BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis

3.1.1 Analisis Masalah

Permasalahan pada penelitian ini adalah masih banyak *event organizer* yang masih mengumumkan *event* secara manual, baik pengumuman yang ditempel di tempat tertentu atau dengan menyewa *space iklan* yang memakan biaya cukup mahal yang menyebabkan penyebaran informasi yang kurang luas sehingga orang-orang kesulitan untuk mengakses pengumuman tersebut.

Serta bagaimana mengimplementasikan *web service* dengan android sebagai solusi permaslahan tersebut.

3.1.2 Analisis Kebutuhan Data

Sistem yang akan dibangun merupakan sistem untuk menginformasikan *event-event* yang ada di kota Bandung. Untuk aplikasi *web service*pada *server* dapatmelakukan proses input,edit,dan delete dan untuk *client* pada android dapat melakukan proses pemanggilan method atau fungsi yang ada pada *web service* untuk mendapatkan data *event*. Maka *web service* harus menyediakan data *event-event* yang ada di kota Bandung agar aplikasi android dapat mengakses data tersebut. Sehinggauntuk dibangunnya sistem ini, dibutuhkan data berupa informasi *event* yang ada Bandung yang di kelola oleh dinas pariwisata kota Bandung.

3.1.3 Analisis Kebutuhan Non Fungsional

Kebutuhan non fungsional yang dibutuhkan untuk membangun sistem ini terdiri dari dua hal, yaitu : kebutuhan perangkat keras dan kebutuhan perangkat lunak.

3.1.3.1 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan untuk membangun sistem ini adalah :

1. *Processor* dengan kecepatan minimal 2 GHz+ *Motherboard*
2. Kapasitas *Harddisk* minimal 40 GB
3. RAM minimal 256 MB
4. VGA *Card* minimal 64 MB
5. *Monitor*
6. *Lan Card 10/100Mbps*
7. Koneksi *internet* minimal64kbps
8. *Mouse*
9. *keyboard*

3.1.3.2 Analisis Kebutuhan Perangkat Lunak

Perangakat lunak yang diperlukan untuk membangun sistem ini adalah :

1. Java Development Kit versi 5 atau 6 , untuk kompilasi kode – kode program.
2. Java Runtime Edition, sebagai *platform* untuk menjalankan sistem.
3. Netbeans IDE, sebagai perangkat lunak yang digunakan untuk membangun *web service*.

4. Eclipse 3.5, sebagai perangkat lunak yang digunakan untuk membangun aplikasi *client* pada android.
5. Glassfish, sebagai *webserver* untuk aplikasi web.
6. Android SDK, untuk pengembang aplikasi android.
7. Android Development Tool, sebagai plugin android pada eclipse.
8. Android Virtual *Device*, sebagai emulator untuk menjalankan sistem android.

3.1.4 Analisis Sistem

3.1.4.1 Arsitektur Sistem

Aplikasi utama pada *event* calendar ini merupakan aplikasi web yang dapat digunakan oleh user melalui *internet*. User yang menggunakan aplikasi utama ini dapat dibedakan menjadi dua jenis, yaitu : administrator dan user android. administrator dapat mengakses fungsi-fungsi lain seperti tambah,edit,hapus*event*. Sedangkan user android adalah user yang menggunakan aplikasi pada android.

Selain aplikasi utama, *event* calendar juga mempunyai dukungan web service dalam bentuk *client – server*. Sistem pada android akan menggunakan dukungan *web service* dengan bantuan WSDL yang telah disediakan. Sistem android selanjutnya dapat mengakses fungsi-fungsi *web service* yang disediakan oleh *server* .

3.1.4.2 Analisis Standar *Web service*

Dalam bab landasan teori telah dijelaskan secara singkat ada dua jenis *web service* yang dapat digunakan, yaitu REST dan SOAP. Tugas akhir ini akan

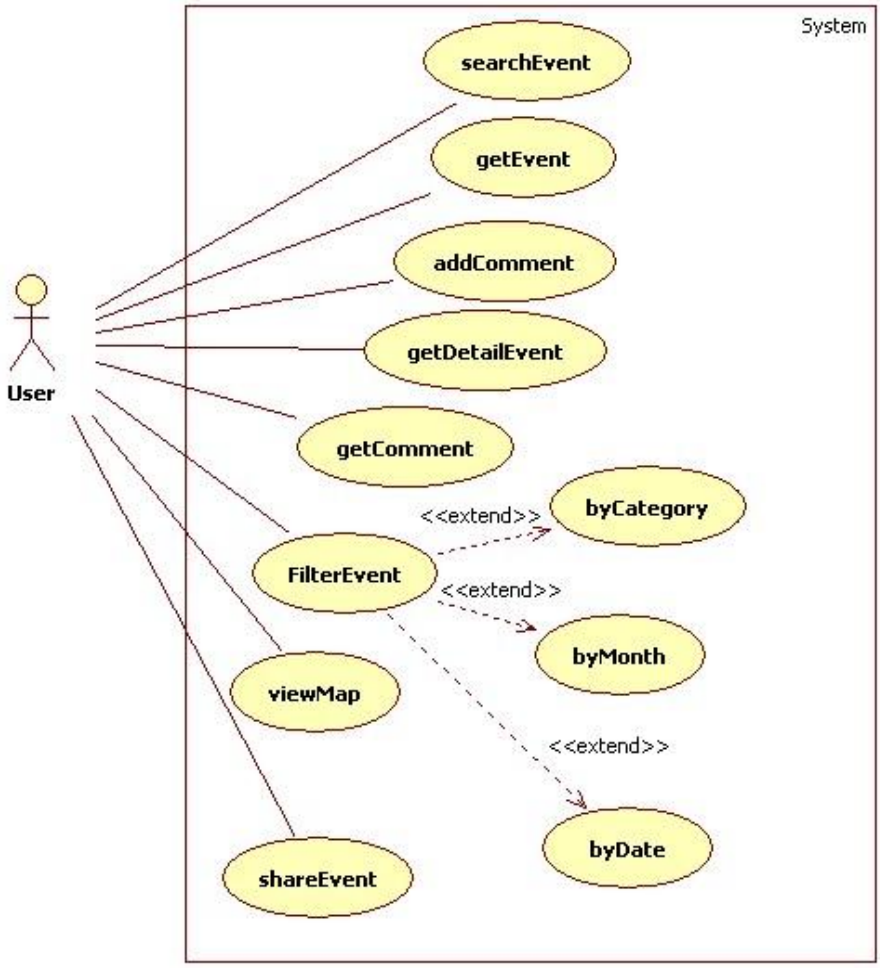
menggunakan *web service*dengan standar SOAP karena pertama, SOAP adalah jenis *web service*yang berorientasi pada *service* yang cocok diterapkan pada aplikasi *event calendar* yang memang menyediakan layanan untuk memanggil metode yang mengolah *resource event*, kedua, SOAP memiliki standar yang baik dan jelas. REST tidak memiliki standar, tetapi hanya sebuah arsitektur *web service*. Untuk aplikasi dimana ia mungkin saja berkomunikasi dengan banyak aplikasi lain yang berbeda bahasa maupun *platform*, standar sangat diperlukan karena menjamin aplikasi-aplikasi ini berbicara dalam bahasa yang sama dan memudahkan pengembangan. Penjabaran SOAP diperjelas dengan disediakannya WSDL, sehingga pengguna *web service*dapat dengan mudah membuat *client*untuk menggunakan SOAP.

3.1.4.3 Analisis WSDL

SOAP sebagai standar pertukaran pesan dalam *web service* mampu berjalan sendiri tanpa bantuan WSDL untuk mendeskripsikannya. Tetapi, hal ini akan menyulitkan pengembangan, karena tanpa bantuan WSDL, para pengembang harus bertanya-tanya seperti apa bentuk pesan yang seharusnya dikirim kepada atau diterima dari penyedia layanan *web service*. Dengan bantuan WSDL, para pengembang telah dibantu karena telah mengetahui seperti apa pesan yang seharusnya dikirim atau diterima. WSDL dalam pembuatannya terlihat sangat menyulitkan dan membingungkan, tetapi dengan bantuan *library* dan *tools* yang telah beredar di *internet*, pembuatannya menjadi mudah dan cepat.

3.1.4.4 Analisis Sistem Pada Aplikasi Android

3.1.4.4.1 Use Case Diagram



Gambar III.1 Use case diagram pada sistem aplikasi pada android

3.1.4.4.1.1 Identifikasi Aktor

Pada tabel 3.1 berisi daftar aktor dan deskripsi peran meraka masing-masing dalam aplikasi ini.

Tabel III.1 Definisi aktor dalam aplikasi android

NO	Aktor	Deskripsi
A-01	User	Merupakan pengguna dari aplikasi event calendar pada android

3.1.4.4.1.2 Identifikasi Use Case

Berdasarkan use case diagram di atas maka di buatlah skenario dari tiap proses yang ada pada use case diagram tersebut.Pada tabel 3.2 menggambarkan skenario mengambil data *event* pada use case getEvent..

Tabel III.2 Skenario Use case getEvent

Identifikasi	
Nomor	UC-AN-01
Nama Use Case	getEvent
Deskripsi	Proses untuk mengambil seluruh data *event* yang ada
Aktor	User
Skenario Utama(UC-AN-01-S01)	
Kondisi awal	Koneksi ke *web service* berhasil
Aksi aktor User	**Respon Sistem**
1. Menjalankan menu utama	2. Memanggil fungsi getEvent
	3. Mengambil *return value* dari *web service* dalam bentuk SOAP
	4. Melakukan parsing data
	5. Menampilkan data hasil parsing
Skenario Alternatif (UC-AN-01-S02) get*Event* namun terdapat pesan error	
	2a. Menampilkan pesan kesalahan, misalnya karena tidak ada koneksi *internet* atau tidak dapat terhubung ke *server*.
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel 3.3 menggambarkan skenario mengambil data *event* berdasarkan id pada use case getDetailEvent.

Tabel III.3skenario use case getDetailEvent.

Identifikasi	
Nomor	UC-AN-02
Nama Use Case	getDetailEvent
Deskripsi	Proses untuk mengambil data *event* berdasarkan id *event*
Aktor	User
Skenario Utama (UC-AN-02-S01)	
Kondisi awal	Koneksi ke *web service* berhasil
Aksi aktor	**Respon Sistem**
1. Memilih event pada listView	2. Menggambil parameter id *event*
	3. Memanggil fungsi getEventById
	4. Mengambil *return value* dari *web service* dalam bentuk SOAP
	5. Melakukan parsing data
	6. Menampilkan data hasil parsing
Skenario Alternatif (UC-AN-02-S01) get*EventByID* namun terdapat pesan error	
	4a. Menampilkan pesan kesalahan, misalnya karena tidak ada koneksi *internet* atau tidak dapat terhubung ke *server*.
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel 3.4 menggambarkan skenario filter data *event* berdasarkan kategori event pada use case byCategory.

Tabel III.4Skenario use case byCategory

Identifikasi	
Nomor	UC-AN-03
Nama Use Case	byCategory
Deskripsi	Proses untuk memfilter data event yang akan di tampilkan berdasarkan kategori event
Aktor	User, web service
Skenario Utama(UC-AN-03-S01)	
Kondisi awal	Koneksi ke *web service* berhasil

Aksi aktor	**Respon Sistem**
1. Memilih menu filter event berdasarkan kategori.	2. Menampilkan form inputan
3. Memilih kategori event akan di tampilkan	4. Memanggil fungsi byCategory
	5. Mengambil *return value* dari *web service* dalam bentuk SOAP
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif (UC-AN-03-S02) byCategory namun terdapat pesan error	
	9.a Menampilkan pesan kesalahan, misalnya karena data yang di filter tidak ada
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel 3.5 menggambarkan skenario filter data *event* berdasarkan bulan tertentu pada use case byMonth.

Tabel III.5 Skenario use case byMonth

Identifikasi	
Nomor	UC-AN-04
Nama Use Case	byMonth
Deskripsi	Proses untuk memfilter data event yang akan di tampilkan berdaskan bulan
Aktor	User
Skenario Utama (UC-AN-04-S01)	
Kondisi awal	Koneksi ke *web service* berhasil
Aksi aktor	**Respon Sistem**
1. Memilih menu filter event berdasarkan bulan.	2. Menampilkan form inputan
3. Memilih nama bulan dan tahun event akan di tampilkan	4. Memanggil fungsi byMonth
	5. Mengambil *return value* dari *web service* dalam bentuk SOAP
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif (UC-AN-04-S02) byMonth namun terdapat pesan error	
	9a. Menampilkan pesan kesalahan, misalnya karena data yang di filter tidak ada.
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel 3.6 menggambarkan skenario filter data *event* berdasarkan tanggal tertentu pada use case byDate.

Tabel III.6 Skenario use case byDate

Identifikasi	
Nomor	UC-AN-05
Nama Use Case	byDate
Deskripsi	Proses untuk menfilter data event yang akan di tampilkan berdaskan tanggal tertentu.
Aktor	User
Skenario Utama(UC-AN-05-S01)	
Kondisi awal	Koneksi ke *web service* berhasil
Aksi aktor	**Respon Sistem**
1. Memilih menu filter event berdasarkan *range* tanggal.	2. Menampilkan form inputan
3. Menginputkan awal tanggal dan akhir tanggal event.	4. Memanggil fungsi byDate
	5. Mengambil *return value* dari *web service* dalam bentuk SOAP
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif(UC-AN-05-S02) byDate namun terdapat pesan error	
	4a. Menampilkan pesan kesalahan, misalnya karena input dari user salah(skenario kembali ke langkah 3)
Skenario Alternatif(UC-AN-05-S03) byDate namun data tidak ditemukan	
	7.a Menampilkan pesan kesalahan, misalnya karena data yang di filter tidak ada.
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel 3.7 menggambarkan skenario memberi komentar *event* pada use case addComment.

Tabel III.7 Skenario use case addComment

Identifikasi	
Nomor	UC-AN-06
Nama Use Case	addComment
Deskripsi	Proses untuk memberikan komentar pada event.
Aktor	User, web service
Skenario Utama(UC-06-S01)	
Kondisi awal	Berada pada halaman detail event
Aksi aktor	**Respon Sistem**
1. Memilih menu komentar	2. Menampilkam halaman komentar
3. Memilih menu tambah komentar	4. Menampilkan form inputan
5. Mengisi form	6. Memanggil fungsi addComment
	7. Memparsing data dari *return value*
	8. Menampilkan pesan komentar berhasil di kirim.
Skenario Alternatif (UC- AN-06-S02) addComment namun terdapat pesan error	
	6a. Menampilkan pesan kesalahan, misalnya karena isi dari form belum lengkap atau salah.(skenario kembali ke langkah 5)
Kondisi akhir	Data berhasil disimpan kedalam database

Pada tabel 3.8 menggambarkan skenario memberi komentar *event* pada use case getComment.

Tabel III.8 Skenario use case getComment

Identifikasi	
Nomor	UC-AN-07
Nama Use Case	getComment
Deskripsi	Proses untuk menampilkan komentar pada event.
Aktor	User
Skenario Utama (UC-AN-07-S01)	
Kondisi awal	Berada pada halaman detail event
Aksi aktor	**Respon Sistem**
1. Memilih menu komentar	2. Memanggil fungsi getComment
	3. Mengambil *return value* dari *web service* dalam bentuk SOAP
	4. Memparsing data dari *return value*
	5. Menampilkan data hasil parsing
Skenario Alternatif(UC-AN-07-S02) addComment namun komentar tidak ada	
	7a. Menampilkan pesan kesalahan, misalnya karena belum ada komentar yang di inputkan
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel 3.9 menggambarkan skenario mencari *event* berdasarkan parameter tertentu pada use case searchEvent.

Tabel III.9 Skenario use case searchEvent

Identifikasi	
Nomor	UC-AN-08
Nama Use Case	searchEvent
Deskripsi	Proses untuk memfilter data event yang akan di tampilkan berdaskan bulan
Aktor	User
Skenario Utama(UC-08-S01)	
Kondisi awal	Koneksi ke *web service* berhasil
Aksi aktor	**Respon Sistem**
1. Memilih menu cari event	2. Menampilkan form inputan
3. Memasukan parameter pencarian	4. Memanggil fungsi searchEvent
	5. Mengambil *return value* dari *web service* dalam bentuk SOAP
	6. Memparsing data dari *return value*
	7. Menampilkan data hasil parsing
Skenario Alternatif (UC-AN-08-S02) searchEvent namun terdapat pesan error	
	4a. Menampilkan pesan kesalahan, inputan pencarian belum diisi atau belum lengkap
Skenario Alternatif (UC-AN-08-S03) searchEvent namun data tidak ditemukan	
	7a. menampilkan pesan kesalahan, misal karena data yang di cari tidak ditemukan.
Kondisi akhir	Data berhasil ditampilkan di android

Pada tabel 3.10 menggambarkan skenario menampilkan peta lokasi event pada use case viewMap.

