BAB 3

ANALISIS DAN PERANCANGAN SISTEM

Analisis dan perancangan sistem memerlukan tahapan yang sistematis untuk mendapatkan aplikasi yang baik dan bersesuaian dengan kegunaan dan tujuannya. Tahap awal dari analisis adalah menganalisis kebutuhan-kebutuhan sistem mulai dari kubutuhan pengguna, kubutuhan non fungsional, dan kebutuhan fungsional. Sedangkan untuk tahap perancangan aplikasi yaitu perancangan database, dan perancangan antarmuka.

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat sesuai dengan kebutuhan.

3.1.1 Analisis Masalah

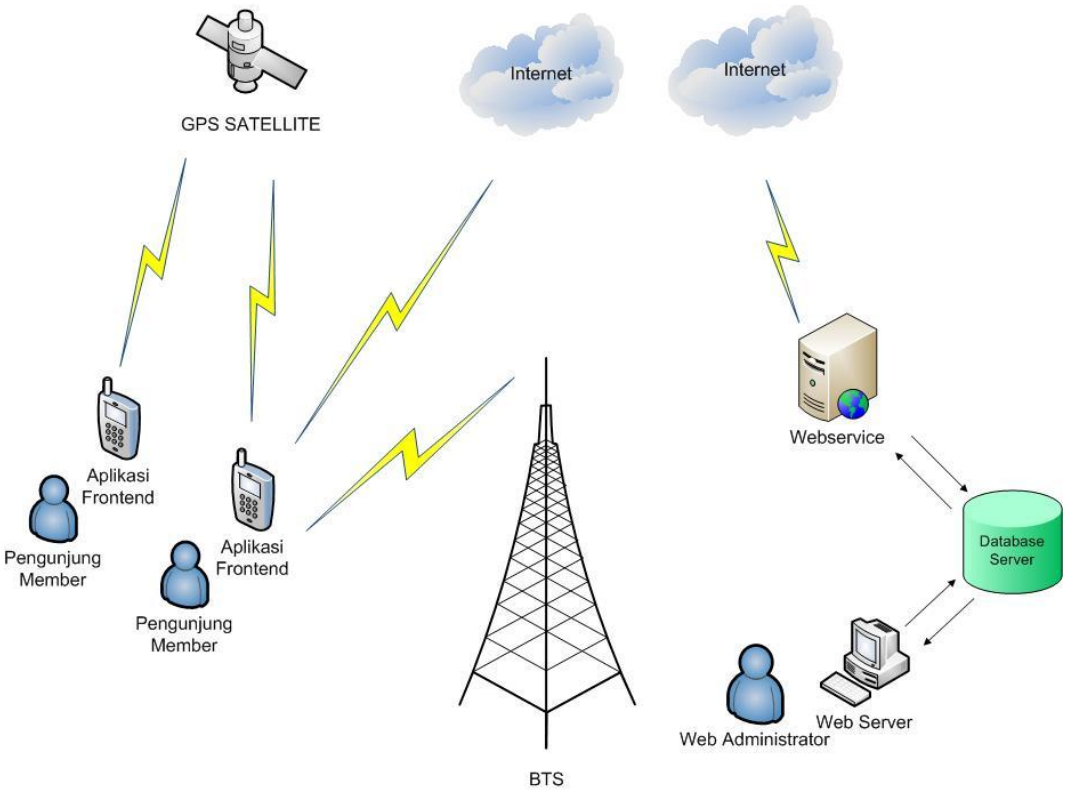
Sulitnya menemukan informasi mengenai Klub olahraga dan kejuaraan olahraga menjadi kendala bagi semua orang yang sedang mencari informasi mengenai lokasi dan data-data klub olahraga serta event kejuaraan olahraga. Ini semua disebabkan karena kurang atau bahkan tidak adanya informasi mengenai lokasi dan data-data klub olahraga serta kejuaraan olahraga. Cara yang dapat dilakukan untuk mendapatkan informasi tersebut adalah dengan bertanya kepada

masyarakat lain atau berkunjung ke kantor KONI untuk mendapatkan informasi yang dicari.

Namun cara tersebut kurang efektif, mengingat bahwa orang di sekitar belum tentu mengetahui informasi yang dibutuhkan serta KONI pun karena sistem yang berjalan masih konvensional, data-data yang ada tersimpan dalam bentuk kertas dan arsip-arsip yang bertumpukan yang pastinya membutuhkan waktu yang lama untuk mencari informasi yang dibutuhkan. Oleh karena itu dibutuhkan sebuah sistem yang efektif, cepat dan mudah digunakan dalam memberikan informasi tersebut yaitu dalam sebuah aplikasi *mobile*. Karena dengan perangkat *mobile* aplikasi ini dapat digunakan kapan saja dan dimana saja asalkan terintegrasi dengan jaringan internet.

3.1.2 Deskripsi Sistem

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



Gambar 3. 1 Arsitektur sistem

Pada Gambar 3.1 menjelaskan alur sistem yang akan dibangun. Pada sistem tersebut terdapat dua aplikasi yang akan dibangun yaitu Aplikasi backend dan Aplikasi Frontend. Pada Aplikasi *Backend* terdapat beberapa sistem pendukung diantaranya yaitu *web administrator*, *web server*, *database server* dan *web services*. Aplikasi backend ini berupa aplikasi web yang akan berfungsi mengolah semua data dan sebagai penghubung dengan Aplikasi *Frontend*. Pada aplikasi *Backend* ini diperlukan koneksi internet untuk proses komunikasi data. Aplikasi *Frontend* digunakan pada SmarthPhone ber-sistem operasi Android, pada Aplikasi *Frontend* ini terdapat semua informasi mengenai klub olahraga, kejuaraan olahraga dan berita olahraga. Selain itu pada aplikasi ini juga tersedia fitur peta lokasi yang akan menampilkan lokasi klub dan kejuaraan olahraga serta

jalur terdekat yang dapat ditempuh untuk mencapai tujuan lokasi. Untuk dapat menjalankan semua fitur yang ada pada aplikasi *frontend* diperlukan koneksi internet untuk mengakses data dan diperlukan juga koneksi *gps satellite* untuk mendapatkan informasi mengenai lokasi tempat. Untuk lebih jelasnya mengenai Aplikasi *Frontend* dan Aplikasi *Backend* dijelaskan pada sub bab berikut.

3.1.2.1 Aplikasi *Frontend*

Aplikasi *frontend* yaitu Aplikasi Klub olahraga dan kejuaraan olahraga. Aplikasi ini berfungsi sebagai antarmuka antara pengguna dan sistem. Untuk aplikasi *frontend* ini terdapat 2 user yaitu pengunjung biasa dan member. Pengunjung biasa (masyarakat umum) hanya dapat mengakses semua data yang telah disediakan secara menyeluruh. Sedangkan untuk member (Pelatih, Atlit dan pecinta olahraga tertentu) dapat melakukan login dan data yang ditampilkan adalah data hasil kustomisasi (pemilihan) data cabor yang telah dilakukan pada aplikasi backend. Untuk menjadi member, pengunjung dapat melakukan registrasi pada aplikasi backend (web) yang telah disediakan. Pada aplikasi *frontend* ini diperlukan koneksi internet untuk mengakses data dan *gps satellite* untuk mengetahui informasi posisi lokasi.

3.1.2.1.1 Analisis Penentuan Lokasi Dengan GPS

Pada penelitian ini metode yang digunakan untuk menentukan posisi suatu lokasi adalah metode *Cell-ID* (*Cell Identification*) yaitu penentuan posisi didasarkan pada daerah geografis yang tercakup oleh sebuah cell berhubungan dengan daerah cakupan dari sinyal radio. Ketika sebuah handphone terhubung secara aktif dengan sebuah base station, berarti handphone tersebut diasumsikan

berada dalam cell dari base station tersebut. Untuk mengukur jarak dan arah handset dari base station tidak dapat diketahui dengan pasti. Oleh karena itu, untuk lebih meningkatkan lagi akurasi hasil pencarian, metode *Cell ID* ini dapat dikombinasikan dengan metode *Timing Advanced* (TA), dengan menggunakan TA ini, metode Cell ID akan ditambahkan sebuah fungsionalitas untuk menghitung *Round Trip Time* (RTT), yaitu waktu transmisi sebuah frame (dari base station ke handphone) dan waktu penerimaan sebuah frame (dari handphone ke base station). Dengan tambahan metode ini, jarak antara handphone dan base station dapat ditentukan dengan keakuratan 50 m.

Berikut ini salah satu rumus yang sering digunakan untuk menentukan jarak lokasi pada metode *CELL ID* dengan mengabaikan bentuk geografis baik bukit maupun lembah.

Jarak = **Arc Cos (Cos(Lat1) x Cos(Lat2) x Cos(Long2-Long1) +**

Sin(Lat1) x Sin(Lat 2)) x R

Dimana

Lat1 = Nilai Latitude dari Lokasi Awal
Long1 = Nilai Longitude dari Lokasi Awal
Lat2 = Nilai Latitude dari Lokasi Tujuan
Long2 = Nilai Latitude dari Lokasi Tujuan
R = Jari-jari rata-rata Bumi (6378.2 KM)
Catatan : Nilai Lat dan Long dalam satuan radian.

Contoh : Misalkan untuk mencari jarak antara Bandara Sultan Iskandar Muda di Banda Aceh ke Bandara Sentani di Jayapura Jika diketahui :

1. Latitude 1 (Banda Aceh) 5°31'1"N dan Longitude 1 (Banda Aceh) 95°25'12"E
2. Latitude 2 (Jayapura) 2°34'19"S dan Longitude 2 (Jayapura)140°30'41"E

Koordinat diatas kan masih dalam bentuk degree minute secon dirubah ke bentuk degree decimal

1. Latitude 1 (Banda Aceh) 5.516944 dan Longitude 1 (Banda Aceh) 95.42
2. Latitude 2 (Jayapura) -2.571944 dan Longitude 2 (Jayapura) 140.511389

Perhitungan :

Jarak = Arc Cos (Cos(5.516944 * π / 180) *

Cos(-2.571944 * π / 180) *

Cos((140.511389 * π / 180) – (95.42* π / 180)) +

Sin(5.516944 * π / 180) *

Sin(-2.571944 * π / 180)) * 6378.2

= Arc Cos (0.6976857) * 6.371

= 5093.849 KM

3.1.2.1.2 Elemen Google Maps API

Beberapa elemen dan objek yang terdapat pada *Google Maps API* yang diterapkan di aplikasi:

1. *Markers*

   Marker adalah fungsionalitas simbol yang menandakan suatu titik koordinat lokasi. marker digunakan untuk menandai lokasi seperti klub olahraga dan kejuaraan olahraga pada peta *Google Maps*.

2. *Directions*

   Adalah *utility* yang menangani masalah pencarian rute. Directions ini digunakan untuk memandu rute jalan dari pengguna menuju tempat lokasi klub olahraga dan kejuaraan olahraga pada peta *Google Maps*.

3. *GeoCoding and Reverse GeoCoding*

   Adalah *utility* untuk memproses konversi alamat yang dapat digunakan untuk menempatkan penanda atau posisi peta.

4. *Location Manager*

   adalah API yang paling penting, dalam memberikan akses ke layanan sistem lokasi dengan pemetaan dan GPS. Dalam hal ini *Location Manager* berfungsi untuk mendapatkan update periodik lokasi geografis perangkat dengan memetakan nilai latitude dan longitude kedalam *Google Maps*.

5. *Overlays Overview*

   Adalah objek di *Google Maps* yang terkait dengan koordinat lintang dan bujur, sehingga objek ini bergerak saat ditarik atau zoom pada peta. *Overlays* mencerminkan benda yang ada pada peta untuk menunjukan titik, garis, area, atau *markers* benda.

6. *Loading Google Maps API* koneksi script yang dibuat keserver Google Map API dengan menggunakan *API key* yang didapatkan pada saat mendaftar ke *Google Maps API*.

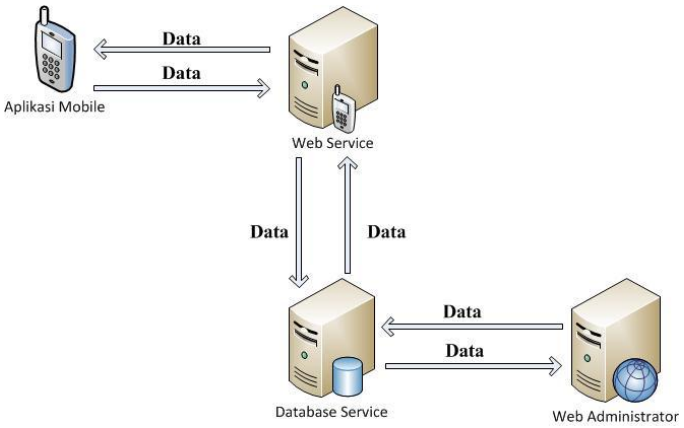
7. *Map DOM Elements*

   Untuk menampilkan map pada halaman web, umumnya menggunakan *div tag* untuk menampungnya. Diharuskan membuat sebuah *div tag* dengan nama "*map_canvas*", dan mereferensikan *element* ini ke *Document Object Model* (DOM).

3.1.2.3 Aplikasi *Backend*

Aplikasi *Backend* berfungsi sebagai penghubung antara aplikasi *frontend* dengan *database server*. Pada aplikasi *backend* ini terdapat tiga bagian yaitu *web administrator, webservice dan database server*. *Web administrator* berfungsi untuk mengolah data yang dioperasikan oleh admin. Pada Aplikasi *Backend* ini admin mengolah berbagai data yaitu data cabor, data induk organisasi, data klub olahraga serta data kejuaraan olahraga. *Webservice* berfungsi untuk menghubungkan berbagai platform aplikasi yang dibangun dengan berbagai platform yang berbeda (contohnya Java, C#, VB, Delphi, PHP, Android) yang memungkinkan seluruh aplikasi untuk bisa berkomunikasi. *Webservice* yang digunakan untuk membangun sistem ini adalah JSON (*Javascript Object Notation*). *Database server* berfungsi untuk menyimpan data-data yang telah diolah. Manajemen data yang digunakan pada aplikasi ini adalah MySQL. Untuk aplikasi *backend* (web) ini terdapat 2 user yaitu administrator dan pengunjung biasa. Administrator bertugas untuk mengolah data, administrator yang bertugas adalah staff dari KONI Kab.Bandung. sedangkan untuk pengunjung biasa untuk melakukan registrasi menjadi member dan meng-kustomisasi data.

3.1.3 Analisis Alur Data Sistem



Gambar 3. 2 Alur Data Sistem

Analisis alur data yang terjadi pada sistem ini seperti berikut:

1. Aplikasi Mobile

   Aplikasi *mobile* dalam hal ini merupakan analisis fungsionalitas aplikasi untuk mengakses data dari *database server* melaui *web service*. Aplikasi ini dibangun diatas *platform* android dan bekerja dengan cara memparsing data dari *web service* untuk diolah pada *platform* android.

2. *Web service*

   Web service memudahkan beberapa aplikasi atau komponennya untuk saling berhubungan dengan aplikasi lain dalam sebuah organisasi maupun diluar organisasi menggunakan standar yang tidak terikat platform (*platform-neutral*) dan tidak terikat akan bahasa pemrograman yang digunakan (*language-neutral*).

   Web service ini berfungsi untuk menghubungkan aplikasi *frontend* dan aplikasi *backend*. Cara kerja *web service* ini yaitu dengan mengambil data

dari *database server* dan kemudian mengkonversikan data tersebut kedalam format pertukaran data JSON.

3. Protokol HTTP

   Protokol HTTP merupakan protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan menggunakan hipermedia. Ada beberapa operasi HTTP yang salah satu contohnya adalah GET dan POST. Operasi GET digunakan untuk meminta sumber dari server. Operasi POST digunakan untuk mengirim informasi ke sumber yang ada di server. Server akan merespon dengan header HTTP yang diikuti dengan bagian data. Pada kasus operasi pemberian GET, respon dari server akan berisi sumber yang diminta klien. Jika server menerima operasi POST, server akan melewatkan data yang ada pada operasi POST ke sumber yang dapat dieksekusi. Setelah server menerima kembali hasil dari sumber tersebut, server akan memasukkan hasil tersebut ke HTTP respon yang dikirim kembali ke klien.Pada penelitian ini HTTP digunakan aplikasi *mobile* dan *web* administrator sebagai protokol yang dapat mendistribusikan data yang bersumber dari *database server*.

3.1.4 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang menitikberatkan pada properti prilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras, perangkat lunak, serta *user* sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.1.4.1 Analisis Kebutuhan Perangkat Keras

Perangkat Keras yang dibutuhkan berdasarkan kebutuhan minimal yang harus terpenuhi antara lain :

1. Perangkat Keras Aplikasi *Backend*

   Pada aplikasi *backend* yang berupa web, aplikasi dijalankan pada sebuah perangkat komputer dengan spesifikasi kebutuhan perangkat keras komputer sebagai berikut :

   1.1 Prosesor Dengan Kecepatan Minimum 2.0 GHZ

   1.2 RAM 512MB

   1.3 Hardisk 20 GB

   1.4 Mouse, Keyboard Monitor dan Network Interface Card

2. Perangkat Keras Aplikasi *Frontend*

   Aplikasi *frontend* dijalankan pada sebuah perangkat *SmartPhone* yaitu *SmartPhone* dengan sistem operasi Android. Kebutuhan perangkat keras untuk *SmartPhone* tersebut yaitu :

   2.1 Resolusi 240 x 320 pixels

   2.2 Memori 100 MB

   2.3 CPU 300 MHz

   2.4 WLAN Wi-fi 802.11 b/g/n

   2.5 Jaringan 3G UMTS

   2.6 GPS A-GPS

3.1.4.2 Analisis Kebutuhan Perangkat Lunak

Analisis perangkat lunak terdiri dari spesifikasi minimum perangkat yang dibutuhkan dan elemen google maps api.

a. Sepesifikasi Minimum Perangkat Lunak

Spesifikasi perangkat lunak untuk membangun aplikasi ini adalah sebagai berikut:

1. Sistem Operasi Windows & Professional.
2. Aplikasi IDE Eclipse Galileo.
3. JDK v 1.6 dan Android SDK windows.
4. ADT v.10.0.0.
5. Platform Android versi. 2.2 keatas.
6. PHP untuk koneksi ke basis data dengan *script* PHP.
7. Pengedit Teks untuk penulisan *script* PHP dengan dreamweaver.
8. Basis Data MySQL.
9. *Browser* Mozilla Firefox.

3.1.4.3 Analisis Pengguna Sistem (User)

Analisis dan kebutuhan pengguna dimaksudkan untuk mengetahui seberapa jauh pengguna sistem ini. Hal ini bertujuan untuk mengetahui karakteristik pengguna dalam mengoperasikan sistem yang telah ada, maupun sistem usulan.

Berdasarkan hasil pengamatan, pengguna yang terlibat dalam sistem yang sedang berjalan ialah:

1. Admin, sebagai pengelola aplikasi backend (web) yang dapat menambah, mengedit dan menghapus data.
2. Pengunjung, sebagai pengguna sistem. Pengunjung ini terdiri dari pengunjung biasa dan member.

