesai di *download* baru dapat dilihat, proses *download* juga mungkin akan mengundang lahirnya modus *cyber crime* yaitu penggandaan *file* yang kemudian diperbanyak dan diperjualbelikan secara ilegal oleh pihak-pihak yang tidak bertanggung jawab. Hal ini kemungkinan menjadi hambatan dan mungkin merupakan suatu alasan mengapa penyedia informasi dalam *website* tidak menyisipkan layanan dalam bentuk

pemutaran *video* karena tentu saja mereka ingin melindungi hak cipta yang mereka miliki.

3.2 Identifikasi Masalah

Berdasarkan deskripsi sistem yang berjalan dan menganalisa proses yang terjadi diatas dapat ditarik kesimpulan bahwa kebutuhan informasi merupakan sesuatu yang penting untuk menunjang kehidupan manusia dalam berbagai hal, namun sayangnya penyedia layanan informasi di web masih jarang yang menyediakan informasi dalam bentuk tayangan *video*.

3.3 Analisa Sistem

Dengan melihat aplikasi dan proses yang sedang berjalan, terlihat beberapa kebutuhan sistem yang belum terpenuhi yaitu bagaimana membuat suatu layanan yang menyediakan kebutuhan informasi tidak hanya dalam bentuk tulisan dan gambar melainkan juga dilengkapi dalam bentuk *video*. Kemudian bagaimana menyampaikan informasi *video* tersebut dengan cepat dan tidak akan menimbulkan pelanggaran hak cipta oleh perilaku kriminal. Permasalahan tersebut merupakan suatu hal mengapa dirasakan perlu adanya pengembangan pada sistem yang sedang berjalan, berupa pelengkap penyajian informasi dalam bentuk *video*.

3.4 Hasil Analisa Sistem

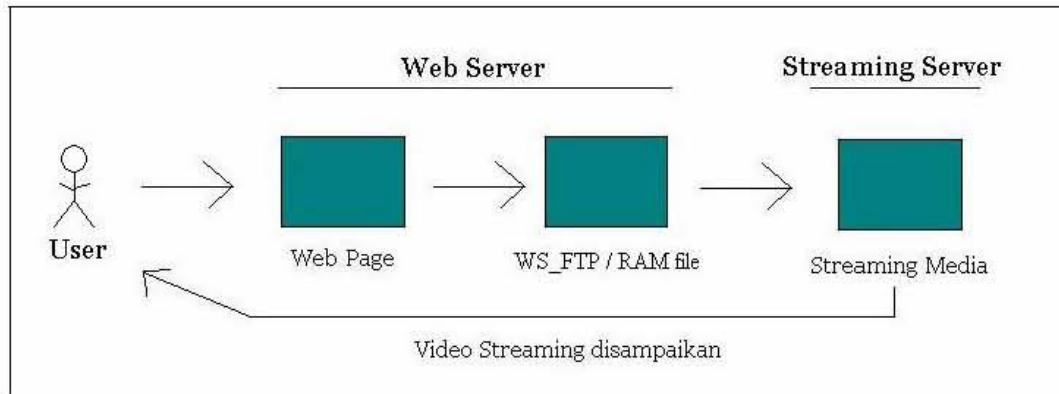
Setelah melakukan analisa sistem yang berjalan, penulis mendapatkan beberapa hasil yang didapat dari analisa tersebut yaitu sebagai berikut:

1. Kebutuhan akan informasi merupakan sesuatu hal yang penting sebagai sumber inspirasi karena dapat membantu manusia dalam menghadapi dan mendapatkan solusi berbagai jenis objek kehidupan.
2. Menyediakan layanan informasi dalam bentuk penayangan *video* sangat mungkin dilakukan dengan *web* melalui internet, namun tidak dilakukan dengan proses *download* melainkan dengan teknologi *streaming*.
3. Dengan teknologi *streaming* audio dan *video*, memungkinkan *user* dapat melihat dan menikmati data berupa suara dan *video* tanpa perlu melakukan

proses *download* (pengambilan *file* dari *server*) ke komputer *user*. Dengan *file video* sebesar 100 MB jika harus *download* ke komputer *user* akan memakan waktu yang cukup lama sesuai dengan kapasitas bandwidth dari jaringan yang digunakan. Namun dengan teknologi *streaming* hanya akan membutuhkan proses *buffering* untuk menayangkan *video* tersebut. Dan dengan proses ini *video* tidak akan ter-*copy* ke komputer *user* sehingga tidak mungkin terjadi *cyber crime* dan tayangan bersifat eksklusif.

4. Cara kerja dari media yang di-*stream* tidak dapat dikirim dari sebuah standar *web server* dengan menggunakan *hypertext transport protokol* atau HTTP yang merupakan protokol untuk pengiriman dokumen dari *web browser* dan HTTP merupakan protokol yang tidak memiliki status. Keadaan ini mengakibatkan kesulitan yang dialami protokol HTTP manakala data *streaming* akan dihentikan sementara (*pause*) saat pengiriman data berlangsung. Hal ini dikarenakan pelaksanaan *pause* membutuhkan pengetahuan (biasanya status) dari *web browser*, sehingga nantinya ketika *user* ingin memulai kembali dikirimkan data melalui *streaming* sistem dapat dengan mudah melanjutkan kembali berdasarkan status yang tersimpan tersebut.
5. Strategi alternatif yang dapat dilakukan untuk menanggulangi hal di atas adalah dengan menggunakan protokol *streaming* khusus yang didesain untuk men-*streaming* media, yaitu real-time streaming protocol (RTSP). RTSP didesain untuk melakukan komunikasi antara *server* yang melakukan *streaming* dengan media player. Keuntungan RTSP adalah bahwa protokol ini menyediakan koneksi yang memiliki status antara *server* dan *user*, yang dapat mempermudah *user* ketika ingin melakukan *pause* atau mencari posisi *random* dalam *stream* ketika memutar kembali data *streaming* tersebut.
6. Untuk menayangkan data *streaming* biasanya digunakan media player, seperti QuickTime, RealOne Player, atau Windows Media Player untuk memutar media yang di-*stream* dari sebuah *web server* yang disertakan pada halaman web. Biasanya, pertama-tama sistem akan meminta *metafile*, yang meliputi keterangan alamat letak data yang di-*stream* berada (URL),

dari data yang di-*stream*. *Metafile* ini nantinya akan dikirim ke *web browser user*, kemudian *browser* akan membuka berkas yang dimaksud dengan memilih media *player* yang sesuai dengan jenis media yang dispesifikasikan oleh *metafile*. Berikut adalah ilustrasi tentang pengiriman data streaming melalui web:



Gambar 3.1 Ilustrasi penyampaian data streaming

3.5 Pembahasan Sistem yang akan dikembangkan

Sesuai dengan hasil analisa sistem, maka akan ditentukan fungsi atau fitur apa saja yang dibuat, selain dari fungsi utama sebagai penyampaian informasi dan hiburan dalam bentuk *video*. Fungsi sistem yang akan dibuat akan dimodelkan dengan *use case diagram*, selanjutnya *use case* akan terhubung dengan *actor*. Berdasarkan kebutuhan sistem yang akan di bangun terdapat beberapa *actor* sebagai berikut:

1. *Acount* yaitu *visitor* yang sudah melakukan proses pendaftaran sehingga dapat menggunakan semua fasilitas yang disediakan oleh aplikasi seperti mempromosikan barang/jasa atau mengirimkan berita untuk ditampilkan di web.
2. *Visitor* yaitu pengunjung biasa yang tidak/belum melakukan registrasi sehingga hanya dapat menggunakan fasilitas *web* secara terbatas.
3. *Administrator* yaitu seseorang yang bertugas dalam mengatur *video* dan informasi yang akan ditayangkan selanjutnya, dan melakukan penyaringan informasi yang diberikan oleh *user* untuk ditayangkan dalam *web*.

Untuk mendeskripsikan *use case* apa saja yang diperlukan dan *actor* mana saja yang terlibat dalam *use case* tersebut maka gambarannya dapat dilihat pada tabel sebagai berikut:

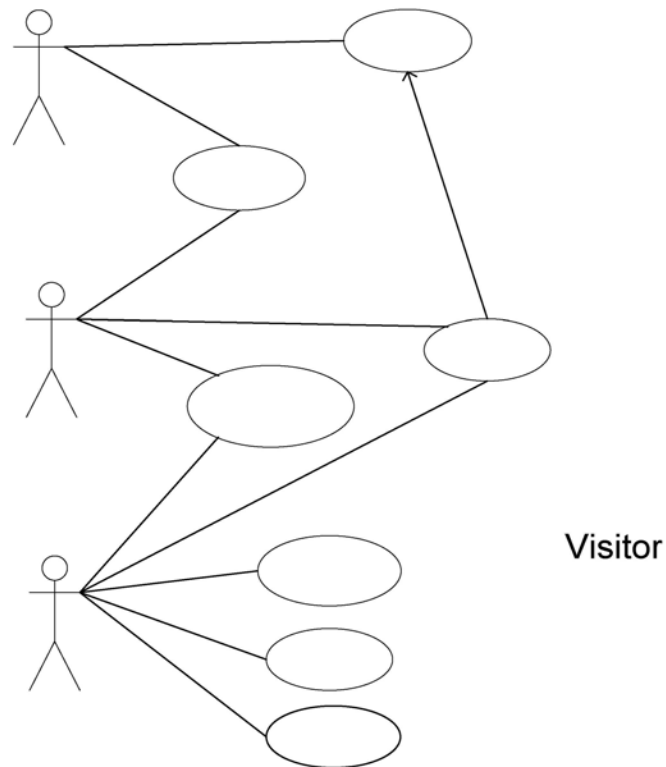
Tabel 3.1 *Requirement actor* dan *use case*

No	Requirement	Actor	Use Case
1	<i>Web</i> menampilkan halaman awal, yang berisi tentang berbagai informasi.	<i>Visitor</i> , <i>account</i>	Tampilkan informasi/index
2	<i>Web</i> dapat menyediakan layanan informasi dalam bentuk <i>video</i>	<i>Visitor</i> , <i>account</i>	<i>Demand video</i>
3	<i>Visitor</i> dapat melakukan pendaftaran sehingga dapat menggunakan fasilitas tambahan <i>web</i> secara maksimal.	<i>Visitor</i>	<i>Create account</i>
4	<i>Account</i> dapat mengirimkan informasi berupa surat pembaca ke dalam web.	<i>Account</i>	Surat Pembaca
5	<i>Account</i> dapat mengirimkan informasi <i>event</i> atau acara yang akan mereka laksanakan.	<i>Account</i>	Event
6	<i>Account</i> dapat mengisi kolom iklan (mempromosikan produk atau jasa) dan kemudian akan di tampilkan pada halaman web. Setelah di <i>approve</i> oleh <i>administrator</i>	<i>Account</i>	Pasang iklan
7	<i>Account</i> dapat mengisi kolom berita yang akan di sampaikan kepada seluruh <i>account</i> dan akan di tampilkan pada halaman web. Setelah di <i>approve</i> oleh <i>admin</i>	<i>Account</i>	Kirim Berita
8	<i>Account</i> dapat mengirim informasi dalam bentuk penayangan <i>video</i> .	<i>Account</i>	Kirim <i>Video</i>
9	<i>Account</i> harus <i>login</i> terlebih dahulu sebelum dapat menggunakan semua fasilitas <i>web</i> .	<i>Account</i>	<i>Account login</i>

10	Terdapat seorang <i>administrator web</i> yang dapat mengatur keseluruhan data yang ada dalam <i>web</i> .	<i>Administrator</i>	Atur Berita, atur <i>video</i> , atur iklan, atur event, atur surat pembaca.
11	Admin juga dapat melakukan berbagai inputan kedalam <i>web</i> .	<i>Administrator</i>	Input Berita, input <i>video</i> , input iklan, input event, input surat pembaca.
12	Agar dapat mengatur keseluruhan data pada <i>web</i> maka <i>administrator web</i> harus terlebih dahulu <i>login</i> ke <i>administrator form</i> .	<i>Administrator</i>	<i>Administrator login</i> .