Tabel III.10 Skenario use case viewMap

Identifikasi		
Nomor	UC-AN-09	
Nama Use Case	viewMap	
Deskripsi	Proses untuk menampilkan peta dari lokasi event	
Aktor	User	
Skenario Utama(UC-AN-09-S01)		
Kondisi awal		Berada pada halaman detail event
Aksi aktor		**Respon Sistem**
1. Memilih menu peta		2. Mengambil latitude dan longitude dari lokasi event.
		3. Mencari posisi latitude dan longitude pada google map
		4. Menampilkan peta
Kondisi akhir	Peta berhasil di tampilkan	

Pada tabel 3.11 menggambarkan skenario untuk *share* event melalui *facebook* pada use case shareEvent.

Tabel III.11 Skenario use case shareEvent

Identifikasi	
Nomor	UC-AN-10
Nama Use Case	shareEvent
Deskripsi	Proses untuk membagikan event melalui facebook
Aktor	User
Skenario Utama(UC-AN-10-S01)	
Kondisi awal	Berada pada halaman detail event
Aksi aktor	**Respon Sistem**
1. Memilih menu *share*	2. Mengambil data event berupa nama,tgl,dan lokasi
	3. Login *facebook*
	4. Menampilkan pesan event berhasil di *posting*
Kondisi akhir	Data event berhasil di posting di facebook

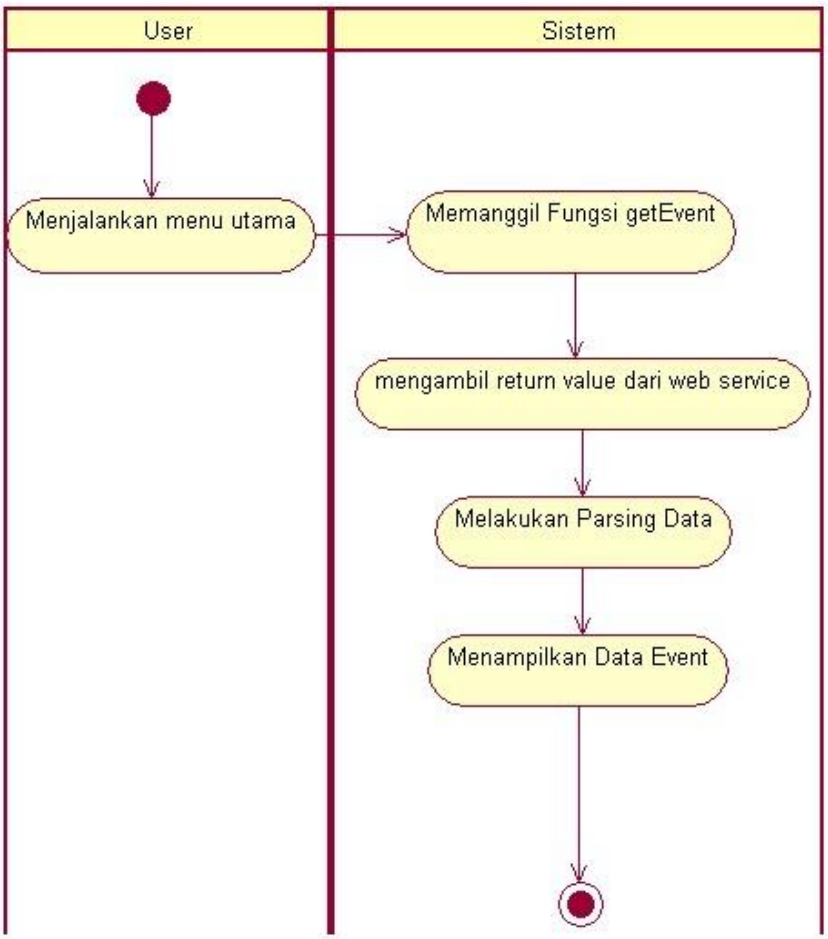
3.1.4.4.2 Activity Diagram

3.1.4.4.2.1 getEvent

Aktifitas getEvent merupakan aktifitas yang dilakukan oleh user untuk mangambil data *event* yang ada pada *database* dengan *web service*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. menjalankan menu utama, pada aktifitas ini user menjalankan menu utama yang ada pada sistem.
2. memanggil fungsi getEvent, pada aktifitas ini sistem memanggil fungsi getEvent yang ada pada *web service*.
3. Mengambil return value dari web service, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh web service berupa hasil dari eksekusi method pada web service.
4. Melakukan parsing data, pada aktifitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
5. Menampilkan data *event*, pada aktifitas ini sistem menampilkan hasil parsing kedalam bentuk list view.

Seluruh aktifitas getEvent terlihat pada *activity diagram* gambar III.2.



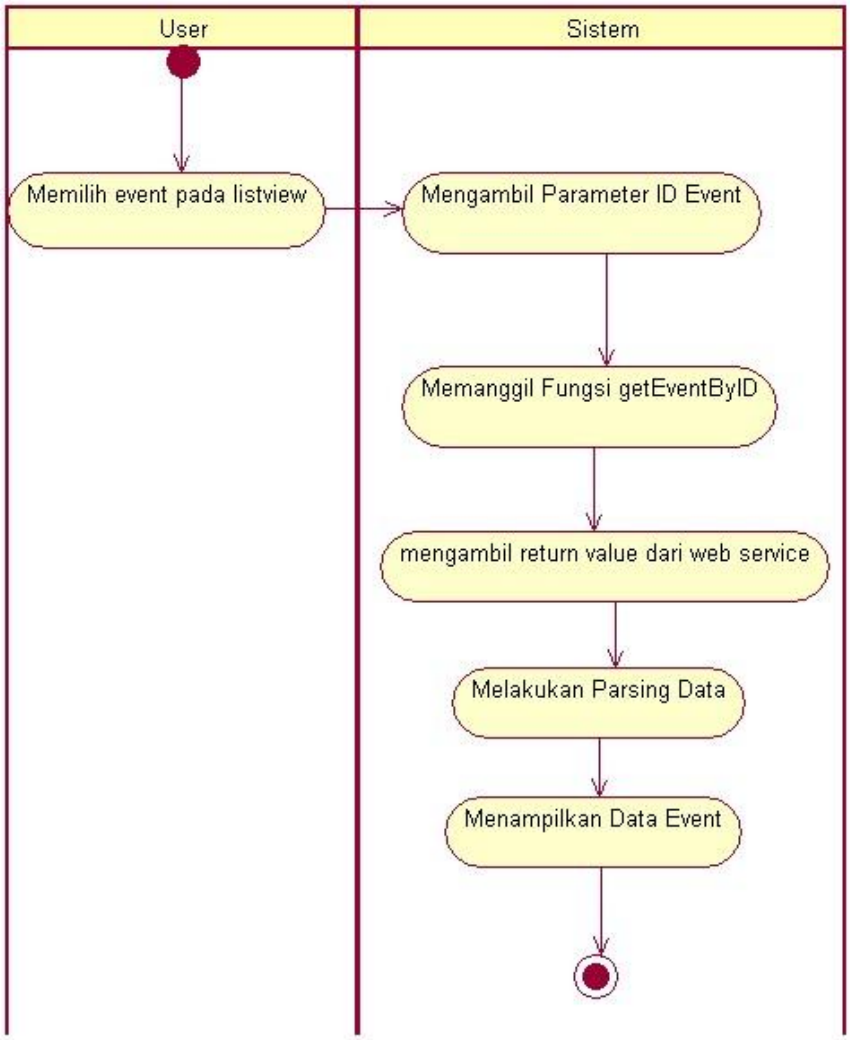
Gambar III.2Activity diagram getEvent

3.1.4.4.2.2 getDetailEvent

Aktifitas getDetailEvent merupakan aktifitas yang dilakukan oleh user untuk mangambil detail dari data *event* yang ada pada *web service* berdasarkan id *event*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. memilih event pada listview, pada aktifitas ini user memilih event untuk menampilkan detail dari event yang dipilih.
2. mengambil parameter id *event*, pada aktifitas ini sistem mengambil id *event*, dimana id *event* didapat ketika user memilih *event* sebelumnya pada list view.
3. memanggil fungsi getEventByID, pada aktifitas ini sistem memanggil fungsi getEventByID yang ada pada *web service*.
4. Mengambil return value dari web service, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh web service berupa hasil dari eksekusi method pada web service.
5. Melakukan parsing data, pada aktifitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
6. Menampilkan data *event*, pada aktifitas ini sistem menampilkan hasil parsing.

Seluruh aktifitas getDetailEventterlihat pada *activity diagram* gambar III.3.



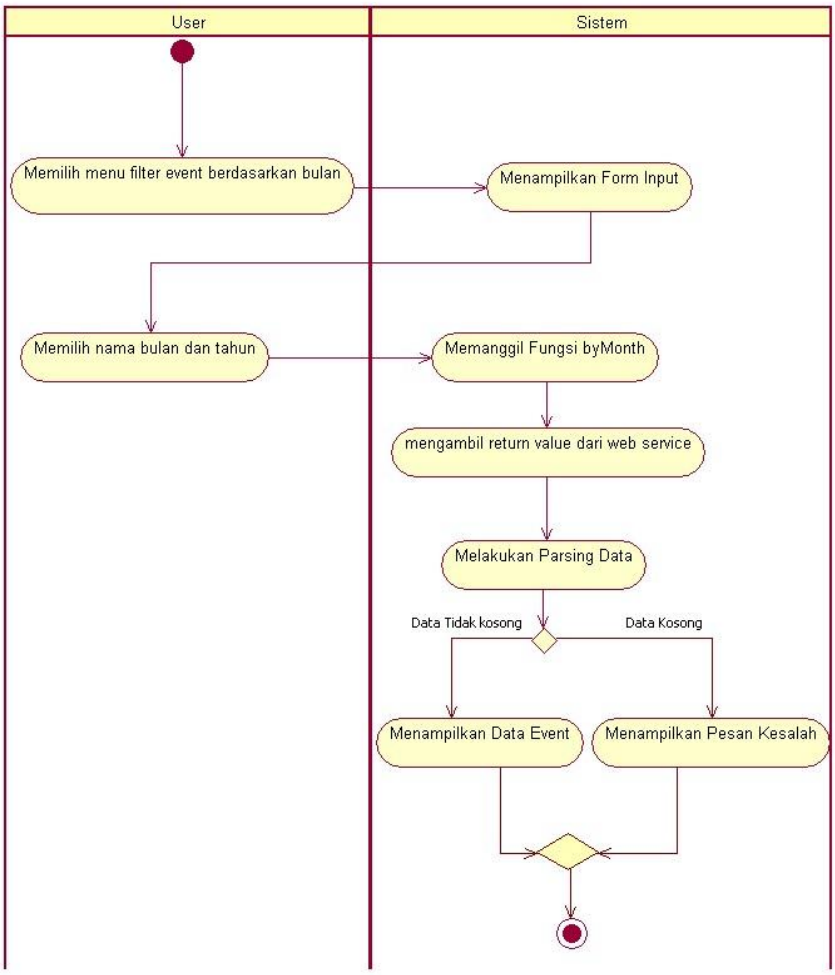
Gambar III.3Activity diagram getDetailEvent

3.1.4.4.2.3 getEventByMonth

Aktifitas get*EventByMonth* merupakan aktifitas yang dilakukan oleh user untuk memfilter data *event*berdasarkan bulan dan tahun dari tanggal yang ada pada *web service*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. memilih menu filter event berdasarkan bulan, pada aktifitas ini user memilih menu untuk memfilter data berdasarkan bulan dari tanggal event.
2. Menampilkan form input, pada aktifitas ini sistem menampilkan form input yang harus user isi.
3. Memilih nama bulan dan tahun , pada aktifitas ini user memilih bulan dan tahun untuk memfilter data event.
4. memanggil fungsi byMonth, pada aktifitas ini sistem memanggil fungsi byMonth yang ada pada *web service*.
5. Mengambil return value dari web service, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh web service berupa hasil dari eksekusi method pada web service.
6. Melakukan parsing data, pada aktifitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
7. Menampilkan data *event*, pada aktifitas ini sistem menampilkan hasil parsing kedalam bentuk list view jika data hasil parsing tidak kosong.
8. Menampilkan pesan kesalahan, pada aktifitas ini sistem menampilkan pesan kesalahan jika data hasil parsing kosong.

Seluruh aktifitas getEventByMonth terlihat pada *activity diagram* gambar III.4.



Gambar III.4Activity diagram getEventByMonth

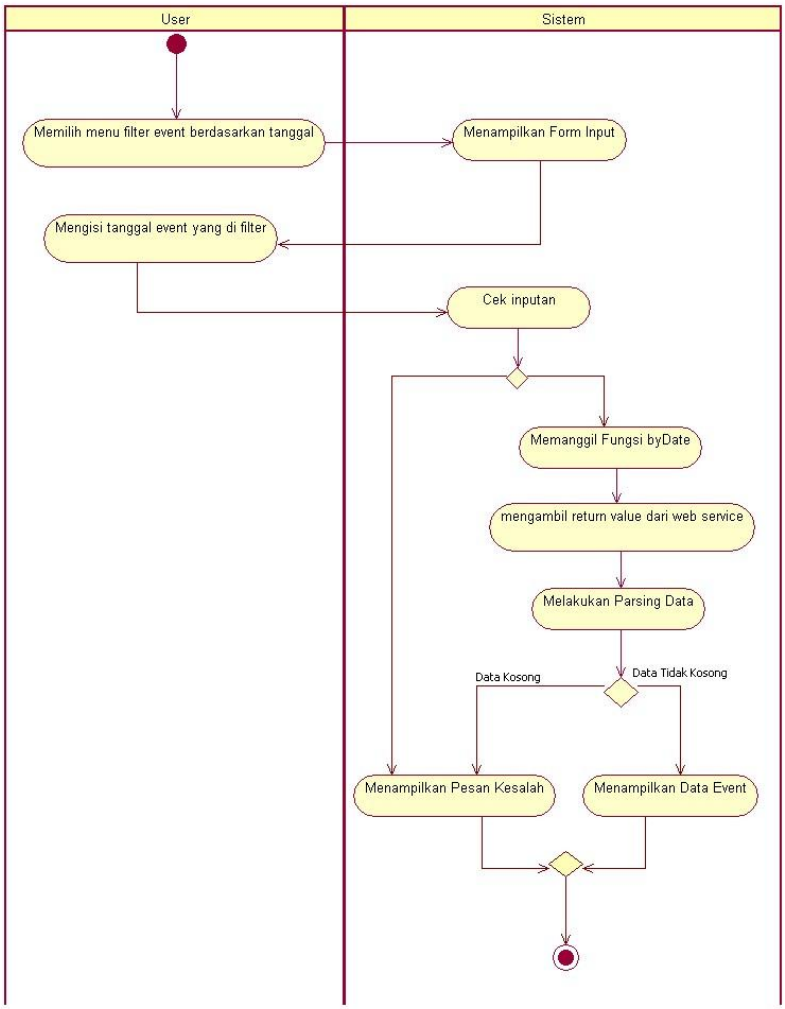
3.1.4.4.2.4 getEventByDate

Aktifitas get*EventByDate* merupakan aktifitas yang dilakukan oleh user untuk memfilter data *event*berdasarkan tanggal tertentu yang ada pada *web service*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. memilih menu filter event berdasarkan tanggal, pada aktifitas ini user memilih menu untuk memfilter data berdasarkan tanggal event.

2. Menampilkan form input, pada aktifitas ini sistem menampilkan form input yang harus user isi.
3. Mengisi tanggal event , pada aktifitas ini user mengisitangal untuk memfilter data event.
4. memanggil fungsi byDate, pada aktifitas ini sistem memanggil fungsi byDate yang ada pada *web service*.
5. Mengambil return value dari web service, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh web service berupa hasil dari eksekusi method pada web service.
6. Melakukan parsing data, pada aktifitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
7. Menampilkan data *event*, pada aktifitas ini sistem menampilkan hasil parsing kedalam bentuk list view jika data hasil parsing tidak kosong.
8. Menampilkan pesan kesalahan, pada aktifitas ini sistem menampilkan pesan kesalahan jika data hasil parsing kosong.

Seluruh aktifitas getEventByDate terlihat pada *activity diagram* gambar III.5.