Berikut karakteristik pengguna sistem yang ditunjukan oleh table dibawah ini

Tabel 3. 1 Karakteristik Pengguna Aplikasi *Backend* (*web*)

Kategori Pengguna	Hak Akses	Tingkat Pendidikan	Tingkat Keterampilan
Pengunjung Biasa	Mengakses informasi di web KONI	-	Mengenal internet dan terbiasa *browsing*
Admin	Mengakses dan mengelola sistem	Yang Menguasai di bidangnya	Mengerti kinerja komputer dan internet

Untuk menggunakan sistem yang dibangun terdapat beberapa karakteristik yang harus dimiliki oleh pengguna, kebutuhan karakteristik dapat dilihat pada table dibawah ini.

Tabel 3. 2 Karakterisitik Pengguna Aplikasi *Frontend*

Kategori Pengguna	Hak Akses	Tingkat Pendidikan	Tingkat Keterampilan
Pengunjung Biasa	Mengakses informasi yang telah disediakan melalui aplikasi	Minimal SMP	Minimal membuka aplikasi pada *mobile* android
Member	Mengakses informasi berdasarkan hasil kustomisasi data	Minimal SMP	Minimal membuka aplikasi pada *mobile* android

3.1.5 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan.

Analisis kebutuhan fungsional ini meliputi analisis kebutuhan sistem, analisis kebutuhan data, spesifikasi sistem dan pemodelan sistem.

Pemodelan sistem dimodelkan dengan menggunakan *UML (Unified Modeling Language)*. Tahap-tahap pemodelan dalam analisis tersebut antara lain *Use case diagram, Class Diagram, Activity Diagram, Sequence Diagram, Component Diagram dan Deployment Diagram.*

3.1.5.1 Analisis Kebutuhan Sistem

Sebelum melakukan perancangan sistem, terlebih dahulu dilakukan analisis kebutuhan sistem yaitu dengan cara melakukan wawancara terhadap responden. Hal ini dimaksudkan agar dapat mengatasi ketidaksesuaian antara aplikasi yang dirancang dengan kebutuhan pengguna. Adapun kebutuhan sistem yang diperlukan antara lain :

1. Menampilkan informasi mengenai data induk organisasi, data klub olahraga dan kejuaraan olahraga.
2. Memberikan informasi data induk organisasi, data klub olahraga dan kejuaraan olahraga dengan lengkap dan rinci.
3. Menampilkan peta lokasi serta direction suatu induk organisasi, klub olahraga dan kejuaraan olahraga
4. Memberikan informasi mengenai Berita Olahraga dari portal – portal penyedia berita olahraga.
5. Memberikan informasi data dan peta lokasi induk organisasi, klub olahraga serta kejuaraan olahraga secara umum dan sesuai kustomisasi data.

3.1.5.2 Analisis Kebutuhan Data

Setelah melakukan proses analisis kebutuhan sistem, maka analisis selanjutnya adalah analisis kebutuhan data. Analisis ini bertujuan untuk memudahkan dalam perancangan informasi. Adapun analisis kebutuhan data yang diperlukan antara lain :

1. Data Cabang Olahraga (cabor)

   Data yang akan ditampilkan adalah nama cabor.

2. Data Induk Organisasi

   Data yang akan ditampilkan adalah nama induk organisasi, nama ketua, nama sekretaris dan logo.

3. Data Klub olahraga

   Data yang akan ditampilkan adalah nama klub, tahun berdiri, nama pelatih, nama assisten, nama penghubung, telepon, prestasi klub dan logo.

4. Data Kejuaraan Olahraga

   Data yang akan ditampilkan adalah nama kejuaraan, waktu pelaksanaan kejuaraan, alamat pendaftaran, fax, deskripsi kejuaraan dan file.

5. Data Lokasi cabor, Klub olahraga dan kejuaraan olahraga

   Menampilkan lokasi cabor, klub olahraga serta kejuaraan olahraga.

6. Data Member

   Menampilkan nama member, status member, tanggal lahir, alamat, email dan foto.

6. Data Kustomisasi

   Menampilkan data kustomisasi meliputi id member dan cabor yang dipilih.

3.1.5.3 Spesifikasi Sistem

Adapun spesifikasi sistem yang dibutuhkan antara lain :

1. Sistem dikembangkan dengan model arsitektur *two tier*. Aplikasi *frontend* sebagai aplikasi *On Device Portal* (ODP). Sedangkan disisi *server* atau *backend* akan dikembangkan aplikasi web.
2. Konektivitas menggunakan koneksi protocol HTTP dengan memanfaatkan jaringan seluler yaitu GPRS/EDGE/UMTS atau WIFI.
3. Sistem informasi yang disediakan dibatasi pada informasi induk organisasi, informasi klub olahraga, kejuaraan olahraga, lokasi dan berita olahraga.
4. Spesifikasi sistem disisi *frontend* yaitu :
   a. Perangkat yang digunakan harus bisa digunakan dimana saja dan kapan saja serta memiliki akses internet.
   b. Aplikasi ini menyediakan informasi tentang lokasi-lokasi yang berhubungan dengan induk organisasi, klub olahraga, kejuaraan olahraga, berita olahraga, alamat lokasi, lokasi pada peta menggunakan *google maps*, dan lokasi *realtime* pengguna dengan perangkat GPS yang terintegrasi.
   c. Untuk pengunjung yang sebelumnya telah melakukan registrasi pada aplikasi *backend* serta telah melakukan kustomisasi data dapat melakukan login dan dapat melihat data sesuai dengan cabor yang dipilih.

d. Aplikasi ini memiliki ukuran yang minimum sehingga mampu disimpan dalam perangkat seluler yang memiliki keterbatasan.

5. Spesifikasi sistem di *backend* yaitu:

a. Sistem dapat memfasilitasi pengelolaan konten informasi.

b. Sistem dapat memfasilitasi Admin untuk melakukan pengolahan data konten informasi cabor, klub olahraga, kejuaraan olahraga.

c. Aplikasi *backend* menyediakan fitur klub olahraga, kejuaraan olahraga serta registrasi untuk pengunjung yang ingin menjadi member agar bisa login pada aplikasi frontend. Registrasi mengisi data yang dibutuhkan untuk menjadi member. Setelah melakukan registrasi pengunjung melakukan kustomisasi data untuk memilih data cabor apa yang diinginkan. Proses kustomisasi ini bertujuan untuk pemilihan data cabor oleh pengguna yang ingin ditampilkan pada aplikasi *frontend.*

3.1.5.4 Pemodelan Sistem

Pada tahap ini dilakukan pemodelan sistem untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem aplikasi serta menentukan kelas yang dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

a. *Use Case* Diagram

Diagram *Use Case* merupakan bagian tertinggi dari fungsionalitas yang dimiliki sistem yang akan menggambarkan bagaimana seseorang atau aktor akan menggunakan dan memanfaatkan sistem. Diagram ini juga mendeskripsikan apa

yang akan dilakukan oleh sistem. *Use Case* terdiri dari tiga bagian yaitu identifikasi aktor, identifikasi *Use Case* dan skenario *Use Case*.

1. Identifikasi Aktor

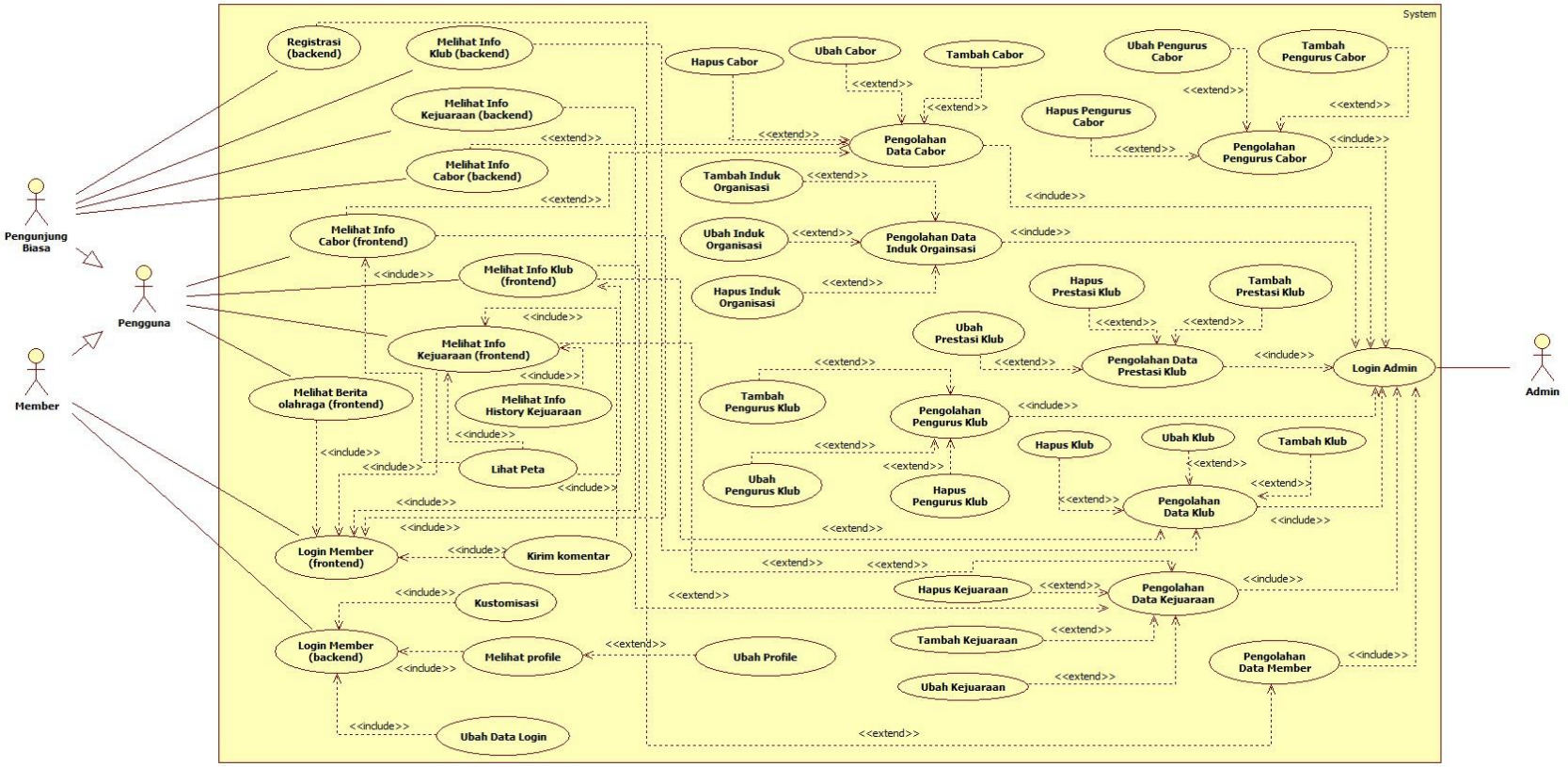
Aktor yang berperan dalam menjalankan sistem dapat dilihat pada Tabel 3.3

Tabel 3. 3 Aktor Sistem

No	Aktor	Deskripsi
P-01	Pengunjung Biasa	Merupakan aktor yang dapat mengakses aplikasi *frontend*. Pengunjung biasa juga dapat melakukan registrasi pada apikasi *backend* untuk menjadi member.
P-02	Admin	Merupakan aktor yang akan mengelola konten. Admin akan melakukan administrasi melalui Aplikasi web.
P-03	Member	Merupakan aktor yang menggunakan aplikasi *frontend* yang telah melakukan proses registrasi dan kustomisasi data pada aplikasi *backend*. Menampilkan informasi induk organisasi, klub olahraga dan kejuaraan olahraga sesuai dengan pilihan data cabor pada proses kustomisasi data.

Pada Gambar 3.3 merupakan interaksi antara aktor dengan sistem. Dimana terdiri dari 3 aktor. Pengunjung biasa, admin dan member. Aktor pertama yaitu Pengunjung biasa berinteraksi dengan aplikasi *frontend* dan *backend*, pengunjung dapat mengakses data dan lokasi induk organisasi, klub olahraga serta kejuaraan olahraga pada aplikasi *frontend* selain itu pengunjung juga dapat melakukan registrasi dan kustomisasi data pada aplikasi *backend*. Aktor kedua yaitu admin berinteraksi dengan aplikasi *backend* pada web untuk pengolahan data. Aktor

ketiga yaitu member berinteraksi dengan aplikasi *frontend*, member ini adalah pengguna biasa yang telah melakukan regitrasi pada aplikasi *backend*, member dapat mengakses data sesuai dengan data cabor yang telah dipilih pada saat proses kustomisasi data.



Gambar 3. 3 *Use case* Diagram

2. Identifikasi *Use Case*

Identifikasi *Use Case* yang terdapat dalam sistem dapat dilihat pada table 3.4 dibawah ini.

Tabel 3. 4 Deskripsi *Use Case*

No	*Use Case*	Deskripsi
UC-01	Melihat Info Cabor (*frontend)*	Fungsionalitas untuk memilih dan melakukan penjelasan cabang olahraga.
UC-02	Melihat Info Klub (*frontend)*	Fungsionalitas untuk memilih dan melakukan penjelasan klub olahraga.
UC-03	Melihat Info kejuaraan (*frontend)*	Fungsionalitas untuk memilih dan melakukan penjelasan kejuaraan olahraga.
UC-04	Melihat Berita Olahraga (*frontend)*	Fungsionalitas untuk memilih dan melakukan penjelasan berita olahraga.
UC-05	Login Member (*frontend)*	Fungsionalitas untuk proses Autentifikasi member pada aplikasi *frontend* yang ingin melihat informasi berdasarkan kustomisasi data.
UC-06	Melihat Info History Kejuaraan	Fungsionalitas untuk memilih dan melakukan penjelasan kejuaraan olahraga yang sudah dilaksanakan.
UC-07	Kirim Komentar	Fungsionalitas untuk mengirim komentar untuk suatu kejuaraan olahraga
UC-08	Login Member (*backend)*	Fungsionalitas untuk proses autentifikasi member pada aplikasi *backend.*
UC-09	Melihat Profile	Fungsionalitas untu melihat profile member.
UC-10	Ubah Profile	Fungsionalitas untuk mengubah data profile member.
UC-11	Kustomisasi	Fungsionalitas untuk kustomisasi data cabor.

UC-12	Ubah Login	Fungsionalitas untuk mengubah data login.
UC-13	Login Admin	Fungsionalitas untuk proses Autentifikasi Admin.
UC-14	Pengolahan Data Induk Organisasi	Fungsionalitas untuk melihat data induk organisasi.
UC-15	Pengolahan Data Cabor	Fungsionalitas untuk melihat data cabor.
UC-16	Pengolahan Pengurus Cabor	Fungsionalitas untuk melihat data pengurus cabor.
UC-17	Pengolahan Data Klub	Fungsionalitas untuk melihat data klub.
UC-18	Pengolahan Pengurus Klub	Fungsionalitas untuk melihat data pengurus klub.
UC-19	Pengolahan Data Prestasi Klub	Fungsionalitas untuk melihat data prestasi klub.
UC-20	Pengolahan Data Kejuaraan	Fungsionalitas untuk melihat data kejuaraan.
UC-21	Pengolahan Data Member	Fungsionalitas untuk melihat data member
UC-22	Tambah Induk Organisasi	Fungsionalitas untuk menambah induk organisasi.
UC-23	Ubah Induk Organisasi	Fungsionalitas untuk mengubah induk organisasi.
UC-24	Hapus Induk Organisasi	Fungsionalitas untuk menghapus induk organisasi.
UC-25	Tambah Cabor	Fungsionalitas untuk menambah cabor.
UC-26	Ubah Cabor	Fungsionalitas untuk mengubah cabor.
UC-27	Hapus Cabor	Fungsionalitas untuk menghapus cabor.
UC-28	Tambah Pengurus Cabor	Fungsionalitas untuk menambah pengurus cabor.

UC-29	Ubah Pengurus Cabor	Fungsionalitas untuk mengubah pengurus cabor.
UC-30	Hapus Pengurus Cabor	Fungsionalitas untuk menghapus pengurus cabor.
UC-31	Tambah Klub	Fungsionalitas untuk menambah klub.
UC-32	Ubah Klub	Fungsionalitas untuk mengubah klub.
UC-33	Hapus Klub	Fungsionalitas untuk menghapus klub.
UC-34	Tambah Pengurus Klub	Fungsionalitas untuk menambah pengurus klub.
UC-35	Ubah Pengurus Klub	Fungsionalitas untuk mengubah pengurus klub.
UC-36	Hapus Pengurus Klub	Fungsionalitas untuk menghapus pengurus klub.
UC-37	Tambah Prestasi Klub	Fungsionalitas untuk menambah prestasi klub.
UC-38	Ubah Prestasi Klub	Fungsionalitas untuk mengubah prestasi klub.
UC-39	Hapus Prestasi Klub	Fungsionalitas untuk menghapus prestasi klub.
UC-40	Tambah Kejuaraan	Fungsionalitas untuk menambah kejuaraan
UC-41	Ubah Kejuaraan	Fungsionalitas untuk mengubah kejuaraan
UC-42	Hapus Kejuaraan	Fungsionalitas untuk menghapus kejuaraan
UC-43	Registrasi (*backend)*	Fungsionalitas untuk mendaftar menjadi member
UC-44	Melihat Info Cabor (*backend)*	Fungsionalitas untuk melihat data cabang olahraga pada aplikasi *backend (web)*
UC-45	Melihat Info Klub (*backend)*	Fungsionalitas untuk melihat data klub olahraga pada aplikasi *backend (web)*
UC-46	Melihat Info Kejuaraan (*backend)*	Fungsionalitas untuk melihat data kejuaraan olahraga pada aplikasi *backend (web)*

3. Skenario *Use Case*

Skenario proses-proses yang terdapat didalam *Use Case* dapat dilihat pada tabel-tabel dibawah ini.

