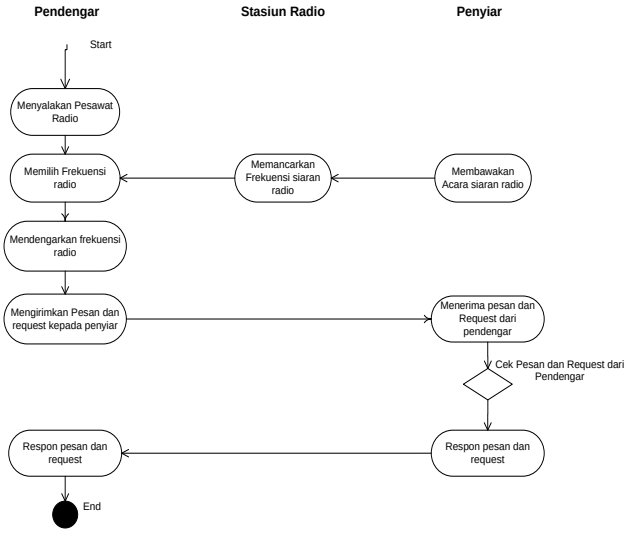
BAB III

PEMBAHASAN

1.1 Analisis Sistem Yang Sedang Berjalan

Analisis sistem yang sedang berjalan bertujuan mempelajari dan mengevaluasi sistem yang berjalan dan mengetahui seperti apa sistem yang sedang berjalan saat ini. Pemahaman terhadap sistem dilakukan dengan mempelajari bagaimana sistem tersebut berjalan. Di Radio Sonata Bandung, sistem yang berjalan hampir sama dengan radio pada umum nya, dimana Stasiun radio Sonata memancarkan siaran radionya dengan menggunakan pemancar dan pendengar mendengarkan radio dengan menggunakan pesawat radio atau alat elektronik yang mempunyai pesawat radio. Namun, pendengar merasa kesulitan untuk mendapatkan sinyal pancar radio manakala berada di daerah yang di luar jarak jangkau pancar radio sonata. Pendengar dapat menlakukan request atau mengirimkan pesan kepada penyiar melalui line telepon dan line sms. Untuk pesan, bisa juga disampaikan melalui line telepon offline yang telah disediakan. Namun kadang pendengar kesulitan berkunikasi dengan penyiar atau *crew* manakala penyiar/ *crew* tidak ada di tempat.



Gambar 1. Activity Diagram

1.2 Fungsi Produk

Secara umum fungsi dari website ini adalah melayani proses pendaftaran *member*, pengelolaan berita, pengelolaan acara, komunikasi pesan dengan penyiar dan radio *Streaming*. Proses pengelolaan berita dan acara dilakukan oleh *admin*istrartor. Dimana *Administrator* dapat melakukan penambahan, penghapusan dan pengeditan berita dan acara. Proses komunikasi pesan hanya bisa dilakukan oleh *member* yang telah melakukan registrasi atau pendaftaran menjadi *member*. Jika pengungjung ingin menlakukan pengiriman pesan kepada penyiar, maka pengunjung harus daftar terlebih dahulu menjadi *member*. Proses radio *Streaming* tidak bergantung pada keanggotaan sehingga setiap pengunjung yang mengakses situs dapat mengakses dan mendengarkan radio *Streaming* tanpa harus mendaftar menjadi anggota.

Fungsi utama dari *website* ini adalah :

a. Pengunjung
    1. Pendaftaran *Member*
    2. Radio *Streaming*

b. Autentifikasi
   1. Autentifikasi *Administrator*
   2. Autentifikasi *Member*
c. *Member*
   1. Komunikasi pesan
   2. Radio *Streaming*
d. *Administrator*
   1. Pengelolaan Berita
      1.1 Memasukan berita baru
      1.2 Menghapus berita
      1.3 Mengedit berita
   2. Pengelolaan Acara
      2.1 Memasukan acara baru
      2.2 Menghapus acara
      2.3 Mengedit acara
   3. Pengelolaan *Crew*
      3.1 Memasukan *crew* baru
      3.2 Menghapus *crew*
      3.3 Mengedit *crew*

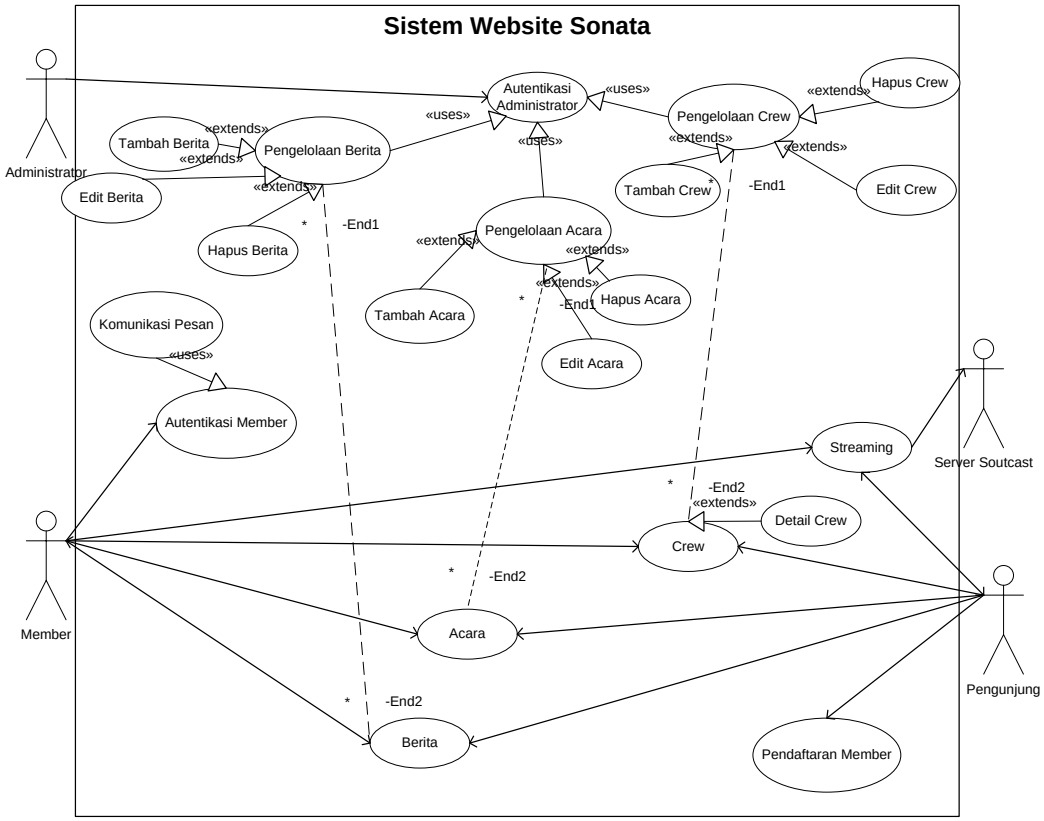
1.3 Karateristik Pengguna

Website digunakan oleh berbaga imacam manusia di seluruh dunia. Namun website ini hanya *member*ikan in*form*asi seputar radio sonata, sehingga pengguna dari *website* ini adalah *member* dari *website* dan pengunjung yang ingin mendengarkan radio *streming* sonata.

Tabel 1. Karakteristik Pengguna

Pengguna	Tanggung Jawab	Hak Akses	Tingkat Pendidikan	Tingkat Keterampilan	Pengalaman	Jenis Pelatihan
Pengunjung	Mendapatkan In*form*asi mengenai Radio sonata	Melakukan pendaftaran menjadi *member*, Mengakses radio *Streaming*	Siswa Sekolah Dasar	Bisa selalu belajar sesuatu hal yang baru	-	
Member	MendapatkanIn*form*asi mengenai Radio Sonata	Berinteraksidengan penyiar melalui pesan , mengakses radio *Streaming*	Minimal lulus sekolah dasar	Mengerti atau mengenal pengoprasian komputer		
Operator	Melakukan Maintenance website.	Memasukan, Mengedit, Menghapus data yang adapada website	Lulus SMA	Mengikuti Teknis Maintenance Website	Pelatihan Operator	Cara Memaintenance Website

1.4 Analis Kebutuhan Fungsional



Gambar 2. Usecase diagram

1.4.1 Pendaftaran Member

Fungsi ini melakukan proses pendaftaran untuk pengunjung/calon *member* yang ingin menjadi *member*. Hal ini harus dilakukan jika pengunjung ingin mengirimkan pesan kepada penyiar.

3.4.1.1 **Skenario**

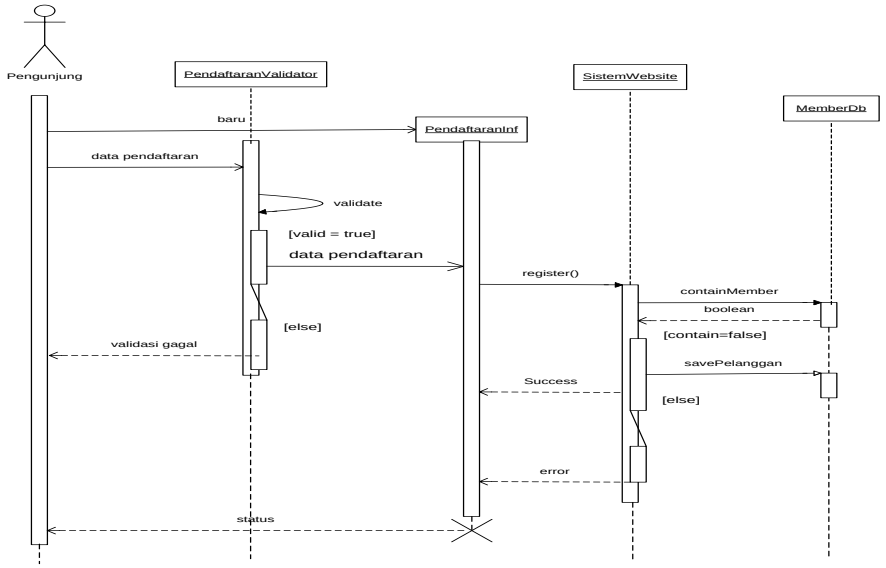
Tabel 2. Skenario Pendaftaran Member

Identifikasi	
Nomor	1
Nama	Pendaftaran *Member*
Tujuan	Melakukan pendaftaran *Member*
Deskripsi	Sistem menerima data dari pengunjung/calon *member* dan sistem akan menyimpan data pengunjung sebagai *member*

Tipe		
Aktor	Pengunjung	
Skenario Utama		
Kondisi Awal	Pengunjung masuk ke halaman pendaftaran	
Aksi Aktor		**Reaksi Sistem**
Pengunjung memasukkan data identitas pengunjung		Mengirim data pengunjung ke sistem untuk di simpan di *database*
		Jika proses berhasil maka akan ditampilkan halaman konfirmasi
Skenario Alternatif – Validasi Gagal		
Aksi Aktor		**Reaksi Sistem**
Pengunjung memasukkan lagi data identitas pengunjung		
Kondisi Akhir	Halaman autentikasi *member* akan ditampilkan	

3.4.1.2 **Dinamika Objek**

Berikut ini adalah dinamika objek dari proses pendaftaran member.



Gambar 3. Dinamika Objek pada Pendaftaran Member

1.4.2 Radio *Streaming*

Fungsi ini melakukan proses mendengarkan siaran radio secara *Streaming* melalui jaringan internet.

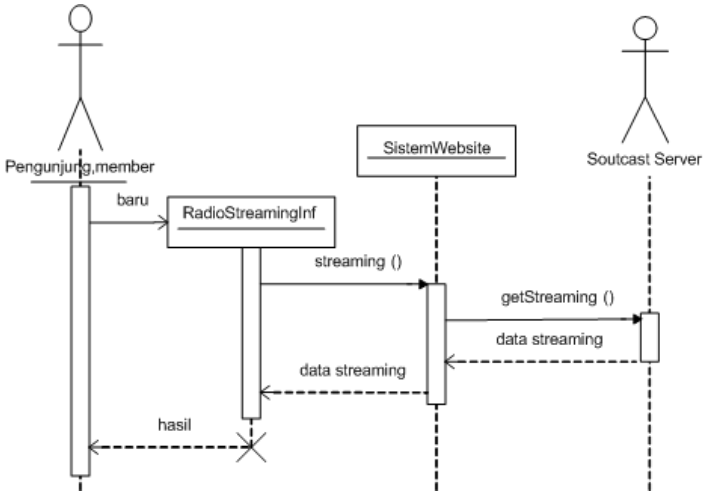
3.4.2.1 **Skenario**

Tabel 3. Skenario Radio *Streaming*

Identifikasi		
Nomor	3.4.2.2 D2	
Nama	i Radio *Streaming*	
Tujuan	nMendengarkan siaran radio secara Streming (Online)	
Deskripsi	aPengunjung dan *member* merequest radio *Streaming* kepada sistem, dan sistem akan *member*ikan siaran radio secara ms*treaming*.	
Tipe	i	
Aktor	kPengunjung, *Member*	
Skenario Utama		
Kondisi Awal	a Pengunjung, *member* masuk ke menu radio *Streaming*	
Aksi Aktor		**Reaksi Sistem**
Pengunjung masuk ke menu radio *Streaming*		Membuka *link server* radio *Streaming*
		Jika proses berhasil maka akan ditampilkan halaman radio *Streaming*
Skenario Alternatif – Validasi Gagal		
Aksi Aktor		**Reaksi Sistem**
Pengunjungmasuk kembali ke menu radio *Streaming* atau me refresh halaman		
Kondisi Akhir	Halaman radio *Streaming* akan di tampilkan.	

3.4.2.3 **Dinamika Objek**

Berikut ini adalah dinamika objek dari proses Radio *Streaming.*



Gambar 4. Dinamika Objek pada proses Radio *Streaming*

1.4.3 Autentifikasi *Administrator*

Fungsi autentikasi *Administrator* merupakan proses pengecekan apakah pengguna seorang *Administrator* atau bukan. Proses ini harus dilalui oleh *Administrator* agar dapat masuk ke halaman *Administrator* dan dapat mengolah isi website.

3.4.3.1 **Skenario**

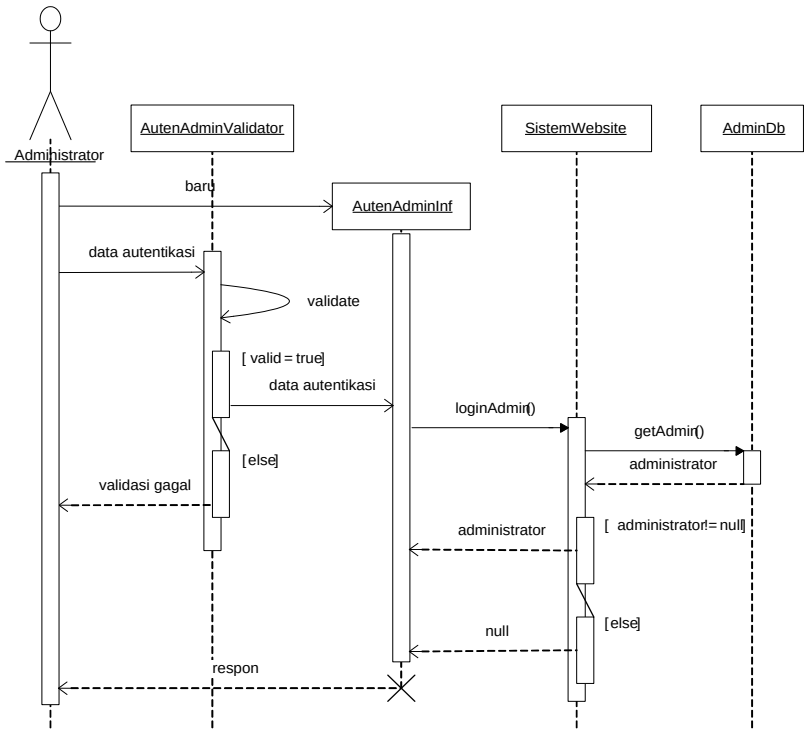
Tabel 4. Skenario Autentifikasi *Administrator*

Identifikasi		
Nomor	3	
Nama	Autentikasi *Administrator*	
Tujuan	Melakukan proses pengecekan ketersediaan *Administrator*	
Deskripsi	Sistem akan melakukan pengecekan apakah pengguna adalah *Administrator*, jika benar maka pengguna dapat mengakses halaman *Administrator*.	
Tipe		
Aktor	*Administrator*	
Skenario Utama		
Kondisi Awal	*Administrator* masuk ke halaman autentikasi	
Aksi Aktor		**Reaksi Sistem**

Administrator memasukkan data *username* dan *password*	Sistem akan mengecek apakah *Administrator* dengan *username* dan *password* tersebut tersedia
	Jika tersedia maka proses autentikasi berhasil dan sistem akan menampilkan halaman *Administrator*
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
Administrator memasukkan data *username* dan *password* lagi	Sistem melakukan validasi data *username* dan *password* lalu melakukan proses pengecekan
Skenario Alternatif – Autentikasi Gagal	
Kondisi Akhir	Sistem menampilkan halaman *Administrator*

3.4.3.2 **Dinamika Objek**

Berikut ini adalah dinamika objek dari proses autentifikasi *Administrator*.



Gambar 5. Dinamika Objek pada Autentifikasi *Administrator*

1.4.4 Lihat Acara

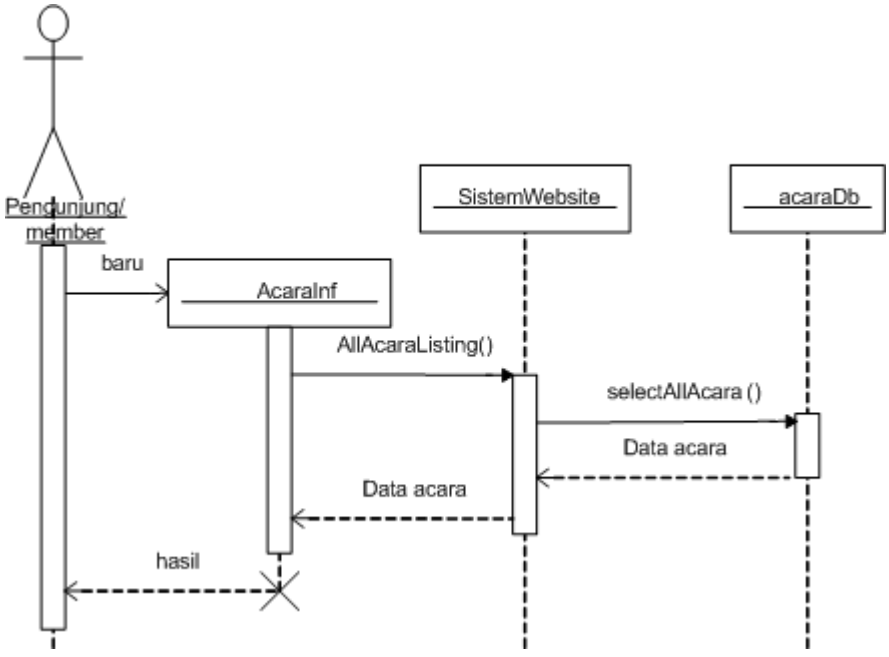
Fungsi ini merupakan perintah untuk menampilkan seluruh data acara yang terdapat pada *database*.