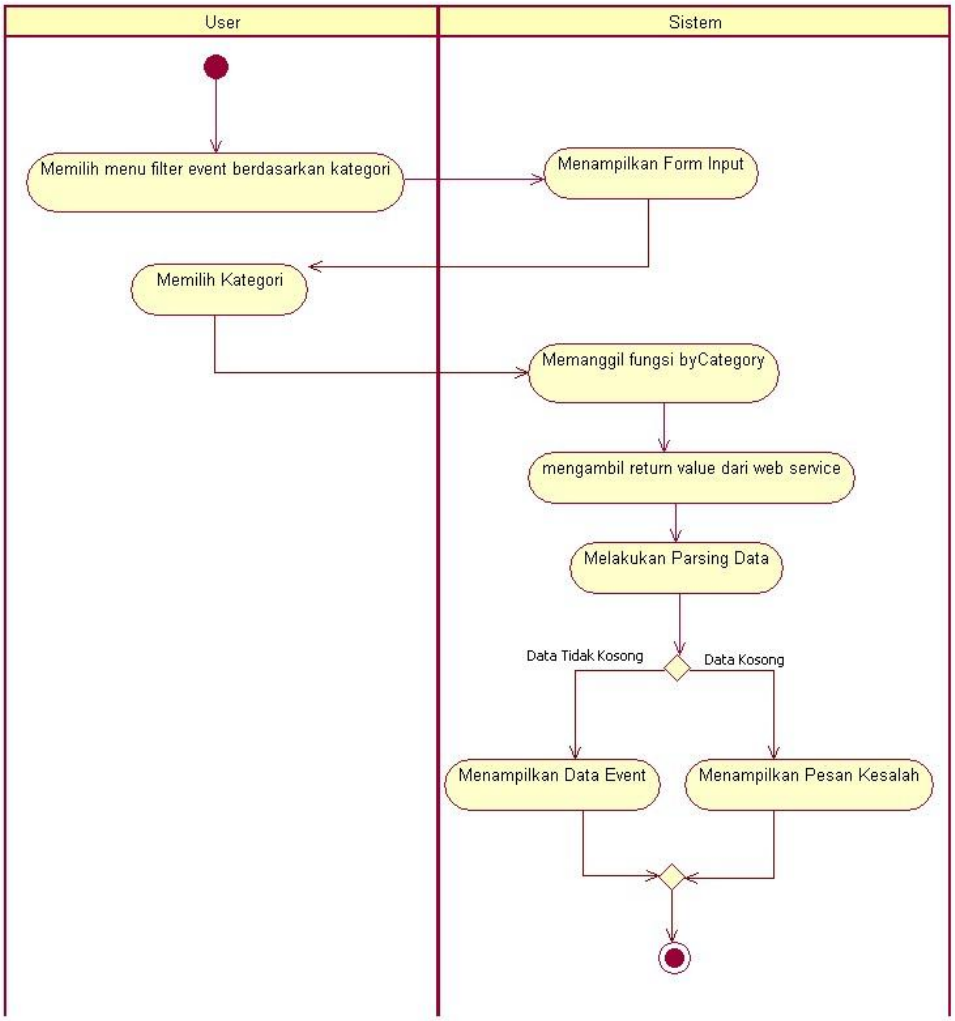
Gambar III.5Activity diagram getEventByDate

3.1.4.4.2.5 getEventByCategory

Aktifitas get*EventByCategory* merupakan aktifitas yang dilakukan oleh user untuk memfilter data *event*berdasarkan kategori event yang ada pada *web service*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. memilih menu filter event berdasarkan kategori, pada aktifitas ini user memilih menu untuk memfilter data berdasarkan kategori event.
2. Menampilkan form input, pada aktifitas ini sistem menampilkan form input yang harus user isi.
3. Memilih kategori, pada aktifitas ini user kategori event untuk memfilter data event.
4. memanggil fungsi byCategory, pada aktifitas ini sistem memanggil fungsi byCategory yang ada pada *web service*.
5. Mengambil return value dari web service, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh web service berupa hasil dari eksekusi method pada web service.
6. Melakukan parsing data, pada aktifitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
7. Menampilkan data *event*, pada aktifitas ini sistem menampilkan hasil parsing kedalam bentuk list view jika data hasil parsing tidak kosong.
8. Menampilkan pesan kesalahan, pada aktifitas ini sistem menampilkan pesan kesalahan jika data hasil parsing kosong.

Seluruh aktifitas getEventByCategory terlihat pada *activity diagram* gambar III.6.



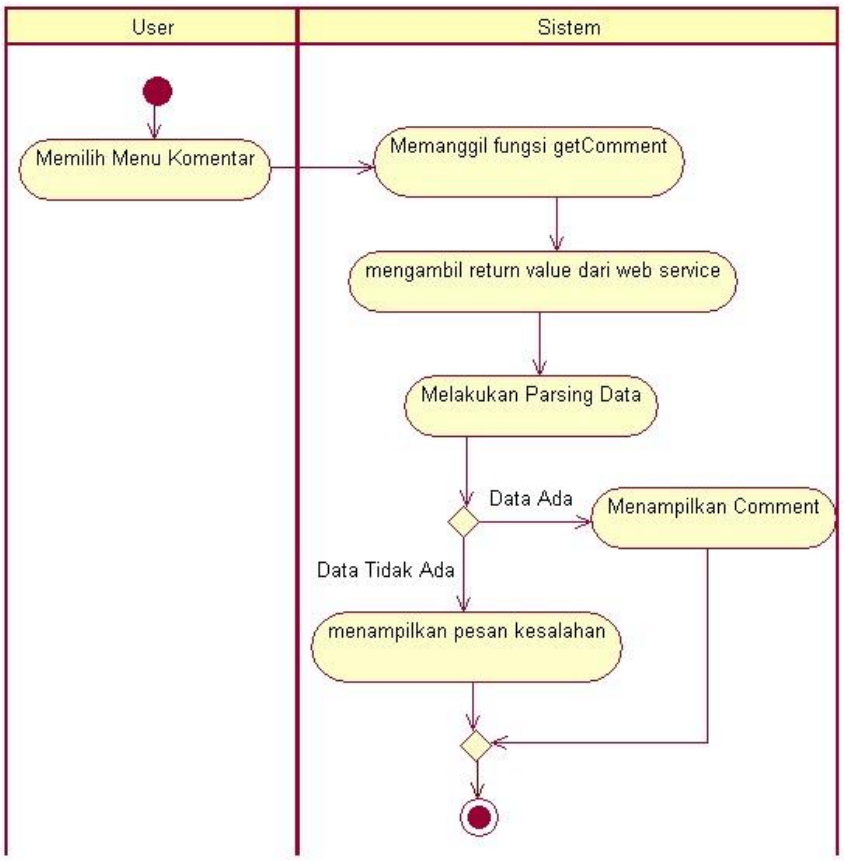
Gambar III.6Activity diagram getEventByCategory

3.1.4.4.2.6 getComment

Aktifitas getComment merupakan aktifitas yang dilakukan oleh user untuk mangambil data komentar yang ada pada *database* dengan *web service*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. memilih menu komentar, pada aktifitas ini user memilih menu komentar yang ada pada sistem.
2. memanggil fungsi getComment, pada aktifitas ini sistem memanggil fungsi getComment yang ada pada *web service*.
3. Mengambil return value dari web service, pada aktifitas ini sistem mengambil nilai yang di kembalikan oleh web service berupa hasil dari eksekusi method pada web service.
4. Melakukan parsing data, pada aktifitas ini sistem memparsing respon dari *web service* agar data yang diterima dapat dibaca.
5. Menampilkan data *event*, pada aktifitas ini sistem menampilkan hasil parsing kedalam bentuk list view.
6. Menampilkan pesan kesalahan, pada aktifitas ini sistem menampilkan pesan kesalahan jika data hasil parsing kosong.

Seluruh aktifitas getComment terlihat pada *activity diagram* gambar III.7.



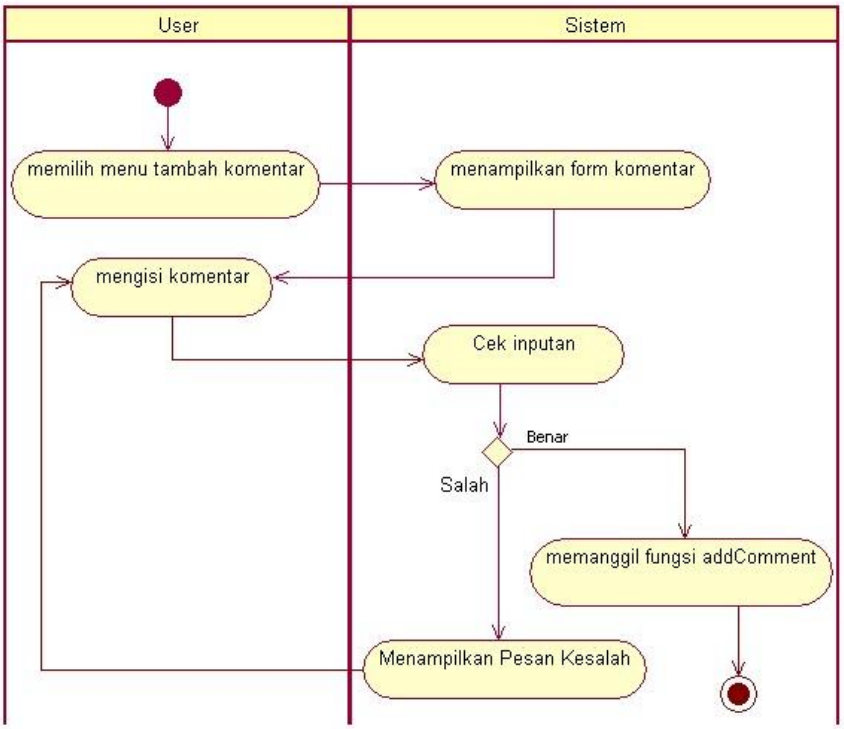
Gambar III.7Activity diagram getComment

3.1.4.4.2.7 addComment

Aktifitas *addComment* merupakan aktifitas yang dilakukan oleh user untuk memberikankomentar pada event. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. memilih menu tambah komentar, pada aktifitas ini user memilih menu tambah komentar yang ada pada sistem.
2. Menampilkan form input, pada aktifitas ini sistem menampilkan form yang harus user isi.
3. Mengisi form, pada aktifitasi ini user mengisi form yang berupa nama,email,dan komentar.
4. Cek inputan, pada aktifitas ini sistem mengecek input dari user.
5. Menampilkan pesan kesalahan, pada aktifitas ini jika inputan dari salah atau tidak lengkap maka sistem akan menampilkan pesan kesalahan.
6. Memanggil fungsi addComment, pada aktifitas ini jka inputan dari user telah lengkap dan benar maka sistem akan memanggil fungsi addcomment yang ada pada *web service.*

Seluruh aktifitas *addComment*terlihat pada *activity diagram* gambar III.8.

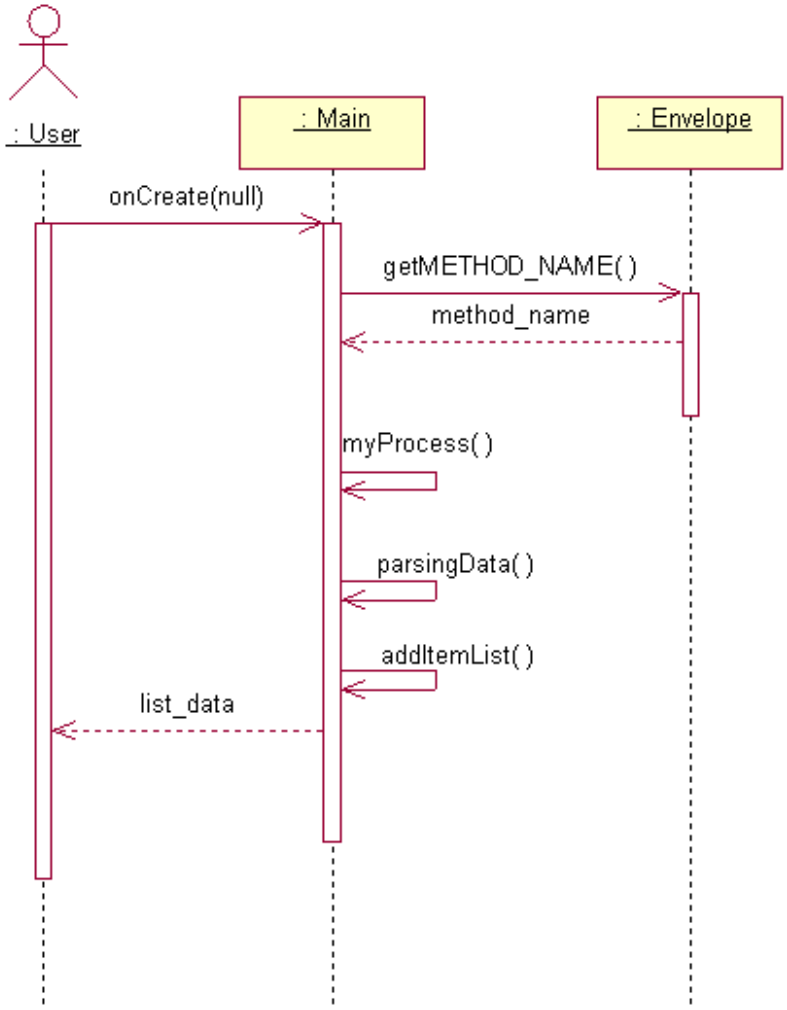


Gambar III.8 Activity diagram addComment

3.1.4.4.3 Sequence Diagram

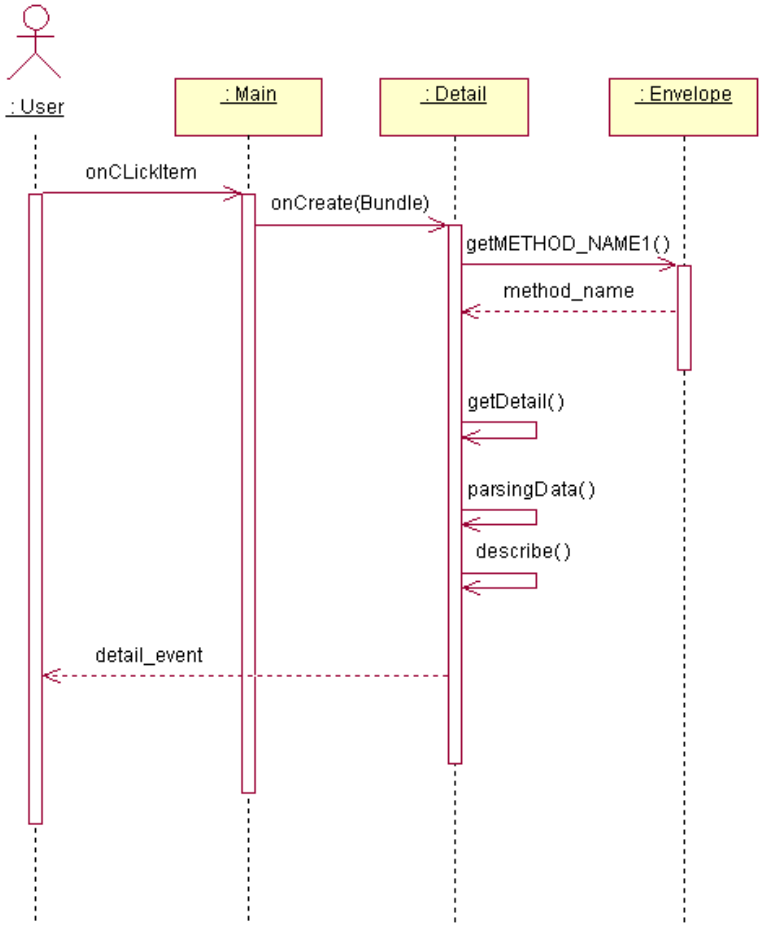
Merupakan bentuk dari interaction diagram, menggambarkan interaksiantar objek di dalam dan di sekitar sistem berupa Message yang di gambarkanterhadap waktu. Masing – masing objek termasuk actor memiliki line vertikal,message digambarkan garis berpanah dari satu objek ke objek lainnya, dan activation bar menunjukan lamanya eksekusi sebuah proses.Sebuah sequence diagram, secara khusus menjabarkan behavior sebuahskenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan – pesan yang melewati objek – objek ini di dalam use case. Berikut ini adalah sequence diagram dari sistem yang akan dibangun.