Tabel 3. 5 Skenario *Use Case* Melihat Info Cabor (*frontend)*

Identifikasi	
Nomor	UC-01
Nama	Melihat Info Cabor (*frontend)*
Tujuan	Untuk melihat detail dan lokasi cabor
Deskripsi	Data cabor dijadikan kategori untuk memilih klub olahraga dan kejuaraan olahraga
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Menampilkan menu utama
Aksi Aktor	Reaksi Sistem
1. Memilih menu info koni 3. Memilih salah satu induk organisasi 5. Memilih salah satu cabor 7. Memilih peta lokasi	2. Menampilkan daftar induk organisasi 4. Menampilkan daftar cabor 6. Menampilkan detail cabor yang dipilih 6. menampilkan peta lokasi cabor
Kondisi Akhir	Menampilkan detail dan lokasi cabor

Tabel 3. 6 Skenario *Use Case* Melihat Info Klub (*frontend)*

Identifikasi	
Nomor	UC-02
Nama	Melihat Info Klub (*frontend)*
Tujuan	Untuk Melihat detail dan lokasi klub olahraga
Deskripsi	Untuk melihat data klub olahraga yang dipilih berdasarkan kategori cabor
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Menampilkan menu utama
Aksi Aktor	Reaksi Sistem
1. Memilih menu info klub 3. Memilih salah satu cabor 5. Memilih salah satu klub 7. Memilih peta lokasi	2. Menampilkan daftar cabor 4. Menampilkan daftar klub 6. Menampilkan detail klub yang dipilih 8. Menampilkan peta lokasi klub
Kondisi Akhir	Menampilkan detail dan lokasi klub olahraga

Tabel 3. 7 Skenario *Use Case* Melihat Info Kejuaraan (*frontend)*

Identifikasi	
Nomor	UC-03
Nama	Melihat Info Kejuaraan (*frontend)*
Tujuan	Untuk Melihat detail dan lokasi kejuaraan olahraga
Deskripsi	Untuk melihat detail dan lokasi kejuaraan olahraga yang dipilih berdasarkan kategori cabor
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Menampilkan menu utama
Aksi Aktor	Reaksi Sistem
1. Memilih menu info kejuaraan 3. Memilih salah satu cabor 5. Memilih salah satu kejuaraan 7. Memilih peta lokasi	2. Menampilkan daftar cabor 4. Menampilkan daftar kejuaraan 6. Menampilkan detail kejuaraan yang dipilih 8. Menampilkan peta lokasi kejuaraan
Kondisi Akhir	Menampilkan detail dan lokasi kejuaraan olahraga

Tabel 3. 8 Skenario *Use Case* Melihat Berita Olahraga (*frontend)*

Identifikasi	
Nomor	UC-04
Nama	Melihat Berita Olahraga (*frontend)*
Tujuan	Melihat Berita Olahraga
Deskripsi	Untuk melihat berita olahraga terkini
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Menampilkan menu utama
Aksi Aktor	Reaksi Sistem
1. Memilih menu berita 3. Memilih salah satu menu berita olahraga 5. Memilih salah satu list berita olahraga	2. Menampilkan beberapa menu berita olahraga 4. Menampilkan list berita olahraga 6. Menampilkan detail berita
Kondisi Akhir	Menampilkan detail berita olahraga

Tabel 3. 9 Skenario *Use Case* Login Member (*frontend)*

Identifikasi	
Nomor	UC-05
Nama	Login Member (*frontend)*
Tujuan	Memastikan pengguna memiliki hak akses ke dalam sistem
Deskripsi	Aplikasi akan memeriksa apakah pengguna memiliki hak akses terhadap aplikasi dengan memasukan user name dan password
Aktor	Member
Skenario Utama	
Kondisi awal	Menampilkan halaman utama
Aksi Aktor	Reaksi Sistem
1. Memilih menu login 3. Member memasukan username dan password	2. menampilkan form login 4. Aplikasi akan memverifikasi apakah data yang dimasukan telah sesuai 5. Jika verifikasi berhasil maka aplikasi akan menampilkan halaman utama member 6. Jika verifikasi gagal maka aplikasi akan menampilkan pesan kesalahan login
Kondisi Akhir	Menampilkan Halaman Utama Member

Tabel 3. 10 Skenario *Use Case* Melihat Info Kejuaraan (*frontend)*

Identifikasi	
Nomor	UC-06
Nama	Melihat Info History Kejuaraan
Tujuan	Untuk Melihat history kejuaraan yang telah dilaksanakan
Deskripsi	Untuk melihat history kejuaraan yang telah dilaksanakan
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Menampilkan daftar kejuaraan
Aksi Aktor	Reaksi Sistem
1. Memilih menu history 3. Memilih salah satu kejuaraan	2. Menampilkan history kejuaraan 4. Menampilkan detail kejuaraan yang dipilih
Kondisi Akhir	Menampilkan detail history kejuaraan

Tabel 3. 11 Skenario *Use Case* Kirim Komentar

Identifikasi	
Nomor	UC-07
Nama	Kirim Komentar
Tujuan	Mengirim komentar terhadap suatu kejuaraan olahraga
Deskripsi	Untuk mengirim komentar seputar suatu kejuaraan olahraga
Aktor	Member
Skenario Utama	
Kondisi awal	Menampilkan detail kejuaraan
Aksi Aktor	Reaksi Sistem
1. Memilih tombol komentar 2. Mengisi isian form komentar 3. Mengklik tombol kirim	2. Menampilkan form komentar 4. Menyimpan Komentar
Kondisi Akhir	Aplikasi Menampilkan Komentar

Tabel 3. 12 Skenario *Use Case* Login Member (*backend)*

Identifikasi	
Nomor	UC-08
Nama	Login Member (*backend)*
Tujuan	Memastikan pengguna memiliki hak akses ke dalam sistem
Deskripsi	Aplikasi akan memeriksa apakah pengguna memiliki hak akses terhadap aplikasi dengan memasukan user name dan password
Aktor	Member
Skenario Utama	
Kondisi awal	Menampilkan halaman utama
Aksi Aktor	Reaksi Sistem
1. Member memasukan username dan password pada form login yang tersedia	2. menampilkan form login 4. Aplikasi akan memverifikasi apakah data yang dimasukan telah sesuai 5. Jika verifikasi berhasil maka aplikasi akan menampilkan halaman utama member 6. Jika verifikasi gagal maka aplikasi akan menampilkan pesan kesalahan login
Kondisi Akhir	Menampilkan Halaman Utama Member

Tabel 3. 13 Skenario *Use Case* Melihat Profile

Identifikasi	
Nomor	UC-09
Nama	Melihat Profile
Tujuan	Melihat profile member
Deskripsi	Fungsionalitas untuk melihat profile member
Aktor	Member
Skenario Utama	
Kondisi awal	Aplikasi menampilkan halaman utama member
Aksi Aktor	Reaksi Sistem
1 Memilih menu profile	2. Menampilkan profile member
Kondisi Akhir	Menampilkan profile member

Tabel 3. 14 Skenario *Use Case* Ubah Profile

Identifikasi	
Nomor	UC-10
Nama	Ubah Profile
Tujuan	Ubah profile member
Deskripsi	Fungsionalitas untuk mengubah profile member
Aktor	Member
Skenario Utama	
Kondisi awal	Aplikasi menampilkan profile member
Aksi Aktor	Reaksi Sistem
1 Memilih ubah profile	2. Menampilkan profile lama member
3. Mengisi data profile baru dan klik simpan	4. Menyimpan dan menampilkan data profile baru
Kondisi Akhir	Menampilkan profile baru member

Tabel 3. 15 Skenario *Use Case* Kustomisasi

Identifikasi	
Nomor	UC-11
Nama	Kustomisasi
Tujuan	Mengkustomisasi data cabor yang dipilih
Deskripsi	Untuk mengkustomisasi data cabor apa saja yang dipilih
Aktor	Member
Skenario Utama	
Kondisi awal	Aplikasi menampilkan halaman utama member
Aksi Aktor	Reaksi Sistem
1. Memilih menu kustomisasi data	2. Menampilkan form kustomisasi
3. Mengisi data kustomisasi	4. Menyimpan data kustomisasi
Kondisi Akhir	Data Kostumisasi tersimpan

Tabel 3. 16 Skenario *Use Case* Ubah Login

Identifikasi	
Nomor	UC-12
Nama	Ubah login
Tujuan	Mengubah data login member
Deskripsi	Fungsionalitas untuk mengubah data login member
Aktor	Member
Skenario Utama	
Kondisi awal	Aplikasi menampilkan halaman utama member
Aksi Aktor	Reaksi Sistem
1. Memilih menu Data Login 3. Mengisi data login baru	2. Menampilkan form data login 4. Menyimpan data login baru
Kondisi Akhir	Data Login tersimpan

Tabel 3. 17 Skenario Use Case Login Admin

Identifikasi	
Nomor	UC-13
Nama	Login Admin
Tujuan	Masuk kedalam sistem
Deskripsi	Admin memasukan data untuk login
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman login admin
Aksi Aktor	Reaksi Sistem
1 Mengisi username dan password pada halaman web login admin	2. Mengautentikasi username dan password yang diinputkan 3. Jika cocok menampilkan halaman web menu utama admin. 4. Jika tidak akan menampilkan pesan kesalahan login
Kondisi Akhir	Halaman Utama Admin ditampilkan

Tabel 3. 18 Skenario *Use Case* Pengolahan Data Induk Organisasi

Identifikasi	
Nomor	UC-14
Nama	Pengolahan Data Induk Organisasi
Tujuan	Melihat fungsionalitas data induk organisasi
Deskripsi	Untuk melihat data induk organisasi
Aktor	Admin

Skenario Utama		
Kondisi awal	Sistem menampilkan halaman utama admin	
Aksi Aktor		Reaksi Sistem
1. Memilih pengolahan data induk		2. Menampilkan halaman data induk
Kondisi Akhir	Menampilkan data induk organisasi	

Tabel 3. 19 Skenario *Use Case* Pengolahan Data Cabor

Identifikasi	
Nomor	UC-15
Nama	Pengolahan Data Cabor
Tujuan	Melihat fungsionalitas data cabor
Deskripsi	Untuk melihat data cabor
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman utama admin
Aksi Aktor	Reaksi Sistem
1. Memilih pengolahan data cabor	2. Menampilkan halaman data cabor
Kondisi Akhir	Menampilkan data cabor

Tabel 3. 20 Skenario *Use Case* Pengolahan Pengurus Cabor

Identifikasi	
Nomor	UC-16
Nama	Pengolahan Pengurus Cabor
Tujuan	Melihat fungsionalitas pengurus cabor
Deskripsi	Untuk melihat data pengurus cabor
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman utama admin
Aksi Aktor	Reaksi Sistem
1. Memilih pengolahan data pengurus cabor	2. Menampilkan halaman data pengurus cabor
Kondisi Akhir	Menampilkan data pengurus cabor

Tabel 3. 21 Skenario *Use Case* Pengolahan Data Klub

Identifikasi	
Nomor	UC-17
Nama	Pengolahan Data Klub
Tujuan	Melihat fungsionalitas data klub
Deskripsi	Untuk melihat data klub
Aktor	Admin

Skenario Utama	
Kondisi awal	Sistem menampilkan halaman utama admin
Aksi Aktor	Reaksi Sistem
1. Memilih pengolahan data klub	2. Menampilkan halaman data klub
Kondisi Akhir	Menampilkan data klub olahraga

Tabel 3. 22 Skenario *Use Case* Pengolahan Pengurus Klub

Identifikasi	
Nomor	UC-18
Nama	Pengolahan Pengurus Klub
Tujuan	Melihat fungsionalitas pengurus klub
Deskripsi	Untuk melihat data pengurus klub
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman utama admin
Aksi Aktor	Reaksi Sistem
1. Memilih pengolahan data pengurus klub	2. Menampilkan halaman data pengurus klub
Kondisi Akhir	Menampilkan data pengurus klub olahraga

Tabel 3. 23 Skenario *Use Case* Pengolahan Data Prestasi Klub

Identifikasi	
Nomor	UC-19
Nama	Pengolahan Data Prestasi Klub
Tujuan	Melihat fungsionalitas dataprestasi klub
Deskripsi	Untuk melihat dataprestasi klub
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman utama admin
Aksi Aktor	Reaksi Sistem
1. Memilih pengolahan data prestasi klub	2. Menampilkan halaman data prestasi klub
Kondisi Akhir	Menampilkan data prestasi klub olahraga

Tabel 3. 24 Skenario *Use Case* Pengolahan Data Kejuaraan

Identifikasi	
Nomor	UC-20
Nama	Pengolahan Data Kejuaraan
Tujuan	Melihat fungsionalitas data kejuaraan
Deskripsi	Untuk melihat data kejuaraan
Aktor	Admin

Skenario Utama		
Kondisi awal	Sistem menampilkan halaman menu utama	
Aksi Aktor		Reaksi Sistem
1. Memilih pengolahan data kejuaraan		2. Menampilkan halaman data kejuaraan
Kondisi Akhir	Menampilkan data kejuaraan olahraga	

Tabel 3. 25 Skenario *Use Case* Pengolahan Data Member

Identifikasi	
Nomor	UC-21
Nama	Pengolahan Data Member
Tujuan	Melihat fungsionalitas data member
Deskripsi	Untuk melihat data member
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman menu utama
Aksi Aktor	Reaksi Sistem
1. Memilih pengolahan data member	2. Menampilkan halaman data member
Kondisi Akhir	Menampilkan data member

Tabel 3. 26 Skenario *Use Case* Tambah Induk Organisasi

Identifikasi	
Nomor	UC-22
Nama	Tambah Induk Organisasi
Tujuan	Menambah data induk organisasi
Deskripsi	Untuk menambah data induk organisasi
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan induk organisasi
Aksi Aktor	Reaksi Sistem
1. Memilih link *Tambah* 3. Mengisi field-field pada form pengisian tambah induk organisasi	2. Menampilkan form pengisian tambah induk organisasi 4. Melakukan proses tambah data pada basis data 5. menyimpan hasil tambah data
Kondisi Akhir	Data induk organisasi bertambah sesuai yang diinginkan

Tabel 3. 27 Skenario *Use Case* Ubah Induk Organisasi

Identifikasi	
Nomor	UC-23
Nama	Ubah Induk Organisasi
Tujuan	Mengubah data induk organisasi
Deskripsi	Untuk Mengubah data induk organisasi
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan induk organisasi
Aksi Aktor	Reaksi Sistem
1. Memilih link *Edit* pada pilihan menu 3. Mengisi field-field pada form edit induk organisasi	2. Menampilkan halaman form edit induk organisasi 4. Melakukan proses ubah data pada basis data 5. Menyimpan hasil perubahan data
Kondisi Akhir	Perubahan data sesuai yang diinginkan

Tabel 3. 28 Skenario *Use Case* Hapus Induk Organisasi

Identifikasi	
Nomor	UC-24
Nama	Hapus Induk Organisasi
Tujuan	Menghapus data induk organisasi
Deskripsi	Untuk Menghapus data induk organisasi
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan induk organisasi
Aksi Aktor	Reaksi Sistem
1. Memilih link *Delete* pada pilihan menu 3. Menjawab konfirmasi dengan menekan tombol “OK”	2. Menampilkan pesan konfirmasi hapus induk organisasi 4. Menghapus induk organisasi 5. Menyimpan hasil perubahan data
Kondisi Akhir	Data induk organisasi yang dipilih telah terhapus

Tabel 3. 29 Skenario *Use Case* Tambah Cabor

Identifikasi	
Nomor	UC-25
Nama	Tambah Cabor
Tujuan	Menambah data cabor
Deskripsi	Untuk menambah data cabor
Aktor	Admin

Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan cabor
Aksi Aktor	Reaksi Sistem
1. Memilih link *Tambah* pada pilihan menu. 3. Mengisi field-field pada form pengisian tambah cabor	2. Menampilkan form pengisian tambah cabor. 4. Melakukan proses tambah data pada basis data 5. menyimpan hasil tambah data
Kondisi Akhir	Data Cabor bertambah sesuai yang diinginkan

Tabel 3. 30 Skenario *Use Case* Ubah Cabor

Identifikasi	
Nomor	UC-26
Nama	Ubah Cabor
Tujuan	Mengubah data cabor
Deskripsi	Untuk Mengubah data cabor
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan cabor
Aksi Aktor	Reaksi Sistem
1. Memilih link *Edit* pada pilihan menu 3. Mengisi field-field pada form edit cabor	2. Menampilkan halaman form edit cabor 4. Melakukan proses ubah data pada basis data 5. Menyimpan hasil perubahan data

Tabel 3. 31 Skenario *Use Case* Hapus Cabor

Identifikasi	
Nomor	UC-27
Nama	Hapus Cabor
Tujuan	Menghapus data cabor
Deskripsi	Untuk Menghapus data cabor
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan cabor
Aksi Aktor	Reaksi Sistem
1. Memilih link *Delete* pada pilihan menu 3. Menjawab konfirmasi dengan menekan tombol “OK”	2. Menampilkan pesan konfirmasi hapus cabor 4. Menghapus cabor 5. Menyimpan hasil perubahan data
Kondisi Akhir	Data cabor yang dipilih telah berhasil dihapus

Tabel 3. 32 Skenario *Use Case* Tambah Pengurus Cabor

Identifikasi	
Nomor	UC-28
Nama	Tambah Pengurus Cabor
Tujuan	Menambah data pengurus cabor
Deskripsi	Untuk menambah data pengurus cabor
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan pengurus cabor
Aksi Aktor	Reaksi Sistem
1. Memilih link *Tambah* pada pilihan menu. 3. Mengisi field-field pada form pengisian tambah pengurus cabor	2. Menampilkan form pengisian tambah pengurus cabor 4. Melakukan proses tambah data pada basis data 5. menyimpan hasil tambah data
Kondisi Akhir	Data pengurus cabor bertambah sesuai yang diinginkan

Tabel 3. 33 Skenario *Use Case* Ubah Pengurus Cabor

Identifikasi	
Nomor	UC-29
Nama	Ubah Pengurus Cabor
Tujuan	Mengubah data pengurus cabor
Deskripsi	Untuk Mengubah data pengurus cabor
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan pengurus cabor
Aksi Aktor	Reaksi Sistem
1. Memilih link *Edit* pada pilihan menu 3. Mengisi field-field pada form edit pengurus cabor	2. Menampilkan halaman form edit pengurus cabor 4. Melakukan proses ubah data pada basis data 5. Menyimpan hasil perubahan data
Kondisi Akhir	Perubahan data sesuai yang diinginkan

Tabel 3. 34 Skenario *Use Case* Hapus Pengurus Cabor

Identifikasi	
Nomor	UC-30
Nama	Hapus Pengurus Cabor
Tujuan	Menghapus data pengurus cabor
Deskripsi	Untuk Menghapus data pengurus cabor
Aktor	Admin

Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan pemgurus cabor
Aksi Aktor	Reaksi Sistem
1. Memilih link *Delete* pada pilihan menu 3. Menjawab konfirmasi dengan menekan tombol “OK”	2. Menampilkan pesan konfirmasi hapus pengurus cabor 4. Menghapus pengurus cabor 5. Menyimpan hasil perubahan data
Kondisi Akhir	Data pengurus cabor yang dipilih telah berhasil dihapus

Tabel 3. 35 Skenario *Use Case* Tambah Klub

Identifikasi	
Nomor	UC-31
Nama	Tambah Klub
Tujuan	Menambah data klub
Deskripsi	Untuk menambah data klub
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan klub
Aksi Aktor	Reaksi Sistem
1. Memilih link *Tambah* pada pilihan menu. 3. Mengisi field-field pada form pengisian tambah klub	2. Menampilkan form pengisian tambah klub 4. Melakukan proses tambah data pada basis data 5. menyimpan hasil tambah data
Kondisi Akhir	Data Klub bertambah sesuai yang diinginkan