3.4.4.1 Skenario

Tabel 5. Skenario Lihat Acara

Identifikasi	
Nomor	4
Nama	Lihat Acara
Tujuan	Menampilkan seluruh data acara
Deskripsi	Sistem akan memanggil data acara setelah itu sistem akan menampilkan seluruh pada halaman acara
Tipe	
Aktor	Pengunjung dan member
Skenario Utama	
Kondisi Awal	Pengunjung dan member masuk ke halaman acara
Aksi Aktor	**Reaksi Sistem**
	Sistem memanggil seluruh data acara yang ada dalam *database*
	Sistem menampilkan data tersebut pada halaman acara
Kondisi Akhir	Sistem menampilkan daftar acara

3.4.4.2 Dinamika Objek



Gambar 6. Dinamika Objek Lihat Acara

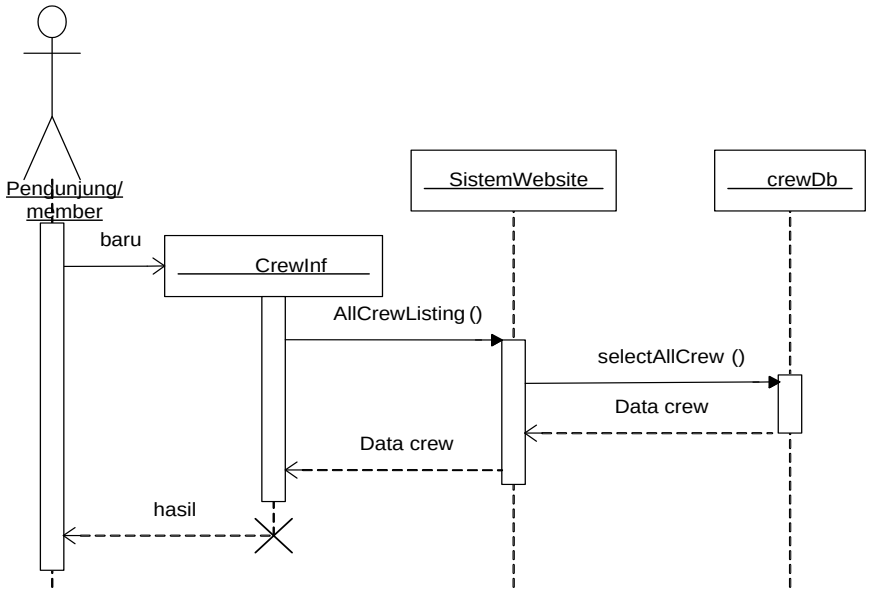
1.4.5 Lihat *Crew*

Fungsi ini merupakan perintah untuk menampilkan seluruh data *crew* yang terdapat pada *database*.

3.4.5.1 **Skenario**

Identifikasi	
Nomor	5
Nama	Lihat *Crew*
Tujuan	Menampilkan seluruh data *crew*
Deskripsi	Sistem akan memanggil data *crew* setelah itu sistem akan menampilkan seluruh pada halaman *crew*
Tipe	
Aktor	Pengunjung dan member
Skenario Utama	
Kondisi Awal	Pengunjung dan member masuk ke halaman *crew*
Aksi Aktor	**Reaksi Sistem**
	Sistem memanggil seluruh data *crew* yang ada dalam *database*
	Sistem menampilkan data tersebut pada halaman *crew*
Kondisi Akhir	Sistem menampilkan daftar *crew*

3.4.5.2 **Dinamika Objek**



Gambar 7. Dinamika Objek Lihat *Crew*

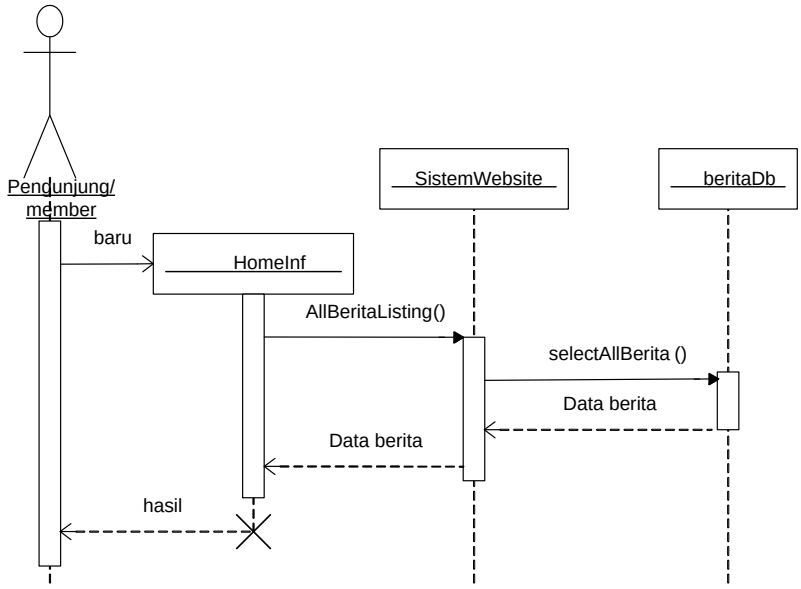
1.4.6 Lihat Berita

Fungsi ini merupakan perintah untuk menampilkan seluruh data berita yang terdapat pada *database*.

3.4.6.1 **Sekenario**

Identifikasi	
Nomor	6
Nama	Lihat berita
Tujuan	Menampilkan seluruh data berita
Deskripsi	Sistem akan memanggil data berita setelah itu sistem akan menampilkan seluruh pada halaman utama (frontpage)
Tipe	
Aktor	Pengunjung dan member
Skenario Utama	
Kondisi Awal	Pengunjung dan member masuk ke halaman utama (frontpage)
Aksi Aktor	**Reaksi Sistem**
	Sistem memanggil seluruh data berita yang ada dalam *database*
	Sistem menampilkan data tersebut pada halaman berita
Kondisi Akhir	Sistem menampilkan daftar berita

3.4.6.2 **Dinamika Objek**



Gambar 8. Dinamika Objek Lihat Berita

1.4.7 Autentifikasi *Member*

Fungsi autentikasi *member* merupakan proses pengecekan apakah pengguna seorang *member* atau bukan. Proses ini harus dilalui oleh *member* agar dapat mengirimkan pesan kepada penyiar dan *crew*.

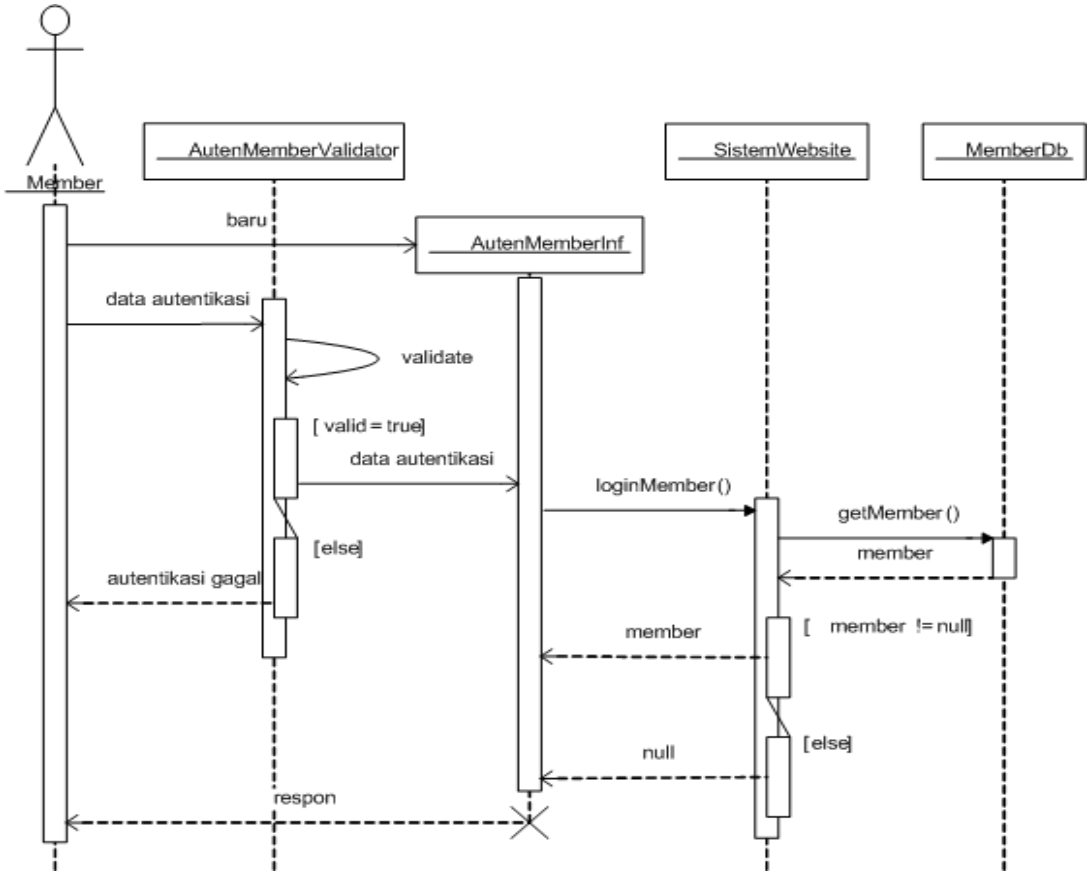
3.4.7.1 **Skenario**

Tabel 6. Skenario Autentifikasi Member

Identifikasi		
Nomor	7	
Nama	Autentikasi *Member*	
Tujuan	Memastikan apakah pengguna telah terdaftar	
Deskripsi	Sistem akan mengecek apakah seorang pengguna telah terdaftar menjadi *member*, sehingga pengguna bisa melakukan proses komunikasi pesan	
Tipe		
Aktor	*Member*	
Skenario Utama		
Kondisi Awal	*Member* masuk ke halaman autentikasi	
Aksi Aktor		**Reaksi Sistem**
Member memasukkan data *username* dan *password*		Sistem melakukan pengecekan apakah ada pengguna dengan *username* dan *password* tersebut dalam *database*
		Jika data yang dimasukkan pengguna valid, maka akan ada pesan autentifikasi valid
Skenario Alternatif – Validasi Gagal		
Aksi Aktor		**Reaksi Sistem**
Member memasukkan data *username* dan *password* lagi		Sistem akan melakukan proses validasi lalu pengecekan pengguna lagi.
Kondisi Akhir	Sistem menampilkan halaman pesan autentifikasi valid	

3.4.7.2 **Dinamika Objek**

Berikut ini adalah dinamika objek dari proses autentifikasi member.



Gambar 9. Dinamika Objek pada Autentifikasi Member

1.4.8 Komunikasi Pesan

Fungsi ini melakukan proses pengiriman pesan dari *member* ke penyiar dan *crew* begitupun sebaliknya.

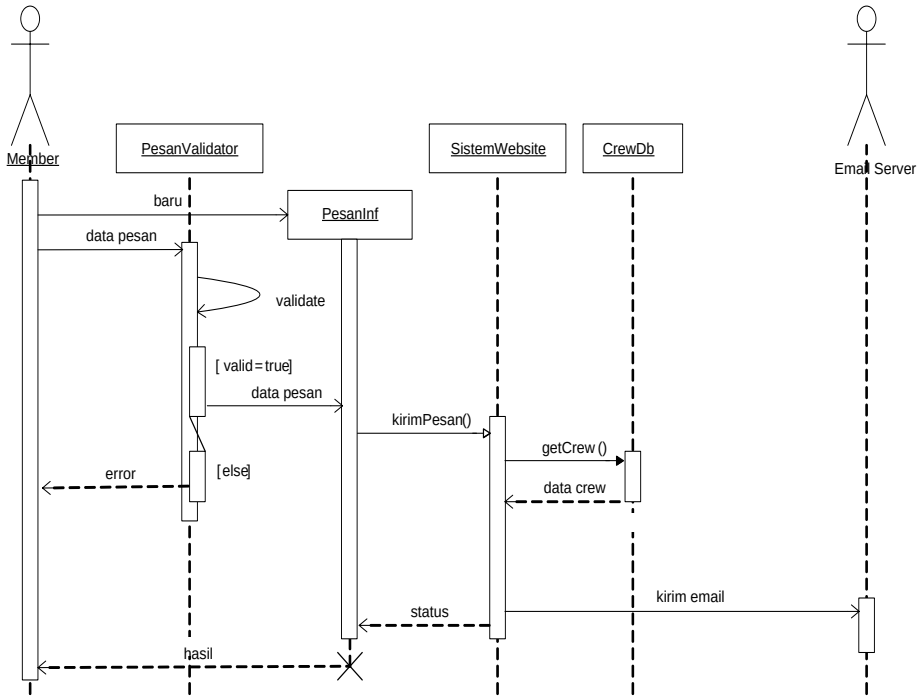
3.4.8.1 **Skenario**

Tabel 7. Skenario Komunikasi Pesan

Identifikasi		
Nomor	8	
Nama	Komunikasi Pesan	
Tujuan	Mengirimkan pesan dari *member* kepada penyiar dan *crew*	
Deskripsi	*Member* menginputkan pesan pada *form* pesan yang telah tersedia pada halaman masing-masing *crew* dan penyiar, lalu sistem akan mengirimkan pesan tersebut ke *email crew* dan penyiar yang bersangkutan.	
Tipe		
Aktor	*Member*	
Skenario Utama		
Kondisi Awal	*Member* masuk ke halaman *crew* yang akan di kirim pesan	
Aksi Aktor		**Reaksi Sistem**
Member masuk ke halaman *crew* yang akan di kirim pesan dan menginputkan pesan yang akan dikirim		Sistem akan mengirimkan pesan ke *email* penyiar dan *crew* yang bersangkutan
		Jika proses berhasil maka akan ditampilkan pesan pengiriman pesan berhasil.
Skenario Alternatif pengiriman pesan gagal		
Aksi Aktor		**Reaksi Sistem**
Member mengetikan kembali pesan yang akan di kirimkan		
Kondisi Akhir	Pesan berhasil di kirim ke krew dan penyiar yang bersangkutan	

3.4.8.3 **Dinamika Objek**

Berikut ini adalah dinamika objek dari proses komunikasi pesan.



Gambar 10. Dinamika Objek pada Komunikasi Pesan

1.4.9 Memasukkan Berita Baru

Fungsi ini melakukan proses memasukkan berita baru ke website.

3.4.9.1 **Skenario**

Tabel 8. Skenario Memasukkan Berita Baru

Identifikasi	
Nomor	9
Nama	Memasukkan berita baru
Tujuan	Menambah berita baru
Deskripsi	Sistem akan menambah berita yang dimasukkan oleh *Administrator* ke *database*
Tipe	

Aktor	*Administrator*
Skenario Utama	
Kondisi Awal	*Administrator* masuk ke halaman berita

Aksi Aktor	**Reaksi Sistem**
Administrator memasukkan berita baru	Sistem menyimpan berita yang dimasukkan ke dalam *database*
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
Administrator memasukkan berita lagi	

Kondisi Akhir	Berita telah tersimpan

3.4.6.1 D

3.4.6.2

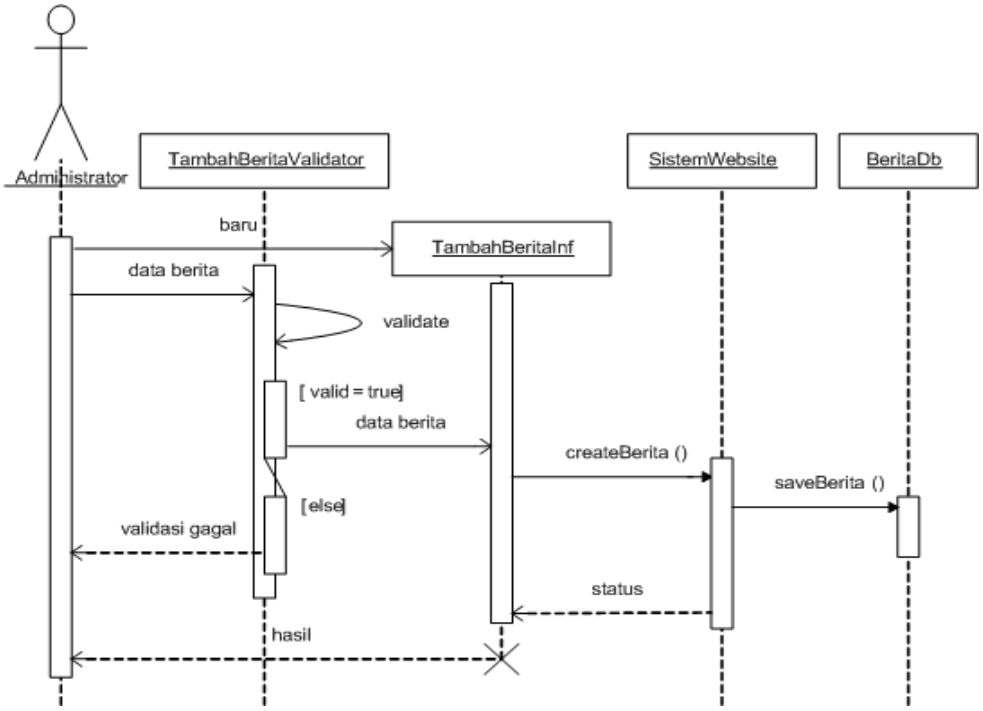
3.4.6.3

3.4.6.4

3.4.9.2

3.4.9.3 **Dinamika Objek**

Berikut ini adalah dinamika objek dari proses memasukan berita baru.



Gambar 11. Dinamika Objek pada Tambah Berita Baru

1.4.10 Menghapus Berita

Fungsi ini melakukan proses penghapusan berita yang sudah lama.

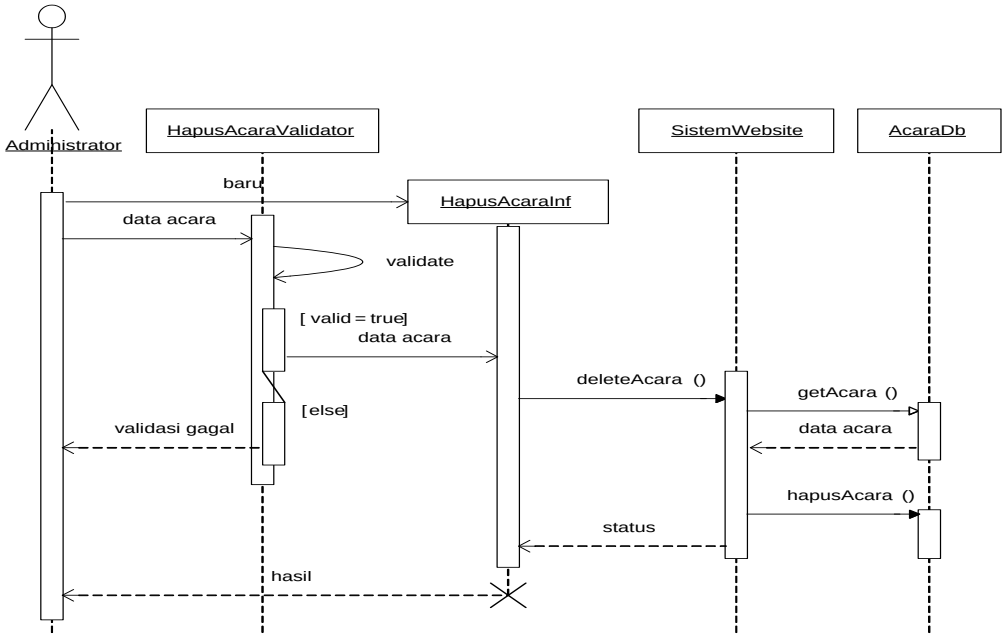
3.4.10.1 **Skenario**

Tabel 9. Skenario Menghapus Berita

Identifikasi		
Nomor	10	
Nama	Menghapus berita	
Tujuan	Menghapus berita yang sudah lama	
Deskripsi	Sistem akan menghapus berita yang dilakukan oleh *Administrator*	
Tipe		
Aktor	*Administrator*	
Skenario Utama		
Kondisi Awal	*Administrator* masuk ke halaman berita	
Aksi Aktor		**Reaksi Sistem**
Administrator memilih berita yang akan di hapus, lalu menghapus berita tersebut		Sistem menghapus berita yang di pilih oleh *Administrator*
Skenario Alternatif – Validasi Gagal		
Aksi Aktor		**Reaksi Sistem**
Administrator menghapus berita lagi		
Kondisi Akhir	Berita telah terhapus	

3.4.10.2 **Dinamika Objek**

Berikut ini adalah dinamika objek dari proses menghapus berita.



Gambar 12. Dinamika Objek pada Menghapus Berita

1.4.11 Mengedit Berita

Fungsi ini melakukan proses pengeditan berita yang sudah disimpan.