3.1.4.4.3.1 getEvent



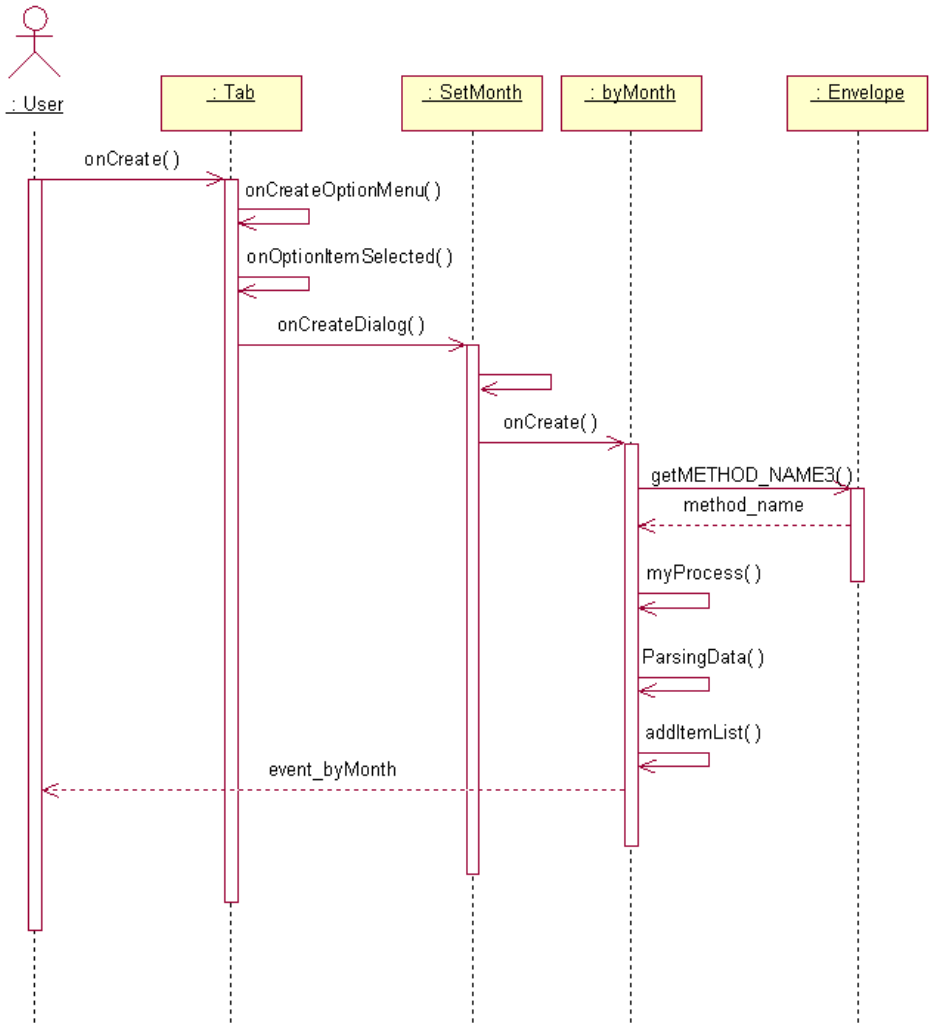
Gambar III.9 Sequence Diagram getEvent

3.1.4.4.3.2 getDetailEvent



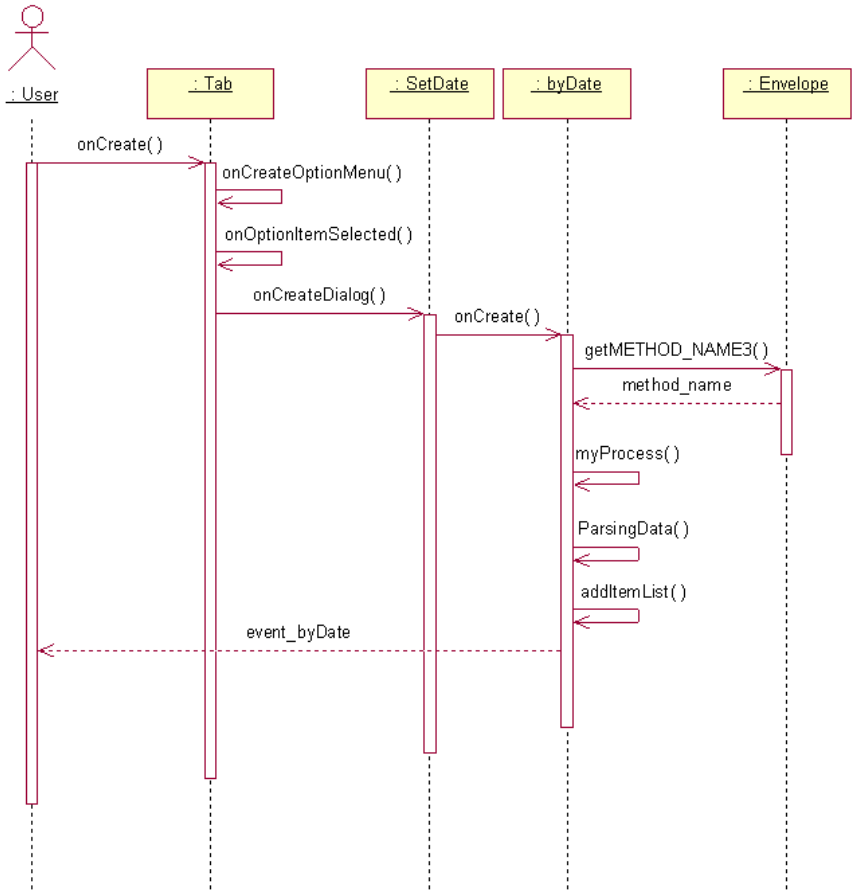
Gambar III.10 sequence diagram untuk mendapatkan detail event

3.1.4.4.3.3 getEventByMonth



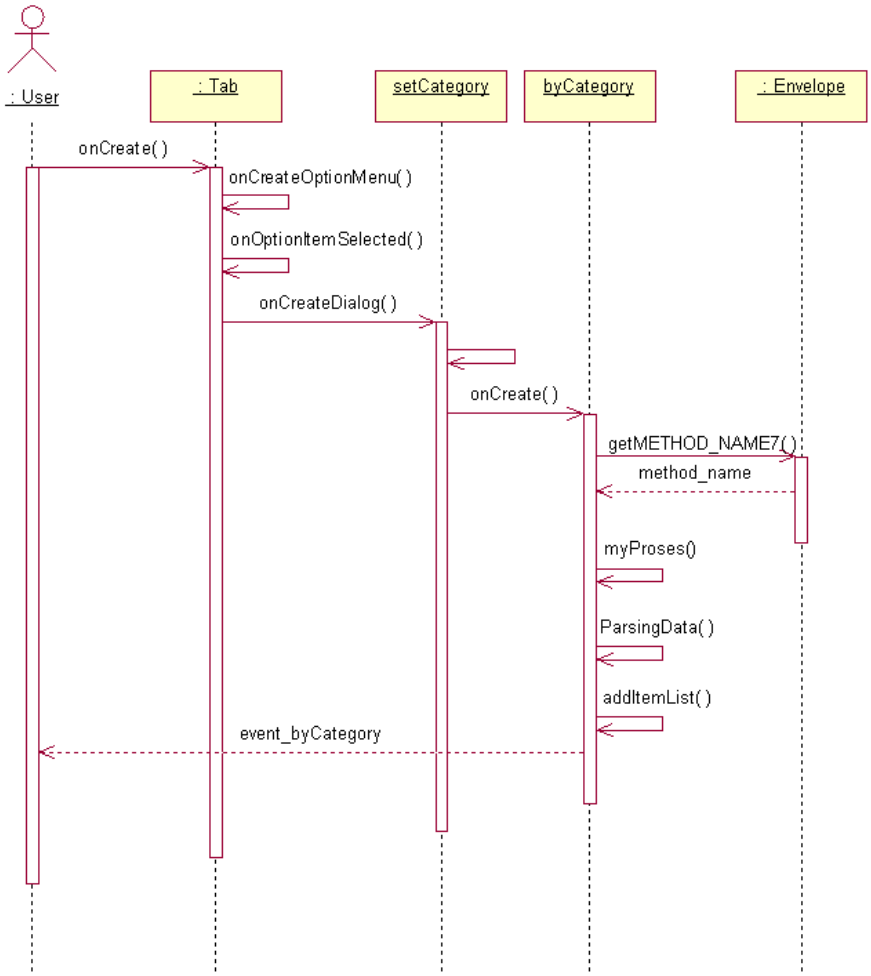
Gambar III.11 sequence diagram filter event berdasarkan bulan

3.1.4.4.3.4 getEventByDate



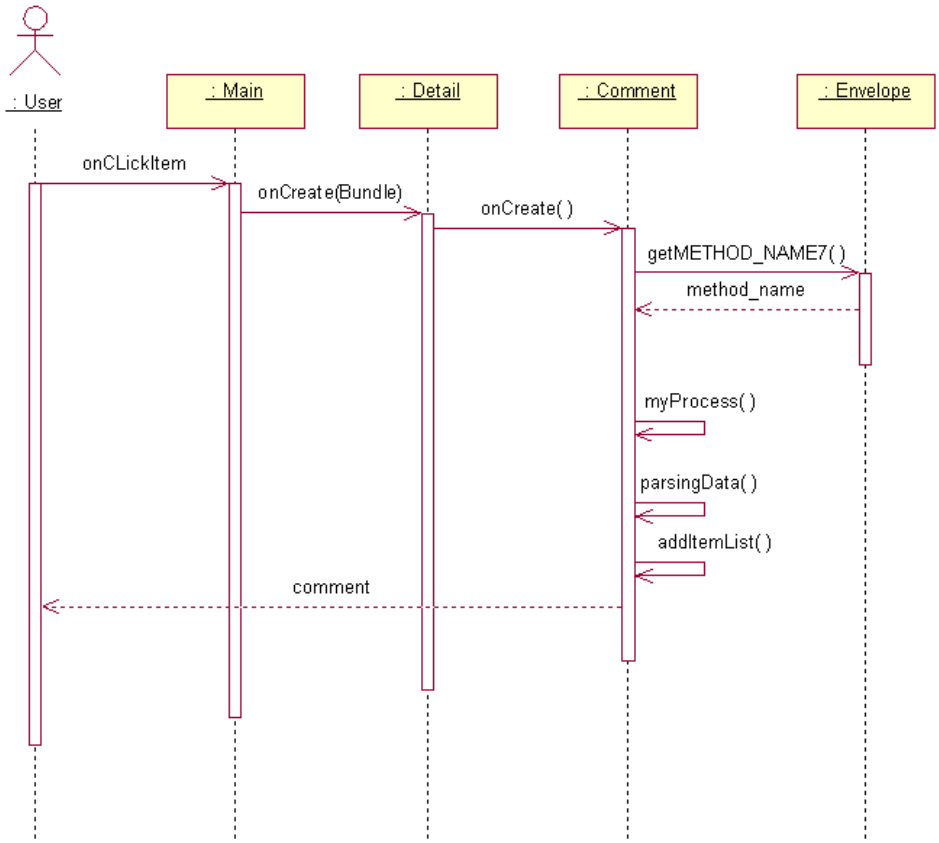
Gambar III.12 sequence diagram filter event berdasarkan tanggal

3.1.4.4.3.5 getEventByCategory



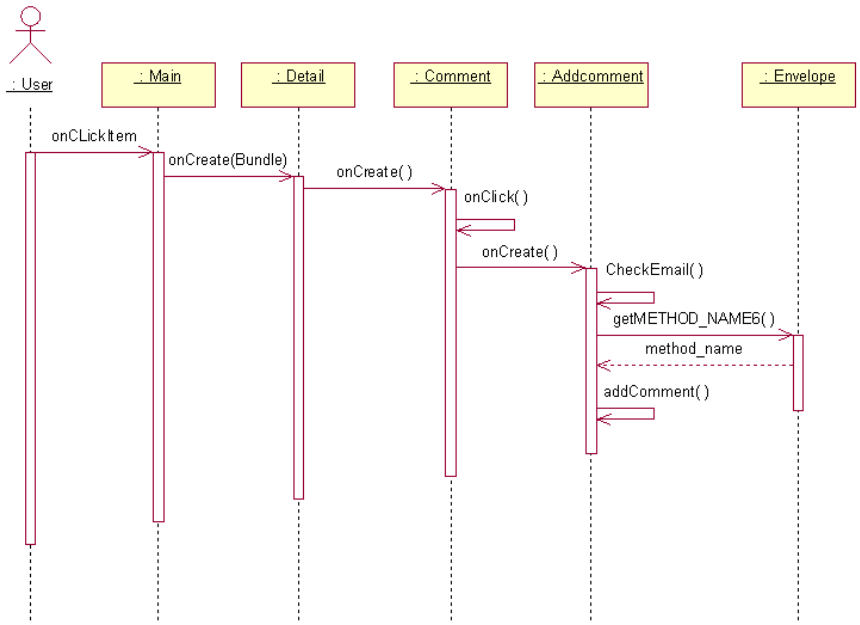
Gambar III.13sequence diagram filter event berdasarkan kategori