Tabel 3. 36 Skenario *Use Case* Ubah Klub

Identifikasi	
Nomor	UC-32
Nama	Ubah Klub
Tujuan	Mengubah data klub
Deskripsi	Untuk Mengubah data klub
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan klub
Aksi Aktor	Reaksi Sistem
1. Memilih link *Edit* pada pilihan menu 3. Mengisi field-field pada form edit klub	2. Menampilkan halaman form edit klub 4. Melakukan proses ubah data pada basis data 5. Menyimpan hasil perubahan data
Kondisi Akhir	Perubahan data sesuai yang diinginkan

Tabel 3. 37 Skenario *Use Case* Hapus Klub

Identifikasi	
Nomor	UC-33
Nama	Hapus Klub
Tujuan	Menghapus data klub
Deskripsi	Untuk Menghapus data klub
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan klub
Aksi Aktor	Reaksi Sistem
1. Memilih link *Delete* pada pilihan menu 3. Menjawab konfirmasi dengan menekan tombol “OK”	2. Menampilkan pesan konfirmasi hapus klub 4. Menghapus klub 5. Menyimpan hasil perubahan data
Kondisi Akhir	Data klub yang dipilih telah berhasil dihapus

Tabel 3. 38 Skenario *Use Case* Tambah Pengurus Klub

Identifikasi	
Nomor	UC-34
Nama	Tambah Pengurus Klub
Tujuan	Menambah data pengurus klub
Deskripsi	Untuk menambah data pengurus klub
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan pengurus klub
Aksi Aktor	Reaksi Sistem
1. Memilih link *Tambah* pada pilihan menu. 3. Mengisi field-field pada form pengisian tambah pengurus klub	2. Menampilkan form pengisian tambah pengurus klub 4. Melakukan proses tambah data pada basis data 5. menyimpan hasil tambah data
Kondisi Akhir	Data pengurus klub bertambah sesuai yang diinginkan

Tabel 3. 39 Skenario *Use Case* Ubah Pengurus Klub

Identifikasi	
Nomor	UC-35
Nama	Ubah Pengurus Klub
Tujuan	Mengubah data pengurus klub
Deskripsi	Untuk Mengubah data pengurus klub
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan pengurus klub

Aksi Aktor	Reaksi Sistem
1. Memilih link *Edit* pada pilihan menu 3. Mengisi field-field pada form edit pengurus klub	2. Menampilkan halaman form edit pengurus klub 4. Melakukan proses ubah data pada basis data 5. Menyimpan hasil perubahan data
Kondisi Akhir	Perubahan data sesuai yang diinginkan

Tabel 3. 40 Skenario *Use Case* Hapus Pengurus Klub

Identifikasi	
Nomor	UC-36
Nama	Hapus Pengurus Klub
Tujuan	Menghapus data pengurus klub
Deskripsi	Untuk Menghapus data pengurus klub
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan pengurus klub
Aksi Aktor	Reaksi Sistem
1. Memilih link *Delete* pada pilihan menu 3. Menjawab konfirmasi dengan menekan tombol “OK”	2. Menampilkan pesan konfirmasi hapus pengurus klub 4. Menghapus pengurus klub 5. Menyimpan hasil perubahan data
Kondisi Akhir	Data pengurus klub yang dipilih telah berhasil dihapus

Tabel 3. 41 Skenario *Use Case* Tambah Prestasi Klub

Identifikasi	
Nomor	UC-37
Nama	Tambah Prestasi Klub
Tujuan	Menambah data prestasi klub
Deskripsi	Untuk menambah data prestasi klub
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan prestasi klub
Aksi Aktor	Reaksi Sistem
1. Memilih link *Tambah* pada pilihan menu. 3. Mengisi field-field pada form pengisian tambah prestasi klub	2. Menampilkan form pengisian tambah prestasi klub 4. Melakukan proses tambah data pada basis data 5. menyimpan hasil tambah data
Kondisi Akhir	Data Prestasi Klub bertambah sesuai yang diinginkan

Tabel 3. 42 Skenario *Use Case* Ubah Prestasi Klub

Identifikasi	
Nomor	UC-38
Nama	Ubah Prestasi Klub
Tujuan	Mengubah data prestasi klub
Deskripsi	Untuk Mengubah data prestasi klub
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan prestasi klub
Aksi Aktor	Reaksi Sistem
1. Memilih link *Edit* pada pilihan menu 3. Mengisi field-field pada form edit prestasi klub	2. Menampilkan halaman form edit prestasi klub 4. Melakukan proses ubah data pada basis data 5. Menyimpan hasil perubahan data
Kondisi Akhir	Perubahan data sesuai yang diinginkan

Tabel 3. 43 Skenario *Use Case* Hapus Prestasi Klub

Identifikasi	
Nomor	UC-39
Nama	Hapus Prestasi Klub
Tujuan	Menghapus data prestasi klub
Deskripsi	Untuk Menghapus data prestasi klub
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan prestasi klub
Aksi Aktor	Reaksi Sistem
1. Memilih link *Delete* pada pilihan menu 3. Menjawab konfirmasi dengan menekan tombol “OK”	2. Menampilkan pesan konfirmasi hapus prestasi klub 4. Menghapus prestasi klub 5. Menyimpan hasil perubahan data
Kondisi Akhir	Data prestasi klub yang dipilih telah berhasil dihapus

Tabel 3. 44 Skenario *Use Case* Tambah Kejuaraan

Identifikasi	
Nomor	UC-40
Nama	Tambah Kejuaraan
Tujuan	Menambah data kejuaraan
Deskripsi	Untuk menambah data kejuaraan
Aktor	Admin

Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan kejuaraan
Aksi Aktor	Reaksi Sistem
1. Memilih link *Tambah* pada pilihan menu. 3. Mengisi field-field pada form pengisian tambah kejuaraan	2. Menampilkan form pengisian tambah kejuaraan 4. Melakukan proses tambah data pada basis data 5. menyimpan hasil tambah data
Kondisi Akhir	Data Kejuaraan bertambah sesuai yang diinginkan

Tabel 3. 45 Skenario *Use Case* Ubah Kejuaraan

Identifikasi	
Nomor	UC-41
Nama	Ubah Kejuaraan
Tujuan	Mengubah data kejuaraan
Deskripsi	Untuk Mengubah data kejuaraan
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan kejuaraan
Aksi Aktor	Reaksi Sistem
1. Memilih link *Edit* pada pilihan menu 3. Mengisi field-field pada form edit kejuaraan	2. Menampilkan halaman form edit kejuaraan 4. Melakukan proses ubah data pada basis data 5. Menyimpan hasil perubahan data
Kondisi Akhir	Perubahan data sesuai yang diinginkan

Tabel 3. 46 Skenario *Use Case* Hapus Kejuaraan

Identifikasi	
Nomor	UC-42
Nama	Hapus Kejuaraan
Tujuan	Menghapus data kejuaraan
Deskripsi	Untuk Menghapus data kejuaraan
Aktor	Admin
Skenario Utama	
Kondisi awal	Sistem menampilkan halaman pengolahan kejuaraan
Aksi Aktor	Reaksi Sistem
1. Memilih link *Delete* pada pilihan menu 3. Menjawab konfirmasi dengan menekan tombol “OK”	2. Menampilkan pesan konfirmasi hapus kejuaraan 4. Menghapus kejuaraan 5. Menyimpan hasil perubahan data

Tabel 3. 47 Skenario *Use Case* Registrasi (*backend)*

Identifikasi		
Nomor	UC-43	
Nama	Registrasi (*backend)*	
Tujuan	Mendaftar untuk menjadi member	
Deskripsi	Pengunjung mengakses web Koni kemudian melakukan registrasi	
Aktor	Pengunjung Biasa	
Skenario Utama		
Kondisi awal	Web menampilkan form registrasi	
Aksi Aktor		Reaksi Sistem
1. Pengunjung memasukkan data yang dibutuhkan		2. Aplikasi akan memverifikasi data 3. Aplikasi menyimpan data yang dimasukkan
Kondisi Akhir	Data member telah tersimpan	

Tabel 3. 48 Skenario *Use Case* Melihat Info Cabor (*backend)*

Identifikasi		
Nomor	UC-44	
Nama	Melihat Info Cabor (*backend)*	
Tujuan	Untuk melihat data cabang olahraga	
Deskripsi	Untuk melihat daftar cabor pada aplikasi *backend* (web)	
Aktor	Pengunjung Biasa	
Skenario Utama		
Kondisi awal	-	
Aksi Aktor		Reaksi Sistem
1. Memilih menu info cabor		2. Menampilkan halaman data cabor
Kondisi Akhir	Menampilkan daftar data cabor	

Tabel 3. 49 Skenario *Use Case* Melihat Info Klub (*backend)*

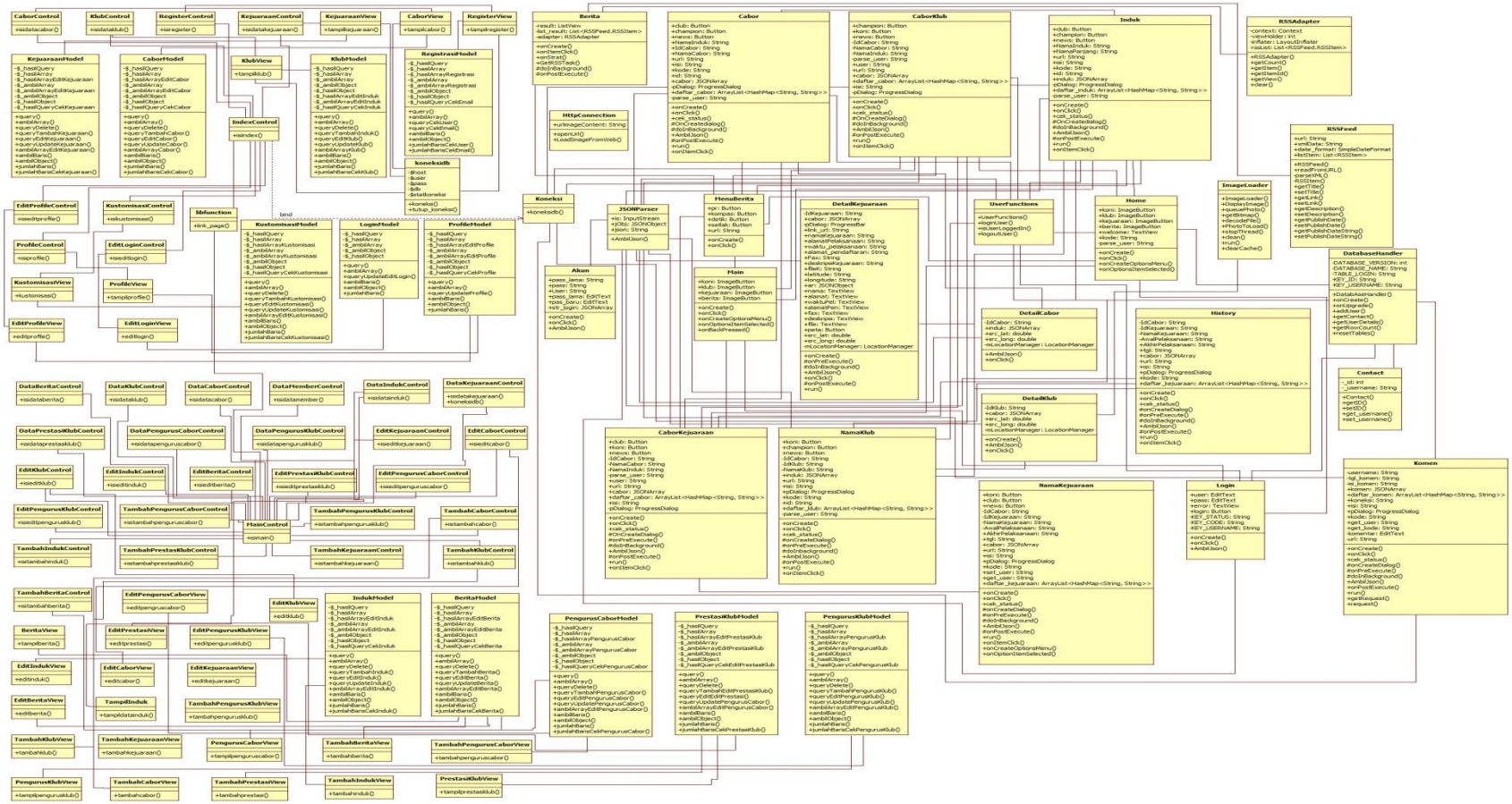
Identifikasi		
Nomor	UC-45	
Nama	Melihat Info Klub (*backend)*	
Tujuan	Untuk melihat data klub olahraga	
Deskripsi	Untuk melihat daftar klub olahraga pada aplikasi *backend* (web)	
Aktor	Pengunjung Biasa	
Skenario Utama		
Kondisi awal	-	
Aksi Aktor		Reaksi Sistem
1. Memilih menu info klub		2. Menampilkan halaman data klub
Kondisi Akhir	Menampilkan daftar data klub	

Tabel 3. 50 Skenario *Use Case* Melihat Info Kejuaraan (*backend)*

Identifikasi		
Nomor	UC-46	
Nama	Melihat Info Kejuaraan (*backend)*	
Tujuan	Untuk melihat data kejuaraan olahraga	
Deskripsi	Untuk melihat daftar kejuaraan olahraga pada aplikasi *backend* (web)	
Aktor	Pengunjung Biasa	
Skenario Utama		
Kondisi awal	-	
Aksi Aktor		Reaksi Sistem
1. Memilih menu info kejuaraan		2. Menampilkan halaman data kejuaraan
Kondisi Akhir	Menampilkan daftar data kejuaraan	

b. Class Diagram

Class Diagram adalah sebuah spesifikasi dari fungsionalitas yang menghasilkan objek dan merupakan inti dari pengembangan aplikasi ini. Diagram ini juga merupakan gambaran keadaan atribut atau properti dari sistem yang melakukan manipulasi fungsi atau metode. Berikut ini gambaran *class* diagram dari sistem Aplikasi Olahraga.



Gambar 3. 4 *Class* Diagram

Tabel 3. 51 Deskripsi *Class Diagram*

Class	Jenis Kelas	Deskripsi
Main	Control	*Class* utama yang berisikan method-method yang berhubungan dengan *class* lain dan berfungsi sebagai *controller*
Induk	Control	*Class* yang berisikan method-method untuk menampilkan list induk organisasi
NamaKlub	Control	*Class* yang berisikan method-method untuk menampilkan list nama-nama klub
DetailKlub	Control	*Class* yang berisikan method-method untuk menampilkan rincian data klub
Cabor	Control	*Class* yang berisikan method-method untuk menampilkan list cabor
NamaKejuaraan	Control	*Class* yang berisikan method-method untuk menampilkan list nama-nama kejuaraan
DetailKejuaraan	Control	*Class* yang berisikan method-method untuk menampilkan rincian data kejuaraan
MenuBerita	Control	*Class* yang berisikan method-method untuk menampilkan pilihan menu-menu berita
Berita	Control	*Class* yang berisikan method-method untuk menampilkan list berita
History	Control	*Class* yang berisikan method-method untuk menampilkan list cabor berdasarkan history
loginMember	Control	*Class* yang berisikan method-method untuk login member
Home	Control	*Class* yang berisikan method-method yang berhubungan dengan *class* lain
Komen	Control	*Class* yang berisikan method-method untuk membuat komentar terhadap suatu kejuaraan

JSONParser	Control	*Class* berisikan method-method untuk memparsing data berformat JSON
ImageLoader	Control	*Class* yang berisikan method-method untuk mengambil dan menampilkan *Image*
HttpConnection	Control	*Class* yang berisikan method-method yang menangani koneksi ke *server*
RSSAdapter	Control	*Class* yang berisikan method-method untuk menentukan item RSS apa saja yang akan diambil
RSSFeed	Control	*Class* yang berfungsi untuk membuat list item
main	Control	*Class* yang berisikan tampilan pilihan menu admin. web
CaborControl	Control	*Class* untuk mengontrol cabor pada halaman depan web
KejuaraanControl	Control	*Class* untuk mengontrol kejuaraan pada halaman depan web
KlubControl	Control	*Class* untuk mengontrol klub pada halaman depan web
RegistrasiControl	Control	*Class* untuk mengontrol registrasi member
CaborModel	Model	*Class* yang berisi model cabor
KejuaraanModel	Model	*Class* yang berisi model kejuaraan
KlubModel	Model	*Class* yang berisi model klub
RegistrasiModel	Model	*Class* yang berisi model registrasi
CaborView	View	*Class* untuk menampilkan data cabor pada halaman depan web
KejuaraanView	View	*Class* untuk menampilkan data kejuaraan pada halaman depan
KlubView	View	*Class* untuk menampilkan data klub pada halaman depan web
IndexControl	View	*Class* berisikan template halaman depan web

RegisterView	View	*Class* untuk menampilkan form registrasi
EditLoginControl	Control	*Class* untuk mengontrol edit login member
EditProfileControl	Control	*Class* untuk mengontrol edit profile member
KustomisasiControl	Control	*Class* untuk mengontrol kustomisasi
ProfileControl	Control	*Class* untuk mengontrol profile member
KustomisasiModel	Model	*Class* yang berisi model kustomisasi
loginModel	Model	*Class* yang berisi model login
ProfileModel	Model	*Class* yang berisi model profile
EditLoginView	View	*Class* untuk menampilkan form edit login
EditProfileView	View	*Class* yang menampilkan form edit profile
KustomisasiView	View	*Class* yang menampilkan kustomisasi data
ProfileView	View	*Class* yang menampilkan profile member
DataIndukControl	Control	*Class* untuk mengontrol data induk
DataKlubControl	Control	*Class* untuk mengontrol data klub
DataCaborControl	Control	*Class* untuk mengontrol data cabor
DataKejuaraan Control	Control	*Class* untuk mengontrol data kejuaraan
DataPengurus CaborControl	Control	*Class* untuk mengontrol data pengurus cabor
DataPengurusKlub Control	Control	*Class* untuk mengontrol data pengurus klub
DataPrestasiKlub Control	Control	*Class* untuk mengontrol data prestasi klub
DataBeritaControl	Control	*Class* untuk mengontrol data berita
DataMemberContro l	Control	*Class* untuk mengontrol data member
TambahaIndukCont rol	Control	*Class* untuk mengontrol tambah data induk
TambahCaborContr ol	Control	*Class* untuk mengontrol tambah data cabor

TambahKlubControl	Control	*Class* untuk mengontrol tambah data klub
TambahKejuaraanControl	Control	*Class* untuk mengontrol tambah data kejuaraan
TambahPengursCaborControl	Control	*Class* untuk mengontrol tambah data pengurus cabor
TambahPengurusKlubControl	Control	*Class* untuk mengontrol tambah data pengurus klub
TambahPrestasiKlubControl	Control	*Class* untuk mengontrol tambah data prestasi klub
TambahBeritaControl	Control	*Class* untuk mengontrol tambah data berita
EditIndukControl	Control	*Class* untuk mengontrol edit data induk
EditCaborControl	Control	*Class* untuk mengontrol edit data cabor
EditKlubControl	Control	*Class* untuk mengontrol edit data klub
EditKejuaraanControl	Control	*Class* untuk mengontrol edit data kejuaraan
EditPengursCaborControl	Control	*Class* untuk mengontrol edit data pengurus cabor
EditPengurusKlubControl	Control	*Class* untuk mengontrol edit data pengurus klub
EditPrestasiKlubControl	Control	*Class* untuk mengontrol edit data prestasi klub
BeritaModel	Model	*Class* yang berisikan model berita
CaborModel	Model	*Class* yang berisikan model cabor
IndukModel	Model	*Class* yang berisikan model induk
KejuaraanModel	Model	*Class* yang berisikan model kejuaraan
KlubModel	Model	*Class* yang berisikan model klub
PengurusCaborModel	Model	*Class* yang berisikan model pengurus cabor