3.5.1 Use Case Diagram

Use case merupakan suatu gambaran atau konstruksi dari sistem yang akan dibuat. *Use case* ditentukan berdasarkan kebutuhan sistem yang telah ditetapkan sebelumnya dari hasil analisa permasalahan yang ada. Berikut adalah *use case* diagram dari sistem yang akan dibuat:



Gambar 3.2 Use case diagram

Dalam *use case* di atas terdapat tiga aktor yaitu *visitor*, *acount* dan *administrator*. Untuk memudahkan dalam menganalisa skenario yang akan digunakan pada *fase-fase* selanjutnya, maka akan dilakukan pemilahan terhadap skenario berdasarkan *usecase* dan aktor yang berhubungan dengan sistem. Berikut adalah skenario-skenario tersebut.

Tabel 3.2 Skenario untuk *use case create acount*

<i>Use case name :</i>	<i>Create Acount</i>
<i>Brief Description :</i>	<i>Use case ini memungkinkan visitor melakukan proses pendaftaran, dengan cara mengisi data-data yang telah disediakan oleh sistem</i>
<i>Actor :</i>	<i>Visitor</i>
<i>Precondition :</i>	<i>Visitor masuk ke halaman create acount dari layanan penyedia informasi (infonet). Web menampilkan halaman create acount yang berisi tentang form isian tentang data-data visitor yang diperlukan oleh sistem.</i>
<i>Main Flow :</i>	<i>Use case ini dimulai saat visitor membuka halaman</i>

	<p><i>create account.</i></p> <p>Aplikasi meminta <i>visitor</i> untuk mengisi beberapa data diri <i>visitor</i> yang diperlukan oleh sistem.</p> <p>Sistem melakukan proses <i>verifikasi</i> data <i>visitor</i>.</p> <p>Sistem mengirimkan pesan proses pendaftaran berhasil dilakukan.</p>
<i>Alternative Flow :</i>	Jika ada data yang tidak lengkap / tidak sesuai, sistem akan mengirimkan pesan kesalahan kepada <i>visitor</i> , dan meminta <i>visitor</i> untuk melakukan koreksi terhadap data tersebut.
<i>Post Condition :</i>	Jika <i>use case</i> berhasil dijalankan, data-data <i>visitor</i> tadi akan disimpan ke dalam <i>database</i> sistem dan tercatat sebagai <i>account</i> .

Tabel 3.3 Skenario untuk *use case* lihat informasi

<i>Use case name :</i>	Lihat Informasi.
<i>Brief Description :</i>	<i>Use case</i> ini memungkinkan <i>visitor/account</i> untuk melihat informasi yang disediakan oleh sistem. Seperti informasi berita, <i>event</i> , iklan dan surat pembaca.
<i>Actor :</i>	<i>Visitor, Account.</i>
<i>Precondition :</i>	<i>Visitor/account</i> masuk ke halaman layanan informasi. <i>Web</i> menampilkan pilihan pilihan informasi yang dapat dilihat oleh <i>visitor/account</i> .
<i>Main Flow :</i>	<p><i>Use case</i> ini dimulai pada saat <i>visitor/account</i> masuk ke halaman <i>index</i> yang berisi tentang pilihan layanan informasi yang dapat dilihat oleh <i>visitor/account</i>.</p> <p><i>Visitor/account</i> memilih jenis informasi yang ingin dilihat (Berita, Iklan, Surat Pembaca, <i>Event</i>).</p> <p>Sistem menayangkan informasi yang dipilih oleh <i>visitor/account</i> dan menayangkan dalam bentuk <i>video</i> untuk kategori berita, iklan atau dalam bentuk teks untuk kategori berita, iklan, surat pembaca dan <i>event</i>.</p>

<i>Post Condition :</i>	<i>Web</i> mengaktifkan halaman informasi sesuai dengan kategori pilihan dari <i>visitor/acount</i> . Dan menayangkan informasi tersebut dalam bentuk teks atau <i>video</i> .
-------------------------	--

Tabel 3.4 Skenario untuk *use case login*.

<i>Use case name :</i>	<i>Login</i> .
<i>Brief Description :</i>	<i>Use case</i> ini memungkinkan seorang <i>acount/administrator</i> untuk masuk ke halaman khusus sesuai dengan hak akses yang mereka miliki. Untuk <i>acount</i> , setelah sukses <i>login acount</i> dapat berpartisipasi secara aktif dengan mengirimkan informasi yang dimilikinya. Untuk <i>administrator</i> setelah berhasil <i>login</i> , <i>administrator</i> dapat mengirimkan informasi dan juga melakukan proses pengaturan berita serta informasi lainnya.
<i>Actor :</i>	<i>Acount, Administrator</i>
<i>Precondition :</i>	<i>Acount/administrator</i> membuka halaman <i>login</i> .
<i>Main Flow :</i>	<i>Use case</i> ini dimulai pada saat <i>acount/administrator</i> memasukkan <i>user name</i> dan <i>password</i> ke <i>login interface</i> . Sistem kemudian melakukan proses <i>validasi</i> berdasarkan <i>user name</i> dan <i>password</i> yang mereka miliki. Sistem kemudian membuka <i>session</i> (halaman <i>web</i>) <i>acount/administrator</i> sesuai dengan hak akses masing-masing.
<i>Alternative Flow :</i>	Jika <i>username</i> dan <i>password</i> yang dimasukkan oleh <i>acount/administrator</i> tidak <i>valid</i> , sistem akan mengeluarkan pesan kesalahan. Sistem mengijinkan <i>acount/administrator</i> melakukan koreksi dan mengulang kembali untuk memasukkan <i>user name</i> dan <i>password</i> .

<i>Post Condition :</i>	Sistem mengaktifkan halaman <i>account/administrator</i> dan selanjutnya <i>account</i> dan <i>administrator</i> dapat berinteraksi dengan sistem untuk mengirim atau mengatur informasi.
-------------------------	---

Tabel 3.5 Skenario untuk *use case* mengirim informasi

<i>Use case name :</i>	Mengirim Informasi
<i>Brief Description :</i>	<i>Use case</i> ini memungkinkan <i>account/administrator</i> untuk mengirimkan informasi seperti pasang iklan, kirim berita, kirim <i>event</i> dan kirim surat pembaca. Selanjutnya informasi tersebut akan ditampilkan oleh web jika telah di <i>verifikasi</i> oleh <i>administrator</i> .
<i>Actor :</i>	<i>Account</i> , <i>administrator</i>
<i>Precondition :</i>	<i>Account/administrator</i> telah berhasil melakukan proses <i>login</i> , kemudian membuka layanan pengiriman informasi dalam sistem.
<i>Main Flow :</i>	<p><i>Use case</i> ini dimulai pada saat <i>account</i> membuka layanan kirim informasi, kemudian memilih salah satu jenis layanan pengiriman informasi.</p> <p><i>Account/administrator</i> mengisi komponen-komponen informasi seperti judul, deskripsi, <i>content</i> dan sebagainya. Sistem akan melakukan proses pengecekan apakah data berhasil terkirim.</p> <p>Sistem mengirimkan pesan proses pengiriman informasi berhasil.</p> <p>Setelah dilakukan proses <i>verifikasi</i> oleh <i>administrator</i> (data memenuhi syarat). Informasi akan ditampilkan dalam <i>web</i>.</p>
<i>Alternative Flow :</i>	Jika data tidak berhasil terkirim sistem akan mengirimkan pesan kesalahan. Kemudian sistem mempersilahkan <i>account/administrator</i> untuk mengulang kembali proses pengiriman informasi.

<i>Post Condition :</i>	Informasi dikirim ke administrator dan disimpan dalam <i>database</i> .
-------------------------	---

Tabel 3.6 Skenario untuk *use case manage content*.

<i>Use case name :</i>	<i>Manage content</i> .
<i>Brief Description :</i>	<i>Use case</i> ini memungkinkan administrator untuk melakukan proses pengaturan isi informasi untuk ditampilkan dalam <i>web</i> , seperti melakukan <i>verifikasi</i> informasi dari <i>account</i> dan juga melakukan proses <i>input video</i> .
<i>Actor :</i>	Administrator.
<i>Precondition :</i>	Administrator berhasil melakukan proses <i>login</i> , kemudian sistem menampilkan menu pengaturan informasi.
<i>Main Flow :</i>	<i>Use case</i> ini dimulai pada saat administrator masuk ke halaman pengaturan informasi. Sistem secara otomatis akan menampilkan informasi yang dikirim ke administrator Administrator melakukan proses pengaturan informasi baik dalam bentuk <i>teks</i> ataupun <i>video</i> . Informasi yang diterima akan ditampilkan dalam <i>web</i> .
<i>Alternative Flow :</i>	Jika data tidak memenuhi syarat akan ditolak oleh administrator dan tidak akan ditampilkan pada <i>web</i> .
<i>Post Condition :</i>	Informasi yang tidak diterima akan di hapus dari <i>database</i> .

Tabel 3.7 Skenario untuk *use case convert video*.

<i>Use case name :</i>	<i>Convert Video</i>
<i>Brief Description :</i>	Sebelum dimasukkan ke dalam <i>web</i> , <i>file video</i> harus di ubah terlebih dahulu kedalam salah satu bentuk <i>file streaming</i> , misalnya bentuk <i>real media file</i> agar dapat disampaikan dengan teknologi <i>streaming</i> . Dalam <i>use</i>

	<i>case</i> inilah proses tersebut dilakukan oleh administrator.
<i>Actor :</i>	Administrator.
<i>Precondition :</i>	Administrator menjalankan <i>software</i> konfersi <i>file</i> , misalnya real producer basic.
<i>Main Flow :</i>	<i>Use case</i> ini dimulai saat administrator memasukkan <i>file video</i> yang akan dikonvert ke dalam salah satu <i>software</i> konverter misalnya real producer basic. <i>Software</i> tersebut kemudian melakukan proses konverter <i>file</i> . <i>Software</i> mengeluarkan pesan proses konversi <i>file</i> berhasil.
<i>Alternative Flow :</i>	Jika proses konversi gagal, sistem akan mengirimkan pesan kesalahan dan administrator dapat kembali mengulang proses konversi <i>file</i> tersebut.
<i>Post Condition :</i>	<i>Video</i> dengan jenis real media <i>file</i> .