3.4.11.1 **Skenario**

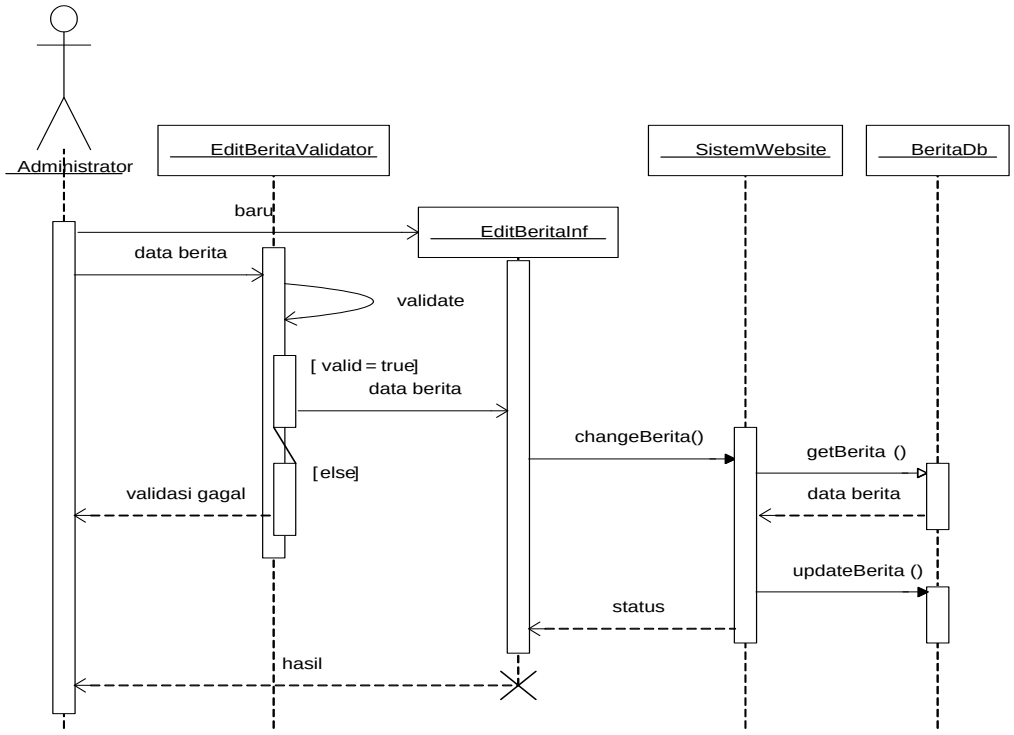
Tabel 10. Skenario Mengedit Berita

Identifikasi	
Nomor	11
Nama	Mengedit berita
Tujuan	Mengedit berita yang sudah disimpan
Deskripsi	Sistem akan mengedit berita yang dilakukan oleh *Administrator*
Tipe	
Aktor	*Administrator*
Skenario Utama	

Kondisi Awal	*Administrator* masuk ke halaman berita
Aksi Aktor	**Reaksi Sistem**
Administrator memilih berita yang akan di edit, lalu mengedit berita tersebut	Sistem menyimpan berita yang telah di edit
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
Administrator mengedit berita lagi	
Kondisi Akhir	Berita telah diubah

3.4.11.2 **Dinamika Objek**

Berikut ini adalah dinamika objek dari proses mengedit berita.



Gambar 13. Dinamika Objek pada Mengedit Berita

1.4.12 Memasukkan Acara Baru

Fungsi ini melakukan proses acara baru ke website.

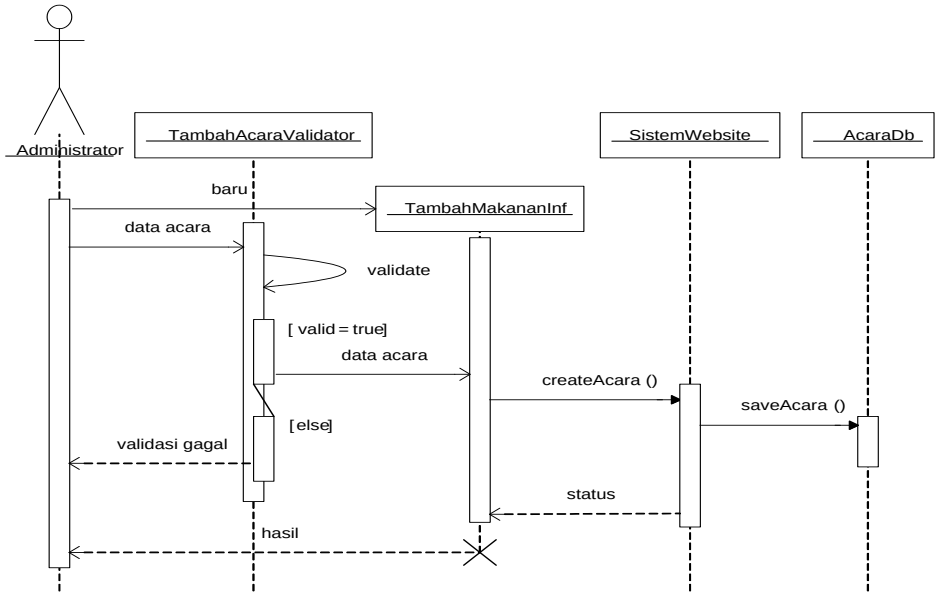
3.4.12.1 **Skenario**

Tabel 11. Skenario Memasukkan Acara Baru

Identifikasi		
Nomor	12	
Nama	Memasukkan acara baru	
Tujuan	Menambah acara baru	
Deskripsi	Sistem akan menambah acara yang dimasukkan oleh *Administrator* ke *database*	
Tipe		
Aktor	*Administrator*	
Skenario Utama		
Kondisi Awal	*Administrator* masuk ke halaman acara	
Aksi Aktor		**Reaksi Sistem**
Administrator memasukkan acara baru		Sistem menyimpan acara yang dimasukkan ke dalam *database*
Skenario Alternatif – Validasi Gagal		
Aksi Aktor		**Reaksi Sistem**
Administrator memasukkan acara lagi		
Kondisi Akhir	Acara telah tersimpan	

3.4.12.2 **Dinamika Objek**

Berikut ini adalah dinamika objek dari proses memasukan acara baru.



Gambar 14. Dinamika Objek pada Penambahan Acara

1.4.13 Menghapus Acara

Fungsi ini melakukan proses penghapusan acara yang sudah lama.

3.4.13.1 **Skenario**

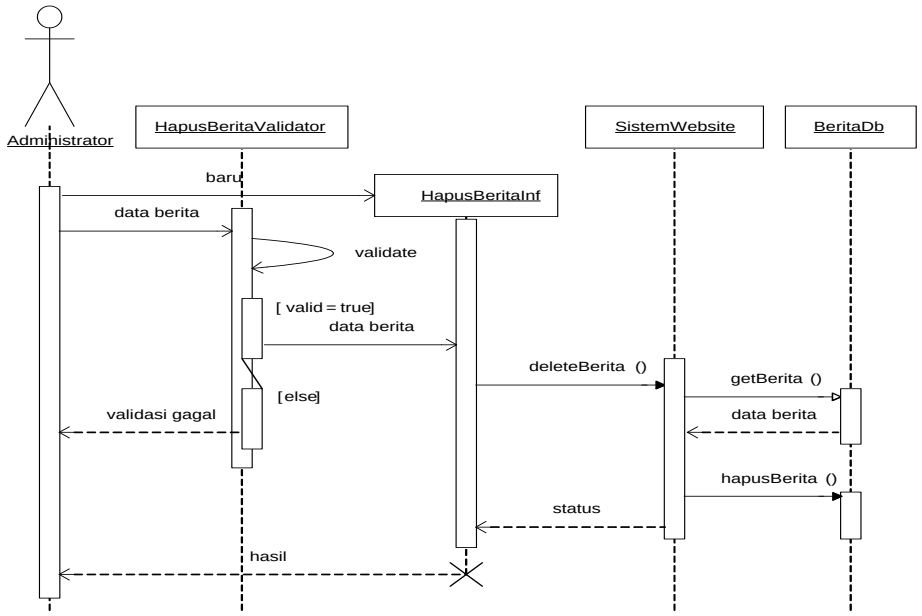
Tabel 12. Skenario Menghapus Acara

Identifikasi	
Nomor	13
Nama	Menghapus acara
Tujuan	Menghapus acara yang sudah lama
Deskripsi	Sistem akan menghapus acara yang dilakukan oleh *Administrator*
Tipe	
Aktor	*Administrator*
Skenario Utama	
Kondisi Awal	*Administrator* masuk ke halaman acara

Aksi Aktor	Reaksi Sistem
Administrator memilih acara yang akan di hapus, lalu menghapus acara tersebut	Sistem menghapus acara yang sudah di pilih oleh *Administrator*
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
Administrator menghapus acara lagi	
Kondisi Akhir	Acara telah dihapus

3.4.13.2 **Dinamika Objek**

Berikut ini adalah dinamika objek dari proses menghapus acara.



Gambar 15. Dinamika Objek pada Penghapusan Acara

1.4.14 Mengedit Acara

Fungsi ini melakukan proses pengeditan acara yang sudah disimpan.

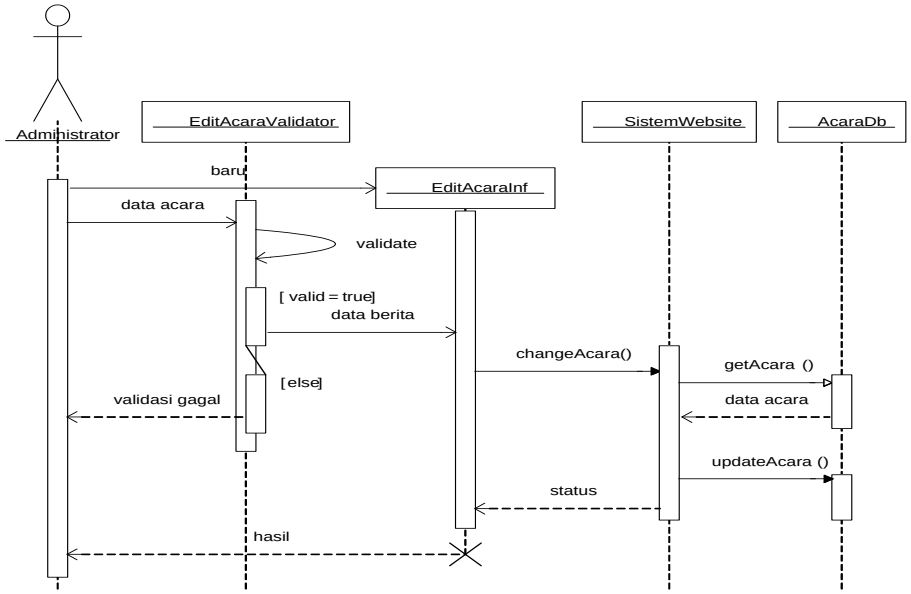
3.4.14.1 **Skenario**

Tabel 13. Skenario Mengedit Acara

Identifikasi		
Nomor	14	
Nama	Mengedit acara	
Tujuan	Mengedit acara yang sudah disimpan	
Deskripsi	Sistem akan mengedit acara yang dilakukan oleh *Administrator*	
Tipe		
Aktor	*Administrator*	
Skenario Utama		
Kondisi Awal	*Administrator* masuk ke halaman acara	
Aksi Aktor		**Reaksi Sistem**
Administrator memilih berita yang akan di edit, lali mengedit acara tesebut		Sistem akan menyimpan acara yang sudah di edit oleh *Administrator*
Skenario Alternatif – Validasi Gagal		
Aksi Aktor		**Reaksi Sistem**
Administrator mengedit acara lagi		
Kondisi Akhir	Acara telah diedit	

3.4.14.2 **Dinamika Objek**

Berikut ini adalah dinamika objek dari proses mengedit acara.



Gambar 16. Dinamika Objek pada Pengeditan Acara

1.4.15 Memasukkan *Crew* Baru

Fungsi ini melakukan proses memasukkan *crew* baru ke website.

3.4.15.1 **Skenario**

Tabel 14. Skenario memasukkan *crew* baru

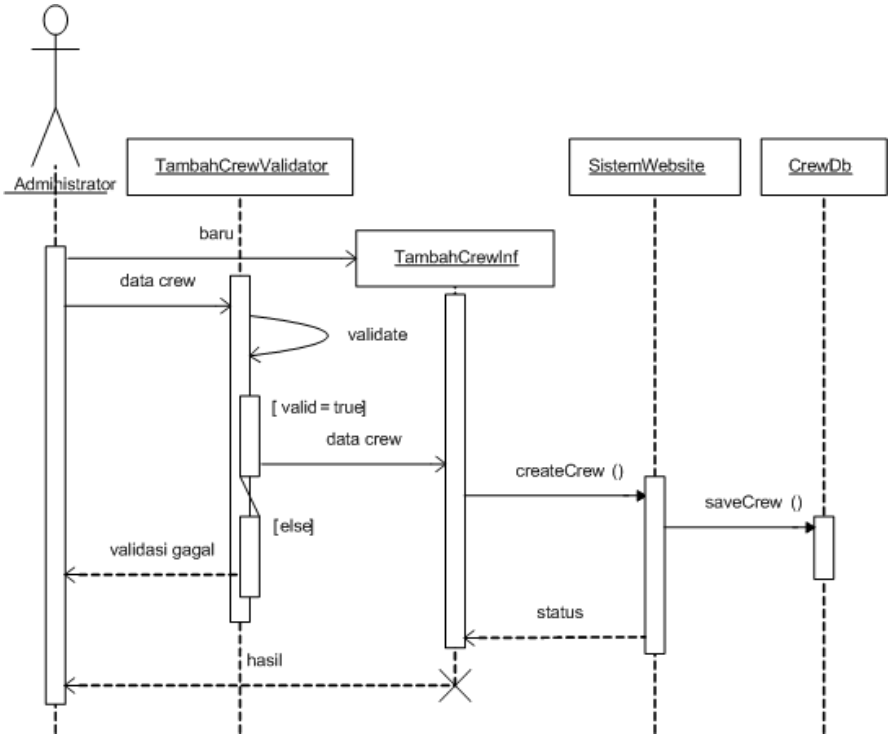
Identifikasi	
Nomor	15
Nama	Memasukkan *crew* baru
Tujuan	Menambah *crew* baru
Deskripsi	Sistem akan menambah *crew* yang dimasukkan oleh *Administrator* ke *database*
Tipe	
Aktor	*Administrator*
Skenario Utama	
Kondisi Awal	*Administrator* masuk ke halaman *crew*

Aksi Aktor	Reaksi Sistem
Administrator memasukkan data *crew* baru	Sistem menyimpan *crew* baru dan dimasukkan ke dalam *database*
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
Administrator memasukkan data *crew* lagi	
Kondisi Akhir	*Crew* telah tersimpan

3.4.15.2

3.4.15.3 **Dinamika Objek**

Berikut ini adalah dinamika objek dari proses memasukan *crew* baru.



Gambar 17. Dinamika Objek pada Memasukkan *Crew*

1.4.16 Menghapus *Crew*

Fungsi ini melakukan proses penghapusan *crew* yang sudah lama.

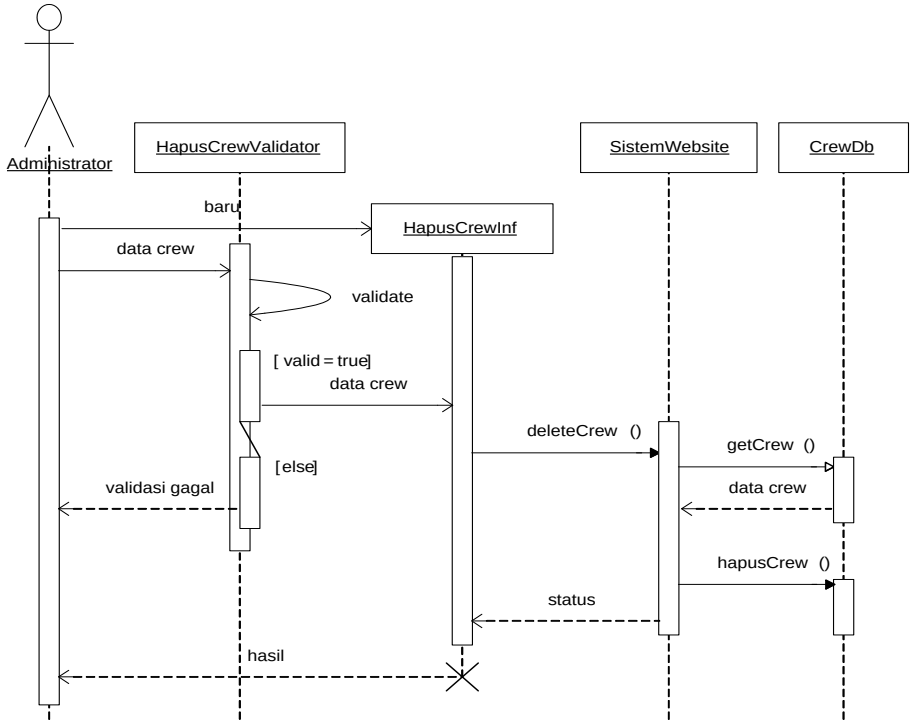
3.4.16.1 **Skenario**

Tabel 15. Skenario Menghapus *Crew*

Identifikasi		
Nomor	16	
Nama	Menghapus *crew*	
Tujuan	Menghapus *crew* yang sudah lama	
Deskripsi	Sistem akan menghapus *crew* yang dilakukan oleh *Administrator*	
Tipe		
Aktor	*Administrator*	
Skenario Utama		
Kondisi Awal	*Administrator* masuk ke halaman *crew*	
Aksi Aktor		**Reaksi Sistem**
Administrator memilih data *crew* yang akan di hapus, lalu menghapusnya		Sistem akan menghapus data *crew* yang telah di pilih *Administrator*
Skenario Alternatif – Validasi Gagal		
Aksi Aktor		**Reaksi Sistem**
Administrator menghapus *crew* lagi		
Kondisi Akhir	*Crew* telah terhapus	

3.4.16.2 **Dinamika Objek**

Berikut ini adalah dinamika objek dari proses menghapus *crew*.



Gambar 18. Dinamika Objek pada Menghapus *Crew*

1.4.17 Mengedit *Crew*

Fungsi ini melakukan proses pengeditan *crew* yang sudah disimpan.