3.1.4.4.3.6 getComment



Gambar III.14sequence diagram getComment

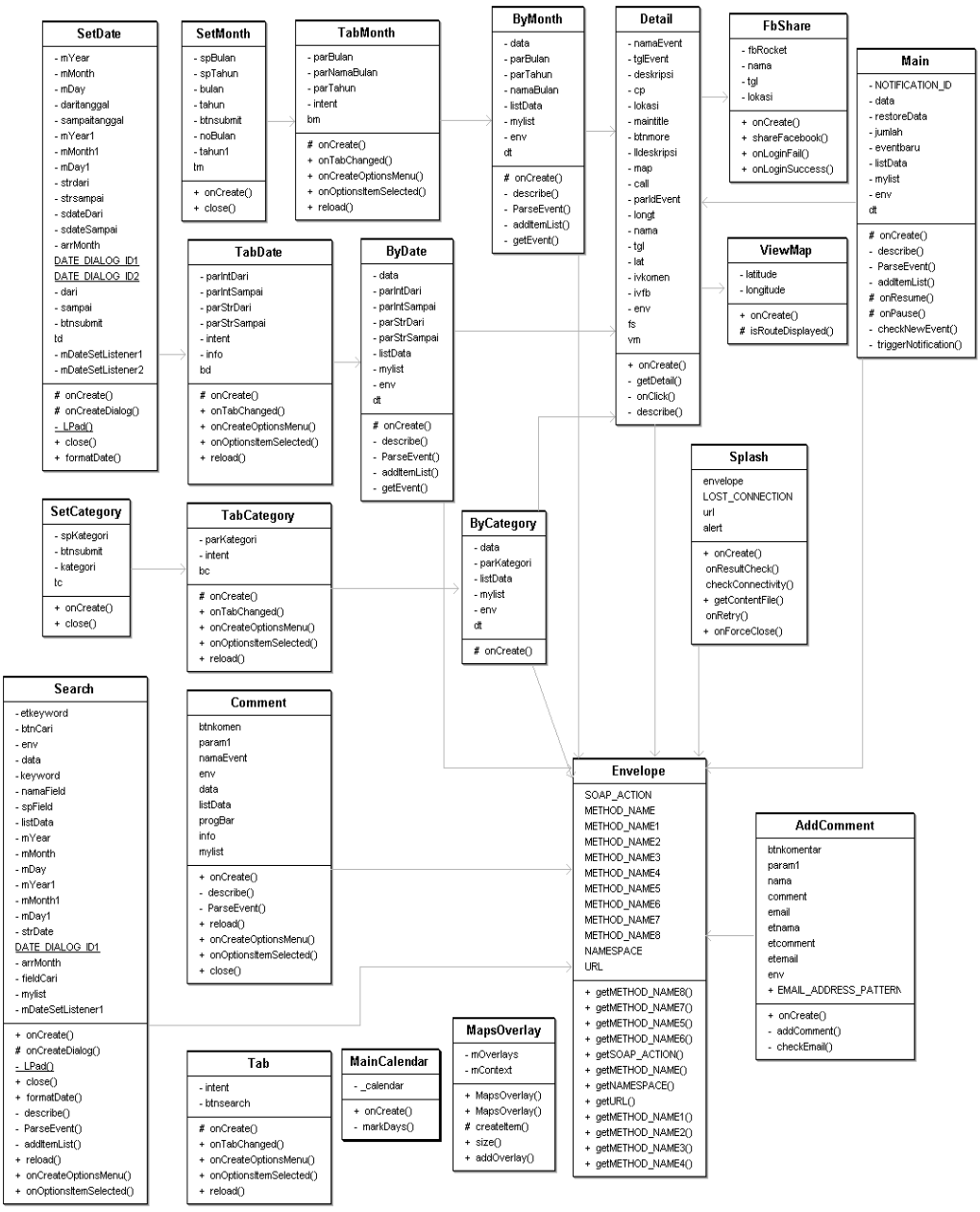
3.1.4.4.3.7 addComment



Gambar III.15sequence diagram untuk menambahkan komentar

3.1.4.4.4 Class Diagram

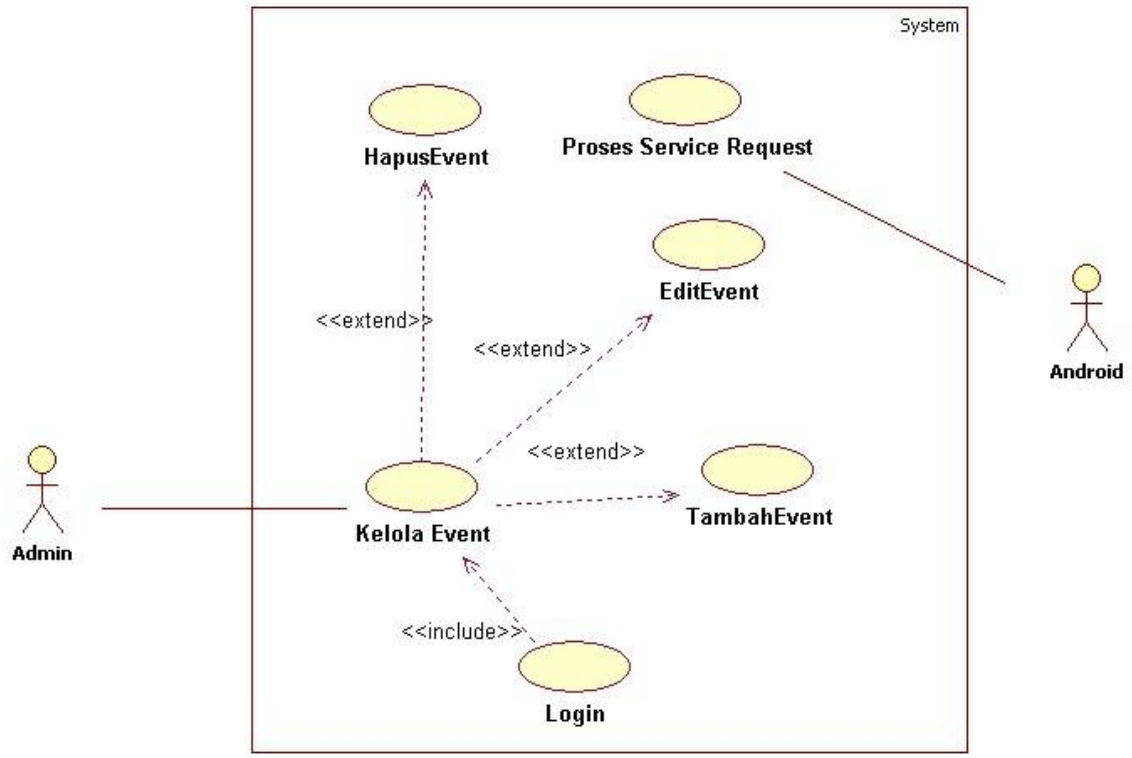
Class diagram menggambarkan struktur dan deskripsi class beserta hubungan satu sama lain seperti Asosiasi, Generalisasi dan lain sebagainya. Dalam gambar 3.16 di bawah ini di gambarkan hubungan antar class yang terjadi pada sistem yang akan dibangun. Sedangkan detail dari masing-masing *class* dapat dilihat pada lampiran A.



Gambar III.16 Class Diagram Aplikasi Android

3.1.4.5 Analisis Sistem Pada Web Service

3.1.4.5.1 Use Case Diagram



Gambar III.17 Use case diagram pada web service

3.1.4.5.1.1 Identifikasi Aktor

Pada tabel 3.12 berisi daftar aktor dan deskripsi peran meraka masing-masing dalam aplikasi ini.

III.12Definisi aktor dalam aplikasi web

NO	Aktor	Deskripsi
B-01	Admin	Merupakan pengguna pada aplikasi web yang mempunyai hak akses untuk tambah,edit,hapus data event
B-02	Android	Merupakan Sistem client yang mengakses web service

3.1.4.5.1.2 Identifikasi Use Case

Berdasarkan use case diagram di atas maka di buatlah skenario dari tiap proses yang ada pada use case diagram tersebut. Pada tabel 3.13 menggambarkan skenario pada use case login.

Tabel III.13Skenario use case login

Identifikasi	
Nomor	UC-WS-01
Nama Use Case	Login
Deskripsi	Proses autentifikasi admin untuk masuk pada sistem
Aktor	Admin
Skenario Utama (UC-WS-01-S01)	
Kondisi awal	Koneksi ke *database* berhasil dan tampil halaman login
Aksi aktor	**Respon Sistem**
1. Membuka halaman utama	2. Menampilkan *form login*
3. Memasukan username dan *password*	4. Mencocokan username dan *password* yang ada pada *database*
	5. Login berhasil.
Skenario Alternatif (UC-WS-01-S02)Login namun terdapat pesan kesalahan	
	5a. Login gagal, sistem menampilkan pesan kesalahan. Misal karena username dan *password* tidak ada pada *database* atau username dan *password* tidak cocok.
Kondisi akhir	Halaman kelola *event* tampil

Pada tabel 3.14 menggambarkan skenario tambah *event* pada use case Tambah*Event*.

Tabel III.14 skenario use case tambah event

Identifikasi	
Nomor	UC-WS-02
Nama Use Case	Tambah*Event*
Deskripsi	Proses untuk menambah data *event*
Aktor	Admin
Skenario Utama(UC-WS-02-S01)	
Kondisi awal	Koneksi ke *database* berhasil
Aksi aktor	**Respon Sistem**
1. Membuka halaman tambah *event*	2. Menampilkan *form* tambah *event*
3. Memasukan data *event* baru	4. Melakukan proses validasi
	5. Menyimpan data yang valid ke dalam *database*.
Skenario Alternatif (UC-WS-02-S02) Tambah *Event* namun terdapat pesan kesalahan	
	4a. Menampilkan pesan kesalahan, karena data yang diinputkan salah.
Kondisi akhir	Jumlah data event pada *database* bertambah

Pada tabel 3.15 menggambarkan skenario hapus *event* yang dilakukan oleh admin pada use case Hapus*Event*

Tabel III.15 skenario use case hapus event

Identifikasi	
Nomor	UC-WS-03
Nama Use Case	Hapus*Event*
Deskripsi	Proses untuk menghapus data *event* dalam *database*
Aktor	Admin
Skenario Utama(UC-WS-03-S01)	
Kondisi awal	Koneksi ke *database* berhasil
Aksi aktor	**Respon Sistem**
1. Membuka halaman hapus *event*	2. Menampilkan *form* hapus *event*
3. Mencari data *event* yang akan di hapus	4. Melakukan proses pencarian data
	5. Menampilkan data *event* yang di cari
	6. Konfirmasi penghapusan data
7. Merespon konfirmasi dari sistem	8. Menghapus data *event* dari *database*
Skenario Alternatif(UC-WS-03-S02) Hapus *Event* namun terdapat pesan error	
	5a. Menampilkan pesan kesalahan, misalnya karena data yang di cari tidak ditemukan.
Kondisi akhir	Jumlah data pada *database* berkurang

Pada tabel 3.16 menggambarkan skenario untuk mengedit event yang dilakukan oleh admin pada use case Edit*Event*

Tabel III.16 skenario use case edit event

Identifikasi	
Nomor	UC-WS-04
Nama Use Case	Edit*Event*
Deskripsi	Proses untuk mengedit data *event* dalam *database*
Aktor	Admin
Skenario Utama(UC-WS-04-S01)	
Kondisi awal	Koneksi ke *database* berhasil
Aksi aktor	**Respon Sistem**
1. Membuka halaman edit *event*	2. Menampilkan *form* edit *event*
3. Mencari data *event* yang akan di edit, dengan memasukan nama *event*.	4. Melakukan proses pencarian data
	5. Menampilkan data *event* yang di cari
6. Mengedit data pada data *event* yang telah dicari sebelumnya	7. Melakukan proses validasi
	8. Menyimpan data yang telah diedit dalam *database*.
Skenario Alternatif(UC-WS-04-S02) Edit *Event* namun event tidak ditemukan	
	5a. Menampilkan pesan kesalahan, misalnya karena data yang di cari tidak ditemukan.
Skenario Alternatif(UC-WS-04-S03) Edit *Event* namun event gagal di simpan	
	8a. Menampilkan pesan kesalahan, misalnya karena data *event* yang di inputkan salah atau karena koneksi ke *database* gagal.
Kondisi akhir	Data *event* yang di edit berhasil di ubah

Pada tabel 3.17 menggambarkan skenario untuk melakukan eksekusi dari service yang android panggil.

Tabel III.17 Skenario proses service request

Identifikasi	
Nomor	UC-WS-05
Nama Use Case	Proses Service Request
Deskripsi	Proses untuk melakukan eksekusi dari sebuah service yang diminta
Aktor	Android
Skenario Utama(UC-WS-05-S01)	
Kondisi awal	Koneksi ke sistem berhasil
Aksi aktor Android	**Respon Sistem**
1. Memanggil method	2. Mencari Method
	3. Menjalankan Method
	4. Menampilkan respon berupa hasil query dalam bentuk SOAP
Kondisi akhir	Sistem menampilkan respon dalam bentuk SOAP

3.1.4.5.2 Activity Diagram

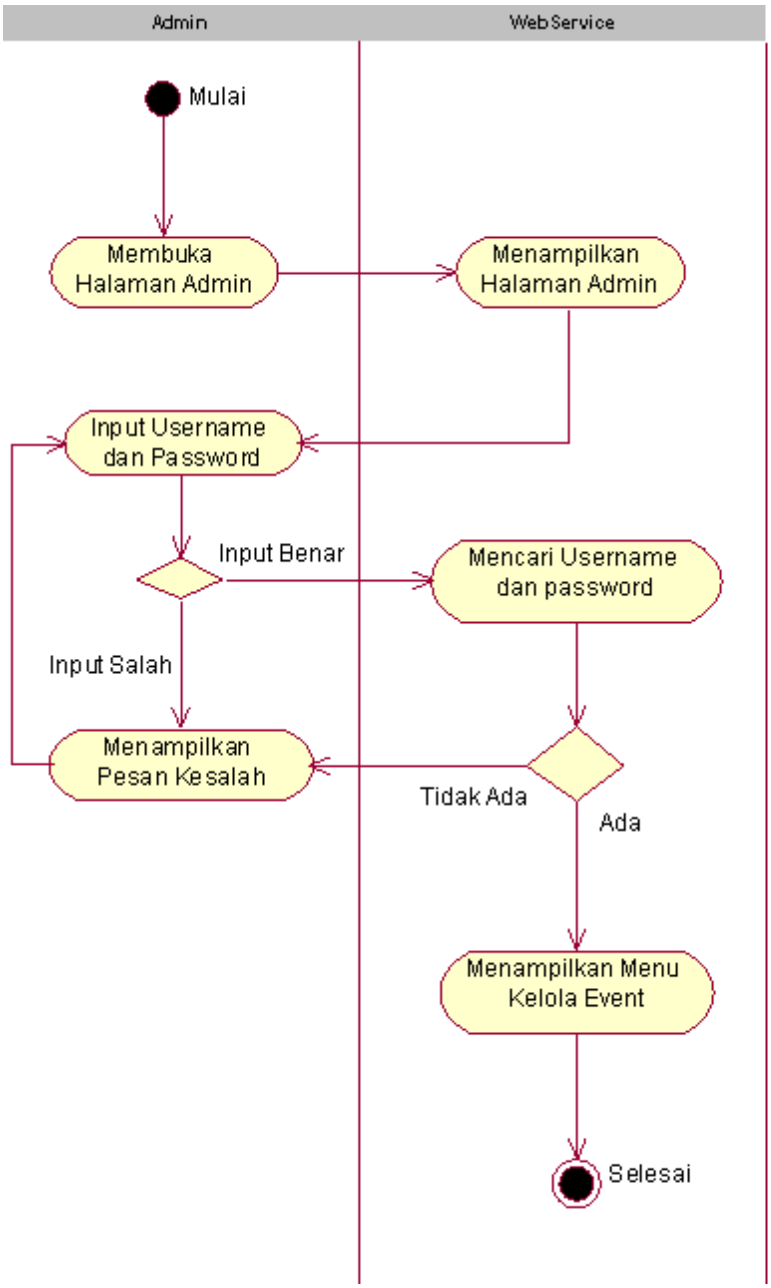
3.1.4.5.2.1 Login Admin

Aktifitas login merupakan aktifitas yang dilakukan admin untuk mandapatkan akses untuk mengelola *event*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Membuka halaman admin, pada aktifitas ini admin membuka halaman untuk mengakses halaman kelola *event*.

2. Menampilkan halaman admin, pada aktifitas ini sistem menampilkan form login terlebih dahulu untuk mengecek apakah admin dapat mengakses atau tidak.
3. Input username dan *password*, pada aktifitas ini admin menginputkan username dan *password*.
4. Mencari username dan *password,* pada aktifitas ini sistem mencocokan username dan *password* yang admin inputkan dengan username dan *password* yang ada pada *database*.
5. Menampilkan pesan kesalahan, aktifitas ini terjadi jika username dan *password* yang admin inputkan tidak ada pada *database*..
6. Menampilkan menu kelola *event*, aktifitas ini terjadi jika username dan *password* yang admin inputkan ada pada *database* lalu sistem menampilkan menu kelola *event*, dimana admin dapat melakukan tambah,edit,hapus terhadap data *event* yang ada pada *database*.

Seluruh aktifitas login terlihat pada *activity diagram* gambar III.18.



Gambar III.18 Activity diagram login

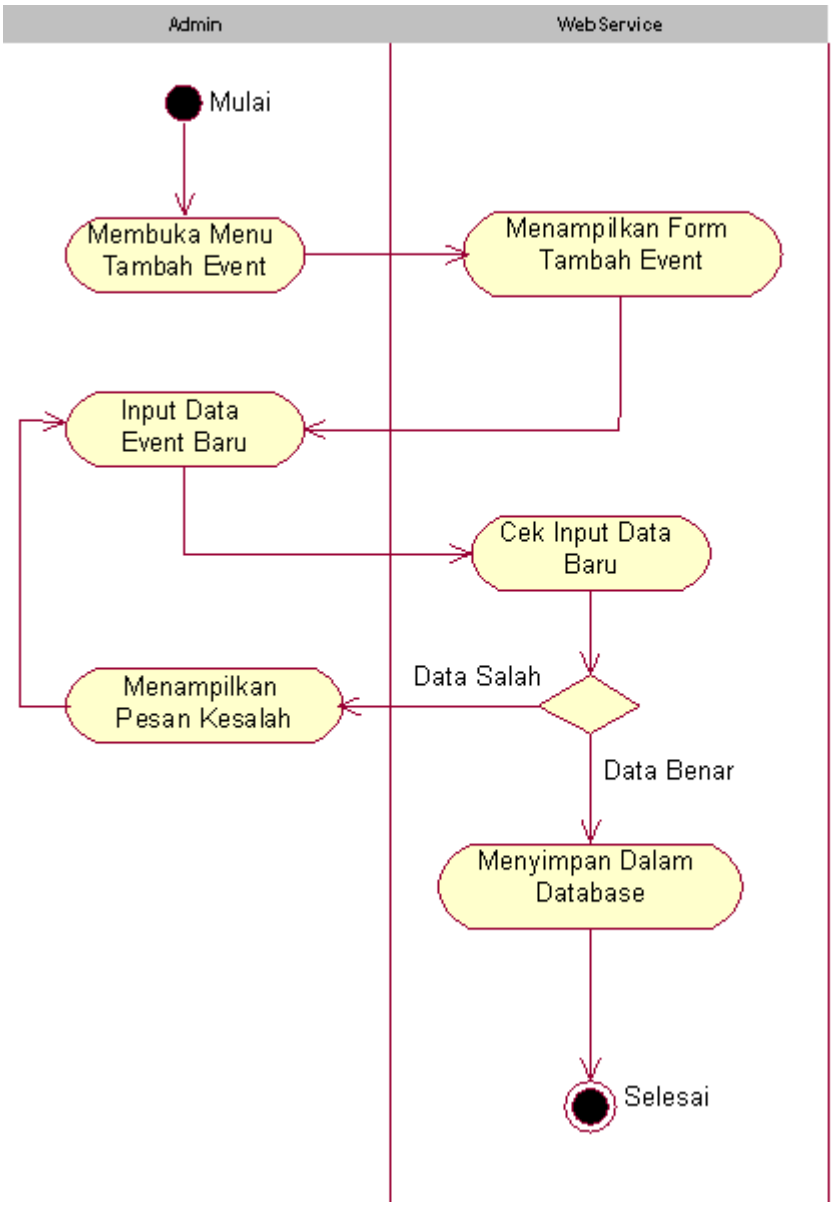
3.1.4.5.2.2 Tambah Data Event

Aktifitas tambah *event* merupakan aktifitas yang dilakukan oleh admin untuk menambahkan data *event* baru. Dalam aktifitas ini terdapat beberapa tahapan aktifitas , yaitu :

1. Membuka menu tambah *event*, dalam aktifitas ini admin memilih menu untuk menambahkan evet.
2. Menampilkan form tambah *event*, aktifitas ini merupakan aktifitas yang dilakukan oleh sistem untuk menampilkan form tambah *event*.
3. Input data *event* baru, pada aktifitas ini admin melakukan input data pada form yang telah disediakan oleh sistem.
4. Cek input data baru, pada aktifitas ini sistem melakukan pengecekan terhadap data yang di inputkan oleh admin, apakah data yang di input kan sudah benar.
5. Menampilkan pesan kesalahan, aktifitas ini terjadi jika data yang admin inputkan salah.
6. Menyimpan dalam *database*, aktifitas ini terjadi jika data yang admin inputkan sudah benar dan sistem menyimpan data tersebut kedalam *database*.