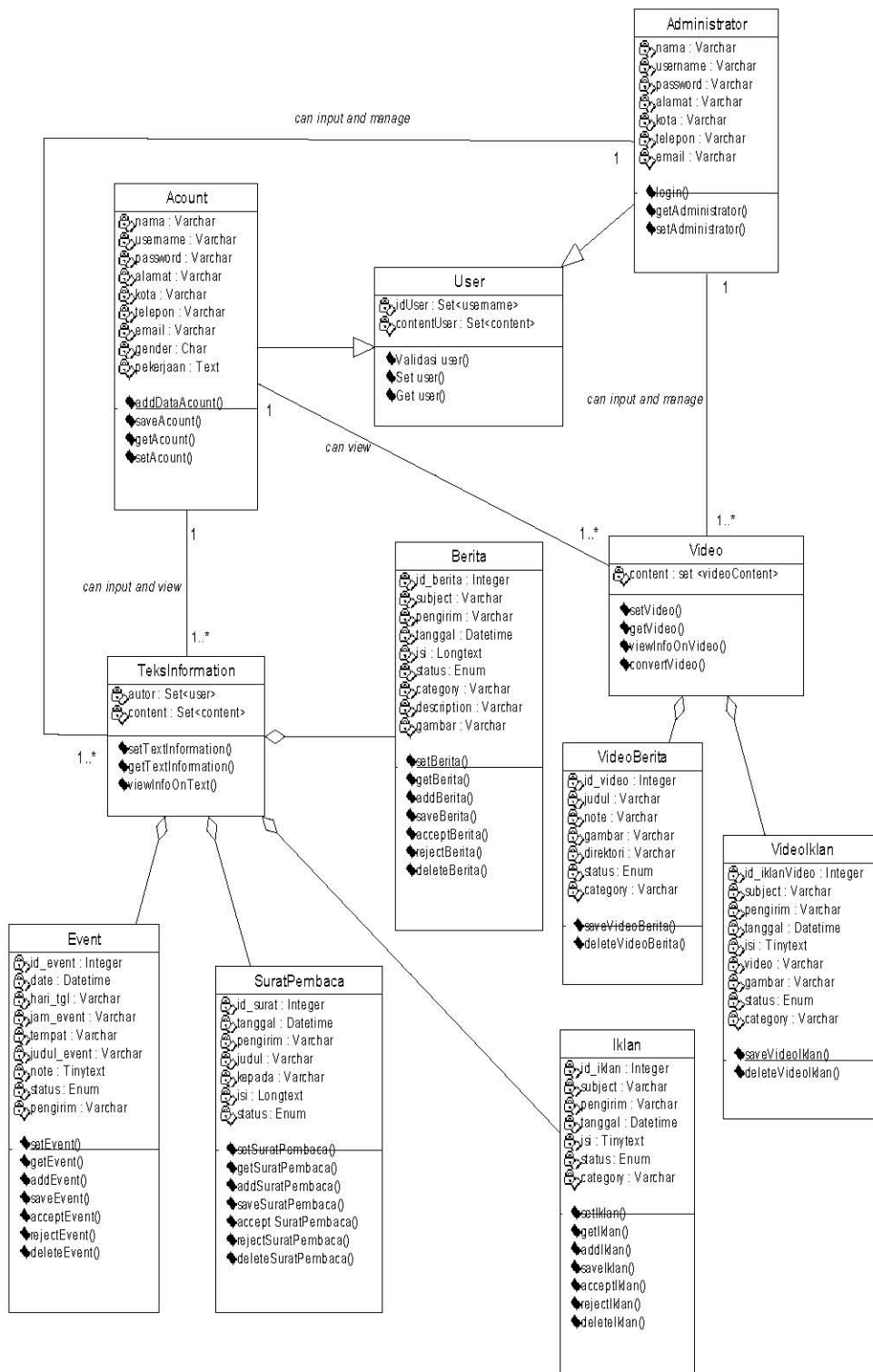
Tabel 3.8 Skenario untuk *use case input video*.

<i>Use case name :</i>	<i>Input Video</i>
<i>Brief Description :</i>	<i>Use case</i> ini memungkinkan administrator untuk mengirimkan informasi dalam bentuk <i>video</i> untuk jenis iklan dan berita.
<i>Actor :</i>	Administrator.
<i>Precondition :</i>	Administrator telah berhasil melakukan proses <i>login</i> , kemudian membuka layanan <i>input video</i> .
<i>Main Flow :</i>	<i>Use case</i> ini dimulai pada saat admistrator membuka layanan <i>input video</i> . Administrator mengisi komponen-komponen <i>video</i> seperti judul, <i>note</i> , <i>file video</i> dan sebagainya. Sistem akan melakukan proses pengecekan apakah data berhasil terkirim. Sistem mengirimkan pesan proses <i>input video</i> berhasil.

<i>Alternative Flow :</i>	Jika data tidak berhasil terkirim sistem akan mengirimkan pesan kesalahan. Kemudian sistem mempersilahkan administrator untuk mengulang kembali proses <i>input video</i> .
<i>Post Condition :</i>	<i>Video</i> tersimpan ke <i>database</i> .

3.5.2 Class Diagram

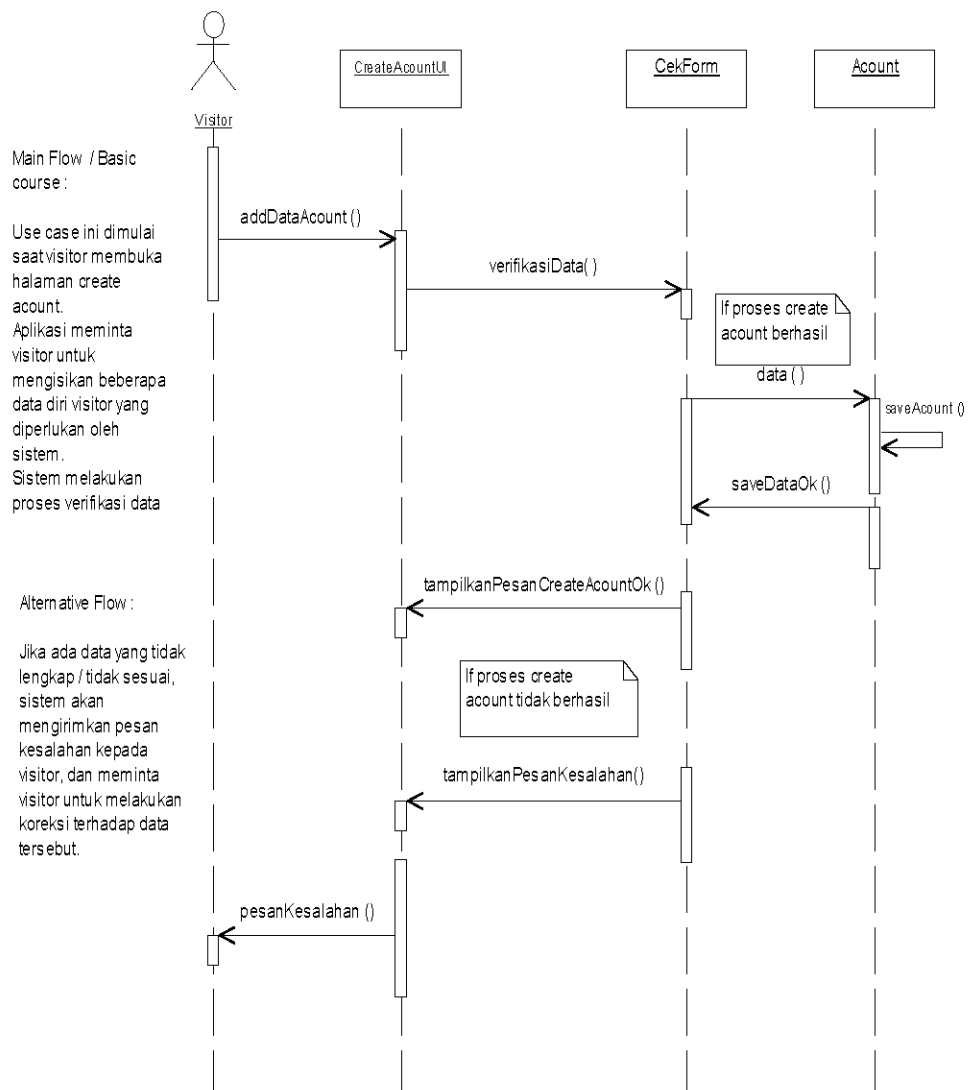
Setelah menganalisa *use case diagram* dan *activity diagram* maka kita dapat mengetahui kebutuhan sistem dan menentukan *class* apa saja yang diperlukan dalam aplikasi ini. Untuk memodelkan *class-class* yang ada maka digunakan *class diagram* yang akan menggambarkan *class-class* yang ada beserta relasi antar *class-class*. Berikut adalah *class diagram* dari *website* yang dibangun:



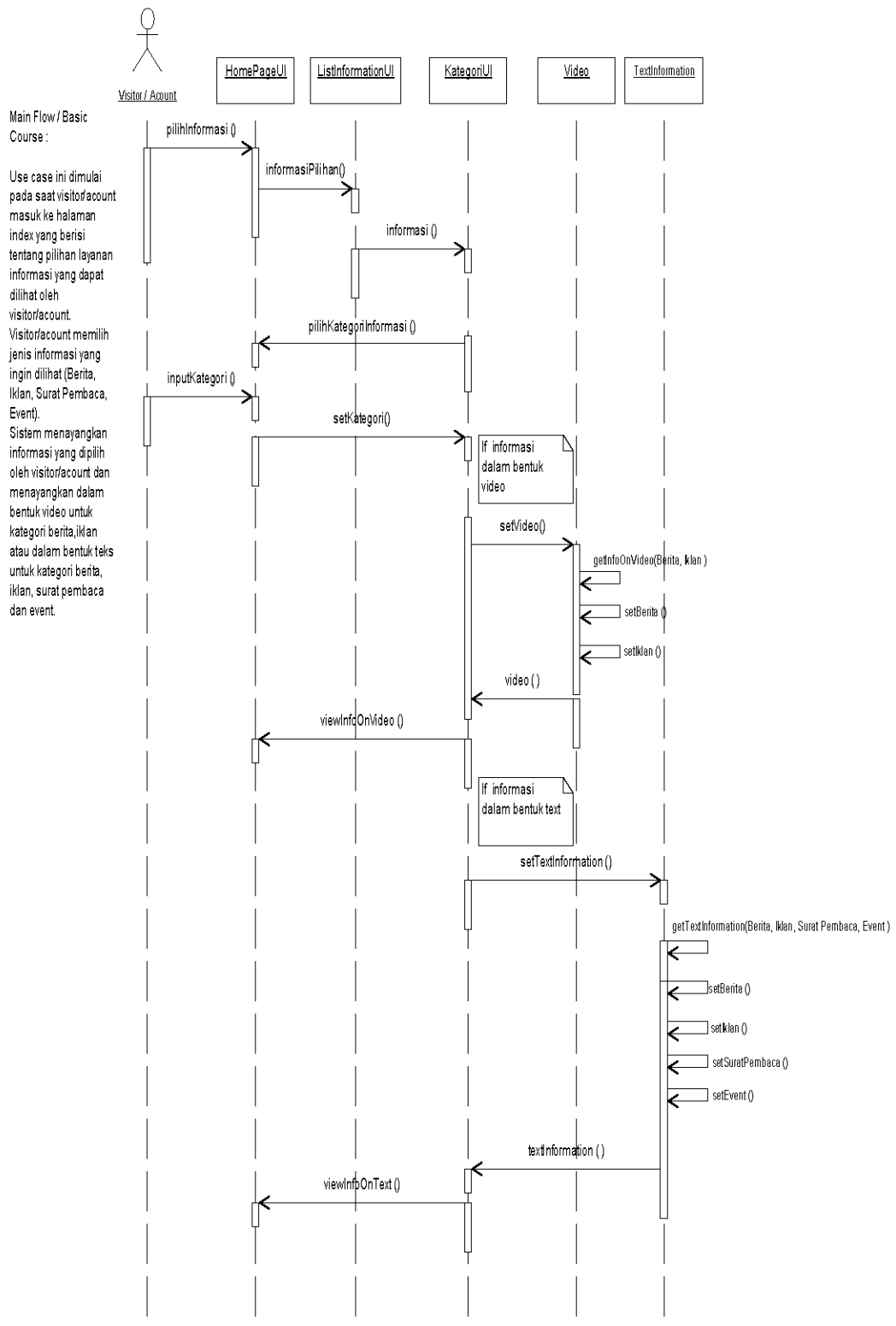
Gambar 3.3 Class diagram dengan attribut dan behaviour untuk sistem yang akan dibangun.