3.4.17.1 **Skenario**

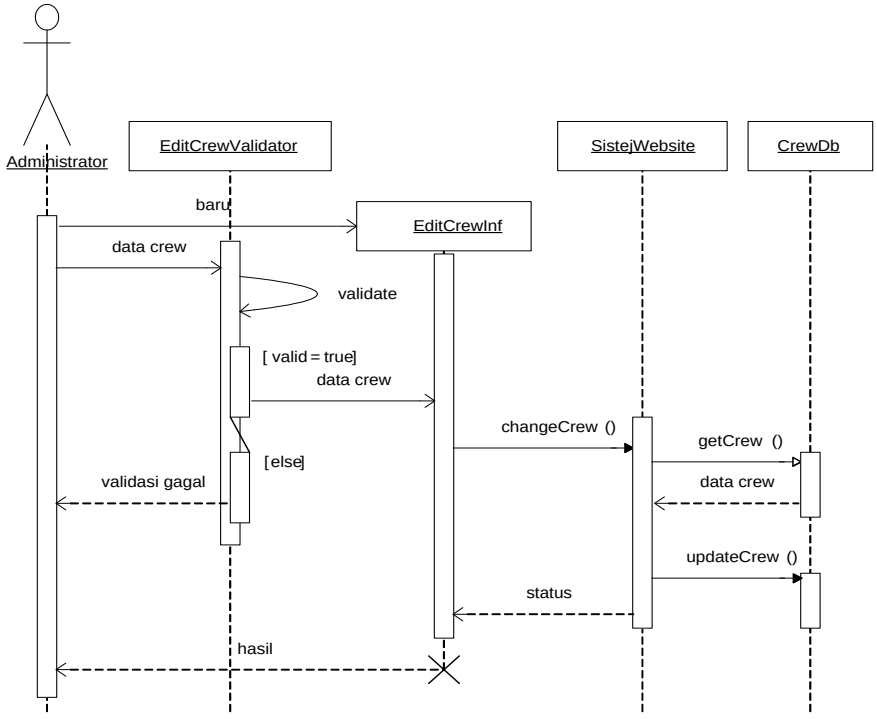
Tabel 16. Skenario Mengedit *Crew*

Identifikasi	
Nomor	17
Nama	Mengedit *crew*
Tujuan	Mengedit *crew* yang sudah disimpan
Deskripsi	Sistem akan mengedit *crew* yang dilakukan oleh *Administrator*
Tipe	

Aktor	*Administrator*	
Skenario Utama		
Kondisi Awal	*Administrator* masuk ke halaman *crew*	
Aksi Aktor		**Reaksi Sistem**
Administrator memilih terlebih dahulu *crew* ang akan di edit, lalu mengeditnya		Sistem menyimpan dta *crew* yang telah di edit
Skenario Alternatif – Validasi Gagal		
Aksi Aktor		**Reaksi Sistem**
Administrator mengedit *crew* lagi		
Kondisi Akhir	*Crew* telah diubah	

3.4.17.2 **Dinamika Objek**

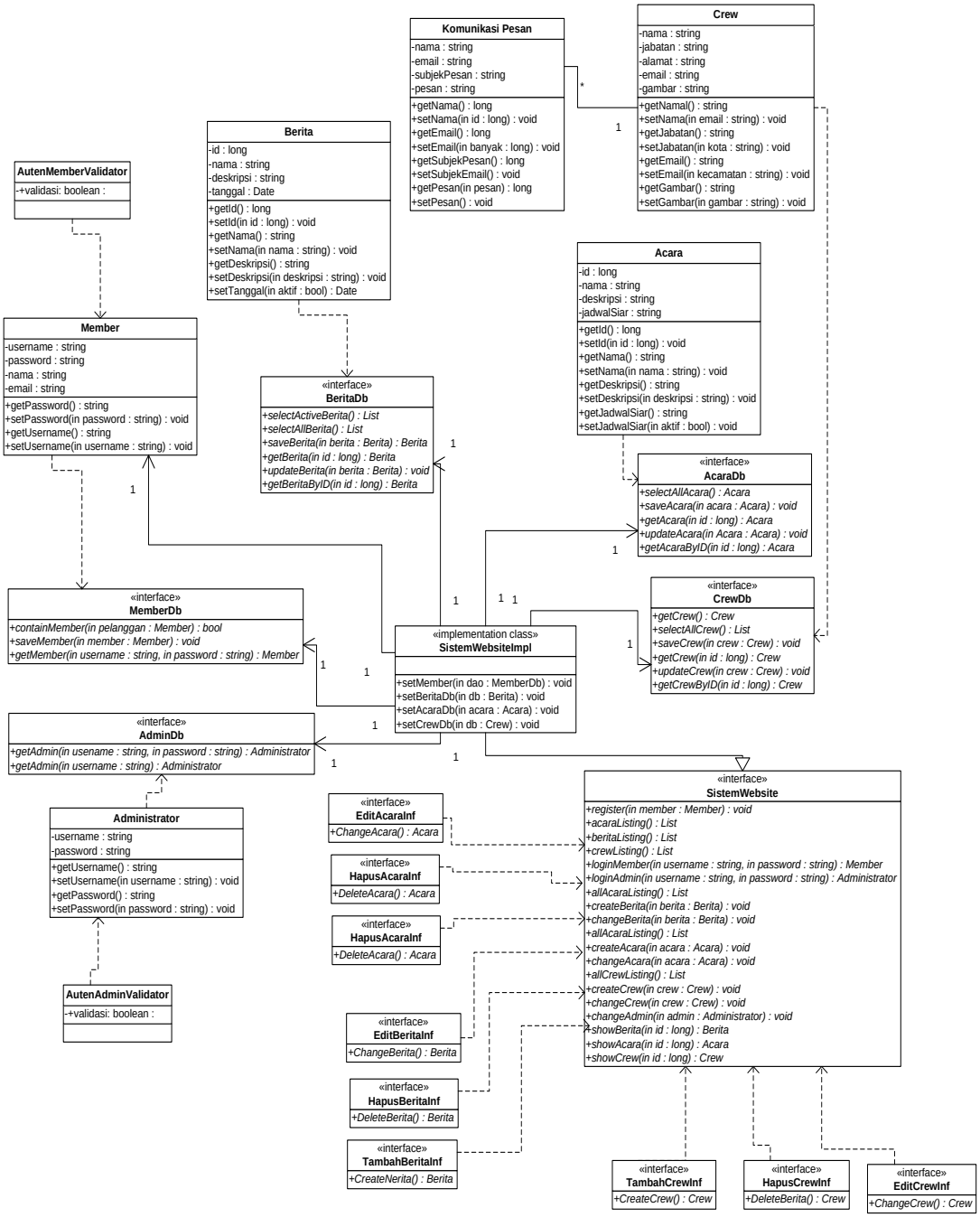
Berikut ini adalah dinamika objek dari proses mengedit *crew*.



Gambar 19. Dinamika Objek pada Mengedit *Crew*

1.5 Deskripsi Kelas – Kelas

1.5.1 Diagram Kelas



Gambar 20. Kelas Diagram

1.5.2 Tabel Kelas

Berikut ini adalah kelas – kelas yang terdapat dalam *website* ini :

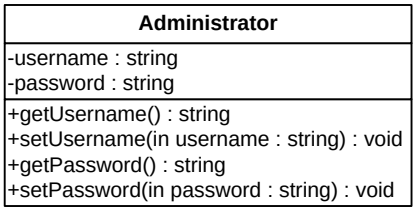
Tabel 17. Keterangan Kelas

Nama Kelas	Tipe
Member	Entitas
Administrator	Entitas
Berita	Entitas
Acara	Entitas
Crew	Entitas
MemberDb	*Database*
*Admin*Db	*Database*
AcaraDb	*Database*
BeritaDb	*Database*
*Crew*Db	*Database*
SistemWebsiteImpl	Implementation Class
SistemWebsite	*Interface*
TambahBeritainf	*Interface*
EditBeritaInf	*Interface*
HapusBeritaInf	*Interface*
TambahAcarainf	*Interface*
EditAcaraInf	*Interface*
HapusAcaraInf	*Interface*
Tambah*Crew*Inf	*Interface*
Edit*Crew*Inf	*Interface*
HapusBeritaInf	*Interface*
Validator	Validator

1.5.3 Spesifikasi Kelas

3.5.3.1 **Kelas *Admini*trator**

Kelas *Administrator* adalah kelas yang merepresentasikan aktor *Administrator*.



Gambar 21. Kelas *Administrator*

3.5.3.1.1 Deskripsi Atribut

Tabel 18. Deskripsi Atribut Kelas *Administrator*

Nama	Tipe	Panjang	Range	Nilai Tetap
Username	String	45	a-z, A-Z	Null
Password	String	45	-	Null

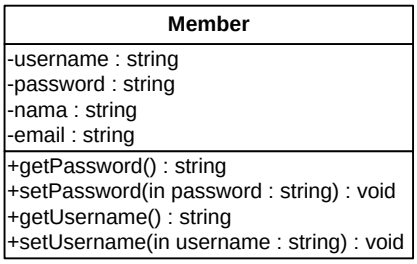
3.5.3.1.2 Deskripsi Layanan

Operasi – operasi yang ada dalam kelas ini adalah :

1. get*Username*() digunakan untuk mendapatkan nilai *username*.
2. set*Username*(String *username*) digunakan untuk mengubah nilai *username*.
3. get*Password*() digunakan untuk mendapatkan nilai *password*.
4. set*Password*(String *password*) digunakan untuk mengubah nilai *password*.

3.5.3.2 **Kelas *Member***

Kelas pelanggan merupakan representasi dari aktor Pelanggan.



Gambar 22. Kelas Member

3.5.3.2.1 Deskripsi Atribut

Tabel 19. Deskripsi Atribut Kelas Member

Nama	Tipe	Panjang	Range	Nilai Tetap
username	String	45	a-z, A-Z	Null
password	String	45	-	Null
nama	String	30	a-z, A-Z	Null
email	String	45	a-z, A-Z	Null

3.5.3.2.2 Deskripsi Layanan

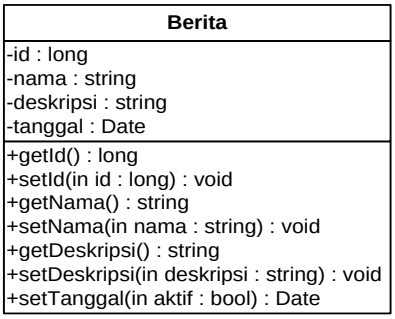
Operasi – operasi yang ada dalam kelas ini adalah :

1. get*Username*() digunakan untuk mendapatkan nilai *username*.
2. set*Username*(String *username*) digunakan untuk mengubah nilai *username*.
3. get*Password*() digunakan untuk mendapatkan nilai *password*.

4. set*Password*(String *password*) digunakan untuk mengubah nilai *password*.
5. getNama() digunakan untuk mendapatkan nilai nama
6. setNama(String nama) digunakan untuk mengubah nilai nama.

3.5.3.3 **Kelas Berita**

Kelas berita merupakan representasi dari berita dari pemkot bandung.



Gambar 23. Kelas Berita

3.5.3.3.1 Deskripsi Atribut

Tabel 20. Deskripsi Atribut Kelas Berita

Nama	Tipe	Panjang	Range	Nilai Tetap
id	Long	-	-	Null
nama	String	45	a-z, A-Z	Null
deskripsi	String	-	-	Null
tanggal	Date	-	-	True

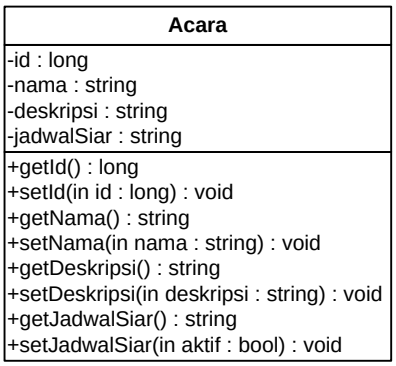
3.5.3.3.2 Deskripsi Layanan

Operasi – operasi yang ada dalam kelas ini adalah :

1. getId() digunakan untuk mendapatkan nilai id.
2. setId(Long id) digunakan untuk mengubah nilai id.
3. getNama() digunakan untuk mendapatkan nilai nama.
4. setNama(String nama) digunakan untuk mengubah nilai nama.
5. getDeskripsi() digunakan untuk mendapatkan nilai deskripsi.
6. setDeskripsi(String deskripsi) digunakan untuk mengubah nilai deskripsi.
7. gettanggal() digunakan untuk mendapatkan nilai tanggal
8. setTanggal digunakan untuk mengubah nilai Tanggal

3.5.3.4 Kelas Acara

Kelas acara merupakan representasi dari acara radio sonata.



Gambar 24. Kelas Acara

3.5.3.4.1 Deskripsi Atribut

Tabel 21. Deskripsi Atribut Kelas Acara

Nama	Tipe	Panjang	Range	Nilai Tetap
id	Long	-	-	Null
nama	String	45	a-z, A-Z	Null
deskripsi	String	-	-	Null
JadwalSiar	Date	-	-	True

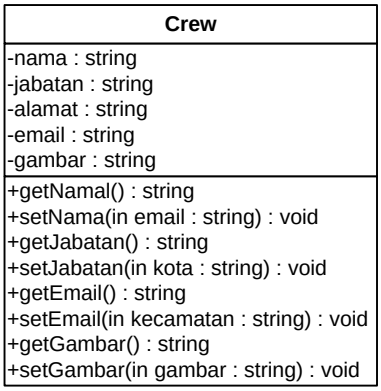
3.5.3.4.2 Deskripsi Layanan

Operasi – operasi yang ada dalam kelas ini adalah :

1. getId() digunakan untuk mendapatkan nilai id.
2. setId(Long id) digunakan untuk mengubah nilai id.
3. getNama() digunakan untuk mendapatkan nilai nama.
4. setNama(String nama) digunakan untuk mengubah nilai nama.
5. getDeskripsi() digunakan untuk mendapatkan nilai deskripsi.
6. setDeskripsi(String deskripsi) digunakan untuk mengubah nilai deskripsi.
7. getJadwalSiarl() digunakan untuk mendapatkan Jadwal Siaran
8. setJadwalSiar digunakan untuk mengubah Jadwal Siar

3.5.3.5 **Kelas *Crew***

Kelas *Crew* merupakan representasi dari *crew* radio sonata



Gambar 25. Kelas *Crew*

3.5.3.5.1 Deskripsi Atribut

Tabel 22. Deskripsi Atribut Kelas *Crew*

Nama	Tipe	Panjang	Range	Nilai Tetap
nama	String	45	a-z, A-Z	Null
jabatan	String	45	a-z, A-Z	Null
gambar	String	255	-	Null
alamat	String	90	-	Null
email	string	45	a-z, A-Z	Null

3.5.3.5.2 Deskripsi Layanan

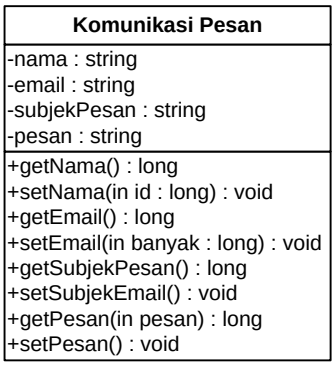
Operasi – operasi yang ada dalam kelas ini adalah :

1. getNama() digunakan untuk mendapatkan nilai nama.
2. setNama() digunakan untuk mengubah nilai nama.
3. getJabatan() digunakan untuk mendapatkan nilai jabatan.

4. setJabatan() digunakan untuk mengubah nilai Jabatan.
5. getAlamat() digunakan untuk mendapatkan nilai alamat.
6. setAlamat() digunakan untuk mengubah nilai alamat.
7. get*Email*() digunakan untuk mendapatkan *Email*
8. set*Email* digunakan untuk mengubah Nilai *email*.

3.5.3.6 **Kelas Komunikasi Pesan**

Kelas komunikasi pesan merupakan kelas pengiriman pesan dari pelanggan ke *crew*



Gambar 26. Kelas Komunikasi Pesan

3.5.3.6.1 Deskripsi Atribut

Tabel 23. Deskripsi Atribut Kelas Komunikasi Pesan

Nama	Tipe	Panjang	Range	Nilai Tetap
nama	String	45	a-z, A-Z	Null
Email	String	45	a-z, A-Z	Null
subjekPesan	String	45	-	Null
pesan	String	300	-	Null
email	string	45	-	Null

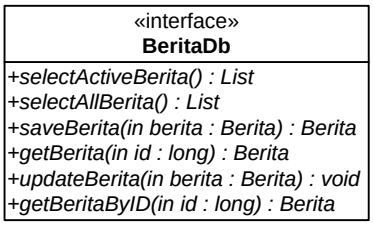
3.5.3.6.2 Deskripsi Layanan

Operasi – operasi yang ada dalam kelas ini adalah :

1. getNama() digunakan untuk mendapatkan nilai nama.
2. setNama() digunakan untuk mengubah nilai nama.
3. getSubjek*Email*() digunakan untuk mendapatkan nilai subjek *email*
4. setJSubjek*Email*() digunakan untuk mengubah subjek *email*.
5. getPesan() digunakan untuk mendapatkan nilai pesan.
6. setPesan() digunakan untuk mengubah nilai pesan.
7. get*Email*() digunakan untuk mendapatkan *Email*
8. set*Email* digunakan untuk mengubah Nilai *email*.

3.5.3.7 **Kelas BeritaDb**

Merupakan bisnis logic untuk kelas BeritaBd



Gambar 27. Kelas BeritaDb

3.5.3.7.1 **Deskripsi Atribut**

Kelas ini tidak memiliki atribut

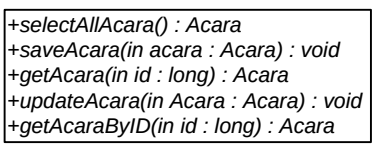
3.5.3.7.2 **Deskripsi Layanan**

Operasi – operasi yang ada dalam kelas ini adalah :

1. selectActiveRestoran() digunakan untuk mendapatkan seluruh data berita yang aktif.
2. selectAllBerita() digunakan untuk mendapatkan seluruh data berita.
3. saveBerita() digunakan untuk menyimpan data berita.
4. updateBerita() digunakan untuk mengubah data berita.
5. getBeriyaById(Long id) digunakan untuk mendapatkan data berita berdasarkan id.

3.5.3.8 **Kelas AcaraDb**

Merupakan Bisnis logic untuk kelas AcaraDb



Gambar 28. Kelas AcaraDb

3.5.3.8.1 Deskripsi Atribut

Kelas ini tidak memiliki atribut

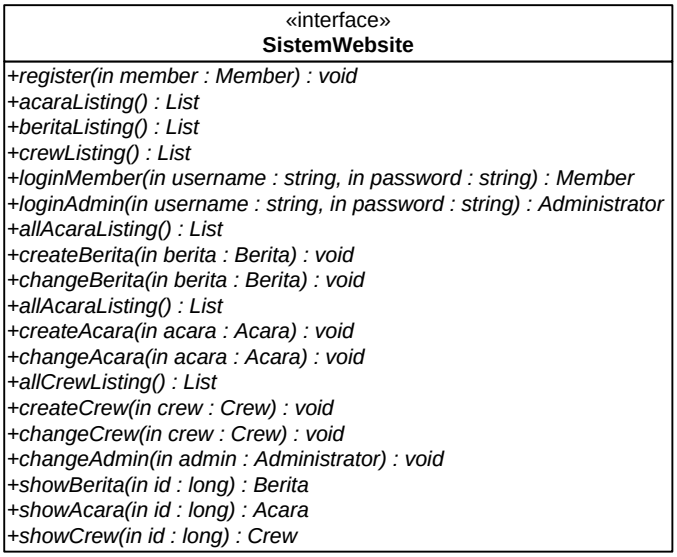
3.5.3.8.2 Deskripsi Layanan

Operasi – operasi yang ada dalam kelas ini adalah :

1. selectAllAcara() digunakan untuk mendapatkan seluruh data berita yang aktif.
2. saveAcara() digunakan untuk menyimpan data acara.
3. getAcara() digunakan untuk mengubah data acara
4. updateAcara() digunakan untuk mengubah data acara
5. getAcaraById() digunakan untuk mendapatkan data acara berdasarkan ID.

3.5.3.9 Kelas Sistem Website

Kelas ini merupakan *interface* untuk service yang di gunakan dalam sistem website sonata



Gambar 29. Kelas Sistem Website

3.5.3.9.1 Deskripsi Atribut

Kelas ini tidak memiliki atribut

3.5.3.9.2 Deskripsi Layanan

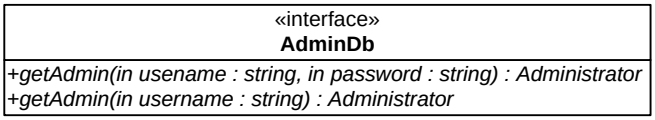
Operasi – operasi yang ada dalam kelas ini adalah :

1. register(*member member*) digunakan untuk mendaftarkan *member* baru.
2. acaraListing() digunakan untuk mendapatkan seluruh daftar acara.
3. beritaListing() digunakan untuk mendapatkan seluruh daftar minuman.
4. *crew*Listing() diguankan untuk mendapatkan seluduh daftar *crew*.
5. *loginMember*(String *username*, String *password*) digunakan untuk melakukan autentikasi *member*.
6. *loginAdmin*(String *username*, String *password*) digunakan untuk melakukan autentikasi *Administrator*
7. allAcaraListing() digunakan untuk mendapatkan seluruh data acara.
8. createAcara(Acara acara) digunakan untuk menambah data acara baru.
9. changeAcara(Acara acara) digunakan untuk mengubah data acara.
10. all*Crew*Listing() digunakan untuk mendapatkan seluruh data *crew*.
11. create*Crew*(*Crew crew*) digunakan untuk menambah data *crew*.
12. change*Crew*(*Crew crew*) digunakan untuk mengubah data acara.