Seluruh aktifitas tambah *event* terlihat pada *activity diagram* gambar III.19

.



Gambar III.19Activity diagram tambah event

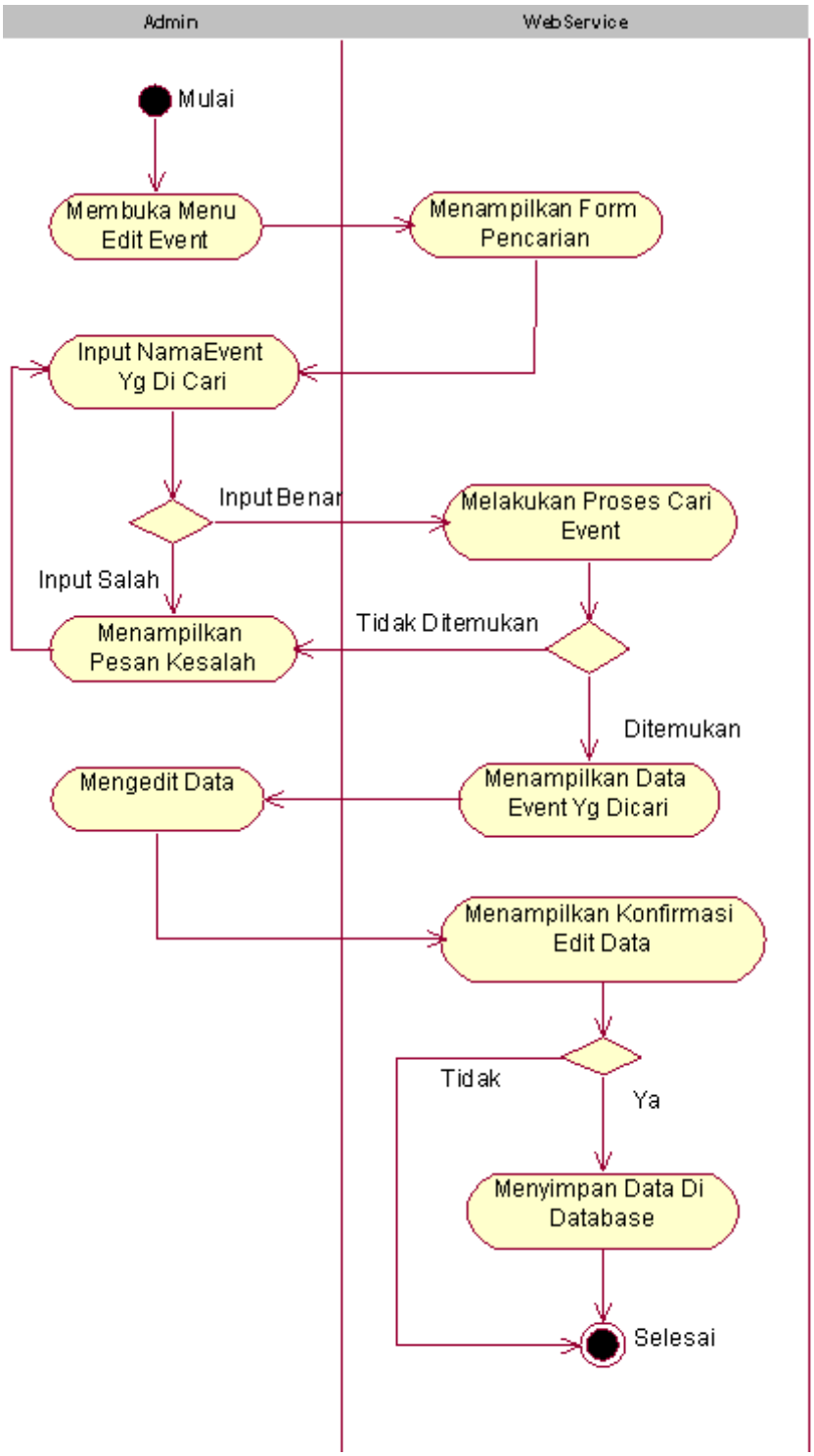
3.1.4.5.2.3 Edit *event*

Aktifitas edit *event* merupakan aktifitas yang dilakukan oleh admin untuk mengedit *event* yang sudah ada dalam *database*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Membuka menu edit *event*, pada aktifitas ini admin memilih menu untuk mengedit data *event*.
2. Menampilkan form pencarian, pada aktifitas ini sistem menampilkan form untuk mencari data *event* yang akan di edit.
3. Input nama*event*, aktifitas ini dilakukan oleh admin untuk menginputkan nama *event* yang akan dicari pada form pencarian.
4. Melakukan proses Cari event, pada aktifitas ini sistem melakukan proses Cari event dengan parameter nama*event* yang admin inputkan sebelumnya.
5. Menampilkan pesan kesalahan, aktifitas ini terjadi jika setelah proses Cari event di jalankan ternyata parameter berupa nama*event* yang admin inputkan tidak ditemukan.
6. Menampilkan data *event* yang dicari, pada aktifitas ini sistem menampilkan data yang admin cari pada form edit *event*.
7. Mengedit data, pada aktifitas ini admin melakukan edit data *event*.
8. Menampilkan konfirmasi edit data, pada aktifitas ini sistem menampilkan pesan konfirmasi apakah admin akan menyimpan hasil dari data yang sudah di edit.

9. Menyimpan data dalam *database*, pada aktifitas sistem akan menyimpan data kedalam *database* jika admin mengkonfirmasi “ya”.

Seluruh aktifitas edit *event* terlihat pada *activity diagram* gambar III.20.



Gambar III.20 Activity diagram edit event

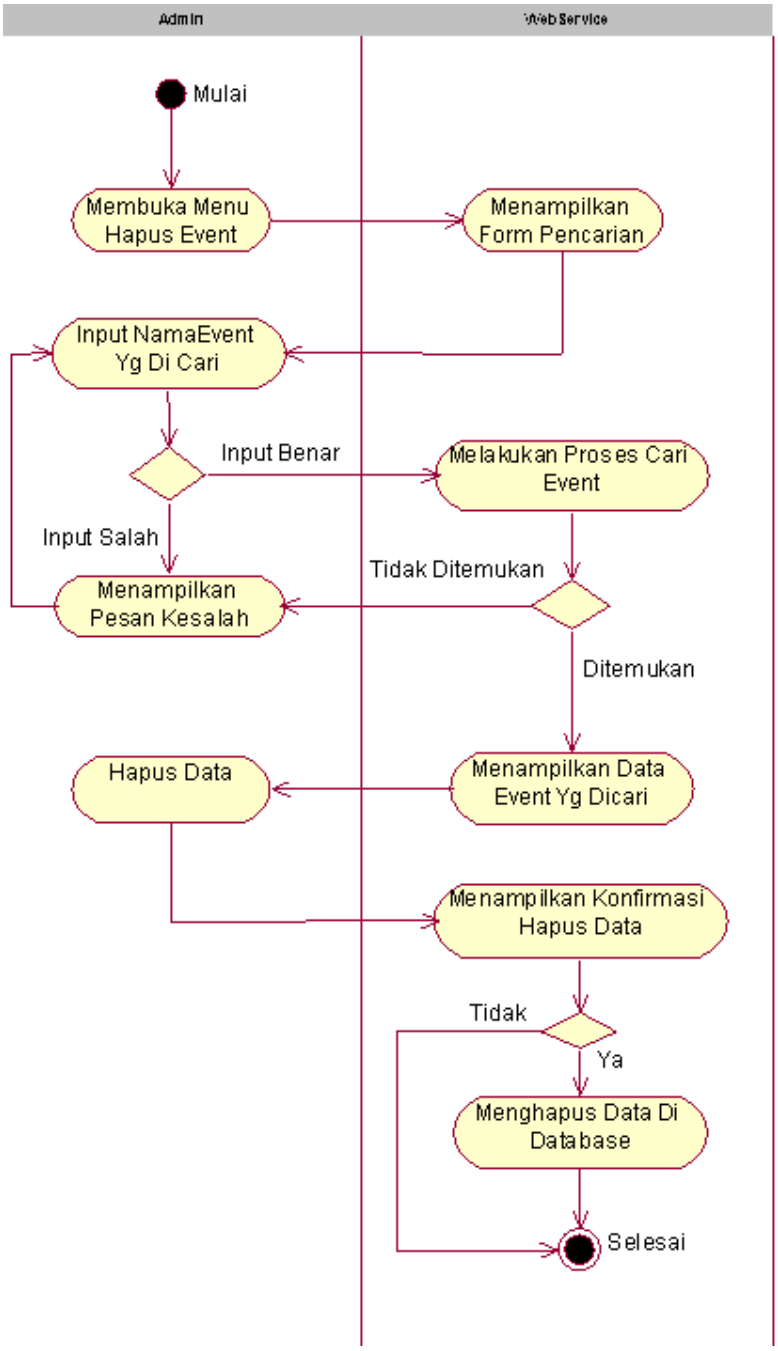
3.1.4.5.2.4 Hapus *Event*

Aktifitas hapus *event* merupakan aktifitas yang dilakukan oleh admin untuk menghapus *event* yang ada dalam *database*. Dalam aktifitas ini terdapat beberapa aktifitas yaitu :

1. Membuka menu edit *event*, pada aktifitas ini admin memilih menu untuk mengedit data *event*.
2. Menampilkan form pencarian, pada aktifitas ini sistem menampilkan form untuk mencari data *event* yang akan di edit.
3. Input nama*event*, aktifitas ini dilakukan oleh admin untuk menginputkan nama *event* yang akan dicari pada form pencarian.
4. Melakukan proses cari event, pada aktifitas ini sistem melakukan proses Cari event dengan parameter nama*event* yang admin inputkan sebelumnya.
5. Menampilkan pesan kesalahan, aktifitas ini terjadi jika setelah proses Cari event di jalankan ternyata parameter berupa nama*event* yang admin inputkan tidak ditemukan.
6. Menampilkan data *event* yang dicari, pada aktifitas ini sistem menampilkan data yang admin cari pada form edit *event*.
7. Menghapus Data, pada aktifitas ini admin melakukan hapus data *event*.
8. Menampilkan konfirmasi hapus data, pada aktifitas ini sistem menampilkan pesan konfirmasi apakah admin akan menghapus data dari *database*.

9. Menghapus data di *database*, pada aktifitas sistem akan menghapus data jika admin mengkonfirmasi “ya”.

Seluruh aktifitas hapus *event* terlihat pada *activity diagram* gambar III.21.



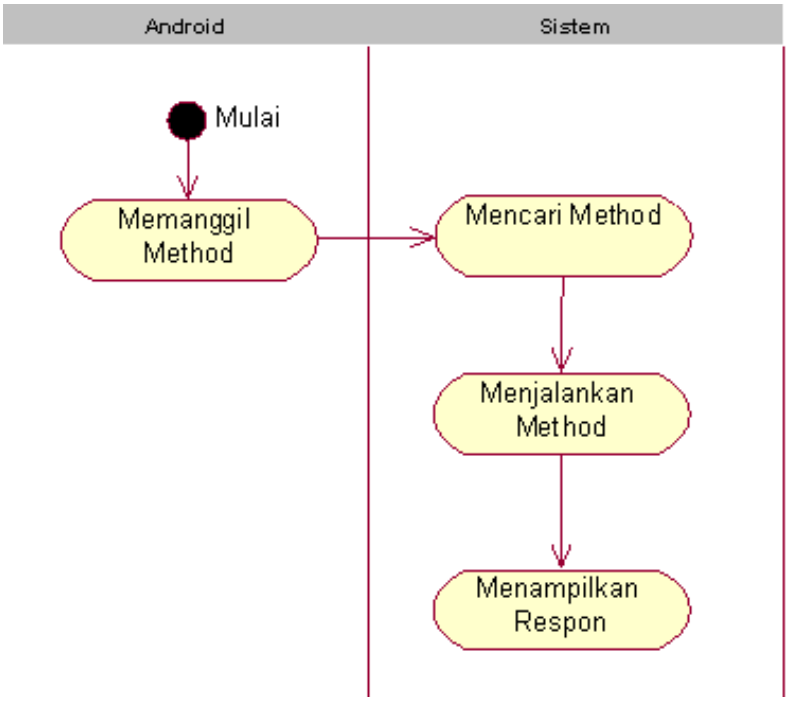
Gambar III.21Activity diagram hapus event

3.1.4.5.2.5 Proses Service Request

Aktifitas proses service request merupakan aktifitas yang dilakukan oleh android untuk memanggil method yang ada pada sistem. Dalam aktifitas ini terdapat beberapa tahapan aktifitas , yaitu :

1. Memanggil method, dalam aktifitas ini android memanggil method dengan mengakses http://adtya.dyndns.org:8080/eventService/DataEventService?wsdl
2. Mencari Method, aktifitas ini merupakan aktifitas yang dilakukan oleh sistem untuk mencari method yang dipanggil oleh android.
3. Menjalankan method, pada aktifitas ini sistem menjalankan method yang android panggil.
4. Menampilkan respon, pada aktifitas ini sistem menampilkan respon berupa hasil query yang di tampilkan dalam bentuk SOAP.

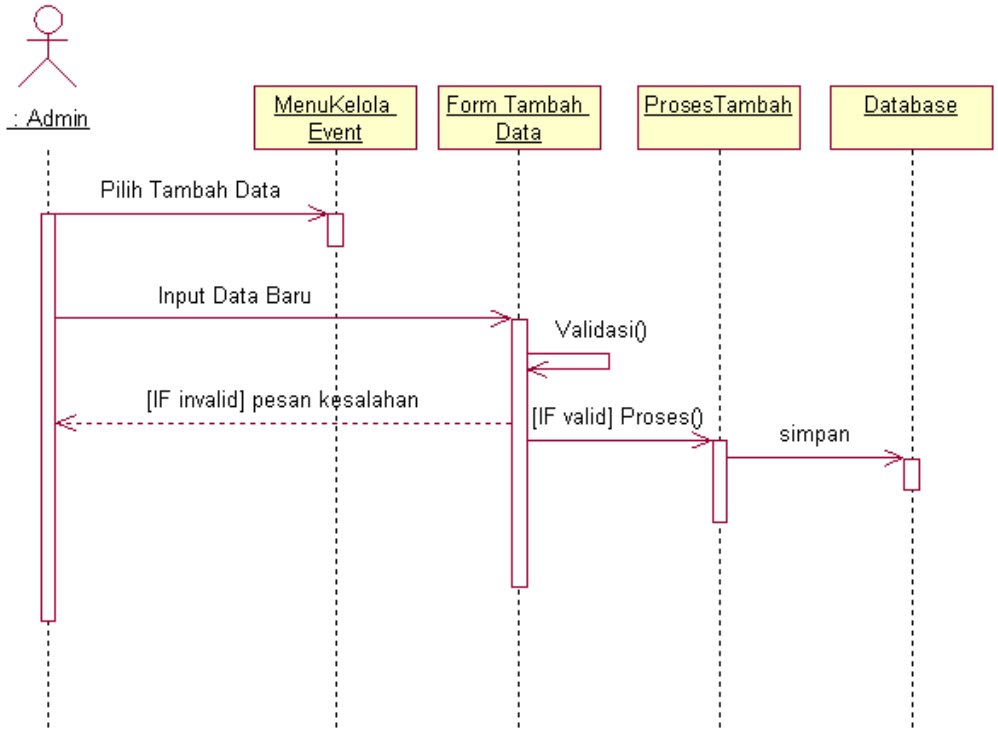
Seluruh aktifitas terlihat pada *activity diagram* gambar III.22