3.5.3 Interaction Diagram

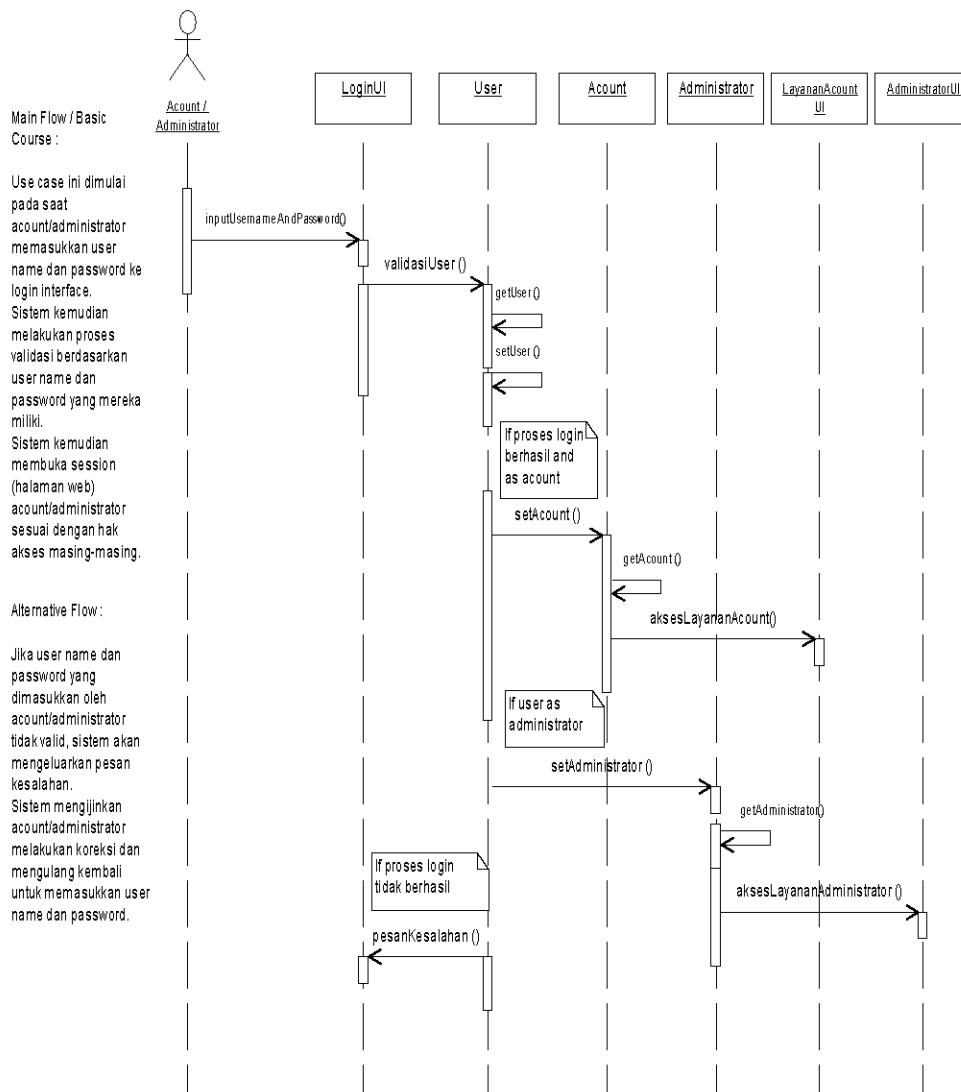
Untuk melihat interaksi antara obyek-obyek yang dibutuhkan untuk menjalankan sebuah *use case* diperlukan sebuah pemodelan interaksi. Pemodelan interaksi antar obyek-obyek dilakukan dengan *sequence diagram*. Terdapat 2 jenis *interaction diagram* yaitu *collaboration diagram* dan *sequence diagram*. Untuk lebih memahami hubungan antara obyek maka penulis memutuskan menggunakan jenis *sequence diagram*. Berikut adalah *sequence diagram* yang dibutuhkan:



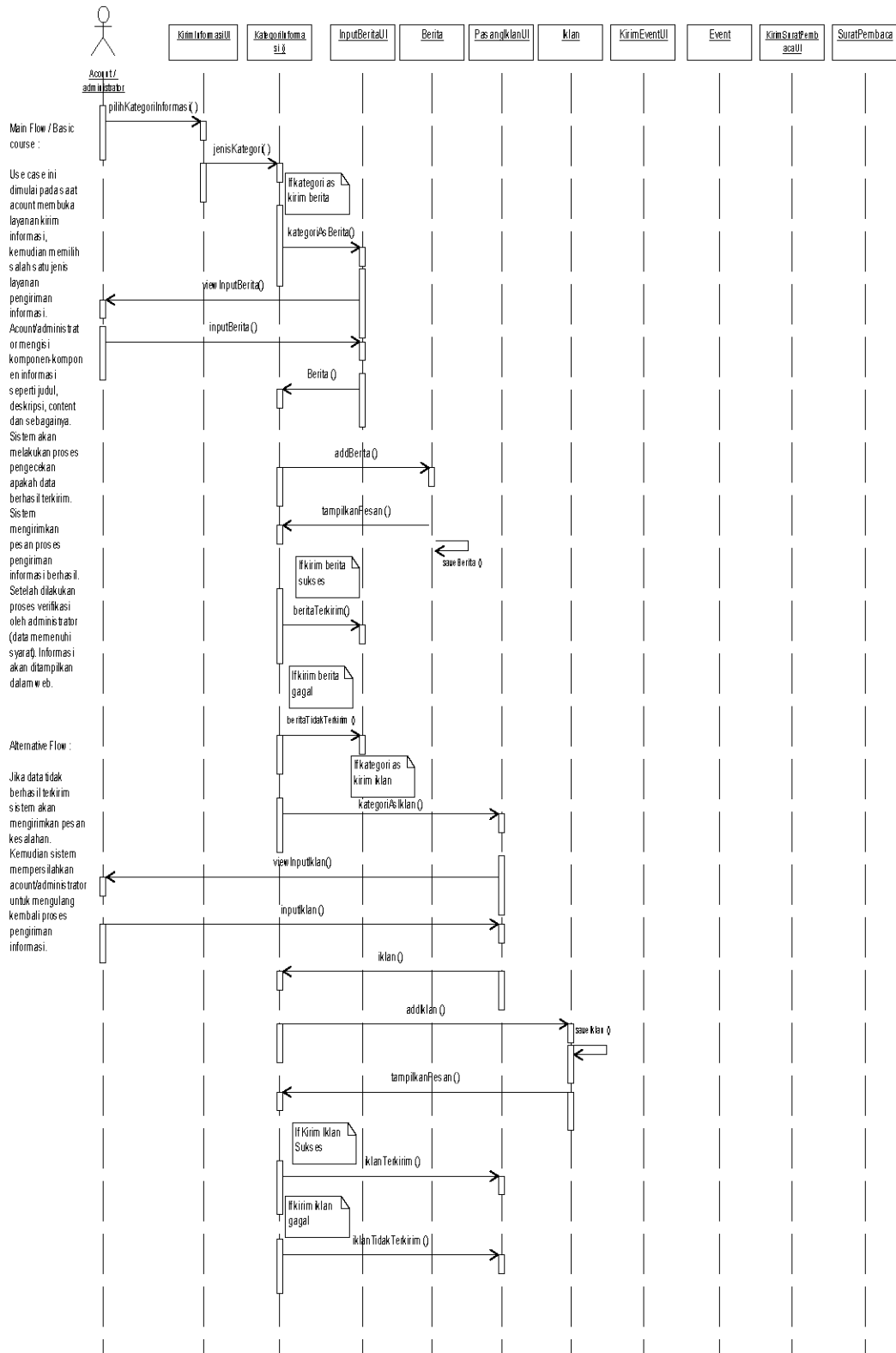
Gambar 3.4 *Sequence diagram* untuk *use case create account*.