13. allRestoranListing() digunakan untuk mendapatkan seluruh data restoran.
14. showBerita(Long id) digunakan untuk mendapatkan data beritaberdasarkan id.
15. showAcara(Long id) digunakan untuk mendapatkan data acara berdasarkan id.
16. show*Crew*(Long id) digunakan untuk mendapatkan data *crew* berdasarkan id.

3.5.3.10 **Kelas *Admin*Db**

Kelas *Admin*DB merupakan kelas bisnis logic untuk kelas *Administrator*



Gambar 30. Kelas *Admin*Db

3.5.3.10.1 Deskripsi Atribut

Kelas ini tidak memiliki atribut

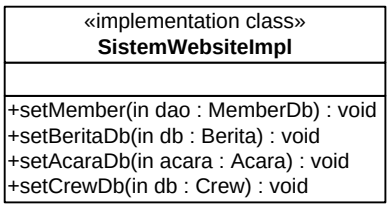
3.5.3.10.2 Deskripsi Layanan

Operasi – operasi yang ada dalam kelas ini adalah :

1. get*Admin*(String *username*, String *password*) digunakan untuk mendapatkan *Administrator* berdasarkan *username* dan *password*.
2. get*Admin*(String *username*) digunakan untuk

3.5.3.11 **Kelas SistemWebsiteImpl**

Kelas SistemWebsiteImpl merupakan kelas implementasi dari *interface* Sistem Website



Gambar 31. Kelas SistemWebsiteImpl

3.5.3.11.1 Deskripsi Atribut

Kelas ini tidak memiliki atribut

3.5.3.11.2 Deskripsi Layanan

Operasi – operasi yang ada dalam kelas ini adalah :

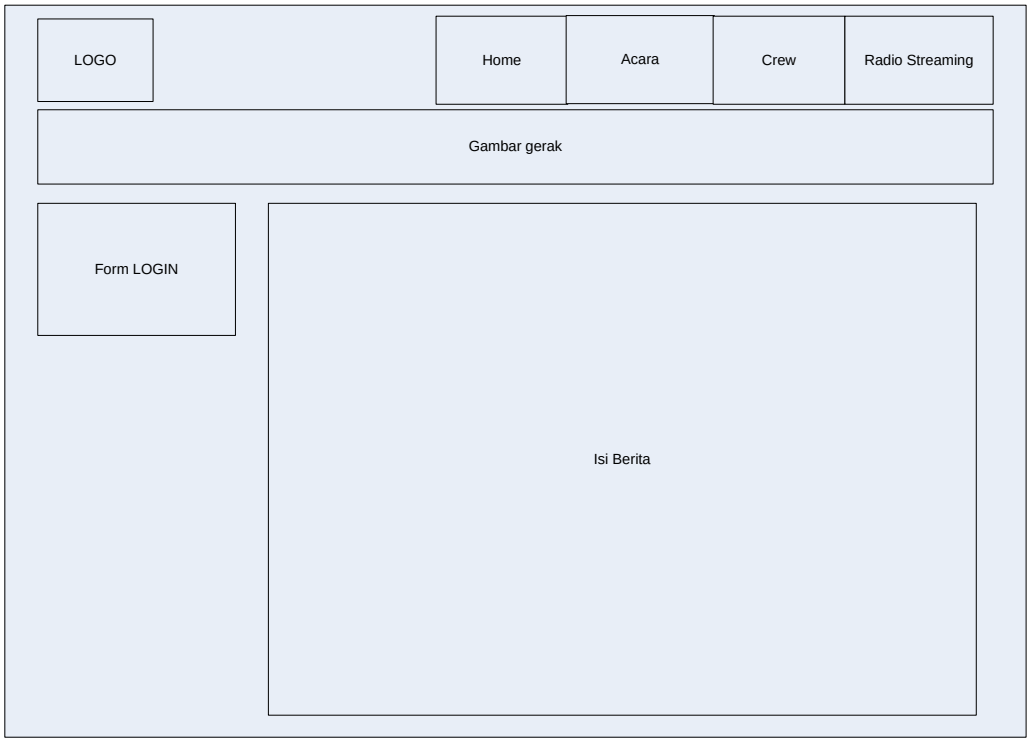
a) set*Member*Bd(*Member*Db db) mengubah nilai *Member*Db.
b) setAcaraDb(AcaraDb db) mengubah nilai acaraDb.
c) setBeritaDb(BeritaDb db) mengubah nilai beritaDb.
d) set*Crew*Db(*Crew*Db db) mengubah nilai *Crew*Db.

1.6 Perancangan Tampilan

Perancangan tampilan aplikasi bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun. Sehingga akan mempermudah dalam mengimplementasikan aplikasi. Dan juga akan memudahkan pembuatan apliaksi yang *user friendly*. Perancangan ini akan diimplementasikan menjadi sebuah program yang utuh dan dapat digunakan. Rancangan aplikasi website radio sonata yang akan di buat adalah sebagai berikut

1.6.1 Layar Utama Home (*Front page*)

Berikut ini adalah tampilan layar untuk Halaman Utama atau *Home*.



Gambar 32. Tampilan Layar Utama

Keterangan :

Tabel 24. Keterangan Tampilan Layar Utama

Tampilan	Jenis	Keterangan
Logo	Image	Gambar Logo Radio Sonata
Home	Menu	Adalah menu untuk masuk ke halaman Utama (*front page*)
Acara	Menu	Adalah menu yang berisi daftar acara yang di siarkan Radio Sonata
Crew	Menu	Adalah menu yang berisi daftar *crew* dari Radio Sonata
Radio *Streaming*	Menu	Adalah menu untuk masuk ke halaman radio *streaming* yang di sediakan *server* listen2myradio.
Form Login	Modul	Adalah modul untuk *login* bagi

		member dan daftar untuk pengunjung
Jadwal hari ini	Modul	Adalah modul untuk menampilkan jadwal acara hari ini
Gambar Gerak	Modul	Adalah gambar untuk menampilkan gambar bergerak
Isi Berita	Artikel	Adalah artikel Frontpage yang menampilkan berita-berita.

1.6.2 L2 Layar Acara

Berikut ini adalah tampilan layar untuk halaman Acara.



Gambar 33. Tampilan Layar Acara

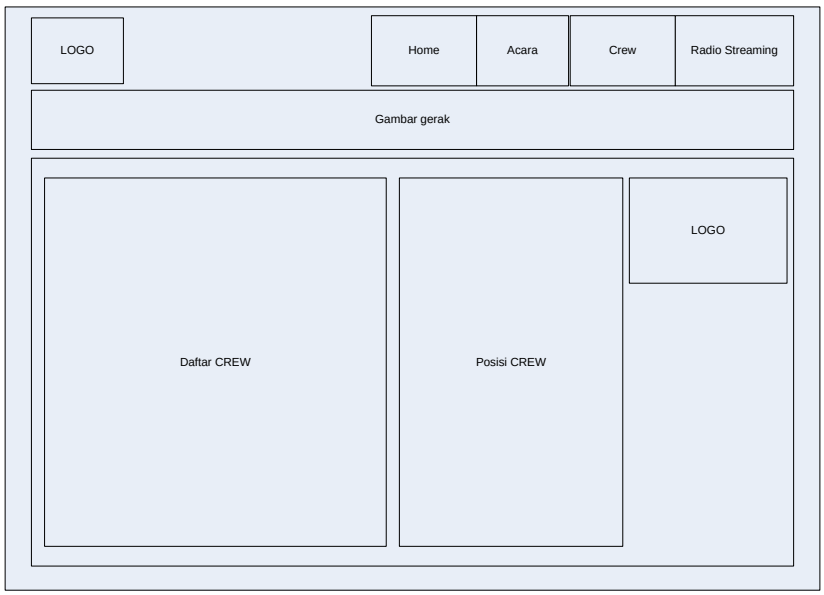
Keterangan :

Tabel 25. Keterangan Tampilan Layar Acara

Tampilan	Jenis	Keterangan
Logo	Image	Gambar Logo Radio Sonata
Home	Menu	Adalah menu untuk masuk ke halaman Utama (*front page*)
Acara	Menu	Adalah menu yang berisi daftar acara yang di siarkan Radio Sonata
Crew	Menu	Adalah menu yang berisi daftar *crew* dari Radio Sonata
Radio *Streaming*	Menu	Adalah menu untuk masuk ke halaman radio *streaming* yang di sediakan *server* listen2myradio.
Gambar Gerak	Modul	Adalah gambar untuk menampilkan gambar bergerak
Isi Deskripsi Acara	Artikel	Adalah artikel yang menampilkan deskripsi masing-masing acara

1.6.3 L3 Layar *Crew*

Berikut ini adalah tampilan layar untuk Halaman *Crew*, yang menampilkan semua *crew*.



Gambar 34. Tampilan Layar *Crew*

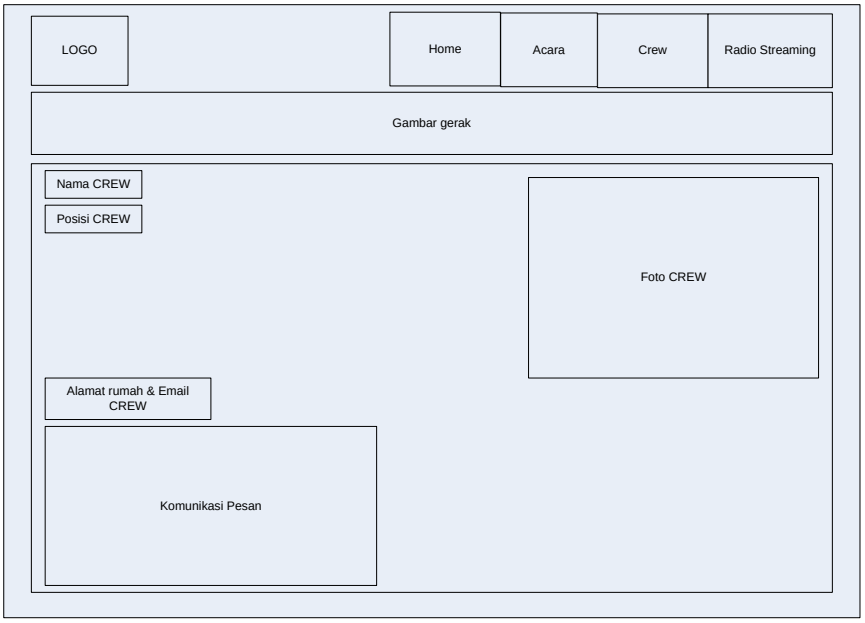
Keterangan:

Tabel 26. Keterangan Tampilan Layar *Crew*

Tampilan	Jenis	Keterangan
Logo	Image	Gambar Logo Radio Sonata
Home	Menu	Adalah menu untuk masuk ke halaman Utama (*front page*)
Acara	Menu	Adalah menu yang berisi daftar acara yang di siarkan Radio Sonata
Crew	Menu	Adalah menu yang berisi daftar *crew* dari Radio Sonata
Radio *Streaming*	Menu	Adalah menu untuk masuk ke halaman radio *streaming* yang di sediakan *server* listen2myradio.
Posisi *Crew*	Component	Adalah menampilkan Posisi setiap *crew* di Radio Sonata
Daftar *Crew*	Component	Adalah menampilkan semua *crew* radio Sonata
Gambar Gerak	Modul	Adalah gambar untuk menampilkan gambar bergerak

1.6.4 L4 Layar *Crew* (Detail)

Berikut ini adalah tampilan layar untuk Halaman Biodata *Crew* secara detail.



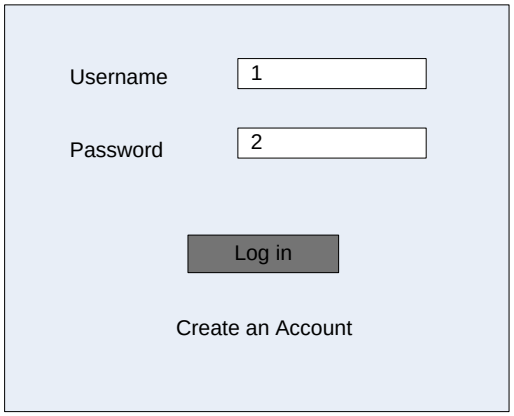
Gambar 35. Tampilan Layar *Crew* (Detail)

Keterangan :

Tabel 27. Keterangan Tampilan Layar *Crew*(Detail)

Tampilan	Jenis	Keterangan
Logo	Image	Gambar Logo Radio Sonata
Home	Menu	Adalah menu untuk masuk ke halaman Utama (*front page*)
Acara	Menu	Adalah menu yang berisi daftar acara yang di siarkan Radio Sonata
Crew	Menu	Adalah menu yang berisi daftar *crew* dari Radio Sonata
Radio *Streaming*	Menu	Adalah menu untuk masuk ke halaman radio *streaming* yang di sediakan *server* listen2myradio.
Gambar Gerak	Modul	Adalah gambar untuk menampilkan gambar bergerak
Nama *Crew*	Text	Menampilkan nama *crew*.
Posisi *Crew*	Text	Menampilkan posisi *crew* di Radio Sonata
Alamat Ruman & *Email*	Text	Menampilkan Alamat rumah *crew* dan *email* dari *crew*
Foto *Crew*	Image	Menampilkan Foto dari *Crew* yang bersangkutan
Komunikasi Pesan	Component	Merupakan suatu component untuk mengirim pesan ke *crew*

1.6.5 L5 *Form Login*



Gambar 36. Tampilan *Form Login*

Keterangan :

Tabel 28. Keterangan Tampilan *Form Login*

Tampilan	Jenis	Keterangan
Create An Acount	*Link*	Adalah *link* untuk masuk ke halaman reistrasi
1	Input Text	Untuk memasukan *Username*
2	Input Text	Untuk memasukan Passord
Log in	*Button*	Untuk melakukan *login*

1.6.6 L6 Halaman Registrasi



Gambar 37. Tampilan Halaman Registrasi

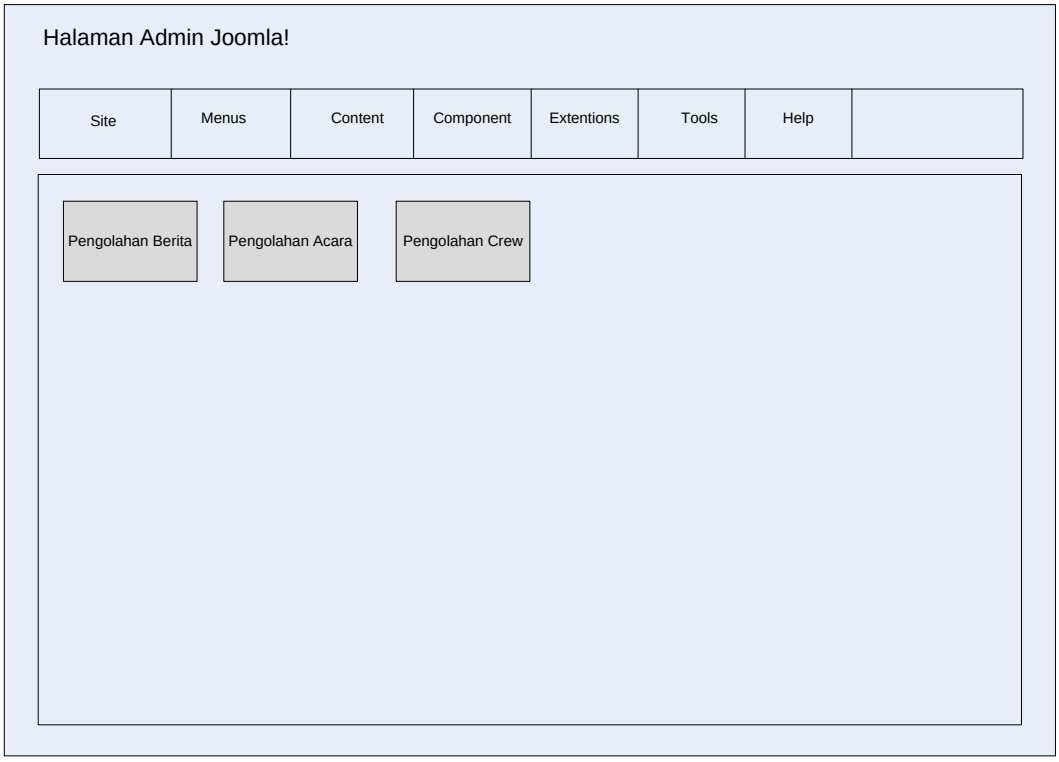
Keterangan :

Tabel 29. Keterangan Tampilan Halaman Registrasi

Tampilan	Jenis	Keterangan
Logo	Image	Gambar Logo Radio Sonata
Home	Menu	Adalah menu untuk masuk ke halaman Utama (*front page*)
Acara	Menu	Adalah menu yang berisi daftar acara yang di siarkan Radio Sonata
Crew	Menu	Adalah menu yang berisi daftar *crew* dari Radio Sonata
Radio *Streaming*	Menu	Adalah menu untuk masuk ke halaman radio *streaming* yang di sediakan *server* listen2myradio.
Gambar Gerak	Modul	Adalah gambar untuk menampilkan gambar bergerak
Form Registrasi	Modul	Adalah modul untuk melakukan registrasi menjadi member
1	Input Text	Untuk mengisikan nama

2	Input Text	Untuk memasukan *username*
3	Input Text	Untuk memasukan *Email*
4	Input Text	Untuk memasukan *Password*
5	Input Text	Untuk memasukan Verify *Password*
Register	*Button*	Untuk meyutujui melakukan registrasi

1.6.7 L7 Halaman *Administrator*



Gambar 38. Tampilan Halaman *Administrator*

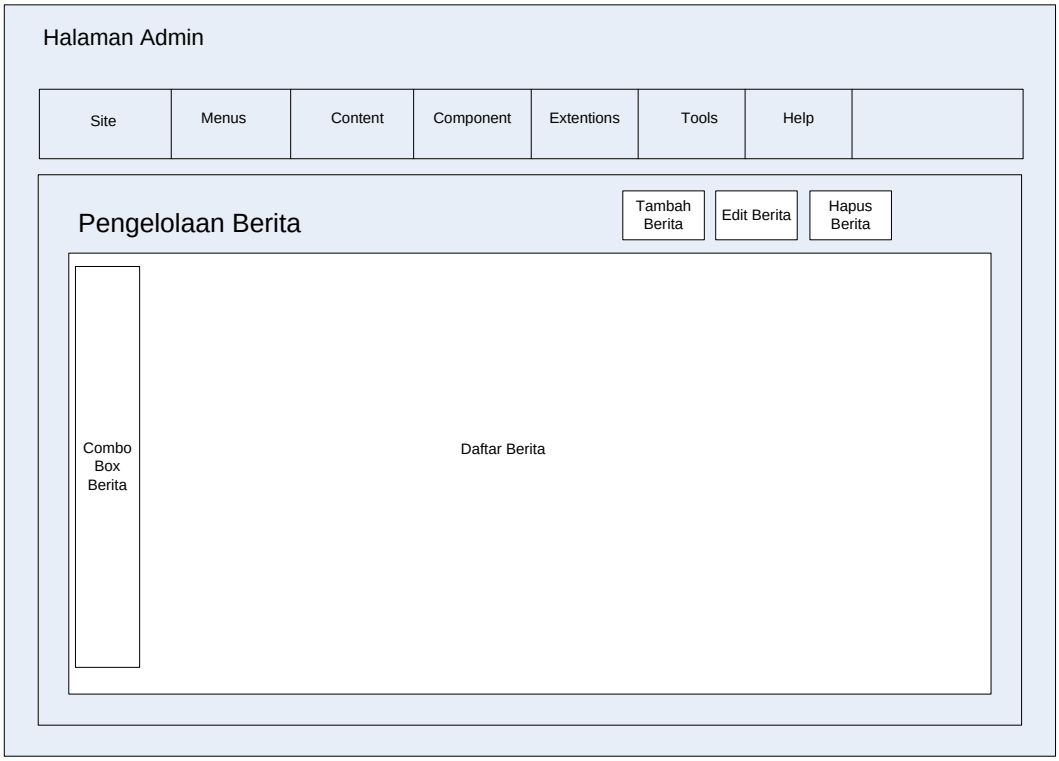
Keterangan :

Tabel 30. Keterangan Tampilan Halaman *Administrator*

Tampilan	Jenis	Keterangan
Site	Menu	Menu *administrator* dari Joomla untuk mengkonfigurasi website
Menus	Menu	Menu *administrator* bawaan Joomla untuk mengelola Menu
Content	Menu	Menu *administrator* bawaan Joomla untuk mengelola content
Component	Menu	Menu *administrator* bawaan Joomla untuk mengelola Extension Joomla

		seperti modul
Tool	Menu	Menu *administrator* bawaan Joomla untuk mengelola Tools Joomla.
Help	Menu	Menu *administrator* bawaan Joomla untuk bantuan
Pengelolaan Acara	*Link*	*Link* untuk masuk ke pengelolaan acara
Pengelolaan Berita	*Link*	*Link* untuk masuk ke pengelolaan berita
Pengelolaan *Crew*	*Link*	*Link* Untuk masuk ke pengelolaan *crew*

1.6.8 L8 Halaman Pengelolaan Berita



Gambar 39. Halaman Pengelolaan Berita

Keterangan :

Tabel 31. Keterangan Tampilan Halaman Pengelolaan Berita

Tampilan	Jenis	Keterangan
Site	Menu	Menu *administrator* dari Joomla untuk mengkonfigurasi website

Menus	Menu	Menu *administrator* bawaan Joomla untuk mengelola Menu
Content	Menu	Menu *administrator* bawaan Joomla untuk mengelola content
Component	Menu	Menu *administrator* bawaan Joomla untuk mengelola Extension Joomla seperti modul
Tool	Menu	Menu *administrator* bawaan Joomla untuk mengelola Tools Joomla.
Help	Menu	Menu *administrator* bawaan Joomla untuk bantuan
Tambah Berita	*Button*	*Link* untuk masuk Halaman tambah berita
Edit Berita	*Button*	*Link* untuk halaman edit berita
Hapus Berita	*Button*	*Button* untuk menghapus berita
Combo box berita	*Combo box*	*Combo box* untuk memilih berita mana yang akan di edit atau di hapus.