Gambar III.22 Activity Diagram Proses Service Request

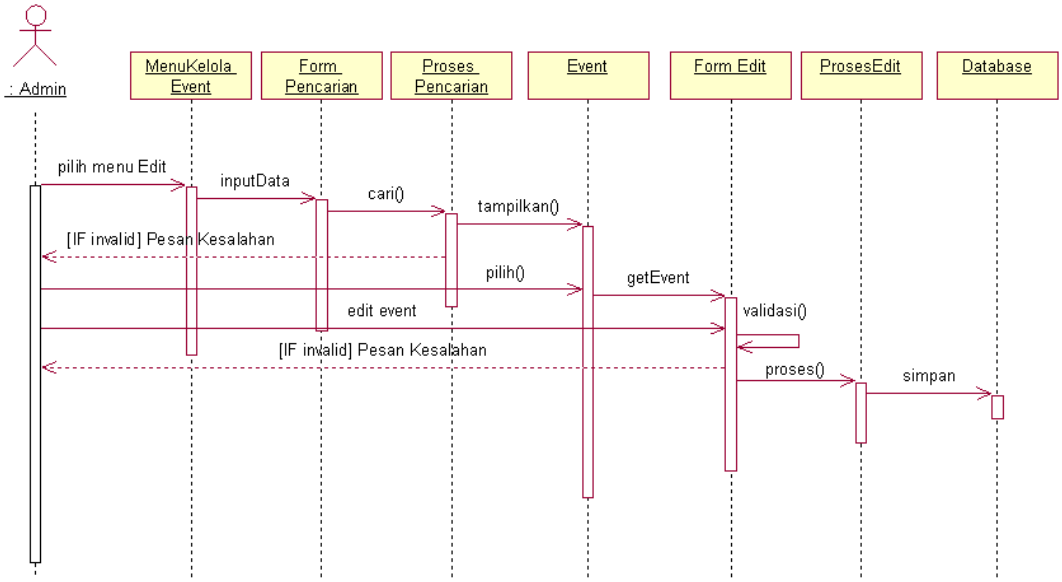
3.1.4.5.3 Sequence Diagram

3.1.4.5.3.1 Tambah *Event*



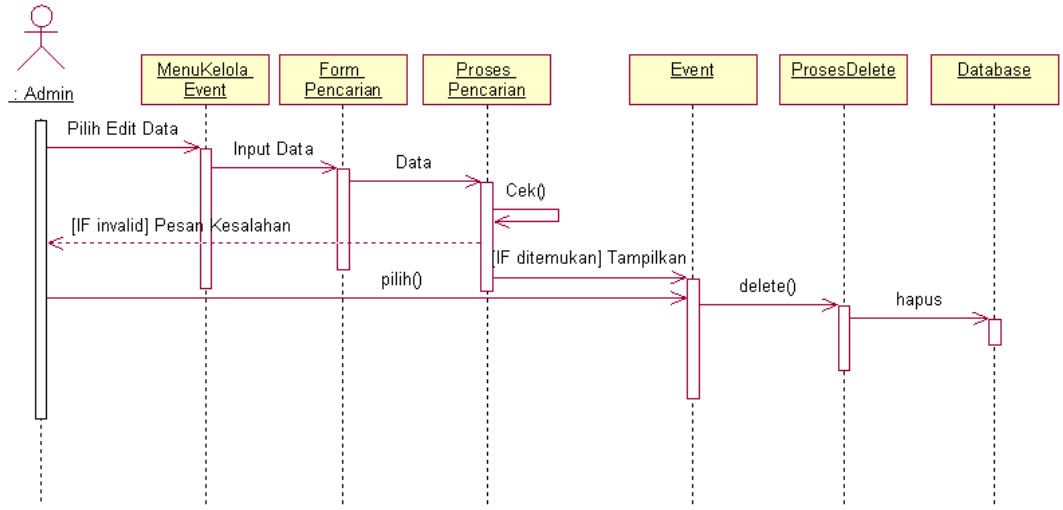
Gambar III.23Sequence diagram tambah event

3.1.4.5.3.2 Edit *Event*



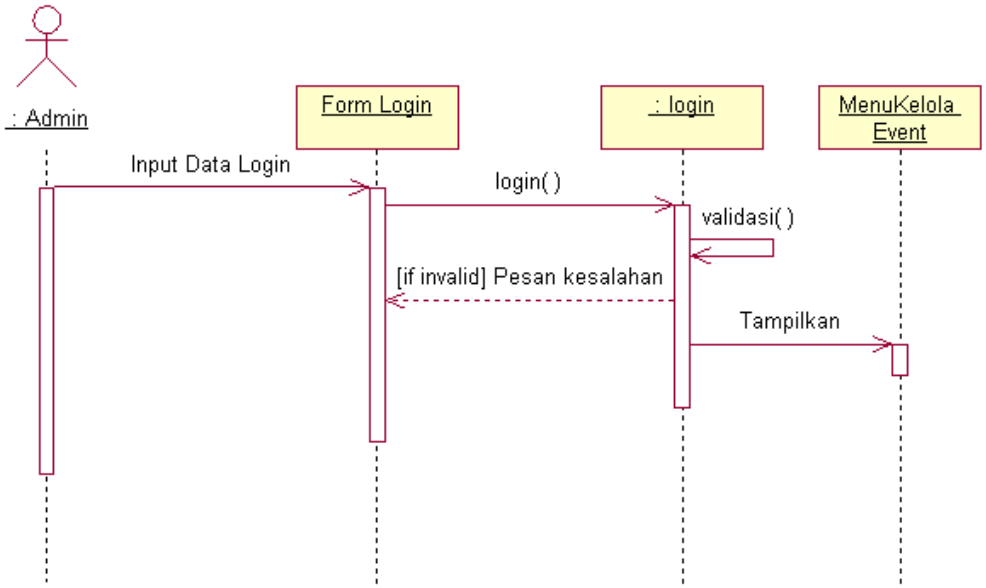
Gambar III.24 sequence diagram edit event

3.1.4.5.3.3 Hapus *Event*



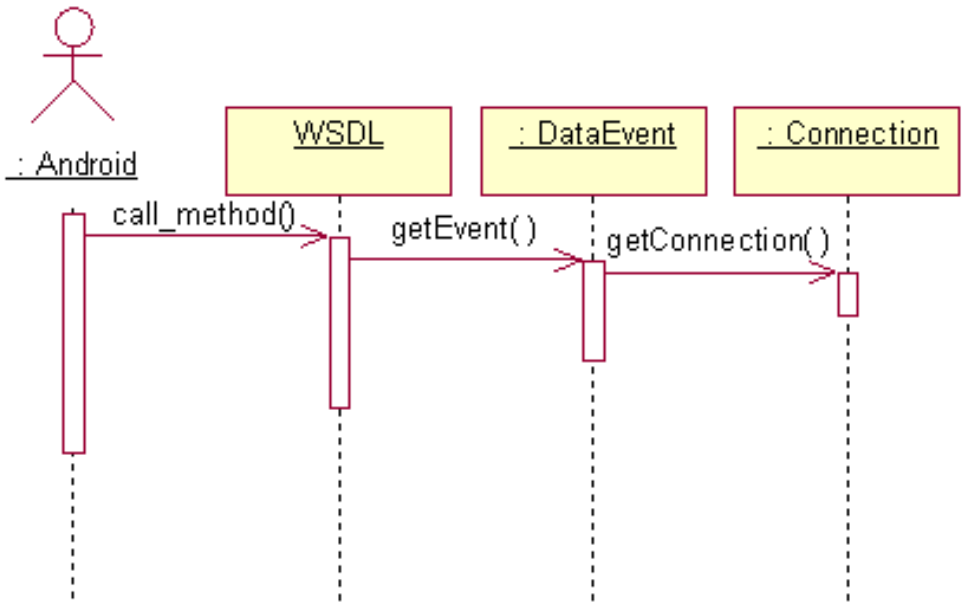
Gambar III.25 sequence diagram hapus event

3.1.4.5.3.4 Login



Gambar III.26 Sequence diagram login

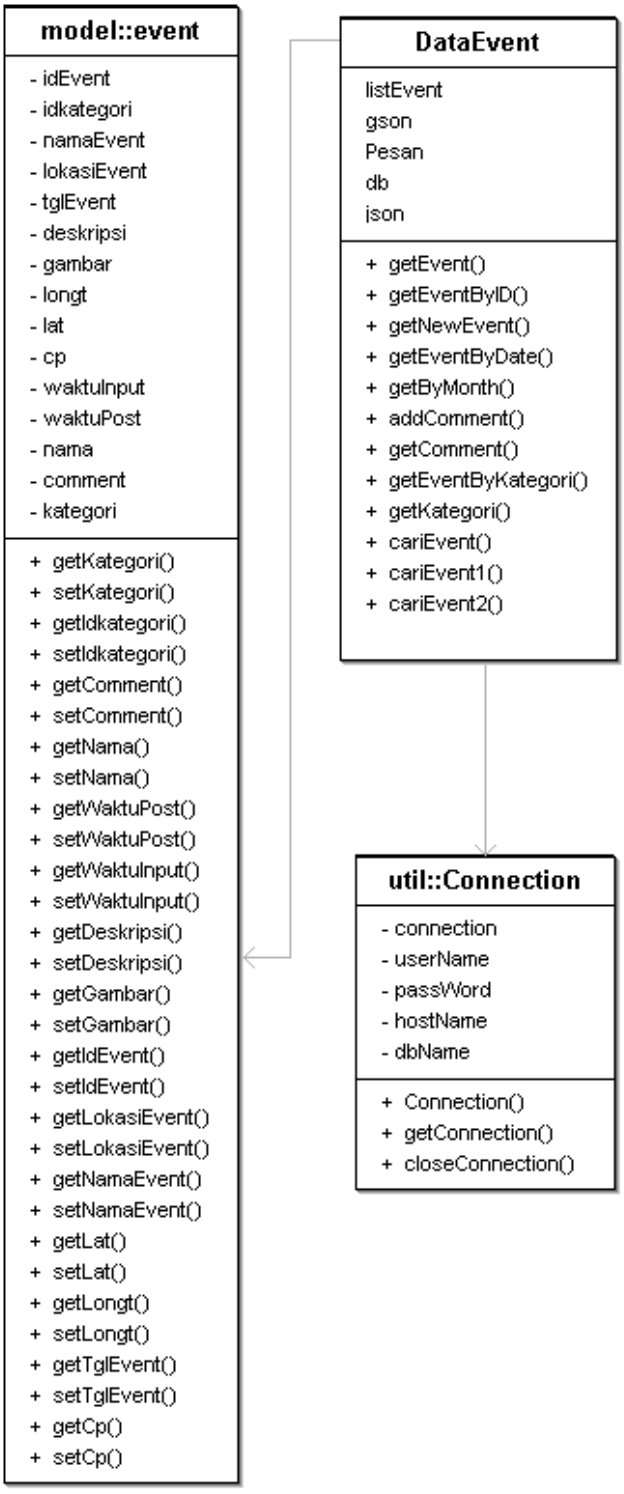
3.1.4.5.3.5 Proses Service Request



Gambar III.27 Sequence diagram proses service request

3.1.4.5.4 Class Diagram

Berikut ini gambaran hubungan antar class yang terjadi pada sistem web service yang akan dibangun. Sedangkan detail dari masing-masing *class*dapat dilihat pada lampiran A.



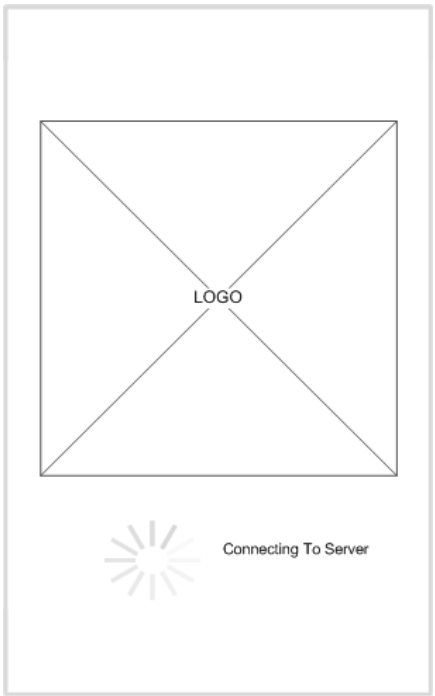
Gambar III.28Class diagram sistem web service

3.2 Perancangan

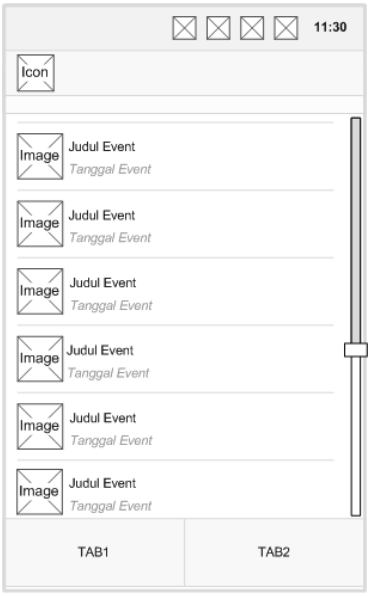
3.2.1 Perancangan Antarmuka

Interface atau antar muka merupakan tampilan dari suau program aplikasi yang berperan sebagai media komunikasi yang digunakan sebagai sarana berdialog antara program dengan *user.* Sistem yang akan dibangun diharapkan menyediakan *interface* yang mudah dipahami dan digunakan oleh *user*. Berikut ini adalah perancangan antarmuka pada sistem yang akan dibuat.

3.2.1.1 Perancangan antarmuka android



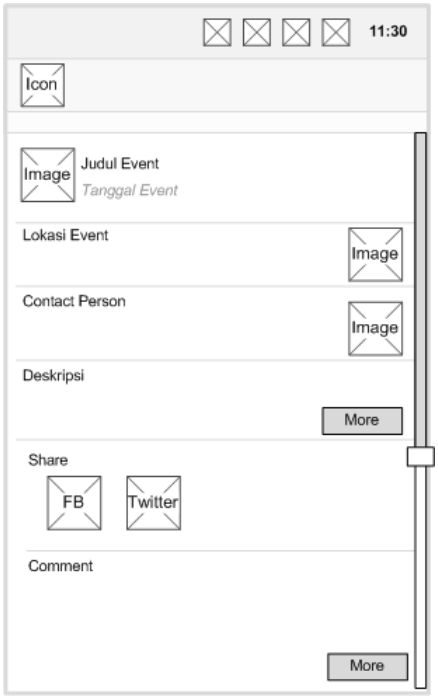
Gambar III.29 antarmuka android splash screen