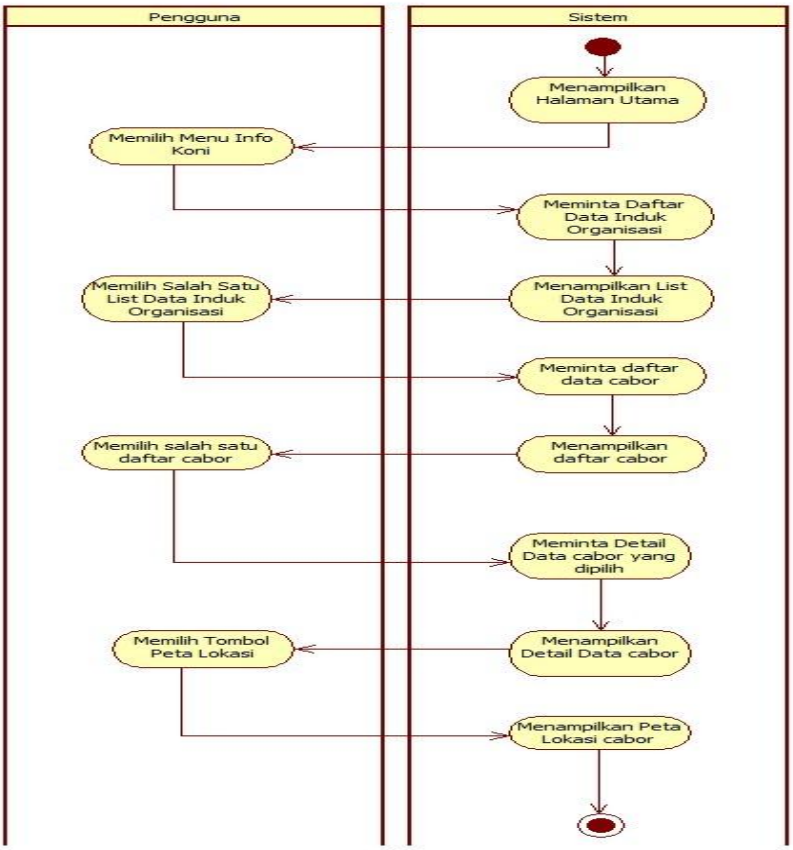
PengurusKlubModel	Model	*Class* yang berisikan model pengurus klub
PrestasiKlubModel	Model	*Class* yang berisikan model prestasi klub
BeritaView	View	*Class* untuk menampilkan data berita
CaborView	View	*Class* untuk menampilkan data cabor
EditBeritaView	View	*Class* untuk menampilkan form edit data berita
EditCaborView	View	*Class* untuk menampilkan form edit data cabor
EditIndukView	View	*Class* untuk menampilkan form edit data induk
EditKejuaraanView	View	*Class* untuk menampilkan form edit data kejuaraan
EditKlubView	View	*Class* untuk menampilkan form edit data klub
EditPengurusCaborView	View	*Class* untuk menampilkan form edit data pengurus cabor
EditPengurusKlubView	View	*Class* untuk menampilkan form edit data pengurus klub
EditPrestasiview	View	*Class* untuk menampilkan form edit data prestasi klub
KejuaraanView	View	*Class* untuk menampilkan data kejuaraan
KlubView	View	*Class* untuk menampilkan data klub
PengurusCaborView	View	*Class* untuk menampilkan data pengurus cabor
PengurusKlubView	View	*Class* untuk menampilkan data pengurus klub
PrestasiKlubView	View	*Class* untuk menampilkan data prestasi klub
TambahBeritaView	View	*Class* untuk menampilkan form tambah data berita
TambahCaborView	View	*Class* untuk menampilkan form tambah data cabor
TambahIndukView	View	*Class* untuk menampilkan form tambah data

		induk
TambahKejuaraanview	View	*Class* untuk menampilkan form tambah data kejuaraan
TambahKlubView	View	*Class* untuk menampilkan form tambah data klub
TambahPengurusCaborView	View	*Class* untuk menampilkan form tambah data pengurus cabor
TambahPengurusKlubView	View	*Class* untuk menampilkan form tambah data pengurus klub
TambahPrestasiView	View	*Class* untuk menampilkan form tambah data prestasi klub
TampilInduk	View	*Class* untuk menampilkan data induk
MainControl	View	*Class* berisikan template halaman admin

c. Activity Diagram

Diagram activity merupakan bagian dari penggambaran sistem secara fungsional menjelaskan proses-proses logika atau fungsi yang terimplementasi oleh kode program. *Activity Diagram* memodelkan event-event yang terjadi didalam suatu *Use Case* dan digunakan untuk pemodelan aspek dinamis dari sistem.

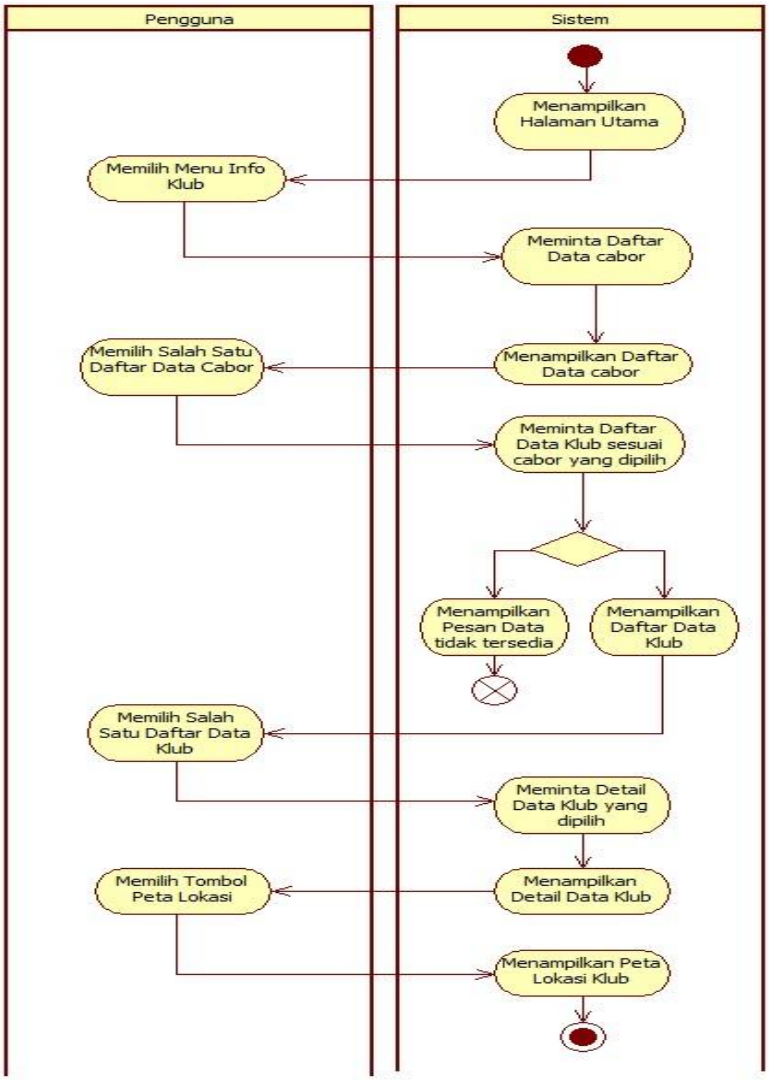
a. *Activity* Diagram Melihat Info Cabor (*frontend)*



Gambar 3. 5 *Activity* Diagram Melihat Info Cabor (*frontend)*

Pada Gambar 3.5 memperlihatkan aktivitas pilih. Pengunjung memilih menu info koni dari halaman utama yang ditampilkan aplikasi. Kemudian aplikasi akan meminta daftar data cabor pada *server*, lalu *server* akan merespon *request* tersebut. Setelah *server* merespon *request* maka aplikasi akan menampilkan daftar data cabor, lalu pengunjung dapat memilih salah satu daftar data cabor yang ditampilkan. Kemudian aplikasi akan meminta detail data induk organisasi yang telah dipilih lalu *server* akan merespon permintaan tersebut dan aplikasi akan menampilkan detail data cabor. Lalu Pengunjung dapat memilih tombol peta lokasi, maka aplikasi akan menampilkan lokasi serta direction cabor yang dipilih.

b. *Activity* Diagram Melihat Info Klub (*frontend)*

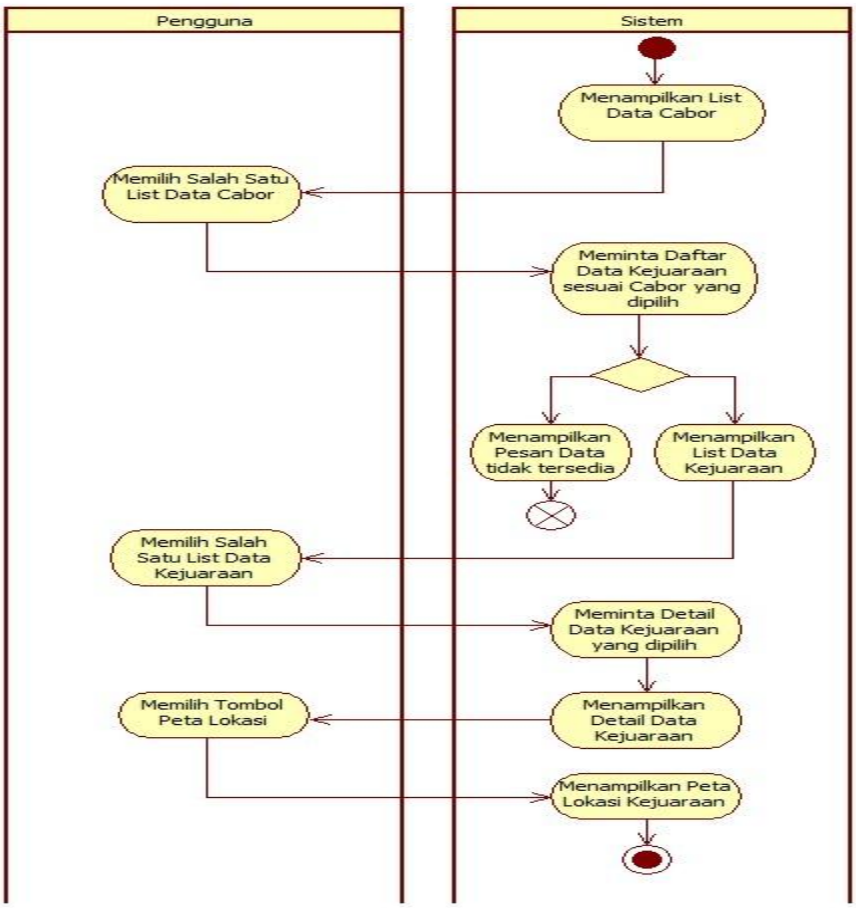


Gambar 3. 6 *Activity* Diagram Melihat Info Klub (*frontend)*

Pada Gambar 3.6 memperlihatkan aktivitas memilih klub olahraga oleh pengunjung biasa. Pengunjung memilih menu info klub dari halaman utama yang ditampilkan aplikasi. Kemudian aplikasi akan meminta daftar data cabor kepada *server,* lalu *server* akan mencari data cabor yang ada. Aplikasi menampilkan list data cabor yang tersedia, lalu pengunjung memilih salah satu daftar data cabor. Setelah itu aplikasi akan meminta daftar klub olahraga sesuai data cabor yang

dipilih kepada *server*, lalu *server* akan merespon request tersebut. Jika data klub olahraga yang dicari tidak ada maka aplikasi akan menampilkan pesan bahwa data klub tidak tersedia, tetapi jika data klub olahraga yang dicari tersedia maka aplikasi akan menampilkan data klub olahraga tersebut. Setelah itu pengunjung dapat memilih salah satu list data klub olahraga untuk dapat melihat detail klub olahraga tersebut. Pengunjung juga dapat memilih peta lokasi untuk melihat lokasi dan direction klub olahraga.

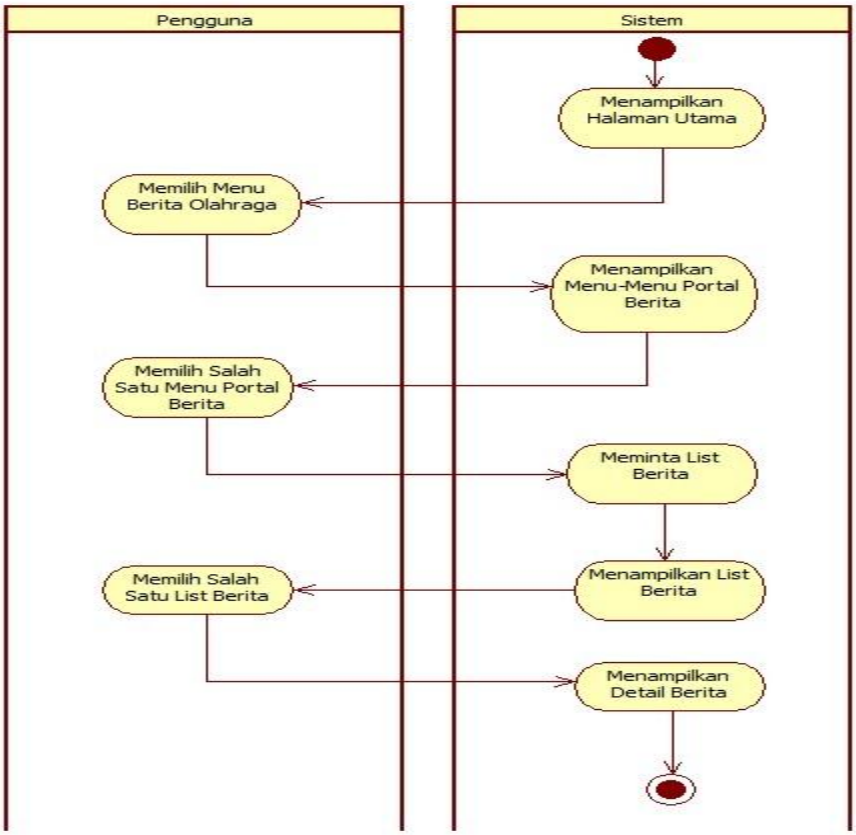
c. *Activity* Diagram Melihat Info Kejuaraan (*frontend)*



Gambar 3. 7 *Activity* Diagram Melihat Info Kejuaraan (*frontend)*

Pada gambar 3.7 memperlihatkan aktivitas memilih Kejuaraan Olahraga oleh pengunjung biasa. Aplikasi akan menampilkan list data cabor lalu pengunjung dapat memilih salah satu list data cabor tersebut maka aplikasi akan meminta daftar data kejuaraan kepada server. Jika data kejuaraan tidak tersedia maka aplikasi akan menampilkan pesan data kejuaraan tidak tersedia tetapi jika data tersedia maka aplikasi akan menampilkan data kejuaraan yang ada. Pengunjung dapat memilih salah satu list data kejuaraan yang ada maka aplikasi akan menampilkan detail dari kejuaraan yang dipilih. Pengunjung juga dapat memilih peta lokasi untuk melihat lokasi serta direction lokasi kejuaraan.

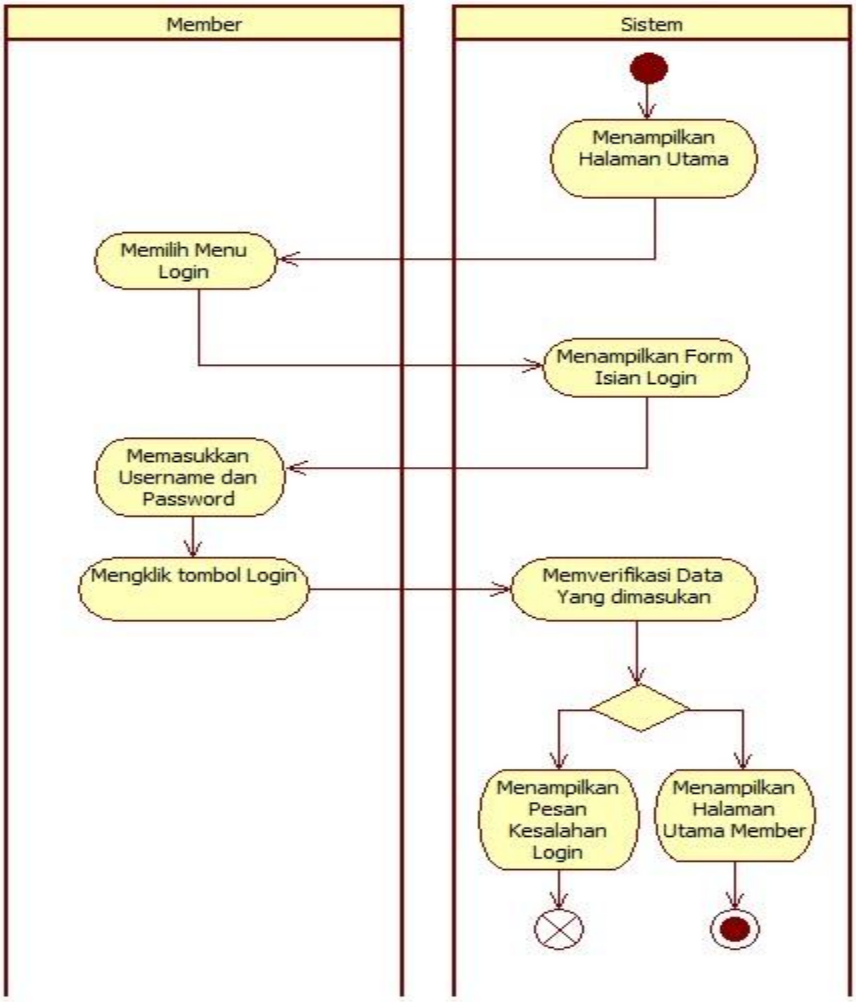
d. *Activity* Diagram Melihat Berita Olahraga



Gambar 3. 8 *Activity* Diagram Melihat Berita Olahraga

Pada gambar 3.8 memperlihatkan aktivitas memilih berita olahraga yang dilakukan oleh pengunjung. Pengunjung memilh menu berita olahraga maka aplikasi akan menampilkan menu portal-portal berita yang tersedia, lalu pengunjung dapat memilih salah satu dari portal berita yang tersedia maka aplikasi akan menampilkan list berita olahraga dari portal berita yang dipilih. Pengunjung dapat memilih salah satu dari list berita yang tersedia lalu aplikasi akan menampilkan detail dari berita yang dipilih.