1.6.9 L9 Tambah Berita dan Edit Berita



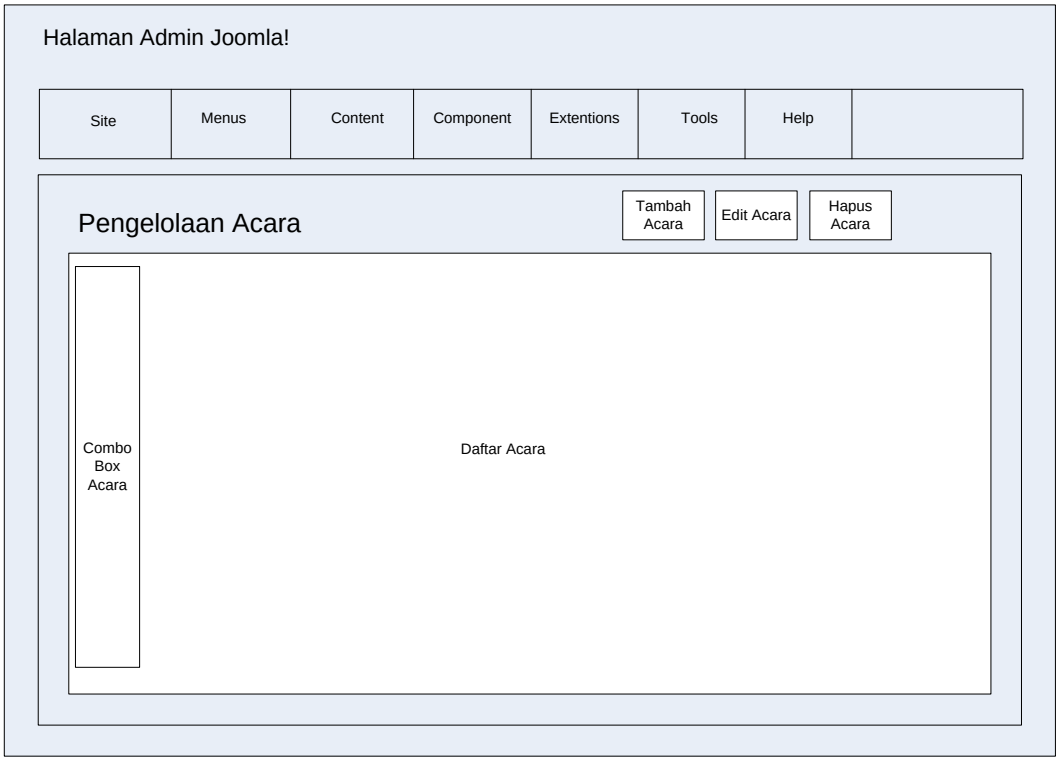
Gambar 40. Tambah Berita dan Edit Berita

Keterangan :

Tabel 32. Keterangan Halaman Tambah Berita dan Edit Berita

Tampilan	Jenis	Keterangan
Site	Menu	Menu *administrator* dari Joomla untuk mengkonfigurasi website
Menus	Menu	Menu *administrator* bawaan Joomla untuk mengelola Menu
Content	Menu	Menu *administrator* bawaan Joomla untuk mengelola content
Component	Menu	Menu *administrator* bawaan Joomla untuk mengelola Extension Joomla seperti modul
Tool	Menu	Menu *administrator* bawaan Joomla untuk mengelola Tools Joomla.
Help	Menu	Menu *administrator* bawaan Joomla untuk bantuan
Save	*Button*	Sebuah tombol untuk menyetujui menyimpan berita
Cancel	*Button*	Sebuah tombol untuk membatalkan penyimpanan berita
Isi Berita	*Text area*	Isi dari berita

1.6.10 L10 Pengelolaan Acara



Gambar 41. Halaman Pengelolaan Acara

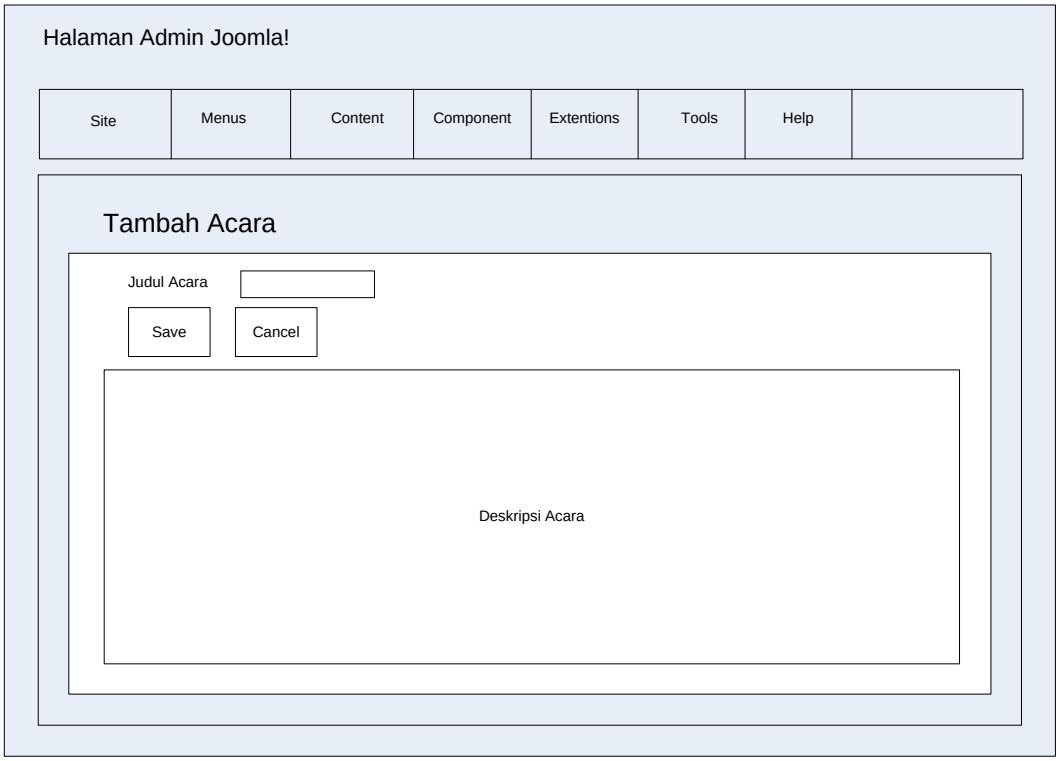
Keterangan :

Tabel 33. Keterangan Halaman Pengelolaan Acara

Tampilan	Jenis	Keterangan
Site	Menu	Menu *administrator* dari Joomla untuk mengkonfigurasi website
Menus	Menu	Menu *administrator* bawaan Joomla untuk mengelola Menu
Content	Menu	Menu *administrator* bawaan Joomla untuk mengelola content
Component	Menu	Menu *administrator* bawaan Joomla untuk mengelola Extension Joomla seperti modul
Tool	Menu	Menu *administrator* bawaan Joomla untuk mengelola Tools Joomla.
Help	Menu	Menu *administrator* bawaan Joomla untuk bantuan
Tambah Acara	*Button*	*Link* untuk masuk Halaman tambah Acara

Edit Acara	*Button*	*Link* untuk halaman edit acara
Hapus Berita	*Button*	*Button* untuk menghapus berita
Combo box Acara	*Combo box*	*Combo box* untuk memilih acara mana yang akan di edit atau di hapus.

1.6.11 L11 Tambah dan Edit Acara



Gambar 42. Halaman Tambah dan Edit Acara

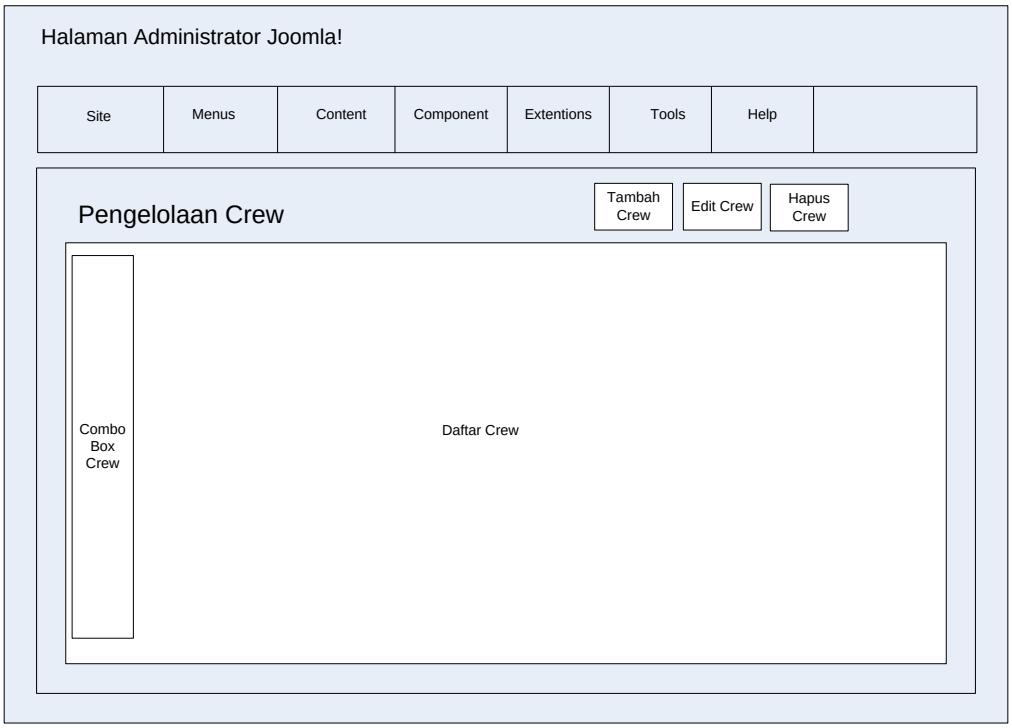
Keterangan :

Tabel 34. Keterangan Halaman Tambah dan Edit Acara

Tampilan	Jenis	Keterangan
Site	Menu	Menu *administrator* dari Joomla untuk mengkonfigurasi website
Menus	Menu	Menu *administrator* bawaan Joomla untuk mengelola Menu
Content	Menu	Menu *administrator* bawaan Joomla untuk mengelola content
Component	Menu	Menu *administrator* bawaan Joomla untuk mengelola Extension Joomla seperti modul

Tool	Menu	Menu *administrator* bawaan Joomla untuk mengelola Tools Joomla.
Help	Menu	Menu *administrator* bawaan Joomla untuk bantuan
Save	*Button*	Sebuah tombol untuk menyetujui menyimpan acara
Cancel	*Button*	Sebuah tombol untuk membatalkan penyimpanan acara
Deskripsi Acara	*Text area*	Deskripsi dari masing-masing acara

1.6.12 L12 Pengelolaan *Crew*



Gambar 43. Halaman Pengelolaan *Crew*

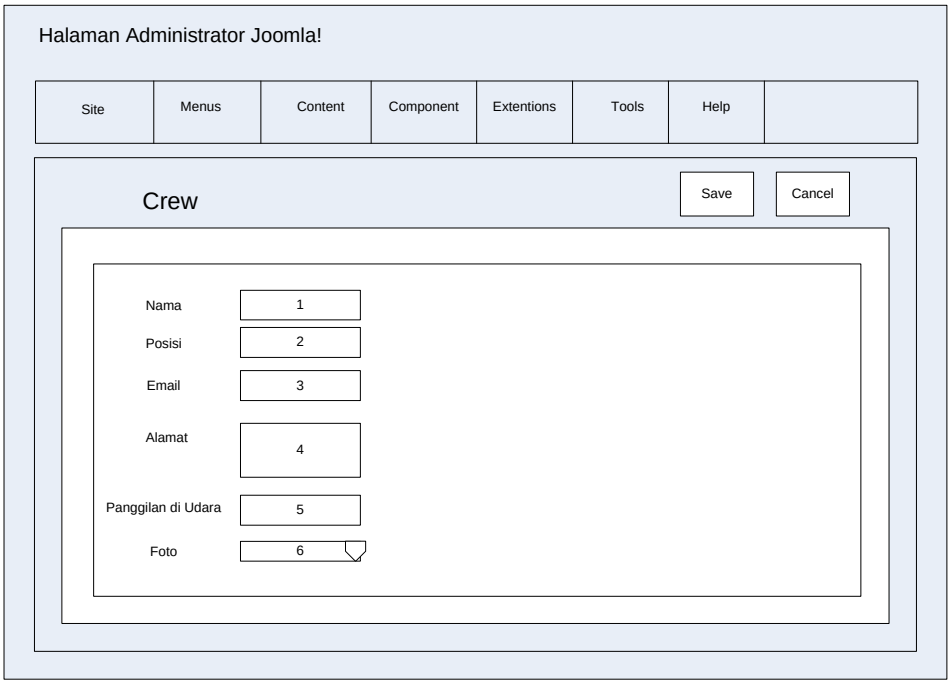
Keterangan :

Tabel 35. Keterangan Halaman Pengelolaan *Crew*

Tampilan	Jenis	Keterangan
Site	Menu	Menu *administrator* dari Joomla untuk mengkonfigurasi website
Menus	Menu	Menu *administrator* bawaan Joomla untuk mengelola Menu
Content	Menu	Menu *administrator* bawaan Joomla

		untuk mengelola content
Component	Menu	Menu *administrator* bawaan Joomla untuk mengelola Extension Joomla seperti modul
Tool	Menu	Menu *administrator* bawaan Joomla untuk mengelola Tools Joomla.
Help	Menu	Menu *administrator* bawaan Joomla untuk bantuan
Tambah *Crew*	*Button*	*Link* untuk masuk Halaman tambah *Crew*
Edit *Crew*	*Button*	*Link* untuk halaman edit *crew*
Hapus *crew*	*Button*	*Button* untuk menghapus *crew*
Combo box crew	*Combo box*	*Combo box* untuk memilih *crew* mana yang akan di edit atau di hapus.

1.6.13 L13 Tambah dan Edit *Crew*



Gambar 44. Halaman Tambah dan Edit *Crew*

Keterangan :

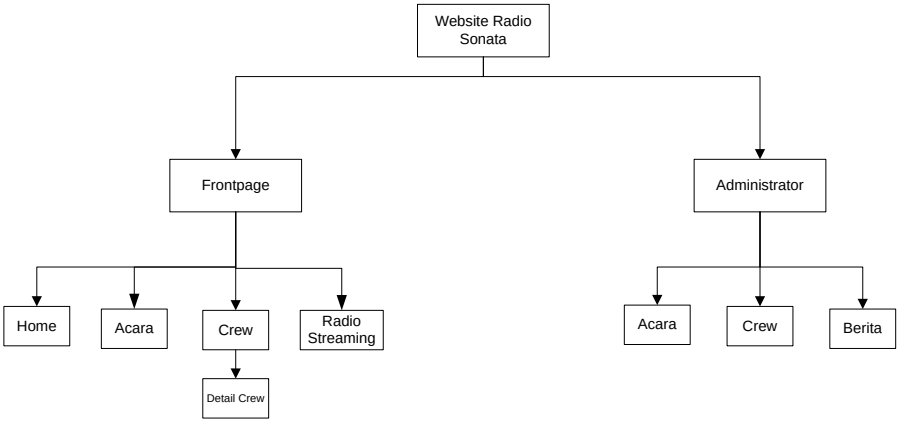
Tabel 36. Keterangan Halaman Tambah dan Edit *Crew*

Tampilan	Jenis	Keterangan
Site	Menu	Menu *administrator* dari Joomla untuk mengkonfigurasi website

Menus	Menu	Menu *administrator* bawaan Joomla untuk mengelola Menu
Content	Menu	Menu *administrator* bawaan Joomla untuk mengelola content
Component	Menu	Menu *administrator* bawaan Joomla untuk mengelola Extension Joomla seperti modul
Tool	Menu	Menu *administrator* bawaan Joomla untuk mengelola Tools Joomla.
Help	Menu	Menu *administrator* bawaan Joomla untuk bantuan
Save	*Button*	Sebuah tombol untuk menyetujui menyimpan acara
Cancel	*Button*	Sebuah tombol untuk membatalkan penyimpanan acara
1	Input Text	Untuk menginputkan nama *crew*
2	Input Text	Untuk menginputkan posisi *crew* di radio sonata
3	Input Teks	Untuk menginputkan *email crew*
4	Teks area	Untuk menginputkan alamat *crew*
5	Input Teks	Untuk menginputkan nama *crew* di udara waktu Onair

1.7 Struktur Menu

Setelah melakukan proses analisis dan merancang Basis data, maka dalam tahap ini, dilakukan perancangan menu sistem. Perancangan menu sistem mendeskripsikan fungsi-fungsi sistem apa saja yang tersedia bagi masing-masing pengguna, sesuai dengan kebutuhan pengguna.