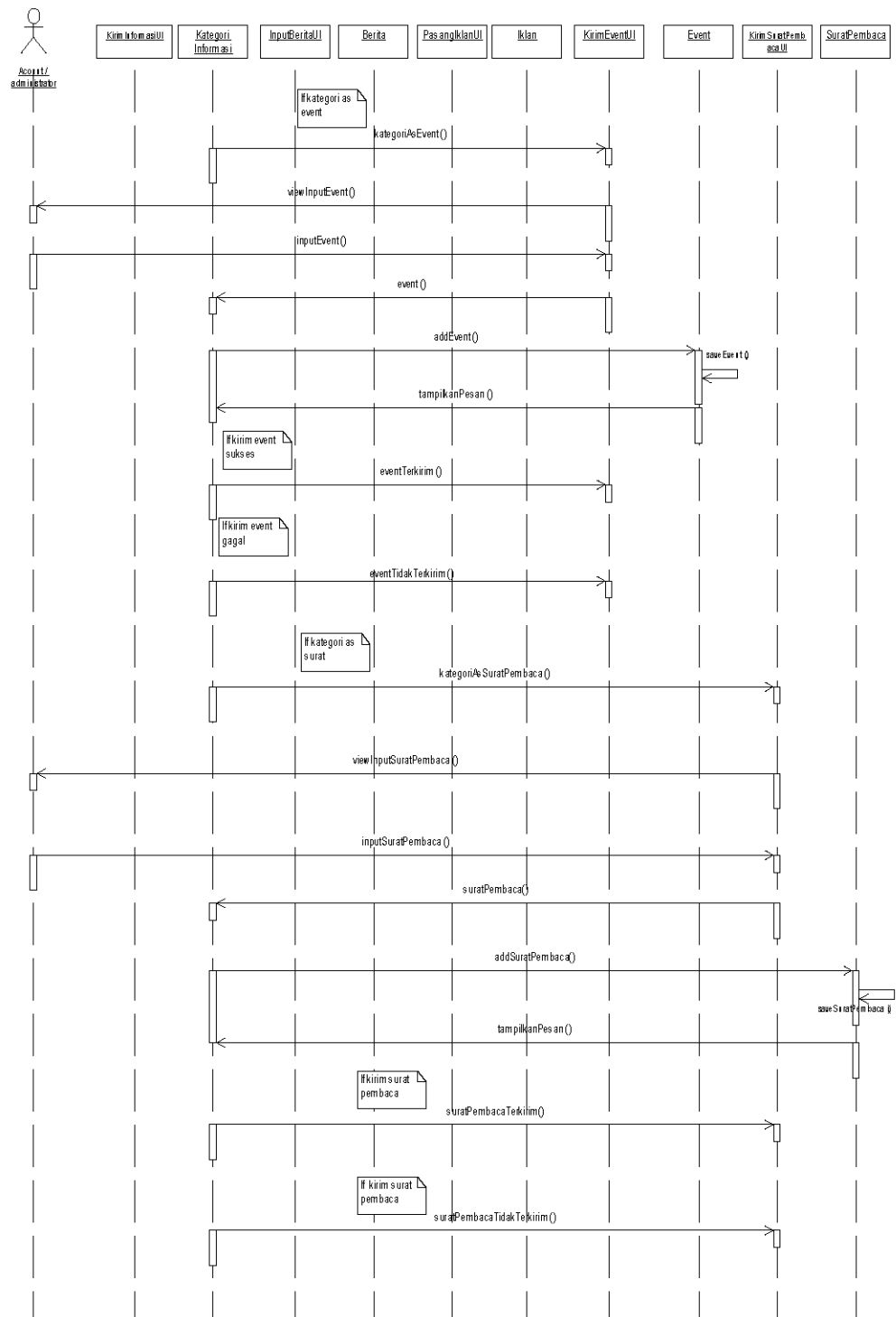
Gambar 3.5 Sequence diagram untuk use case lihat informasi.



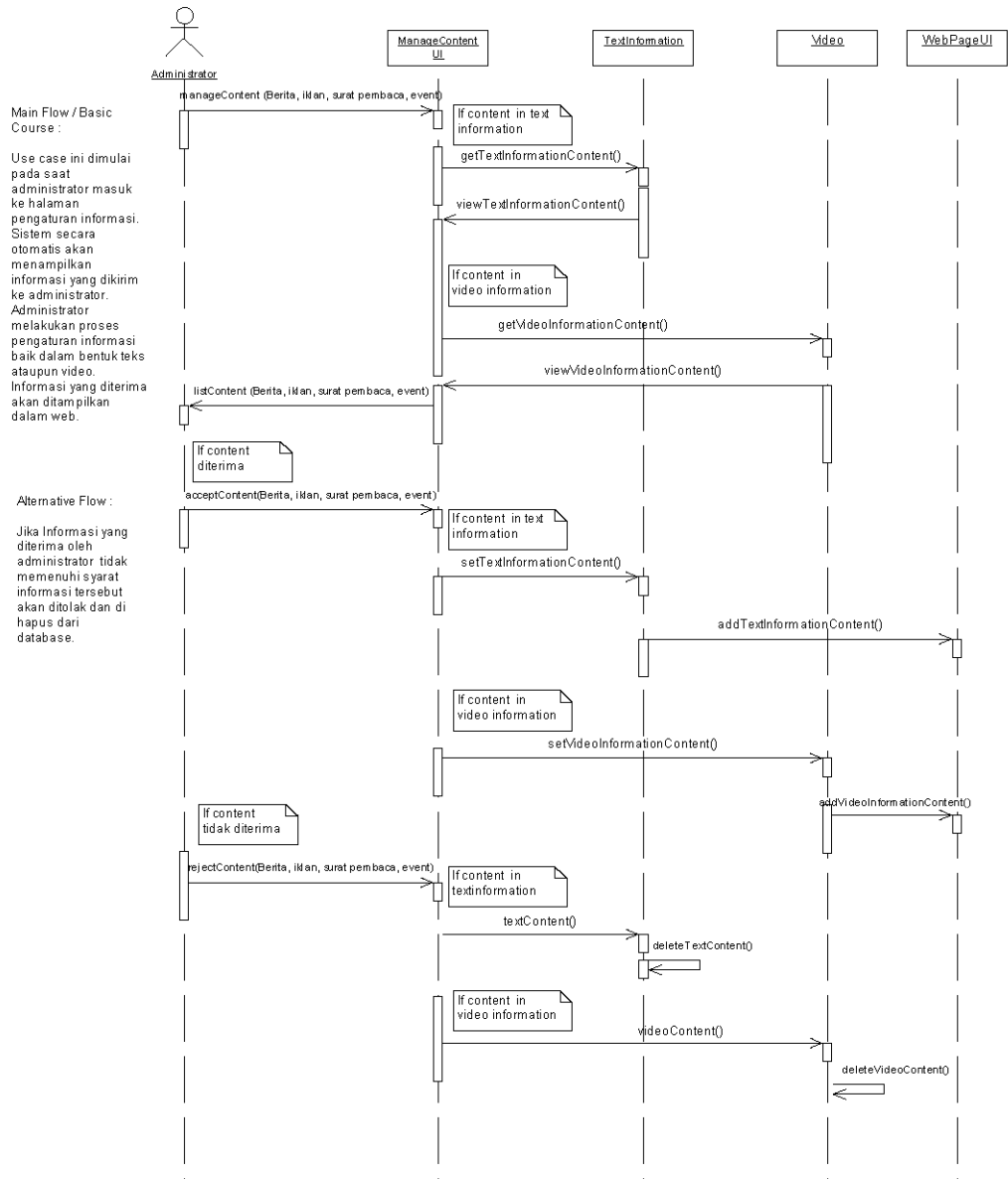
Gambar 3.6 Sequence diagram untuk use case login.



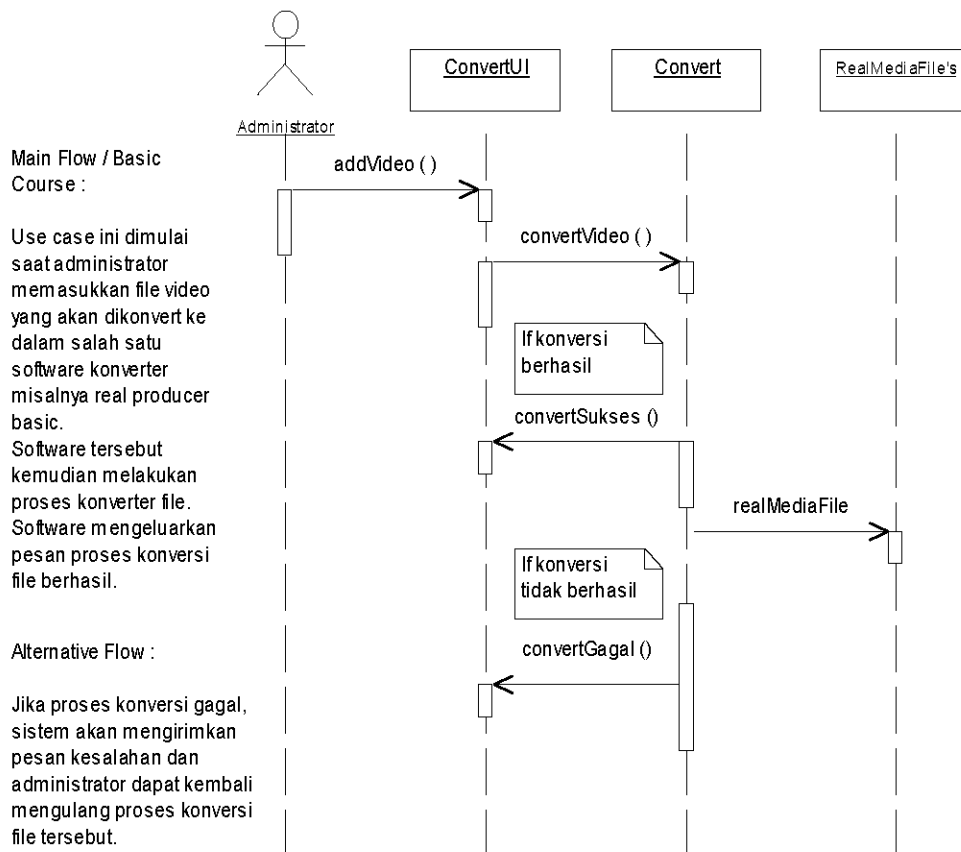
*Gambar lanjutan.



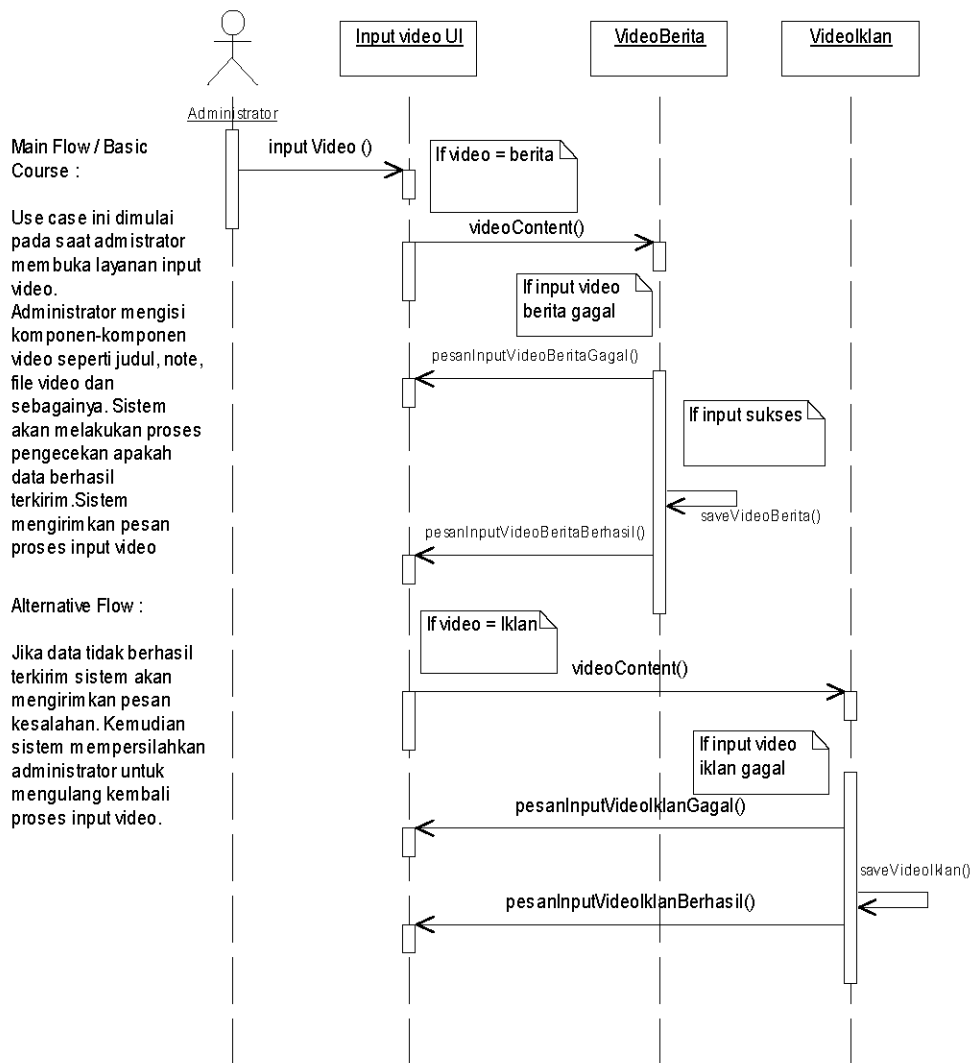
Gambar 3.7 Sequence diagram untuk usecase mengirim informasi.