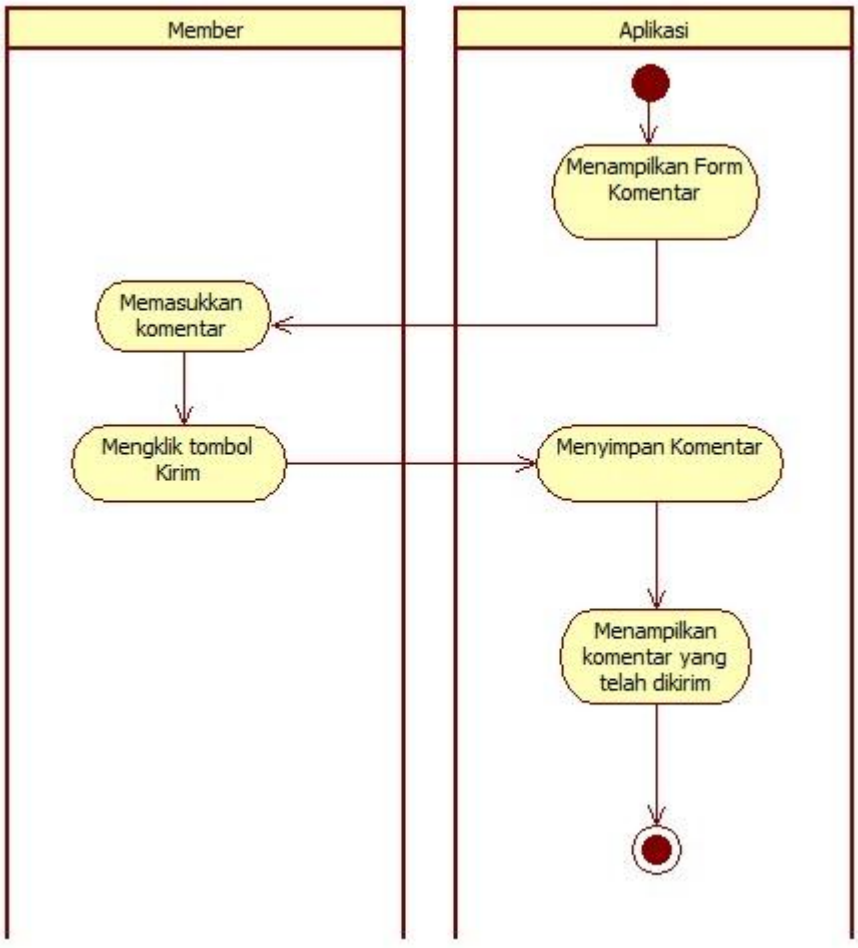
e. *Activity* Diagram Login Member (*frontend)*



Gambar 3. 9 *Activity* Diagram Login Member (*frontend)*

Pada gambar 3.9 memperlihatkan aktivitas Login member. Saat aplikasi menampilkan halaman utama, kemudian member memilih tombol *login* kemudian aplikasi menampilkan *form* isian *login*. Kemudian member memasukan *username* dan *password* . kemudian sistem akan mengautentifikasi data yang dimasukan apakah sudah sesuai atau tidak dengan data yang ada di *database*, jika sesuai maka akan menampilkan halaman utama member, jika tidak maka akan menampilkan pesan kesalahan login.

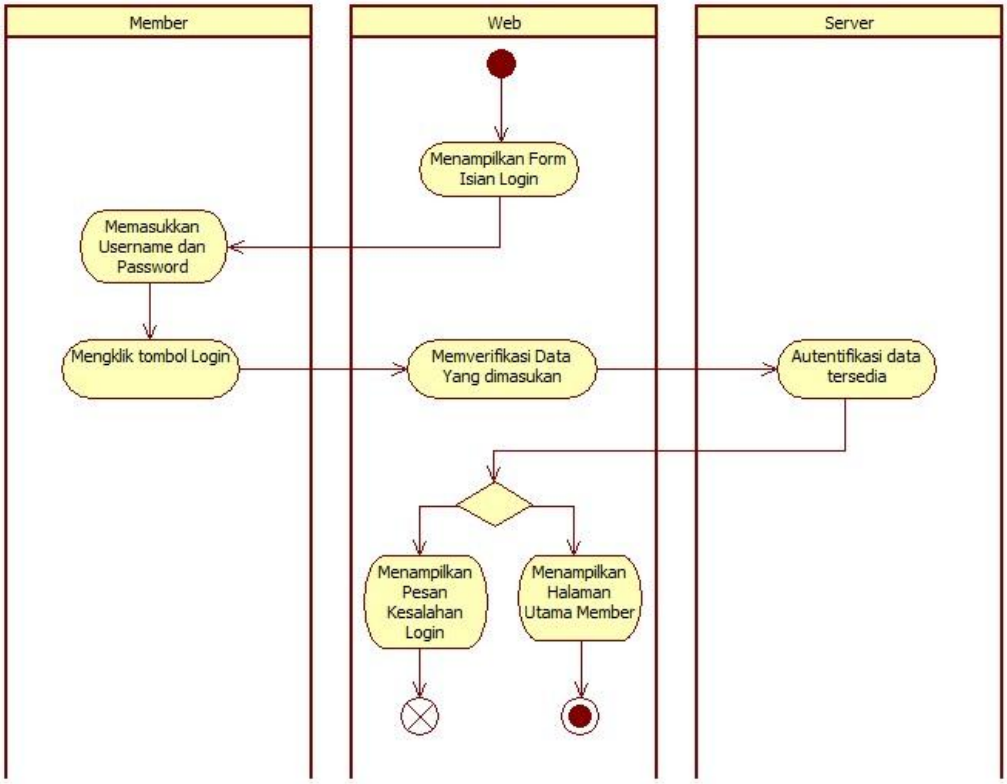
f. *Activity* Diagram Kirim komentar



Gambar 3. 10 *Activity* Diagram Kirim Komentar

Pada Gambar 3.10 memperlihatkan aktivitas kirim komentar pada sebuah kejuaraan olahraga yang dilakukan oleh member. Setelah aplikasi menampilkan detail kejuaraan maka member dapat memilih menu komentar lalu aplikasi akan menampilkan form komentar. Kemudian member dapat memasukkan komentar lalu klik tombol kirim. Maka aplikasi akan menyimpan data komentar ke database server lalu aplikasi akan menampilkan komentar yang telah kirim dan tersimpan

g. Activity Diagram Login Member (*backend)*

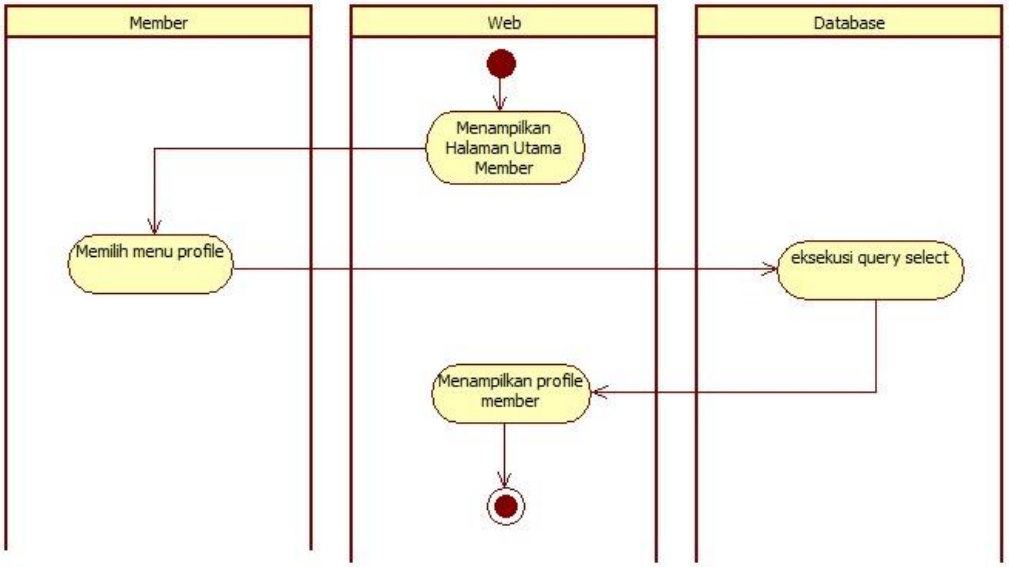


Gambar 3. 11 *Activity* Diagram Login Member (*backend)*

Pada gambar 3.11 memperlihatkan aktivitas Login member pada aplikasi web. Saat aplikasi menampilkan halaman utama, kemudian member memasukan *username* dan *password* . kemudian sistem akan mengautentifikasi data yang dimasukan apakah sudah sesuai atau tidak dengan data yang ada di *database*, jika

sesuai maka akan menampilkan halaman utama member, jika tidak maka akan menampilkan pesan kesalahan login.

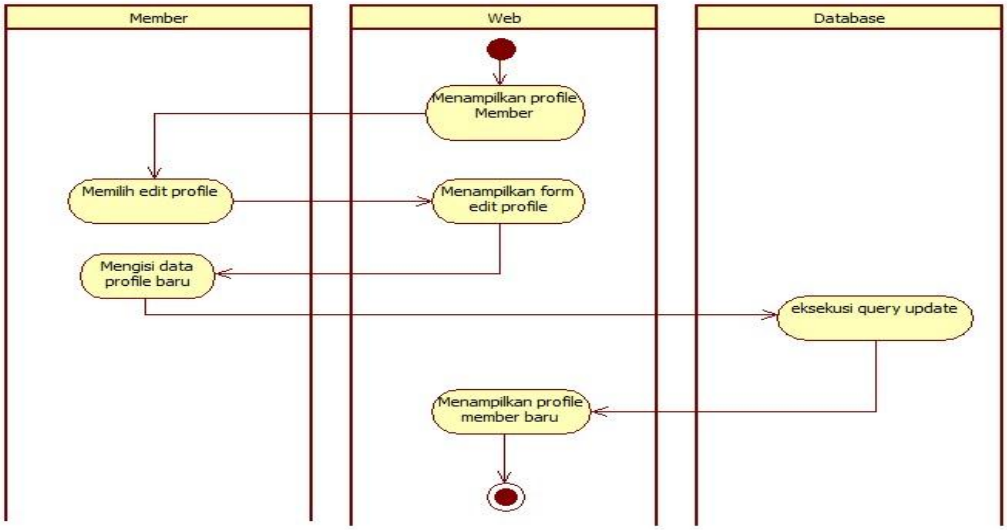
h. *Activity* Diagram Melihat Profile



Gambar 3. 12 *Activity* Diagram Melihat Profile

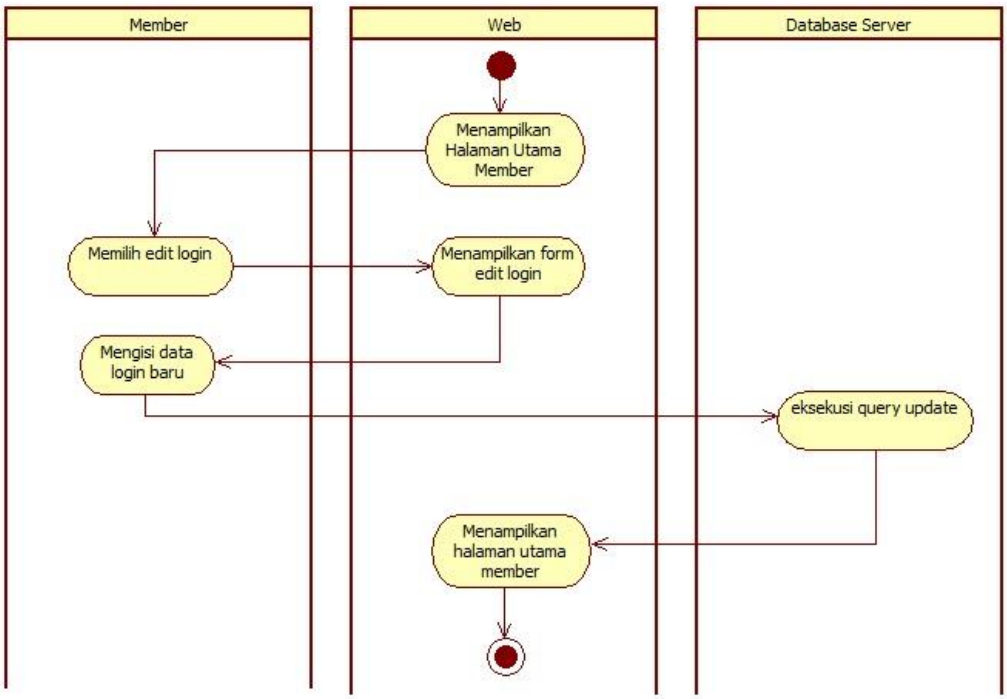
Pada Gambar 3.12 memperlihatkan aktivitas melihat profile member pada aplikasi web. Aplikasi akan menampilkan halaman utama member kemudian member memilih menu profile lalu system akan menampilkan profile member.

i. *Activity* Diagram Ubah Profile



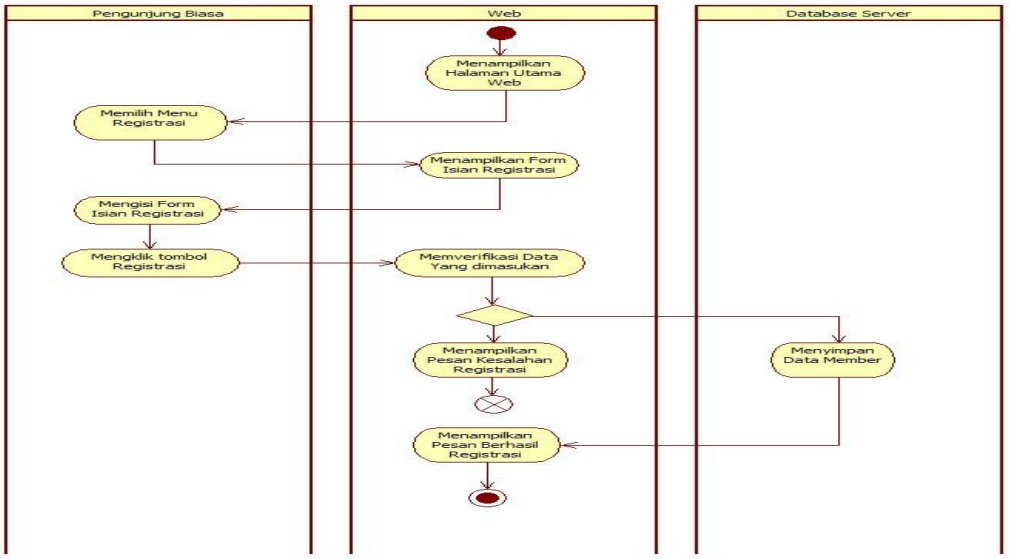
Gambar 3. 13 *Activity* Diagram Ubah Profile

j. *Activity* Diagram Ubah Data Login



Gambar 3. 14 *Activity* Diagram Ubah Data Login

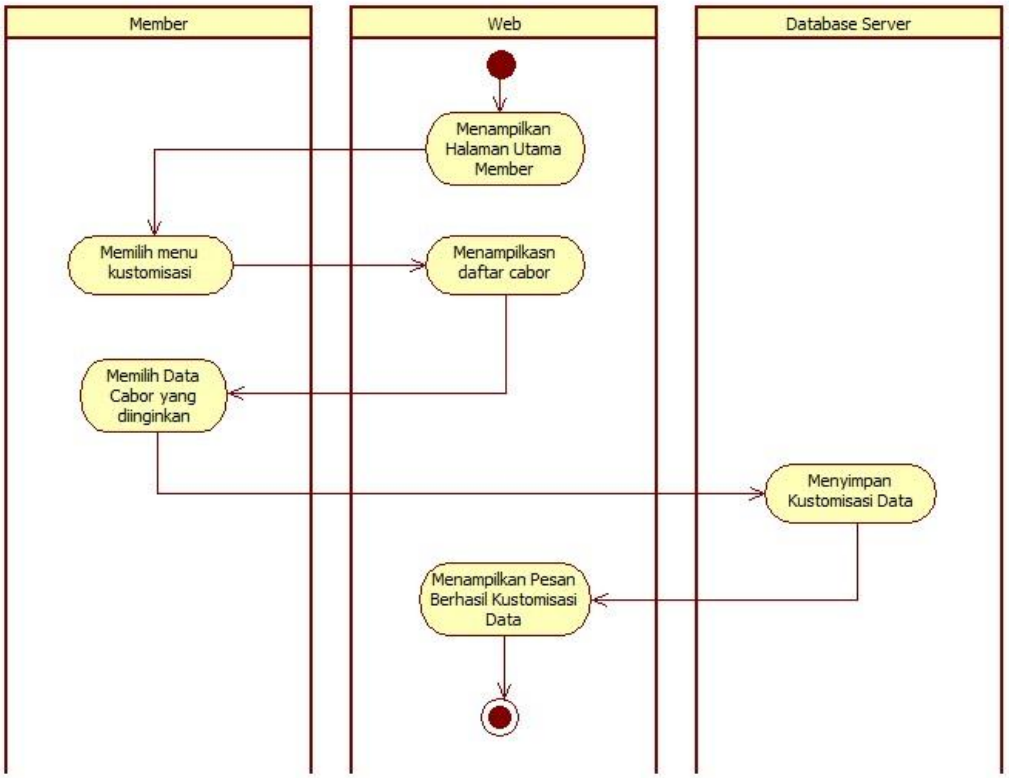
k. *Activity* Diagram Registrasi (*backend)*



Gambar 3. 15 *Activity* Diagram Registrasi (*backend)*

Pada gambar 3.15 memperlihatkan aktivitas *Registrasi* oleh pengunjung biasa.Registrasi ini dilakukan pada aplikasi web, pertama-tama pengunjung mengakses web koni lalu web akan menampilkan halaman utama web, pengunjung dapat memilih menu registrasi. Web akan menampilkan form *registrasi* bagi pengunjung. Pengunjung mengisi *form* isian registrasi dan mengklik tombol *register*, kemudian web akan memverifikasi data registrasi yang dimasukan. Jika sesuai maka akan menampilkan pesan berhasil registrasi dan data akan disimpan kedalam database, jika salah maka akan menampilkan pesan kesalahan registrasi.