Gambar 45. Struktur Menu Website Sonata

1.8 Jaringan Semantik

Jaringan semantik pada sistem Website Radio Sonata adalah sebagai berikut :

a) F01 : Home (Frontpage)

b) F02 : Acara

c) F03 : *Crew*

d) F04 : Radio *Streaming*

e) F05 : Detail *Crew*

f) F06 : Halaman *Administrator*

g) F07: Acara *Administrator*

h) F08: Berita *Administrator*

i) F09: *Crew administrator*

j) F10 : *Login*

k) F11: Komunikasi Pesan

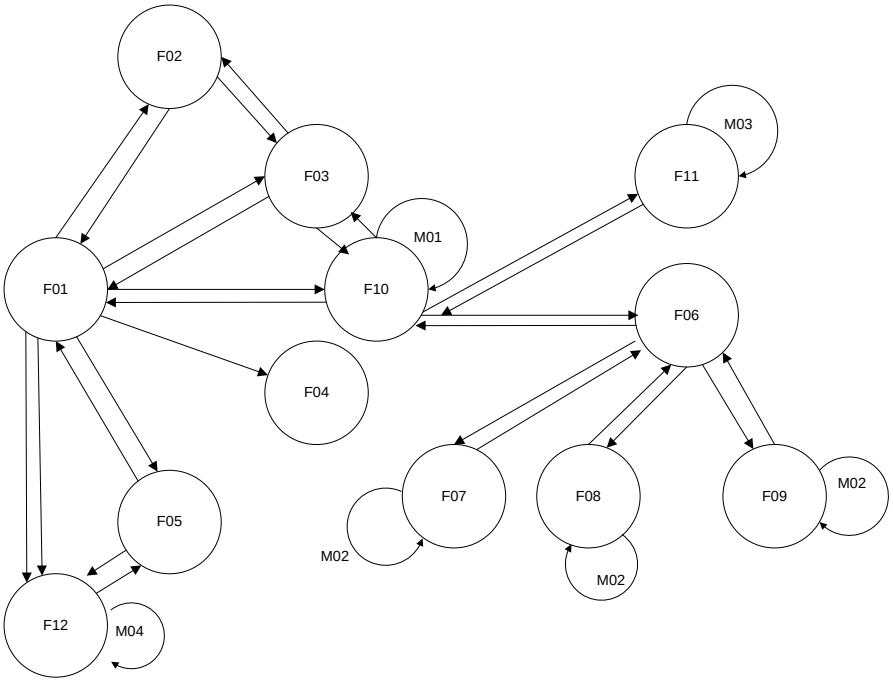
l) F12 : *Form* Pendafraran

m) M01 : pesan *login* valid atau tidak

n) M02 : pesan hasil proses, tambah, edit, hapus

o) M03 : Pesan terkirim atau tidak

p) M04 : Pesan berhasil atu tidak melakukan pendaftaran



Gambar 46. Jaringan Semantik

1.9 Implementasi

Tahap implementasi sistem merupakan tahap menterjemahkan perancangan Radio *Streaming* Sonata Berbasis Website berdasarkan hasil analisis dalam bahasa yang dapat dimengerti oleh mesin serta penerapan perangkat lunak pada keadaan yang sesungguhnya. Seluruh kode program Radio *Streaming* Sonata Berbasis Website menggunakan *Joomla!*.

1.9.1 Perangkat Lunak Pendukung

Perangkat lunak yang dipasang pada sistem komputer yang digunakan untuk membangun Radio *Streaming* Sonata Berbasis Website ini adalah sebagai berikut :

1. Sistem Operasi Windows Xp.
2. Joomla!

1.9.2 Kebutuhan Perangkat Keras

Kebutuhan minimal perangkat keras (*hardware*) yang diperlukan untuk mengimplementasikan program aplikasi ini adalah perangkat keras komputer PC kompatibel dengan spesifikasi yang tersebut dibawah ini. Semakin tinggi spesifikasi komputer semakin baik untuk menjalankan program apilkasi ini :

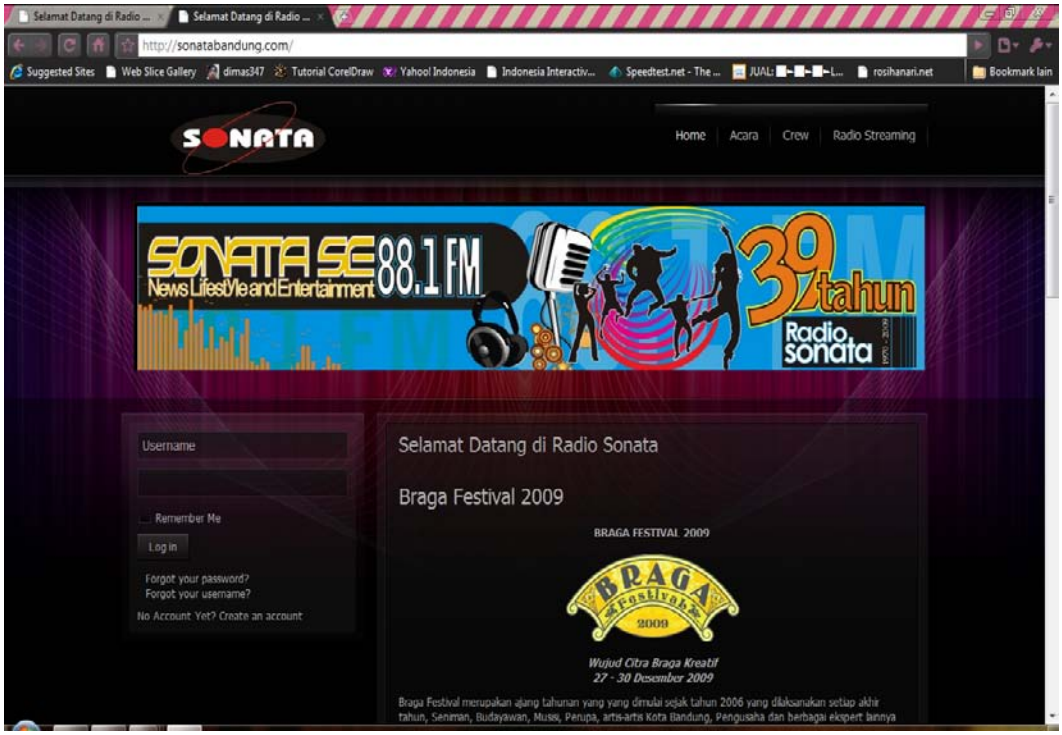
a. Processor Pentium II sampai dengan Pentium IV
b. RAM sebesar 128 MB
c. Hard disk space minimum 500 Mb
d. Lan Card
e. Monitor SVGA

1.9.3 Implementasi Prosedur dan Data

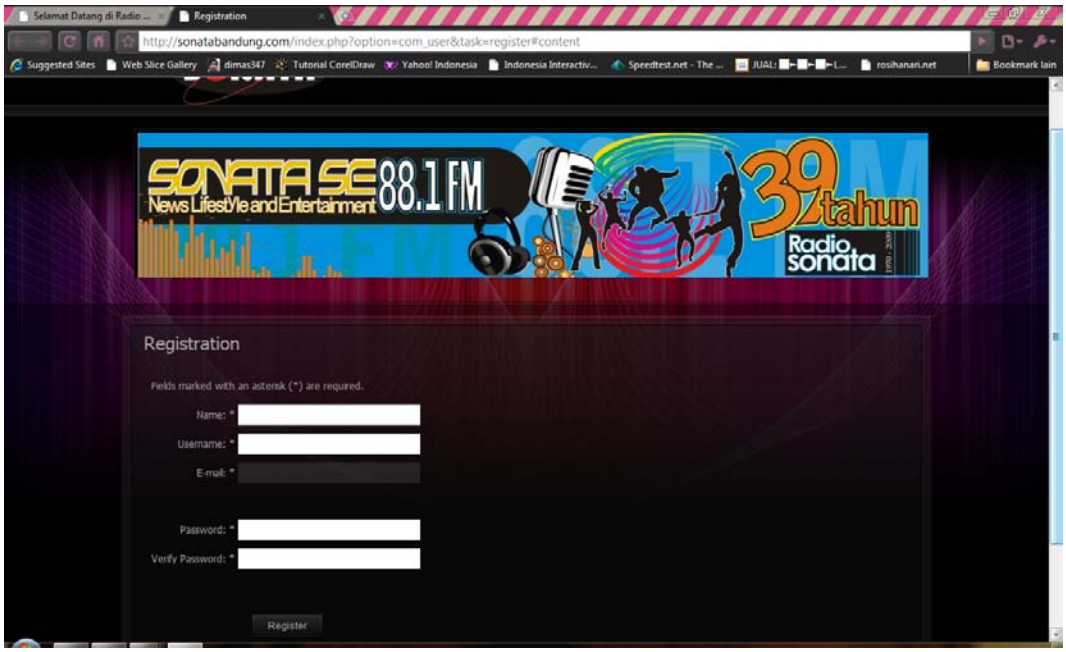
Dalam pembuatan Radio *Streaming* Sonata Berbasis Website menggunakan CMS dari *Joomla!* untuk medesain setiap tampilan, *module* dan *componen*t.

1.9.4 Implementasi Antarmuka

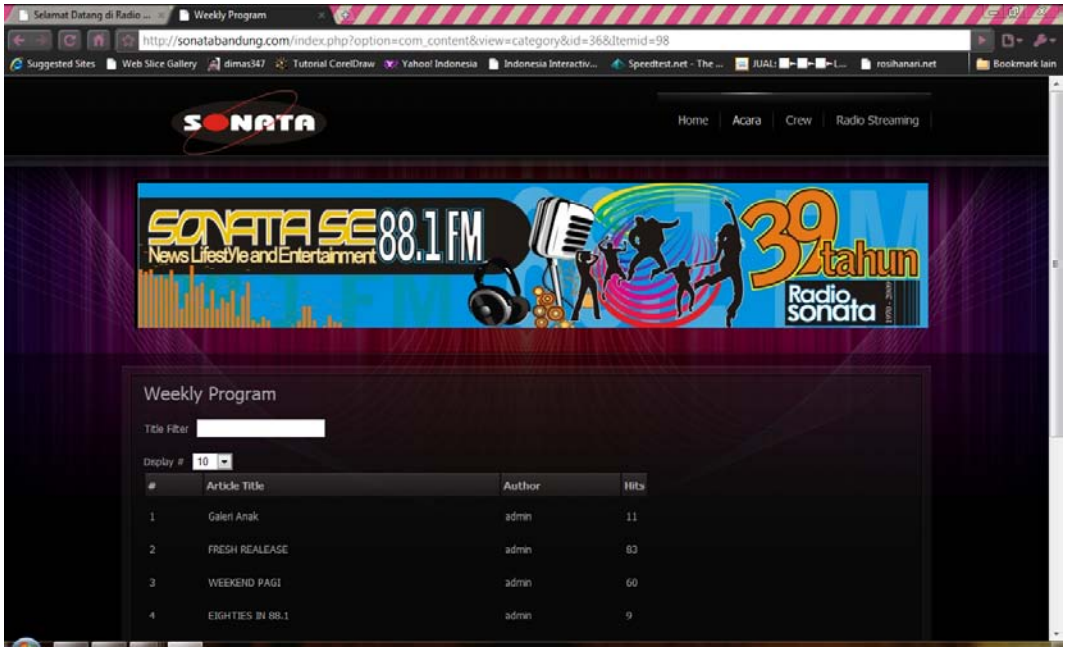
Berikut ini beberapa contoh tampilan antarmuka pengguna yang telah dibuat dengan CMS dari *Joomla!*.



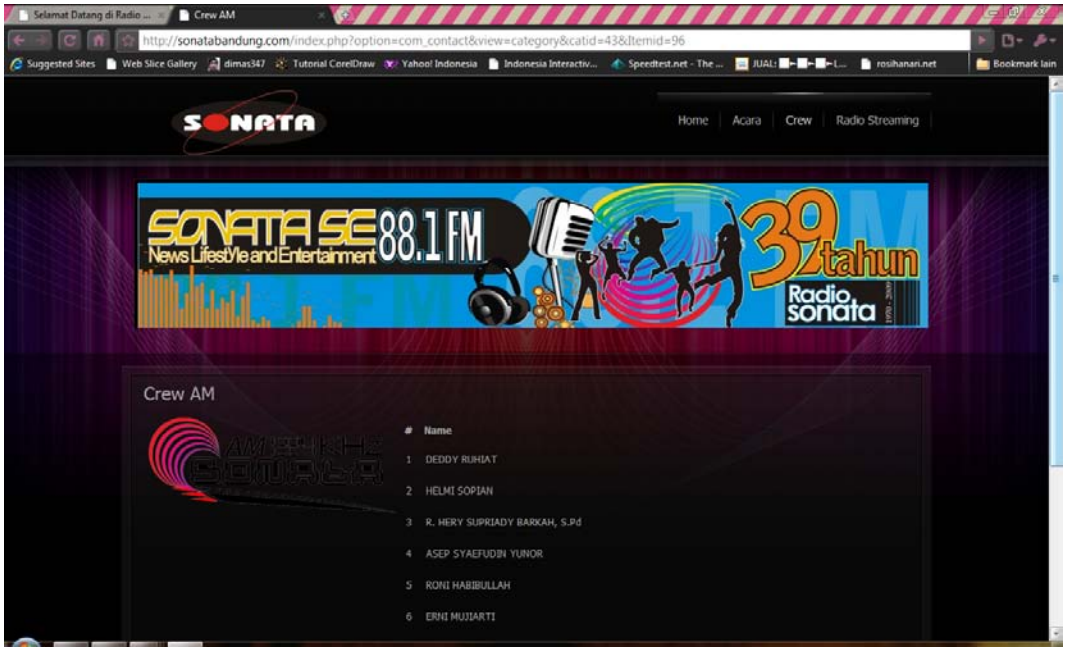
Gambar 47. Tampilan Halaman Home



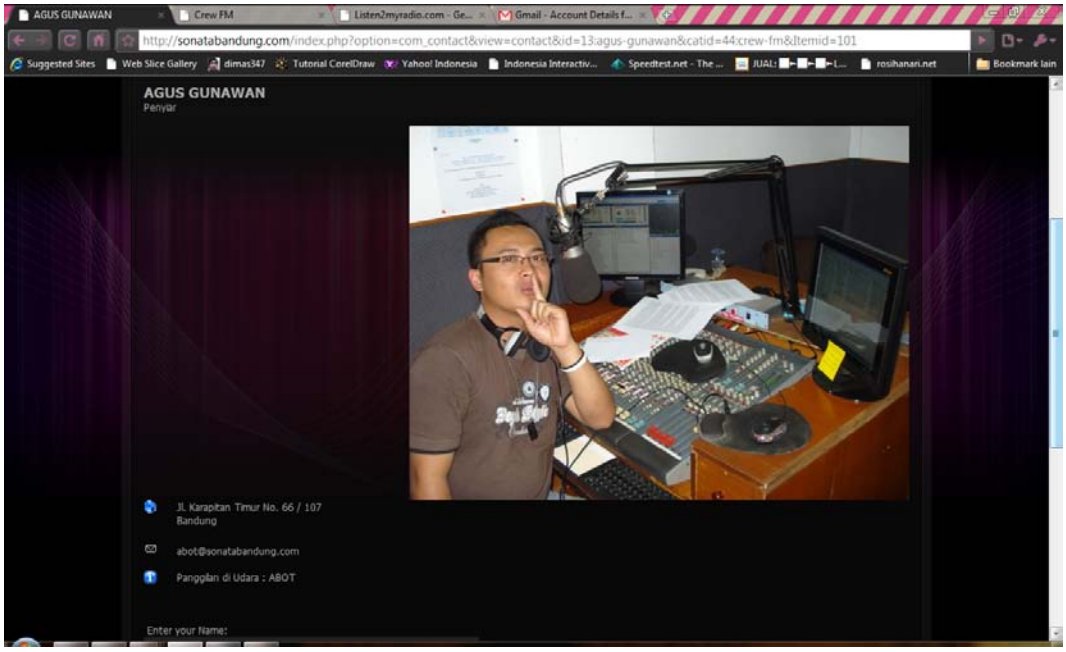
Gambar 48. Tampilan Halaman Registrasi member



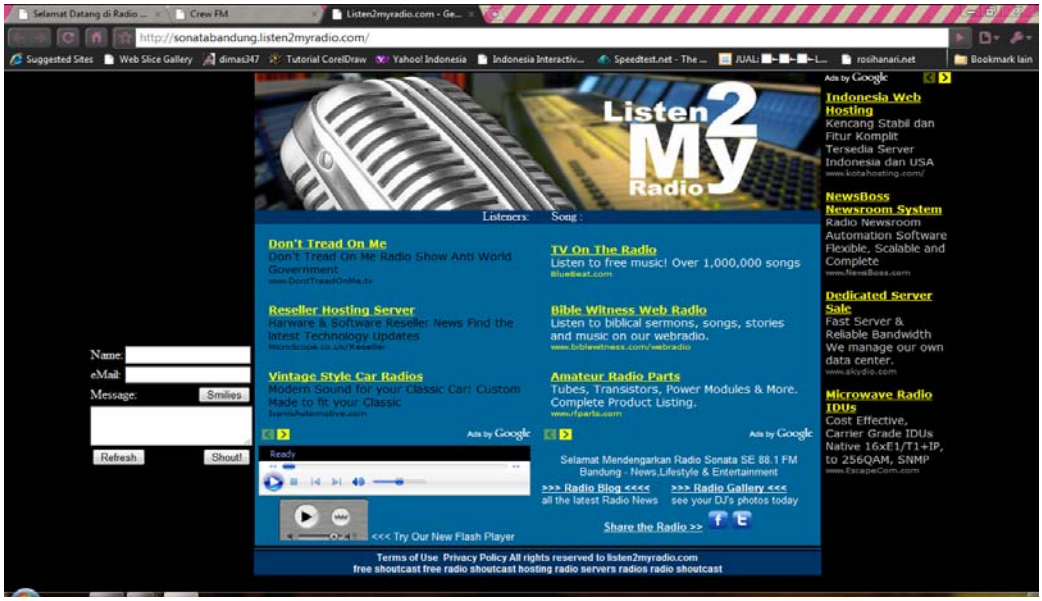
Gambar 49. Tampilan Halaman Acara



Gambar 50. Tampilan Halaman *Crew*



Gambar 51. Tampilan Halaman Detail *Crew*



Gambar 52. Tampilan Halaman Radio *Streaming*

1.10 Pengujian

Proses pengujian adalah mencoba program dengan memasukan data kedalam *form* - *form* masukan yang telah disediakan. Pada tahap ini merupakan kelanjutan dari tahap implementasi yaitu melakukan pengujian - pengujian terhadap aplikasi yang dibangun.