Gambar 3.8 Sequence diagram untuk usecase manage content.



Gambar 3.9 Sequence diagram untuk usecase convert video.



Gambar 3.10 Sequence diagram untuk usecase input video.

BAB IV

PERANCANGAN

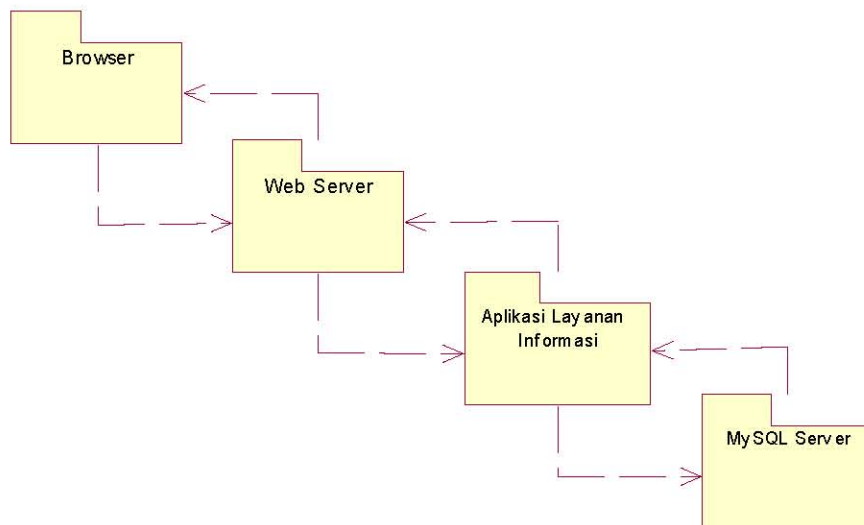
4.1 Deskripsi Sistem

Sistem yang akan dibangun merupakan sebuah aplikasi penyedia layanan informasi berbasis *web* yang memungkinkan setiap pengunjung dapat menikmati informasi tidak dalam bentuk tulisan dan gambar saja melainkan juga dalam bentuk video. Untuk proses pengiriman data video sendiri penulis lakukan dengan teknologi *streaming* sehingga informasi video dapat dengan cepat tersampaikan kepada *user*.

Dalam aplikasi ini juga *user* diajak berinteraktif dengan ikut serta mengirimkan informasi yang mereka miliki kedalam aplikasi namun sebelumnya *user* harus mendaftar terlebih dahulu sebagai *account*. Setelah *user* terdaftar sebagai *account* mereka dapat menikmati seluruh layanan dalam aplikasi ini, selain mendapatkan informasi yang dibutuhkan mereka juga dapat mengirimkan berita, mempromosikan usaha/produk atau jasa, mengirimkan informasi acara(*event*) dan mengirimkan keluhan terhadap suatu sistem dalam surat pembaca.

4.2 Arsitektur Sistem

Sistem akan dirancang dengan berbasis *web* dimana *program* akan disimpan dan dijalankan pada *server* sedangkan pada sisi *client* hanya tampilan. Semua data akan disimpan pada *database*. *Database* yang digunakan adalah *database MySQL*, dan untuk pemrograman ditulis dengan bahasa pemrograman PHP. Berikut adalah diagram untuk arsitektur sistem yang digunakan.



Gambar 4.1 Arsitektur Sistem

4.3 Perancangan *Database* Berdasarkan *Class Diagram*

Setelah memiliki *class* diagram maka kita akan membuat tabel-tabel dalam *database* berdasarkan *class-class* yang ada pada *class* diagram. Tabel-tabel ini digunakan untuk menyimpan data-data yang digunakan dalam *website*. Berikut ini adalah tabel-tabel tersebut:

Tabel 4.1 Tabel *account*

<i>Field</i>	<i>Type</i>	<i>Constraint</i>	<i>Comments</i>
<i>user_name</i>	<i>varchar(32)</i>	<i>Primary key</i>	Nama <i>user</i> untuk <i>login</i> ke aplikasi
<i>password</i>	<i>varchar(4)</i>	<i>Not null</i>	<i>Password</i> untuk <i>login</i> <i>account</i>
<i>email</i>	<i>varchar(50)</i>	<i>Unique</i>	Alamat <i>email</i> <i>account</i>
<i>nama</i>	<i>varchar(30)</i>	<i>Not null</i>	Nama <i>account</i>
<i>alamat</i>	<i>varchar(50)</i>	<i>Not null</i>	Alamat <i>account</i>
<i>kota</i>	<i>varchar(20)</i>	<i>Not null</i>	Nama kota tempat tinggal <i>account</i>
<i>telepon</i>	<i>varchar(20)</i>	<i>Not null</i>	Nomor telepon <i>account</i>
<i>Gender</i>	<i>char(1)</i>	<i>Not null</i>	<i>Gender</i> <i>account</i>
<i>pekerjaan</i>	<i>Text</i>	<i>Not null</i>	Pekerjaan <i>account</i>

Tabel 4.2 Tabel *administrator*

<i>Field</i>	<i>Type</i>	<i>Constraint</i>	<i>Comments</i>
<u><i>user_name</i></u>	<i>varchar(32)</i>	<i>Primary key</i>	Nama administrator untuk <i>login</i> ke aplikasi
<i>password</i>	<i>varchar(4)</i>	<i>Not null</i>	<i>Password</i> administrator
<i>nama</i>	<i>varchar(30)</i>	<i>Not null</i>	Nama Administrator
<i>alamat</i>	<i>varchar(50)</i>	<i>Not null</i>	Alamat administator
<i>kota</i>	<i>varchar(20)</i>	<i>Not null</i>	Nama kota administrator
<i>telp</i>	<i>varchar(20)</i>	<i>Not null</i>	No.Telp Administrator
<i>email</i>	<i>varchar(50)</i>	<i>Unique</i>	Alamat email administrator

Tabel 4.3 Tabel berita.

<i>Field</i>	<i>Type</i>	<i>Constraint</i>	<i>Comments</i>
<u><i>id_berita</i></u>	<i>int(11)</i>	<i>Primary key</i>	Nomor <i>id_berita</i>
<i>subject</i>	<i>varchar(100)</i>	<i>Not null</i>	Judul berita
<i>pengirim</i>	<i>varchar(30)</i>	<i>Not null</i>	Pengirim berita
<i>tanggal</i>	<i>Datetime</i>	<i>Not null</i>	Tanggal dikirim berita
<i>isi</i>	<i>Longtext</i>	<i>Not null</i>	Isi berita
<i>status</i>	<i>enum('0', '1', '2')</i>	<i>Not null</i>	Status berita, jika berita terkirim maka status=0, jika diterima status=1, jika di <i>delete</i> atau ditolak status=2.
<i>category</i>	<i>varchar(20)</i>	<i>Not null</i>	Kategori berita yang akan dimasukkan
<i>Description</i>	<i>varchar(20)</i>	<i>Not null</i>	Deskripsi dari berita yang akan dimasukkan
<i>gambar</i>	<i>varchar(50)</i>	<i>Not null</i>	Nama <i>file</i> gambar yang dimasukkan

Tabel 4.4 Tabel *event*.

<i>Field</i>	<i>Type</i>	<i>Constraint</i>	<i>Comments</i>
<u><i>id_event</i></u>	<i>int(11)</i>	<i>Primary key</i>	Nomor <i>id event</i>
<i>date</i>	<i>Datetime</i>	<i>Not null</i>	Tanggal dimasukkan <i>event</i>
<i>hari_tgl</i>	<i>varchar(25)</i>	<i>Not null</i>	Hari dan tanggal <i>event</i>
<i>jam_event</i>	<i>varchar(25)</i>	<i>Not null</i>	Waktu <i>event</i> di selenggarakan.
<i>tempat</i>	<i>varchar(25)</i>	<i>Not null</i>	Tempat <i>event</i> diselenggarakan.
<i>judul_event</i>	<i>varchar(30)</i>	<i>Not null</i>	Judul <i>event</i>
<i>note</i>	<i>Tinytext</i>	<i>Not null</i>	Tempat <i>event</i> diselenggarakan
<i>status</i>	<i>enum('0', '1', '2')</i>	<i>Not null</i>	Status <i>event</i> , jika <i>event</i> terkirim maka status=0, jika diterima status=1, jika di <i>delete</i> atau ditolak status=2.
<i>pengirim</i>	<i>varchar(25)</i>	<i>Not null</i>	Pengirim <i>event</i> .

Tabel 4.5 Tabel iklan.

<i>Field</i>	<i>Type</i>	<i>Null</i>	<i>Comments</i>
<u><i>id_iklan</i></u>	<i>int(11)</i>	<i>Primary key</i>	Nomor <i>id iklan</i>
<i>subject</i>	<i>varchar(50)</i>	<i>Not null</i>	Judul iklan
<i>pengirim</i>	<i>varchar(30)</i>	<i>Not null</i>	Nama pengirim iklan
<i>tanggal</i>	<i>Datetime</i>	<i>Not null</i>	Tanggal dikirim iklan
<i>isi</i>	<i>Tinytext</i>	<i>Not null</i>	Isi iklan
<i>status</i>	<i>enum('0', '1', '2')</i>	<i>Not null</i>	Status iklan, jika berita terkirim maka status=0, jika diterima status=1, jika di <i>delete</i> atau ditolak status=2.
<i>category</i>	<i>varchar(15)</i>	<i>Not null</i>	Kategori iklan yang akan dimasukkan.

Tabel 4.6 Tabel surat pembaca.

<i>Field</i>	<i>Type</i>	<i>Constraint</i>	<i>Comments</i>
<u>id_surat</u>	int(11)	Primary key	Nomor <i>id</i> surat pembaca
tanggal	datetime	Not null	Tanggal dikirim surat
pengirim	varchar(30)	Not null	Nama pengirim surat
judul	varchar(75)	Not null	Judul surat
kepada	varchar(50)	Not null	Surat ditujukan untuk
isi	text	Not null	Isi surat
status	enum('0', '1', '2')	Not null	Status surat, jika berita terkirim maka status=0, jika diterima status=1, jika di <i>delete</i> atau ditolak status=2.