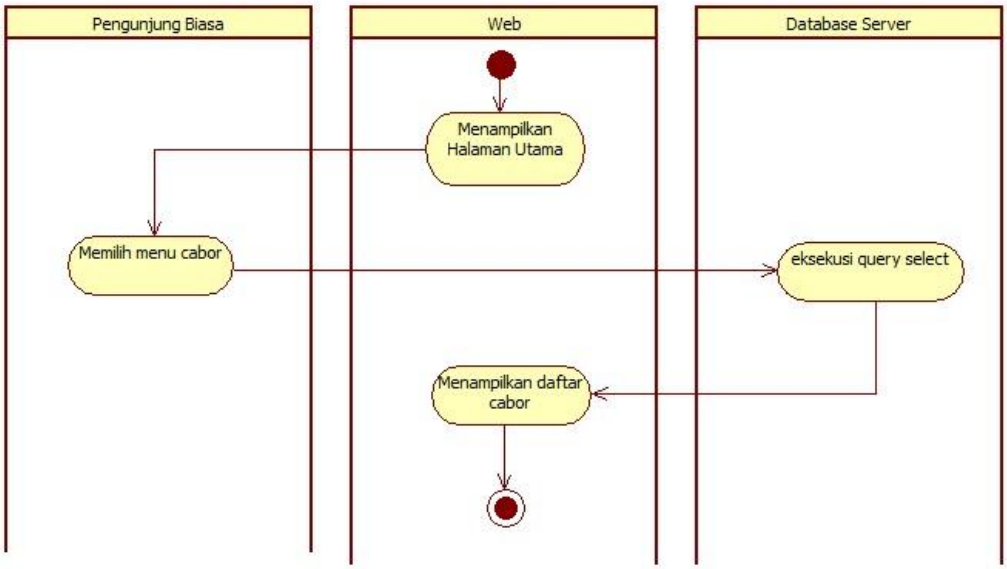
l. *Activity Diagram* Kustomisasi



Gambar 3. 16 *Activity* Diagram Kustomisasi

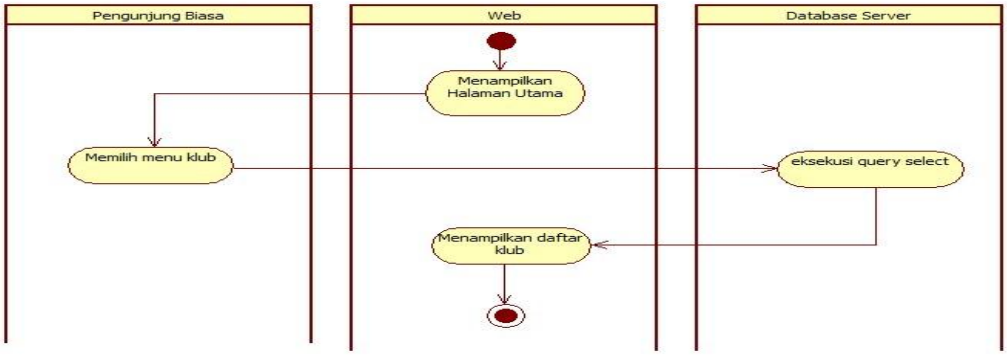
Pada gambar 3.16 menjelaskan aktivitas Kustomisasi Data oleh Member. Setelah Registrasi, Member melakukan Kustomisasi Data pada web. Pada aktivitas ini member melakukan login, setelah berhasil login maka aplikasi akan menampilkan form kustomisasi data. Lalu member memilih data cabor apa saja yang diinginkan, setelah itu data akan disimpan pada database lalu aplikasi akan menampilkan pesan berhasil kustomisasi data.

m. Activity Diagram Melihat Info Cabor (*backend)*



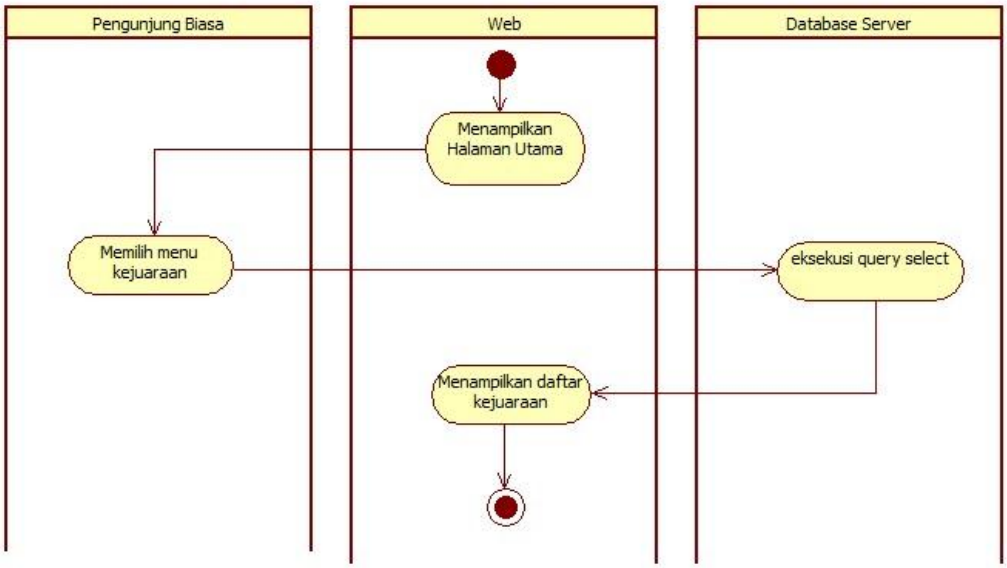
Gambar 3. 17 *Activity* Diagram Melihat Info Cabor (*backend)*

n. *Activity* Diagram Melihat Info Klub (*backend)*



Gambar 3. 18 *Activity* Diagram Melihat Info Klub (*backend)*

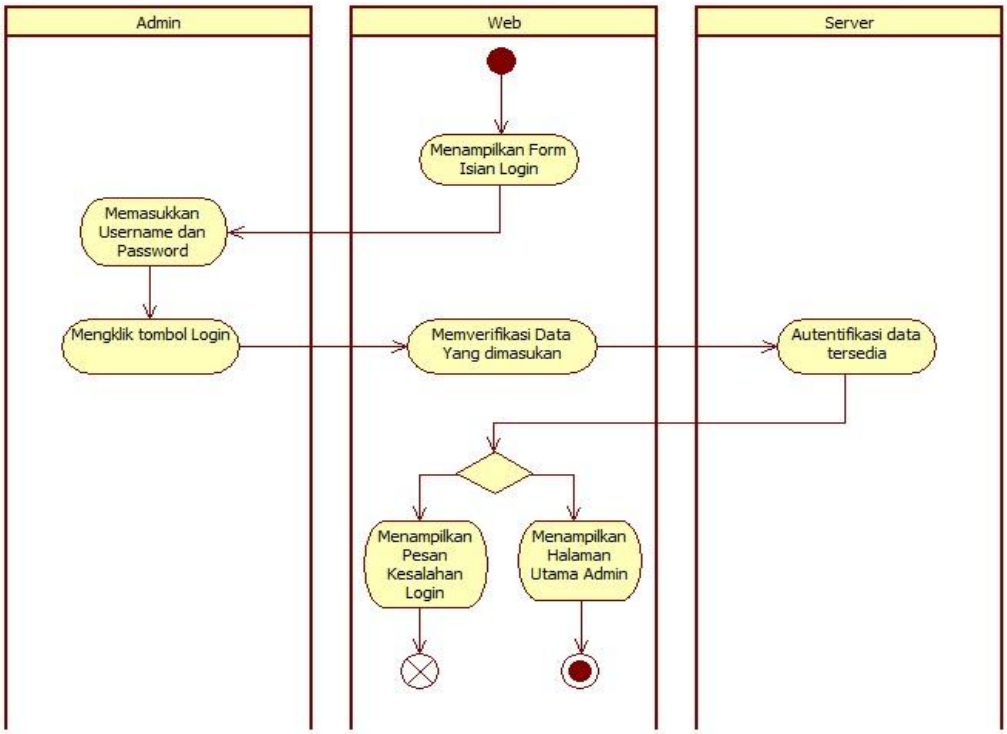
o. Activity Diagram Melihat Info Kejuaraan (*backend)*



Gambar 3. 19 *Activity* Diagram Melihat Info Kejuaraan (*backend*)

2. *Activity Diagram* Untuk User Admin

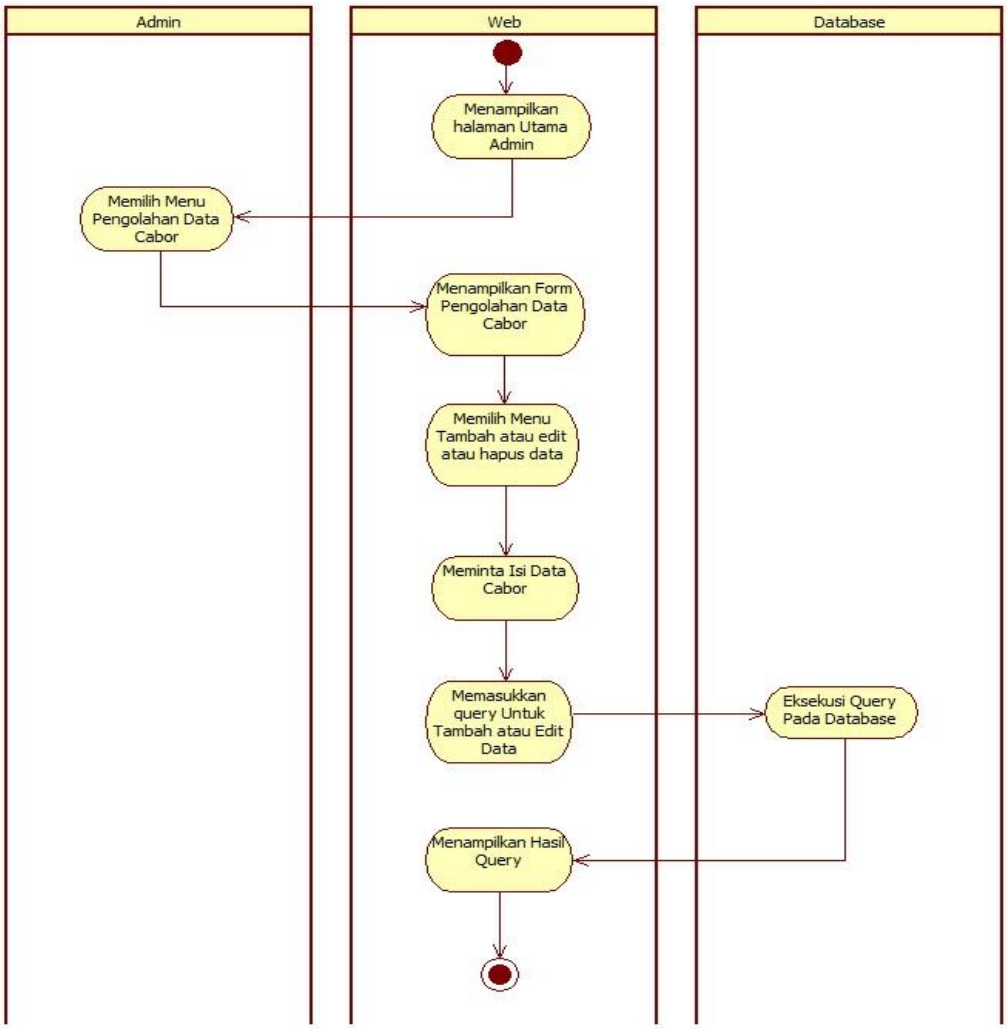
a. *Activity Diagram* Login Admin



Gambar 3. 20 *Activity* Diagram Login Admin

Pada gambar 3.20 memperlihatkan aktivitas *Login Admin*. Web menampilkan *form* isian *login*. Kemudian Admin memasukan *username* dan *password* . kemudian sistem akan mengautentifikasi data yang dimasukan apakah sudah sesuai atau tidak dengan data yang ada di *database*, jika sesuai maka akan menampilkan halaman Admin, jika tidak maka akan menampilkan pesan kesalahan login.

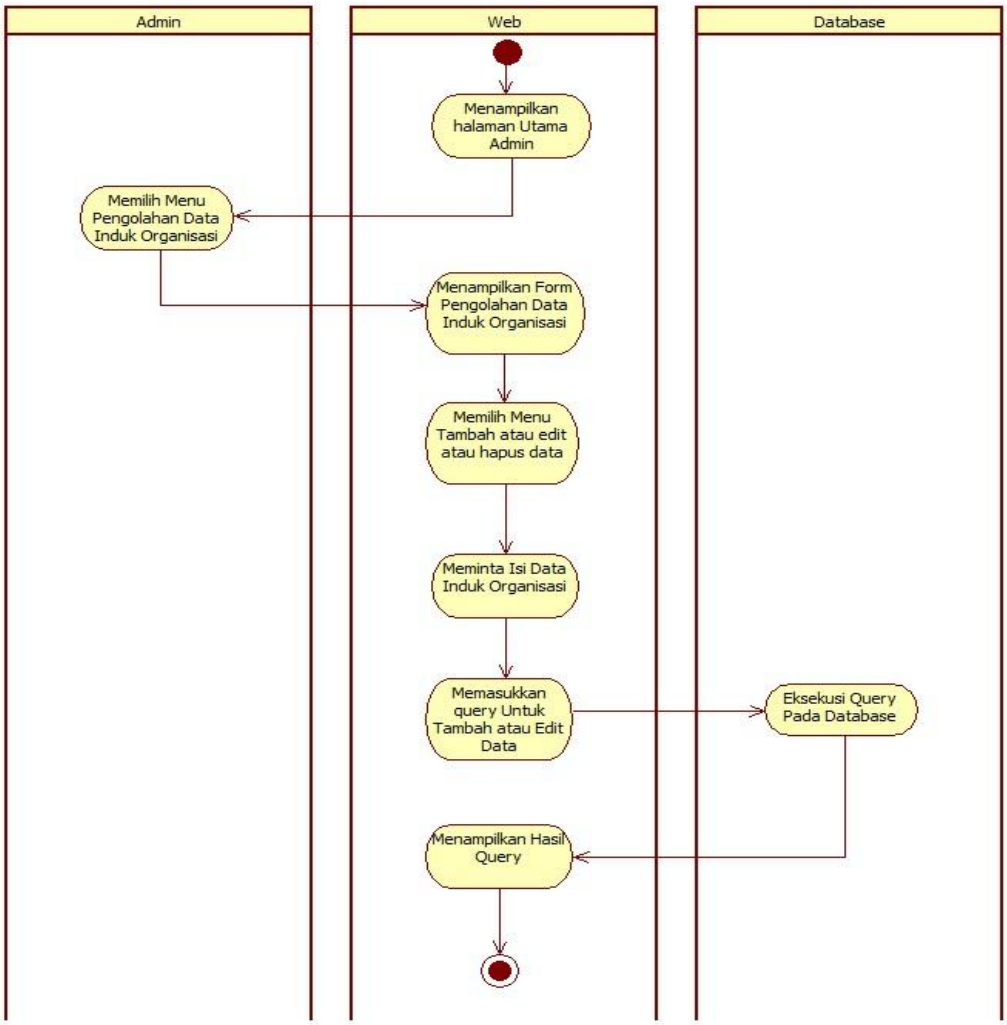
b. *Activity* Diagram Pengolahan Data Cabor



Gambar 3. 21 *Activity* Diagram Pengolahan Data Cabor

Pada gambar 3.21 memperlihatkan aktivitas pengolahan data cabor. Web menampilkan halaman utama admin, Admin memilih menu pengolahan data cabor, kemudian Web akan menampilkan form pengolahan data cabor. Lalu admin dapat memilih sublink untuk tambah atau ubah data cabor lalu Web akan menampilkan Form isian data. *Server* mengeksekusi hasil query kemudian mengirimkannya ke aplikasi *Backend* (web).

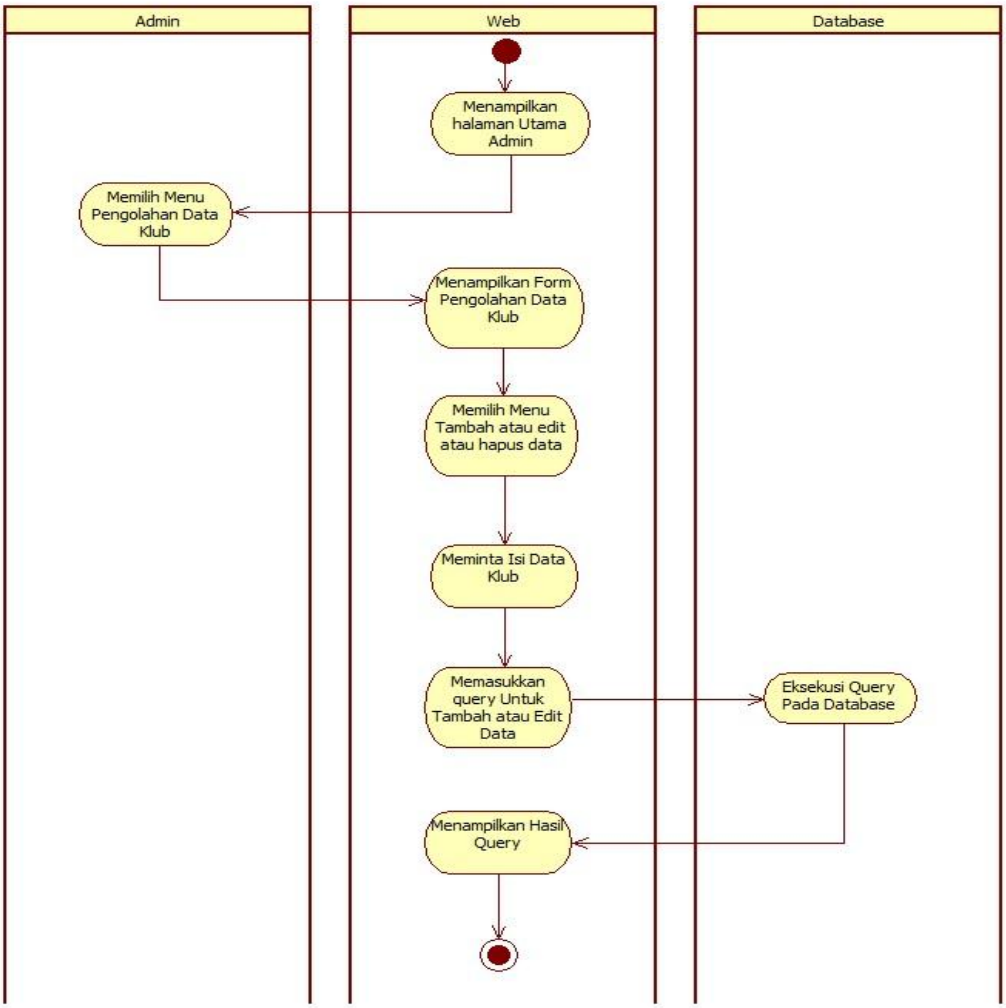
c. *Activity* Diagram Pengolahan Data Induk Organisasi



Gambar 3. 22 *Activity* Diagram Pengolahan Data Induk Organisasi

Pada gambar 3.22 memperlihatkan aktivitas pengolahan data induk organisasi. Web menampilkan halaman utama admin, Admin memilih menu pengolahan data induk organisasi, kemudian Web akan menampilkan form pengolahan data induk organisasi. Lalu admin dapat memilih sublink untuk tambah atau ubah data induk organisasi lalu Web akan menampilkan Form isian data. *Server* mengeksekusi hasil query kemudian mengirimkannya ke aplikasi *Backend* (web).