Tabel 37. Rencana Pengujian Aplikasi Radio *Streaming* Sonata Berbasis Website

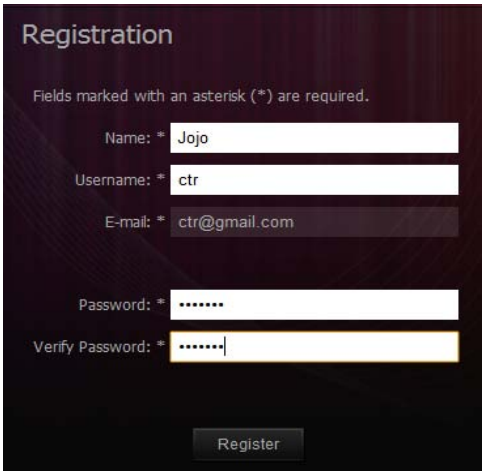
Kelas Uji	Butir Uji	Jenis Pengujian
Registrasi	Validasi	*Black Box*
Login	Verifikasi *user id, password*	*Black Box*
Komunikasi Pesan	validasi	*Black Box*
Tambah Berita	Validasi	*Black Box*
Tambah Acara	Validasi	*Black Box*

1.10.1 Kasus dan Hasil Pengujian Alpha

3.10.1.1 **Pengujian Registrasi**

Pengujian Registrasi Validasi pada setiap *textbox*

a. Data Normal



Gambar 53. *Form* Registrasi Data Normal



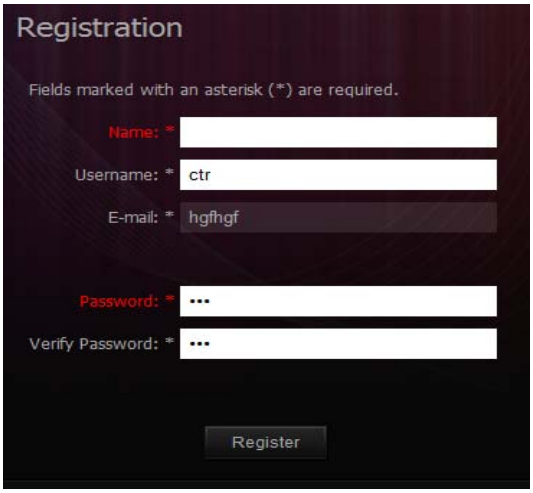
Gambar 54. Sukses Registrasi

Keterangan :

Tabel 38. Keterangan Hasil Uji (Data Normal)

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nama : jojo *Username* : ctr *Email* : ctr@gmail.com *Password* : 1234567 *Verify Password* : 1234567	Data harus terisi pada *textbox*. Data *Email* harus sesuai dengan penulisan *Email*. Penulisan *Password* dan *verify password* minimal 4 character	Dapat mengisi penulisan *Email*, *password* dan verify *password*	Diterima
Klik tombol Register	Data nama , *username*, *email* dan *password* berhasil ditambahkan	Tombol *register* dapat berfungsi. Sesuai yang diharapkan.	Diterima

b. Data Salah



Gambar 55. *Form* Registrasi Salah

Keterangan :

Tabel 39. Keterangan Hasil Uji (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nama : *Username* : ctr *Email* : hgfhgf *Password* : 123 *Verify Password* : 123	Tidak dapat registrasi dan memberikan tanda merah pada setiap label yang tidak sesuai dengan validasi	User tidak dapat registrasi dan diberikan tanda merah pada setiap label	Diterima

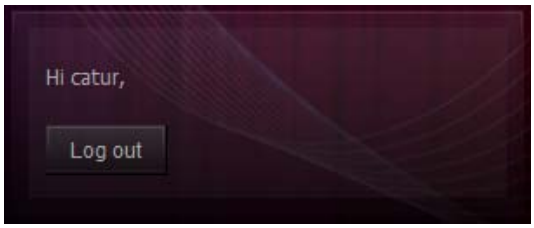
3.10.1.2 **Pengujian *Login***

Pengujian *login* verifikasi *user id, password.*

a. Data Normal



Gambar 56. *Form Login*



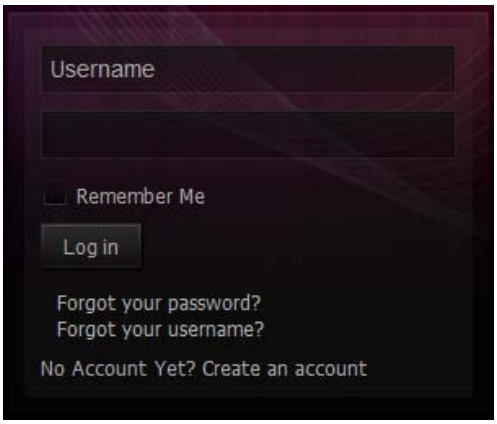
Gambar 57. *Login* sukses

Keterangan :

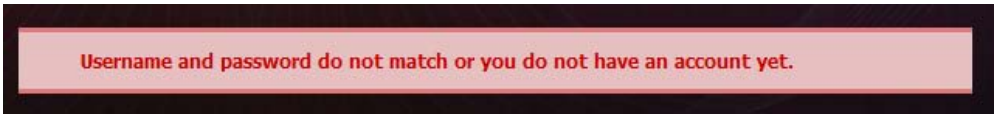
Tabel 40. Hasil Uji (Data Normal)

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nama User : Catur *Password* : 1234	Tercantum pada *textbox user id* dan *textbox password*	Dapat mengisi *login user id* dan *password*. Sesuai yang diharapkan.	Diterima
Klik tombol *login*	Data nama *user* dan *password* dicari di tabel pengguna	Tombol *login* dapat berfungsi. Sesuai yang diharapkan.	Diterima

b. Data Salah



Gambar 58. *Form Login* Salah



Gambar 59. Tampilan Jika *Login* Salah

Keterangan :

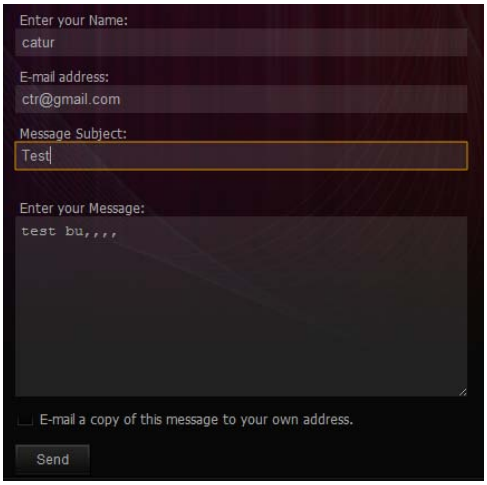
Tabel 41. Hasil Uji (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data *user id* dan *password* tidak terdaftar	Tidak dapat *login* dan menampilkan pesan "*Username and password do not match or you do not have an account yet....*"	User tidak dapat *login* dan memberikan pesan "*Username and password do not match or you do not have an account yet....*"	Diterima

3.10.1.3 **Pengujian Komunikasi Data**

Pengujian komunikasi pesan validasi pada setiap *textbox*:

a. Data Normal



Gambar 60. *Form* Komunikasi Pesan



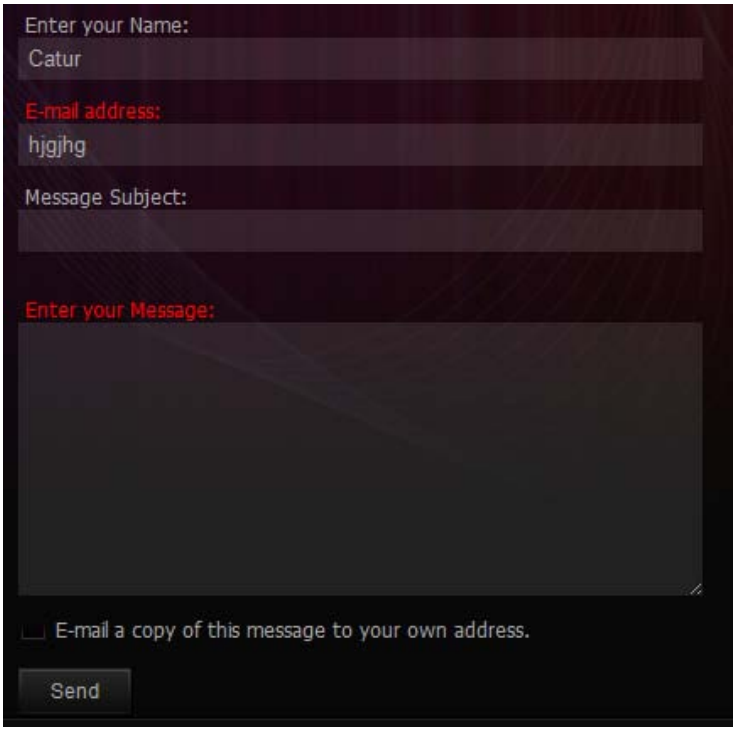
Gambar 61. Sukses mengrim komunikasi pesan

Keterangan :

Tabel 42. Hasil Uji (Data Normal)

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Name : Catur *Email* : ctr@gmail.com *Message Subject : Test* *Message :* Test bu,,,,	Data *Email* dan Message harus terisi. Data *Email* harus sesuai dengan penulisan *Email*.	Dapat mengisi penulisan *Email* dan *Message* yang akan dikirim	Diterima
Klik tombol Send	Mengirimkan pesan	Tombol *Send* dapat berfungsi. Sesuai yang diharapkan.	Diterima

b. Data Salah



Gambar 62. *Form* komunikasi pesan salah

Keterangan :

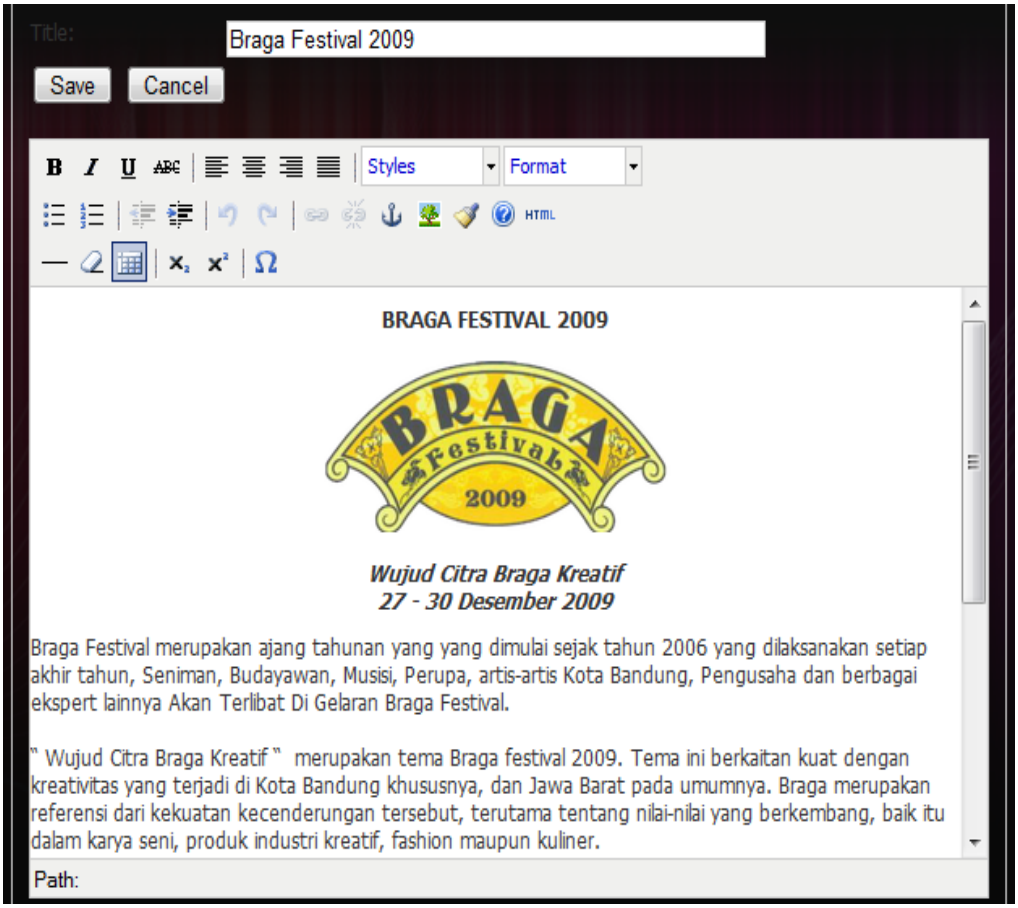
Tabel 43. Hasil Uji (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Nama : Catur *Email* : hgfhgf *Message Subject :Test Message :*	Tidak dapat mengirimkan pesan dan memberikan tanda merah pada setiap label yang tidak sesuai dengan validasi	User tidak dapat mengrimkan pesan dan diberikan tanda merah pada setiap label	Diterima

3.10.1.4 **Pengujian Tambah Berita**

Pengujian tambah berita pada pengisian judul di *textbox* title dan keberhasilan penambahan berita:

a. Data Normal



Gambar 63. Form tambah berita



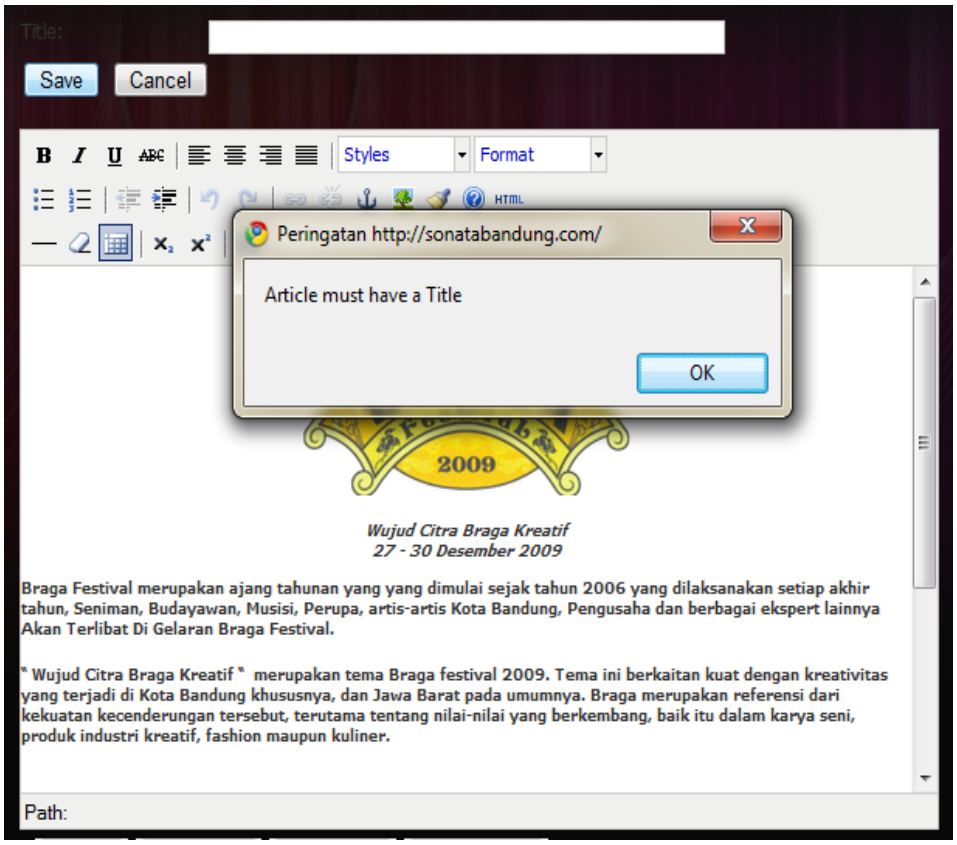
Gambar 64. Sukses tambah berita

Keterangan :

Tabel 44. Hasil Uji (Data Normal)

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Title : Braga Festival 2009 Article : berisikan artikel atau berita yang akan disimpan	Data terisi pada *textbox* terutama pada title tidak boleh kosong.	Dapat mengisi title dan menambahkan berita ke dalam *database*	Diterima
Klik tombol save	Data title dan berita akan ditambahkan kedalam *database*.	Tombol *save* dapat berfungsi. Sesuai yang diharapkan.	Diterima

b. Data salah



Gambar 65. Form Tambah Berita Salah

Keterangan :

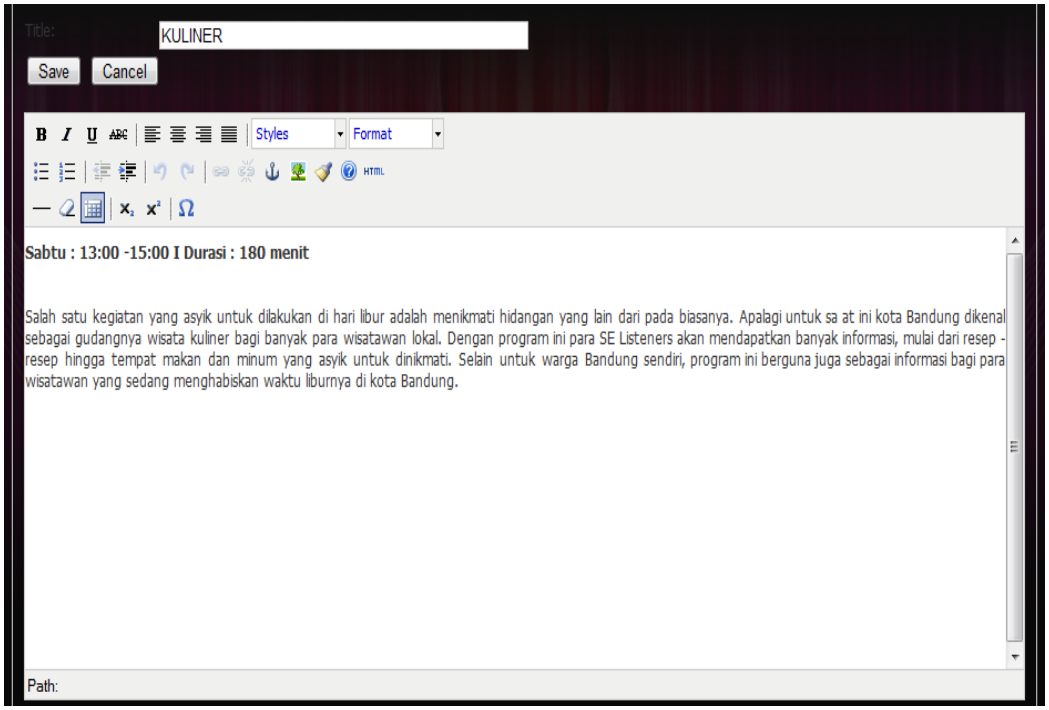
Tabel 45. Data Salah

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Title : Article : berisikan artikel atau berita yang akan disimpan	Tidak dapat menambahkan berita dan memberikan tanda pesan yeng berbunyi" *Article must have a title* "	*admin* tidak dapat menambahkan berita dan diberikan tanda pesan	Diterima

3.10.1.5 Pengujian Tambah Acara

Pengujian Tambah Acara pada pengisian judul di *textbox* title dan keberhasilan penambahan berita:

a. Data Normal



Gambar 66. Form Tambah Acara



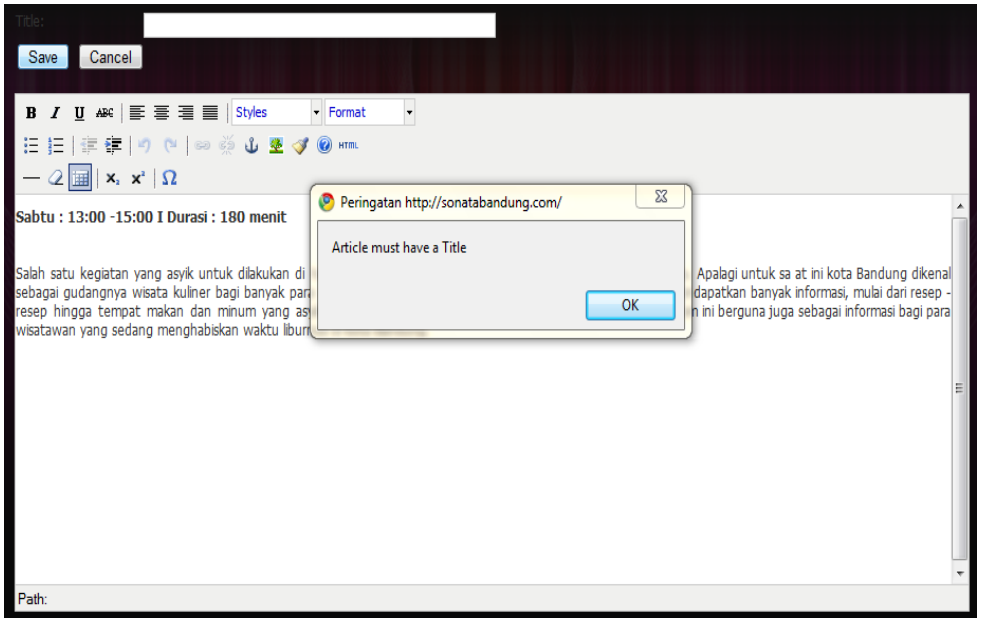
Gambar 67. Sukses Tambah Acara

Keterangan :

Tabel 46. Hasil Uji (Data Normal)

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Title : Acara Article : berisikan jadwal acara yang akan disimpan	Data terisi pada *textbox* terutama pada title tidak boleh kosong.	Dapat mengisi title dan menambahkan jadwal acara ke dalam *database*	Diterima
Klik tombol save	Data title dan Jadwal acara akan ditambahkan kedalam *database*.	Tombol *save* dapat berfungsi. Sesuai yang diharapkan.	Diterima

b. Data salah



Gambar 68. Form Tambah Berita Salah

Keterangan :

Tabel 47. Hasil Uji (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Title : Article : berisikan artikel atau jadwal acara yang akan disimpan	Tidak dapat menambahkan jadwal acara dan memberikan tanda pesan yeng berbunyi" *Article must have a title "*	*admin* tidak dapat menambahkan jadwal acara dan diberikan tanda pesan	Diterima