Tabel 4.7 Tabel *video* informasi.

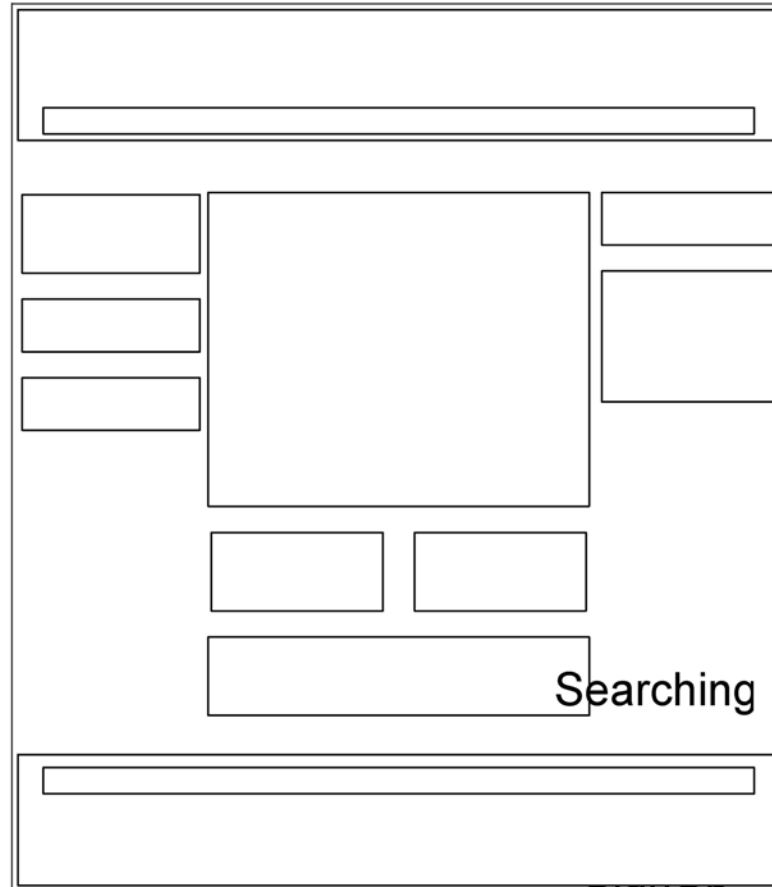
<i>Field</i>	<i>Type</i>	<i>Null</i>	<i>Comments</i>
<u>id_video</u>	int(11)	Primary key	Nomor <i>id video</i> .
judul	varchar(30)	Not null	Judul <i>video</i> .
tanggal	datetime	Not null	Tanggal dimasukkan <i>video</i> .
note	varchar(100)	Not null	Komentar <i>video</i> .
Gambar	varchar(50)	Not null	Nama <i>file video</i> .
direktori	varchar(50)	Not null	Nama <i>file video</i> .
status	enum('0', '1', '2')	Not null	Status <i>video</i> , jika <i>video</i> terkirim ke <i>web</i> maka status=0, jika diterima status=1, jika di <i>delete</i> atau ditolak status=2.
category	varchar(15)	Not null	Kategori <i>video</i> .

Tabel 4.8 Tabel *video* iklan.

<i>Field</i>	<i>Type</i>	<i>Null</i>	<i>Comments</i>
<u><i>id_iklanvideo</i></u>	<i>int(11)</i>	<i>Primary key</i>	Nomor <i>id video</i>
<i>Subject</i>	<i>varchar(30)</i>	<i>Not null</i>	Judul <i>video</i>
Pengirim	<i>varchar(30)</i>	<i>Not null</i>	Pengirim iklan.
Tanggal	<i>datetime</i>	<i>Not null</i>	Tanggal dimasukkan <i>video</i> .
isi	<i>tinytext</i>	<i>Not null</i>	Komentar <i>video</i> .
<i>video</i>	<i>varchar(50)</i>	<i>Not null</i>	Nama file <i>video</i> .
Gambar	<i>varchar(50)</i>	<i>Not null</i>	Nama file <i>video</i> .
status	<i>enum('0', '1', '2')</i>	<i>Not null</i>	Status <i>video</i> , jika <i>video</i> iklan terkirim ke <i>web</i> maka status=0, jika diterima status=1, jika di <i>delete</i> atau ditolak status=2.
<i>category</i>	<i>varchar(15)</i>	<i>Not null</i>	Kategori <i>video</i> .

4.4 Perancangan Antarmuka

Berikut merupakan rancangan antarmuka untuk tiap halaman *web* yang akan dibuat. Antar muka menggunakan sistem *template* dimana tampilan keseluruhan hampir memiliki struktur yang sama. Tiap halaman hanya dibedakan bagian isi (*content*)-nya saja. Pada halaman bagian atas (*header*) dan bawah (*footer*) terdapat bagian yang akan selalu muncul pada tiap halaman. Hal ini untuk lebih memudahkan navigasi *website*. Berikut adalah rancangan antar muka yang dibuat.



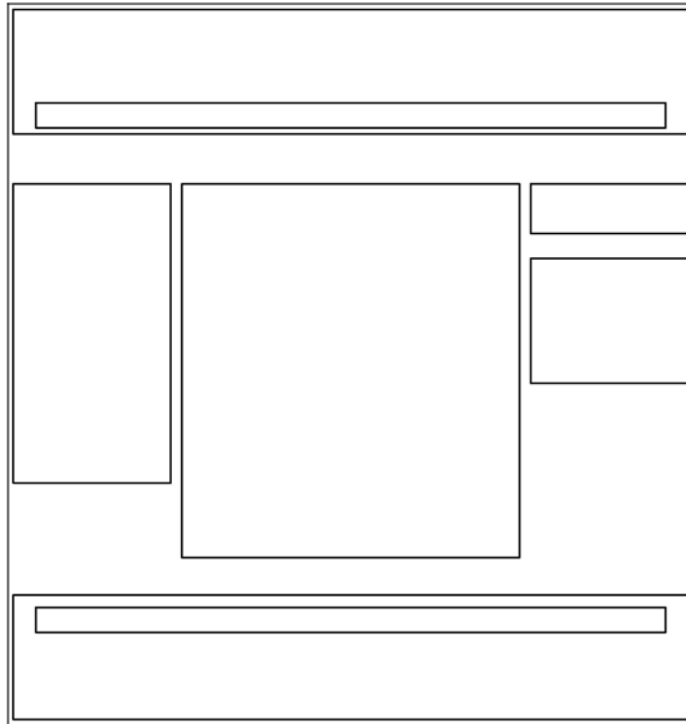
Gambar 4.2 Rancangan Halaman Awal

Pada rancangan diatas dapat dilihat struktur tampilan awal pada *web*, bagian yang akan berubah hanyalah pada bagian *preview* dan menu saja (sebelah kiri rancangan) sesuai dengan informasi yang akan dilihat. Jika memilih menu pada navigasi, *category* informasi akan aktif dan berubah secara dinamis sesuai dengan informasi yang ingin dilihat. Berikut ini adalah gambaran rancangan *web* secara keseluruhan.

Kolom Iklan

Kolom 5

n

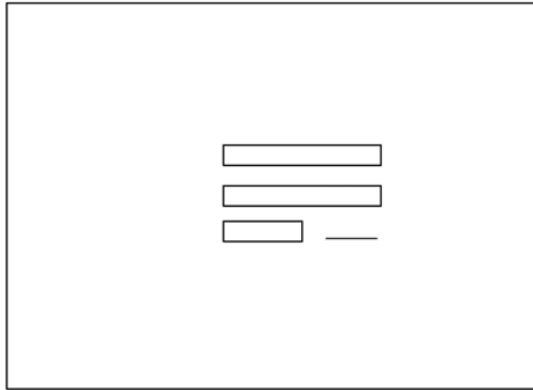


Gambar 4.3 Rancangan menu informasi selengkapnya.

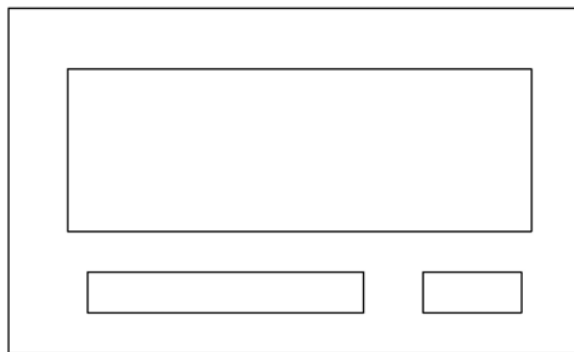
Gambar diatas adalah *template* untuk melihat isi **Menu Informasi** secara detail, dan menu informasi akan berubah sesuai dengan kategori informasi yang dilihat.



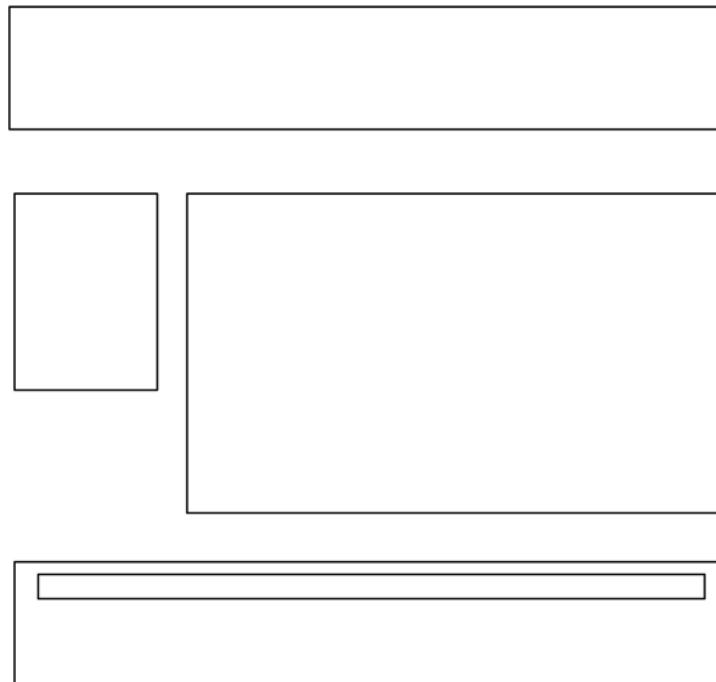
Gambar 4.4 Rancangan *signup*.



Gambar 4.5 Rancangan *login*.



Gambar 4.6 Rancangan *searching*.



Gambar 4.7 Rancangan halaman layanan *account*.