Gambar III.30antarmuka android menu utama list view



Gambar III.31antarmuka android menu utama calendar view



Gambar III.32antarmuka android detail

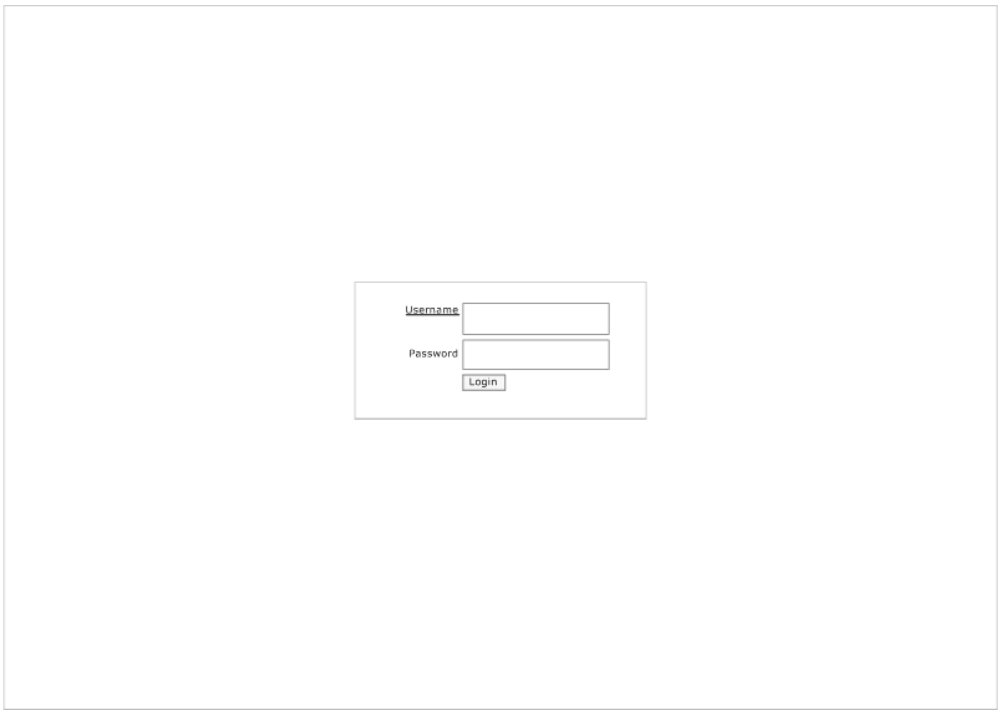


Gambar III.33antar muka menu reminder

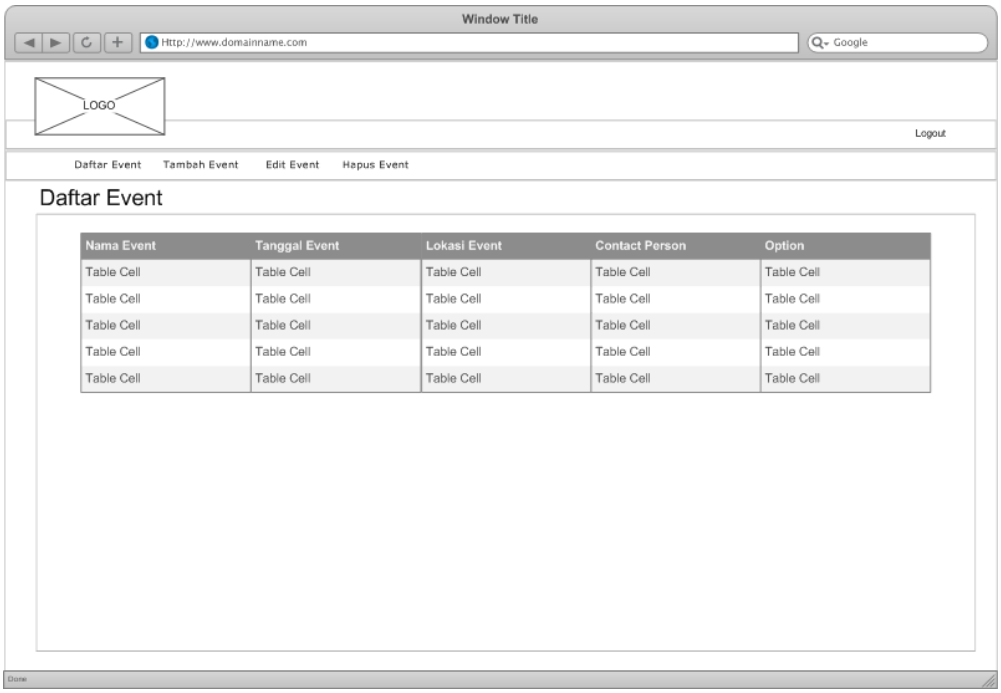


Gambar III.34antarmuka menu map view

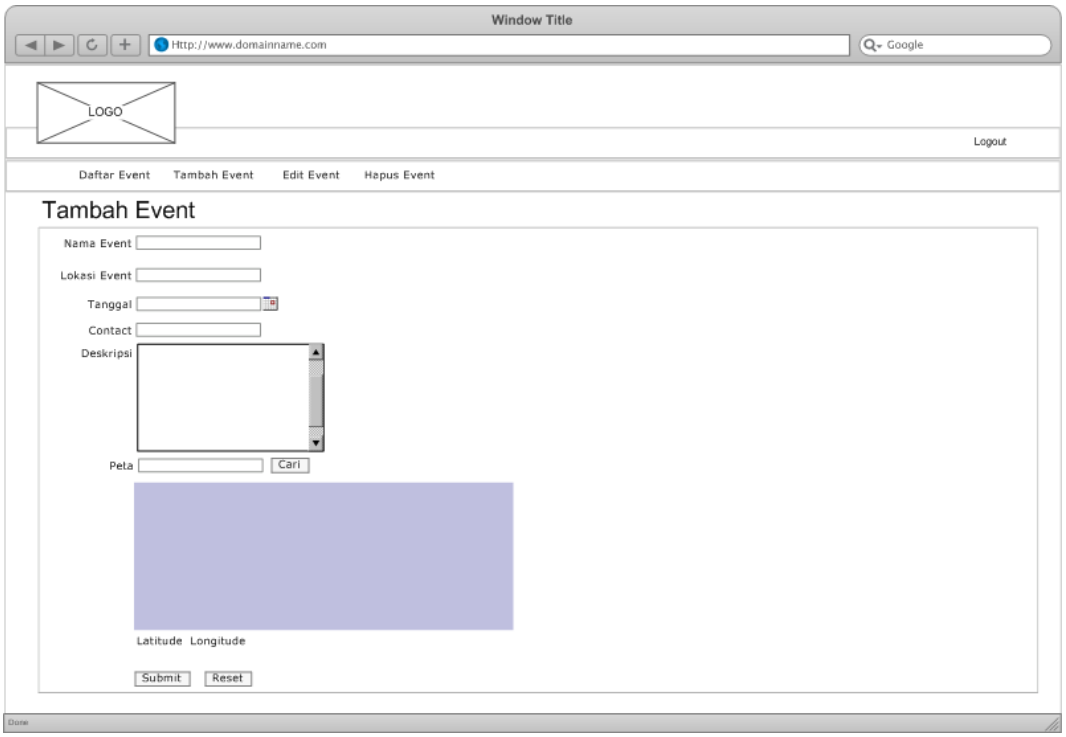
3.2.1.2 Perangcangan antarmuka admin



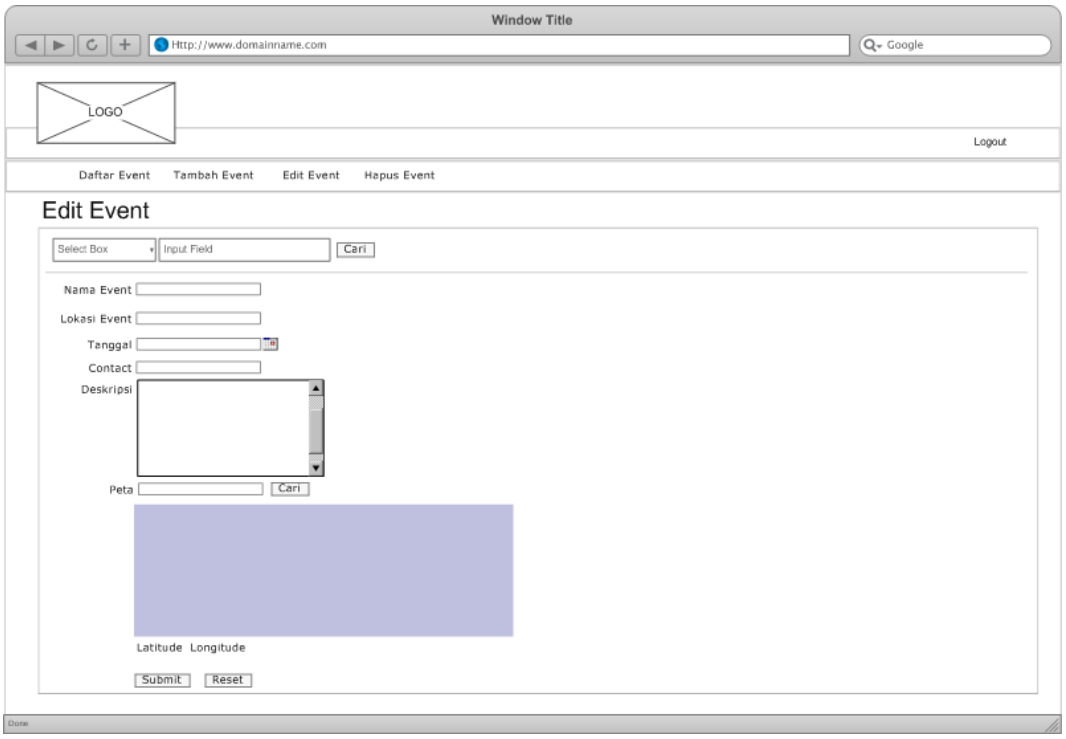
Gambar III.35 antarmuka admin login



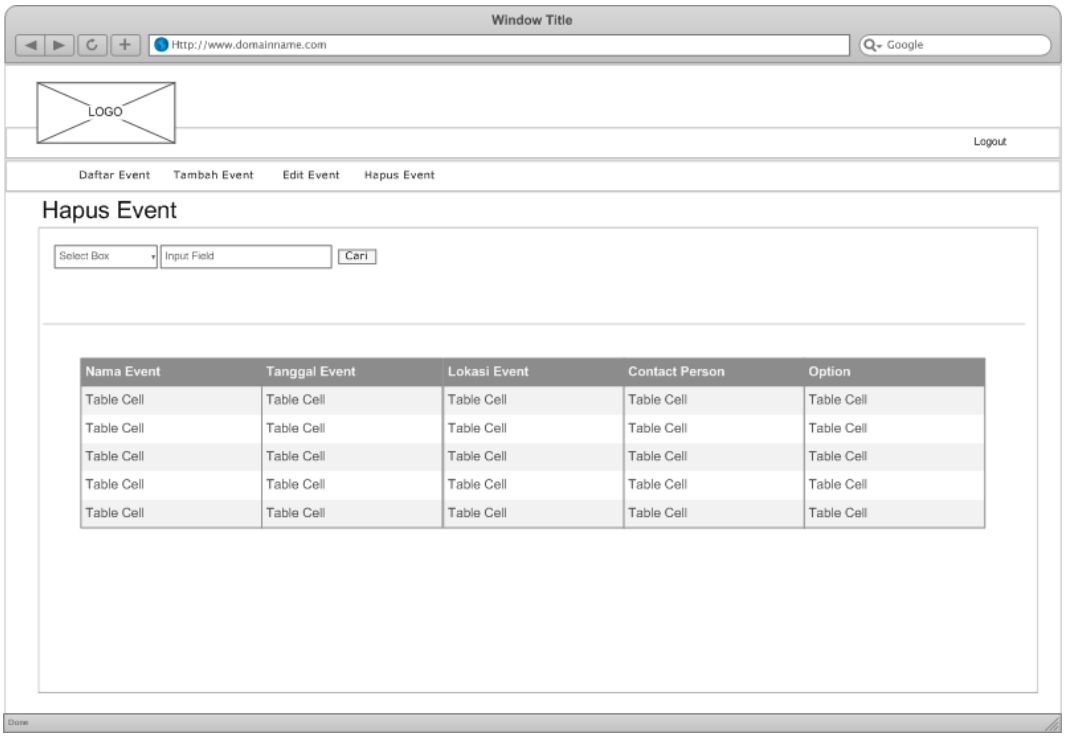
Gambar III.36 antarmuka admin halaman utama



Gambar III.37 antarmuka admin tambah event



Gambar III.38 antarmuka admin edit event



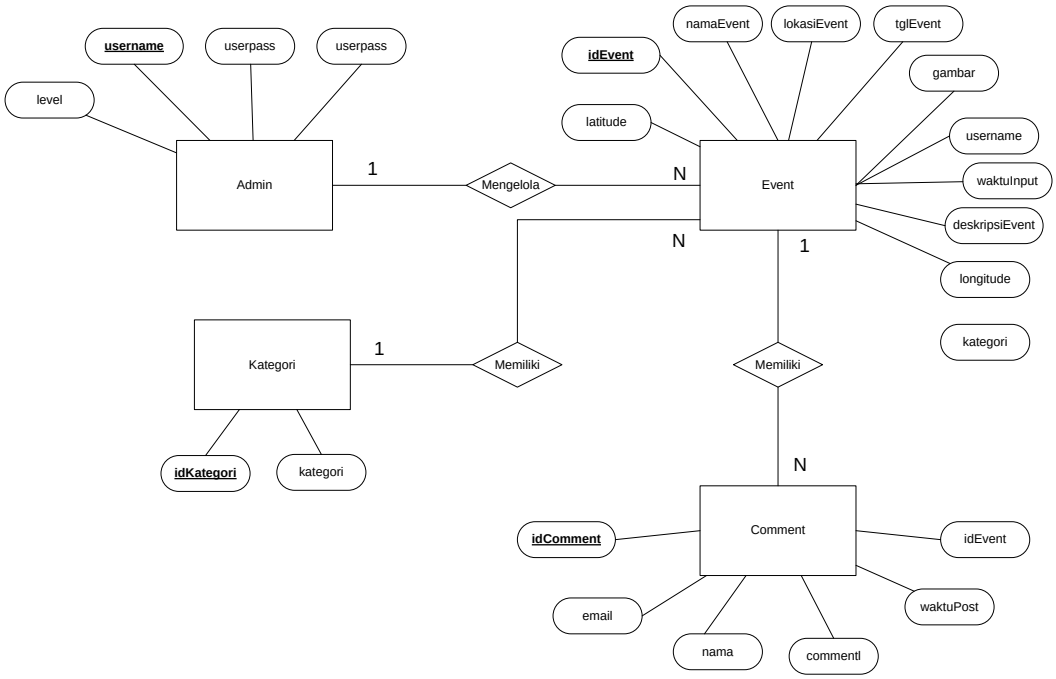
Gambar III.39 antarmuka admin hapus event

3.2.2 Perancangan Basis Data

Perancangan basis data adalah menciptakan atau merancang kumpulan data yang terhubung dan tersimpan secara bersama – sama.

3.2.2.1 Entity relationship Diagram

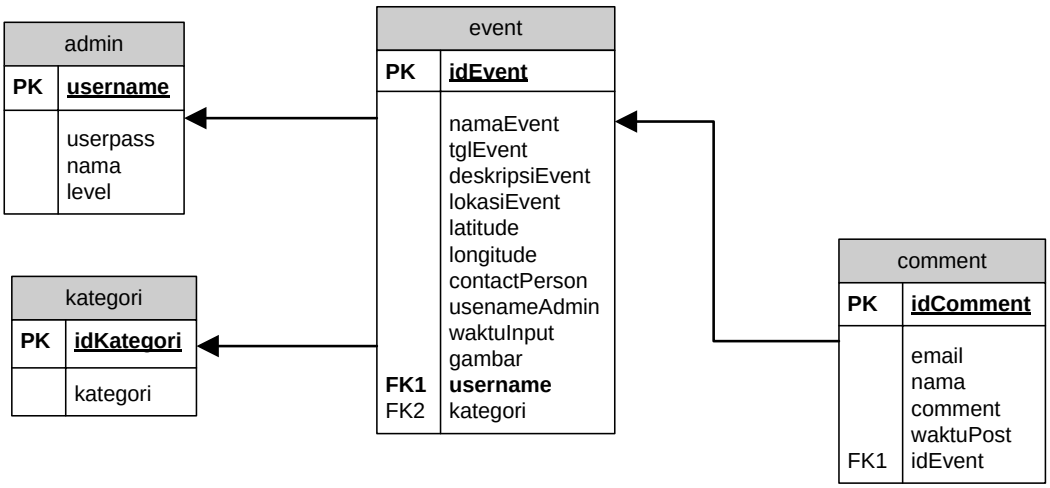
Dalam memodelkan data dan menggambarkan hubungan antara data yang ada pada sistem digunakan alat bantu yaitu diagram E-R. Sistem yang akan di bangun dapat dilihat hubungan antar entitas. Untuk itu sistem yang akan dirancang memiliki usulan ERD yaitu:



Gambar III.40 ER Diagram

3.2.2.2 Skema Relasi

Skema relasi menggambarkan hubungan antar data, arti data dan batasannya dijelaskan dengan baris dan kolom.



Gambar III.41 Skema Relasi

3.2.2.3 Perancangan Struktur Tabel

3.2.2.3.1 Struktur Tabel Pada Server

Tabel-tabel yang terdapat dalam basis data yang digunakan dalam sistem ini adalah sebagai berikut :

1. Tabel Admin

Tabel ini adalah perancangan untuk tabel admin yang akan digunakan pada pembangunan sistem. Struktur tabel admin adalah sebagai berikut :

Tabel III.18 Struktur tabel admin

No	Nama Field	Type	Ukuran	Keterangan
1	Username	Varchar	8	*Primary Key
2	Userpass	Varchar	20	
3	Nama	Varchar	50	
4	Level	Enum	(‘ADMIN’, ’SUPERADMIN’)	

2. Tabel Event

Tabel ini adalah perancangan untuk tabel event yang akan digunakan untuk menampung seluruh data event. Struktur tabel event adalah sebagai berikut:

Tabel III.19 Struktur tabel event

No	Nama Field	Type	Ukuran	Keterangan
1	idEvent	Integer	5	*Primary Key
2	namaEvent	Varchar	50	
3	tglEvent	Date		
4	lokasiEvent	Varchar	100	
5	deskripsiEvent	Text		
6	gambar	Varchar	50	
7	latitude	Decimal	8,5	
8	longitude	Decimal	8,5	
9	contactPerson	Varchar	14	
10	waktuInput	Timestamp		
11	usernameAdmin	Varchar	8	**Foreign key
12	Kategori	Integer	5	**ForeignKey

3. Tabel Comment

Tabel ini adalah perancangan untuk tabel comment yang akan digunakan untuk menampung seluruh komentar yang user masukan. Struktur tabel comment adalah sebagai berikut:

Tabel III.20 Struktur tabel comment

No	Nama Field	Type	Ukuran	Keterangan
1	idComment	Integer	8	*Primary Key
2	Email	Varchar	50	
3	Nama	Varchar	50	
4	Comment	Varchar	200	
5	waktuPost	Timestamp		
6	idEvent	Integer	5	**ForeignKey

4. Tabel Kategori

Tabel ini adalah perancangan untuk tabel kategori yang akan digunakan untuk menampung kategori dari event. Struktur tabel kategori adalah sebagai berikut:

Tabel III.21 Struktur tabel kategori

No	Nama Field	Type	Ukuran	Keterangan
1	idComment	Integer	8	*Primary Key
2	kategori	Varchar	50	