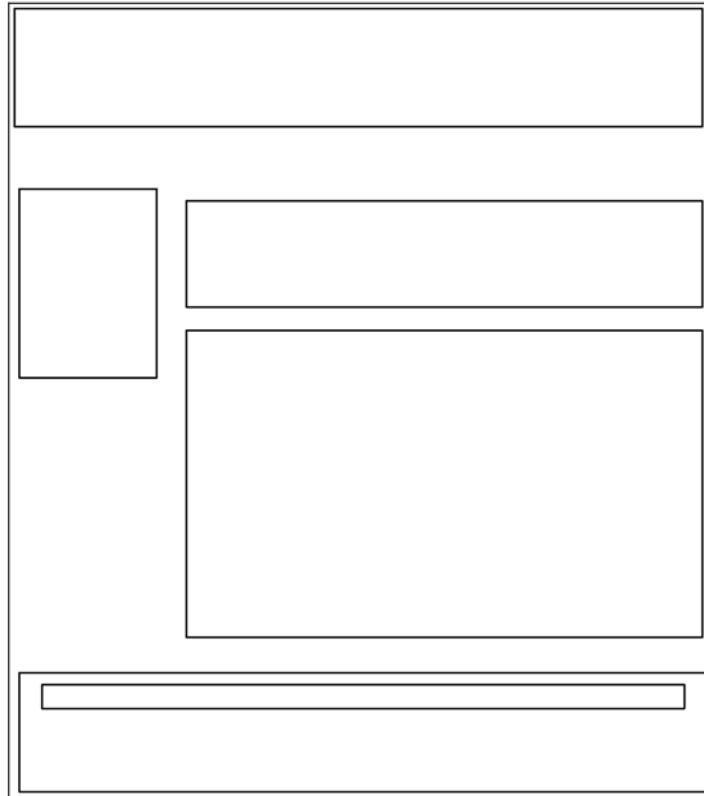
User name

Password

Search

Keyword

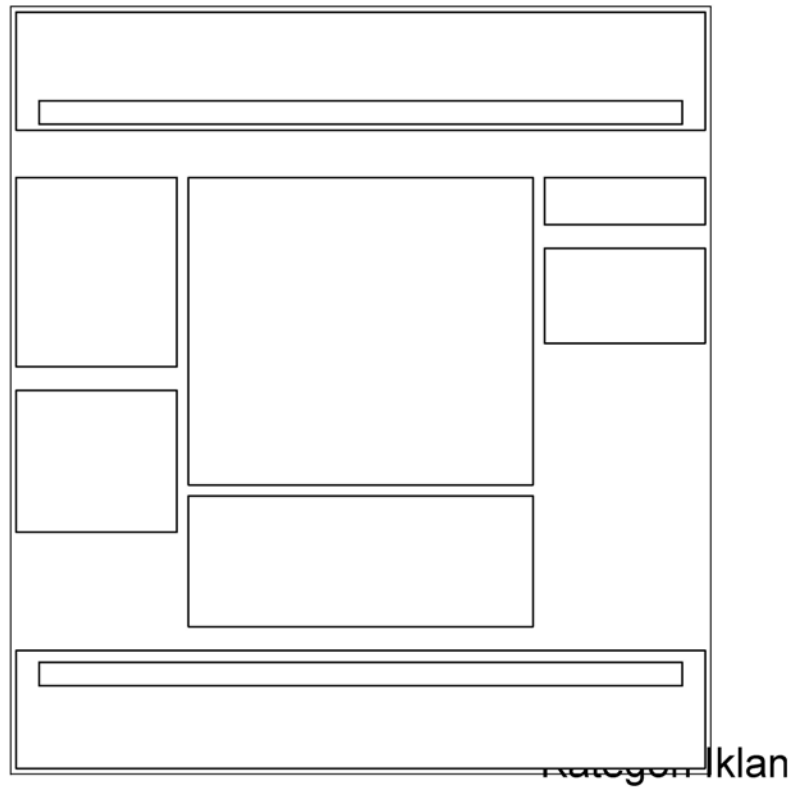
Pada rancangan ini juga bersifat *template*, karena tampilan yang berubah hanya pada bagian isi layanan saja sesuai dengan menu layanan yang dipilih oleh *account*.



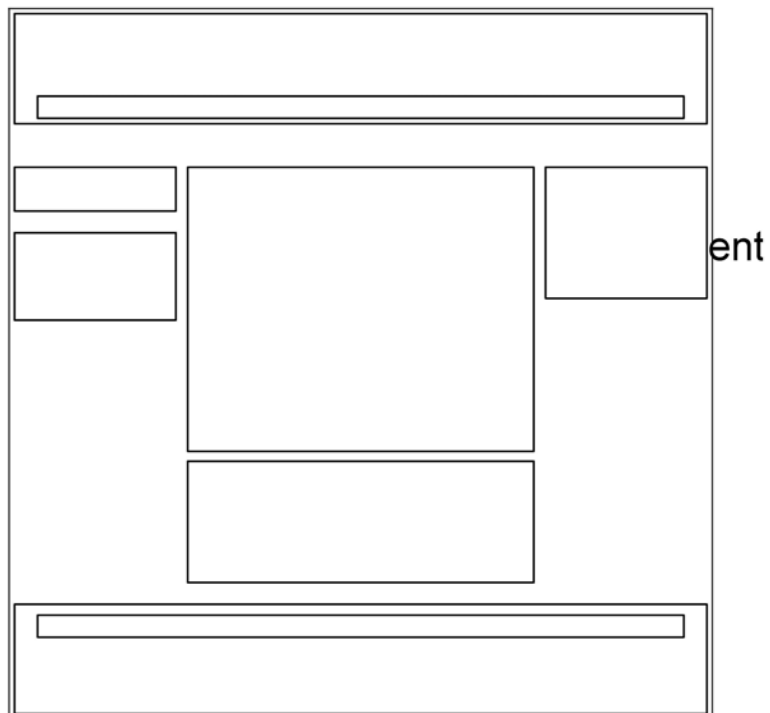
Gambar 4.8 Rancangan halaman administrator.

Seperti layanan *account*, untuk halaman administrator bagian yang akan berubah hanya pengaturan informasi dan *input* layanan saja sesuai dengan menu pengaturan yang dipilih.

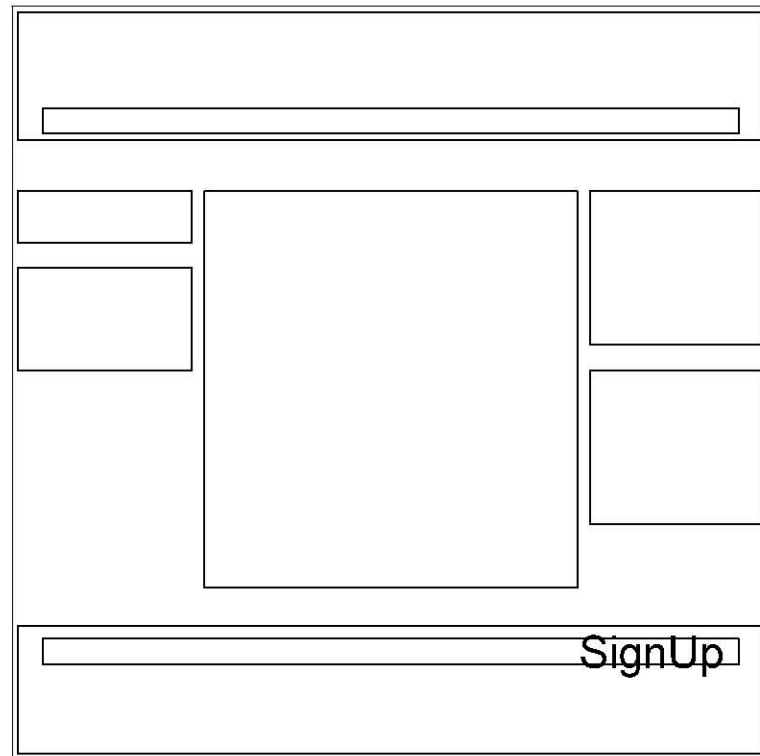
Menu
Pengaturan



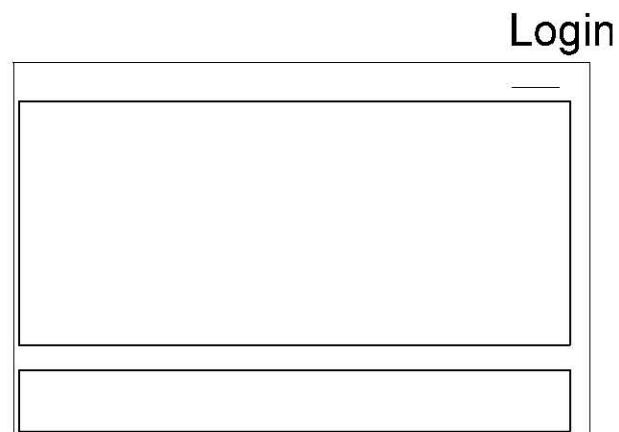
Gambar 4.9 Rancangan halaman *view iklan*.



Gambar 4.10 Rancangan halaman *view event*.



Gambar 4.11 Rancangan halaman *view* surat pembaca.



Gambar 4.12 Rancangan halaman penayangan video.

Perancangan aplikasi menggunakan sistem template. Sehingga, bagian yang berubah hanya pada *content*(isi) saja. Berikut ini adalah perancangan-perancangan *content* dalam aplikasi :

A wireframe diagram of a news submission form. It features a large central rectangular area for the main content. Above this area are two horizontal input fields. Below the main content area are two pairs of smaller horizontal input fields, one pair positioned above the other. The form is enclosed in a rectangular border.

Gambar 4.13 Rancangan *content form* kirim berita.

Category
Description

Judul

Pengirim Berita

A wireframe diagram of an advertisement placement form. It features a large central rectangular area for the main content. Above this area is a single horizontal input field. Below the main content area are two pairs of smaller horizontal input fields, one pair positioned above the other. The form is enclosed in a rectangular border.

Gambar 4.14 Rancangan *content form* pasang iklan.

nama gambar

The wireframe shows a form layout within a rectangular border. At the top center, there is a vertical stack of five rectangular input fields. Below this stack is a large, wide rectangular text area. To the right of the text area, the label "Pengirim" is positioned. Below the text area, there are two small rectangular input fields side-by-side. To the right of these fields, the labels "Hari dan tanggal event" and "Waktu event (jam)" are positioned. At the bottom right of the form, the label "Tempat" is positioned.

Gambar 4.15 Rancangan *content form* pasang event

Judul

The wireframe shows a form layout within a rectangular border. At the top center, there is a vertical stack of three rectangular input fields. Below this stack is a large, wide rectangular text area. At the bottom center of the form, there are two small rectangular input fields side-by-side.

Gambar 4.16 Rancangan *content form* kirim surat pembaca.

A wireframe diagram of a content form for video information. It features a large central rectangular area for video content. Above this area are three input fields: a long one at the top, and two smaller ones below it, one to the left of a small square. Below the video area are four input fields arranged in two rows of two, and two more input fields at the bottom.

egory

Judul

Masukkan gambar

Gambar 4.17 Rancangan *content form input video* informasi.

A wireframe diagram of a content form for video advertisement. It features a large central rectangular area for video content. Above this area are three input fields: a long one at the top, and two smaller ones below it, one to the left of a small square. Below the video area are four input fields arranged in two rows of two, and two more input fields at the bottom.

video

Gambar 4.18 Rancangan *content form input video* iklan.