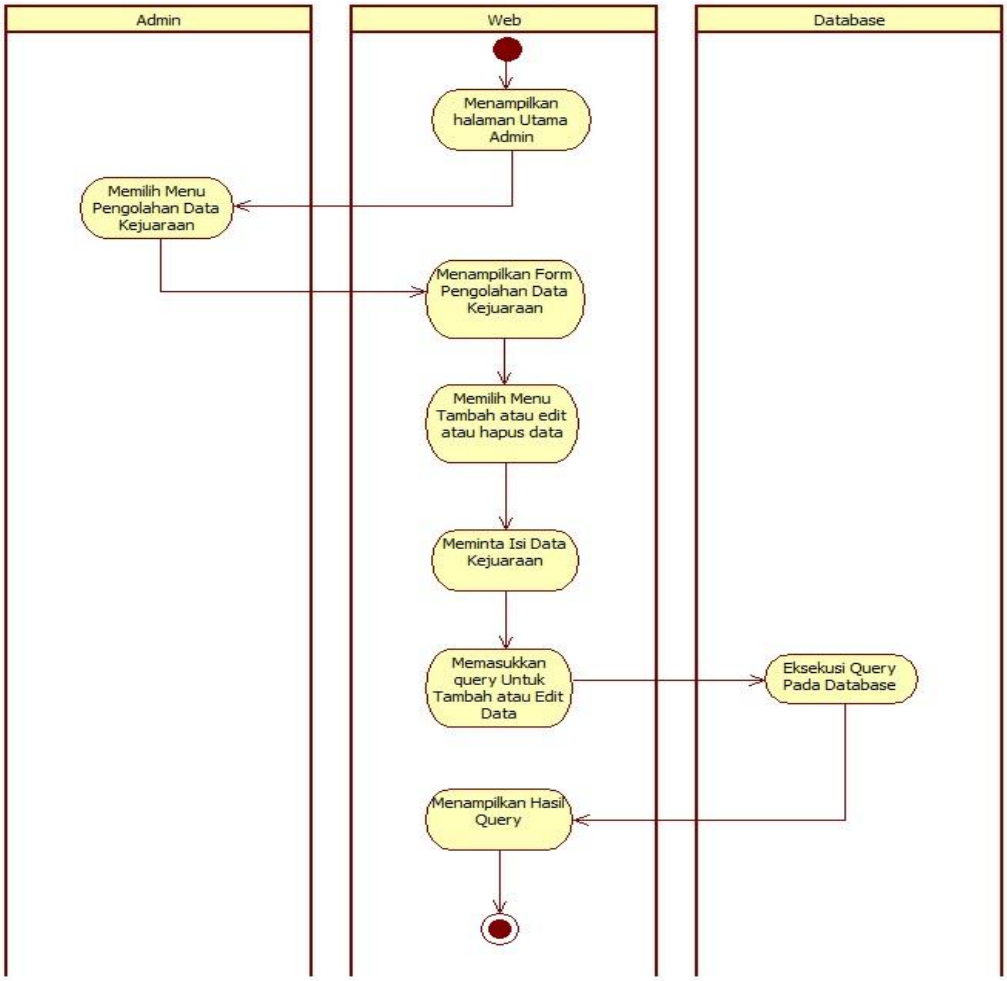
d. *Activity* Diagram Pengolahan Data Klub



Gambar 3. 23 *Activity* Diagram Pengolahan Data Klub

Pada gambar 3.23 memperlihatkan aktivitas pengolahan data klub. Web menampilkan halaman utama admin, Admin memilih menu pengolahan data klub, kemudian web akan menampilkan form pengolahan data klub. Lalu admin dapat memilih sublink untuk tambah atau ubah data klub lalu web akan menampilkan Form isian data. *Server* mengeksekusi hasil query kemudian mengirimkannya ke aplikasi *Backend* (web).

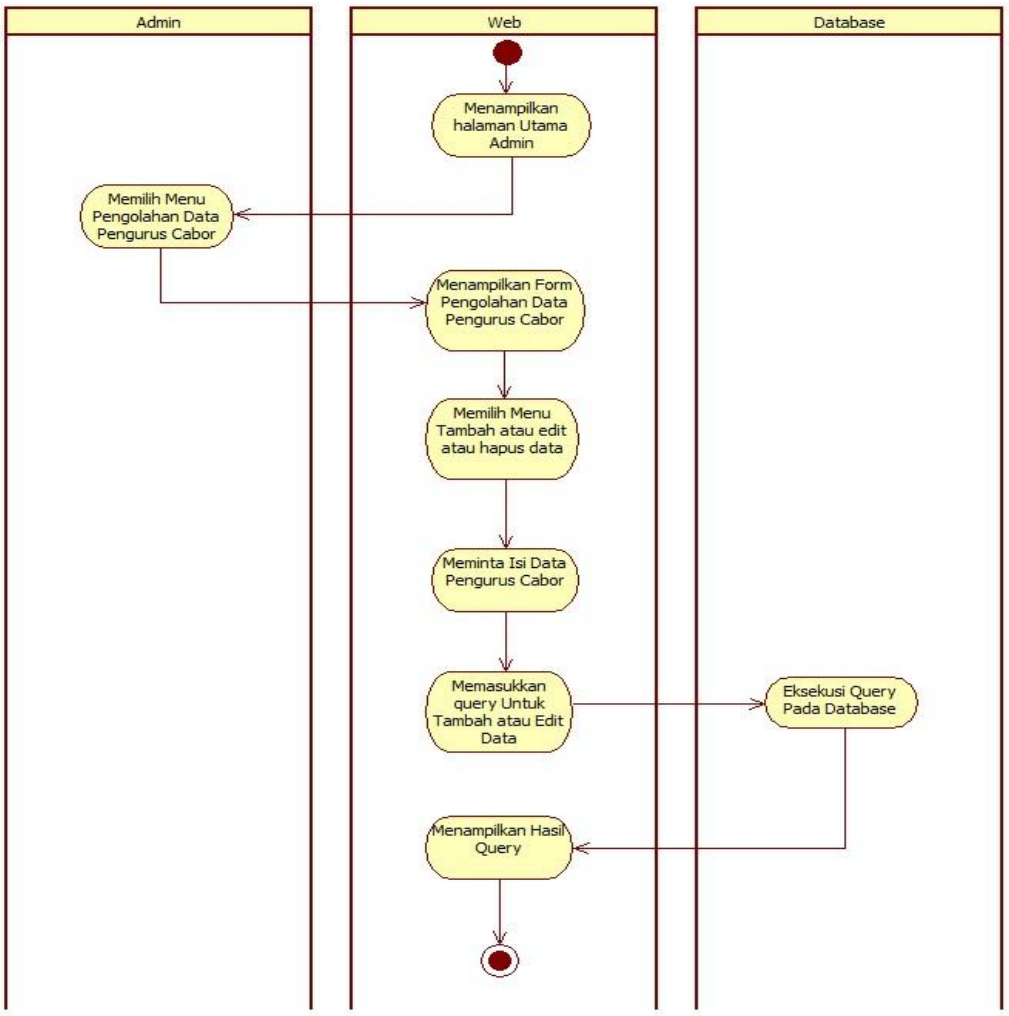
e. *Activity* Diagram Pengolahan Data Kejuaraan



Gambar 3. 24 *Activity* Diagram Pengolahan Data Kejuaraan

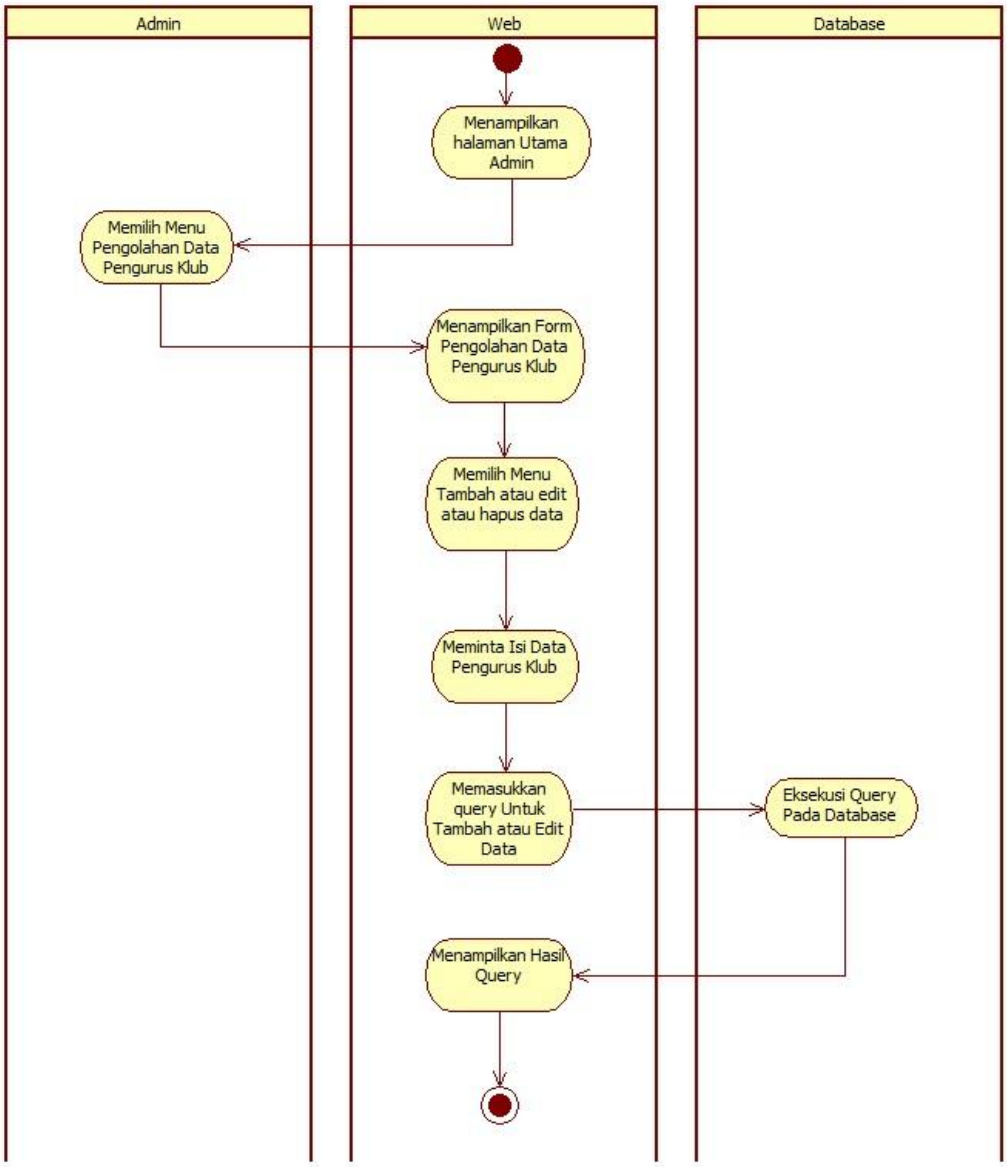
Pada gambar 3.24 memperlihatkan aktivitas pengolahan data kejuaraan. Web menampilkan halaman utama admin, Admin memilih menu pengolahan data kejuaraan, kemudian Web akan menampilkan form pengolahan data kejuaraan. Lalu admin dapat memilih sublink untuk tambah atau ubah data kejuaraan lalu Web akan menampilkan Form isian data. *Server* mengeksekusi hasil query kemudian mengirimkannya ke aplikasi *Backend* (web).

f. *Activity* Diagram Pengolahan Data Pengurus Cabor



Gambar 3. 25 *Activity* Diagram Pengolahan Data Pengurus Cabor

g. *Activity* Diagram Pengolahan Data Pengurus Klub



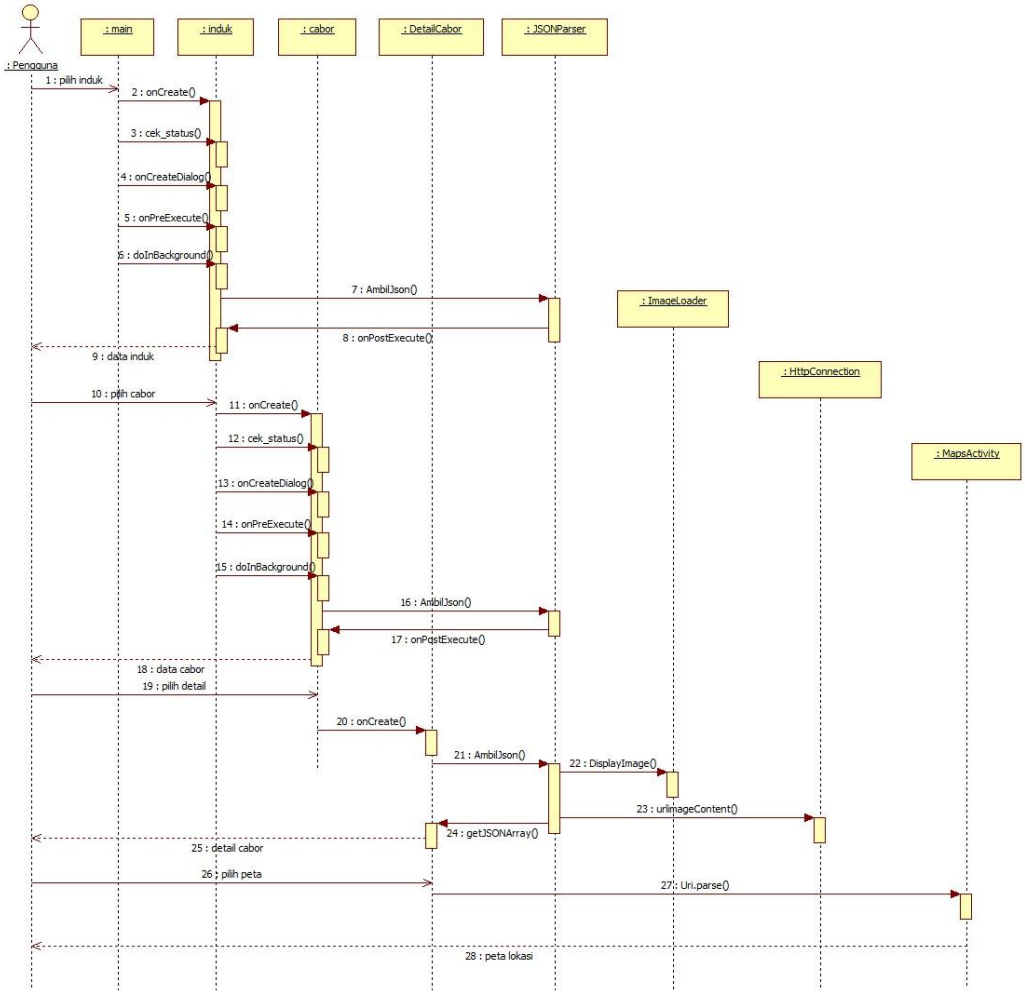
Gambar 3. 26 *Activity* Diagram Pengolahan Data Pengurus Klub

d. *Sequence* Diagram

Pada tahap ini *Sequence Diagram* menjelaskan secara detail urutan proses yang dilakukan sistem untuk mencapai tujuan dari *Use Case*. interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

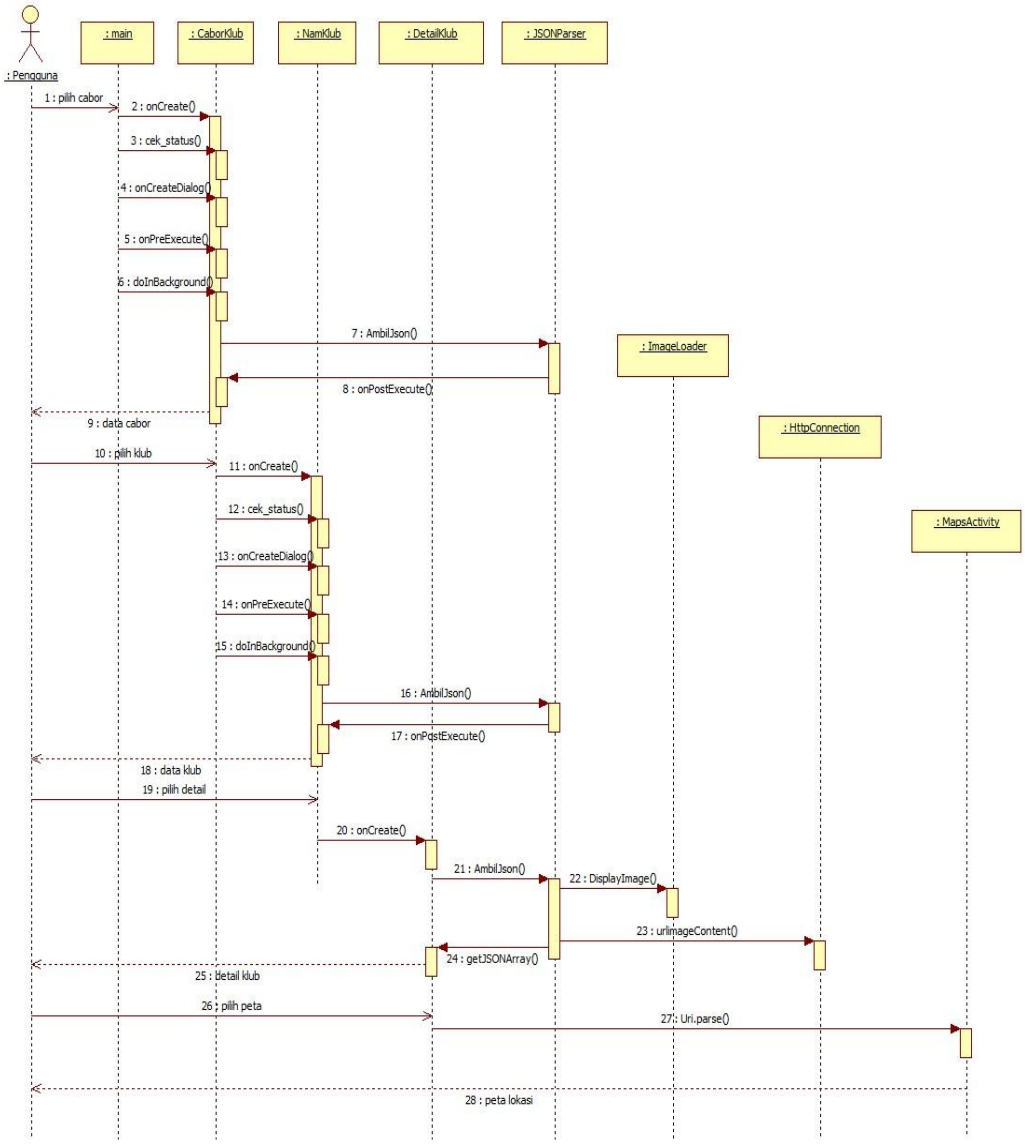
1. *Sequence* Diagram Aplikasi *Frontend*

a. *Sequence* Diagram Melihat Info Cabor (*frontend)*



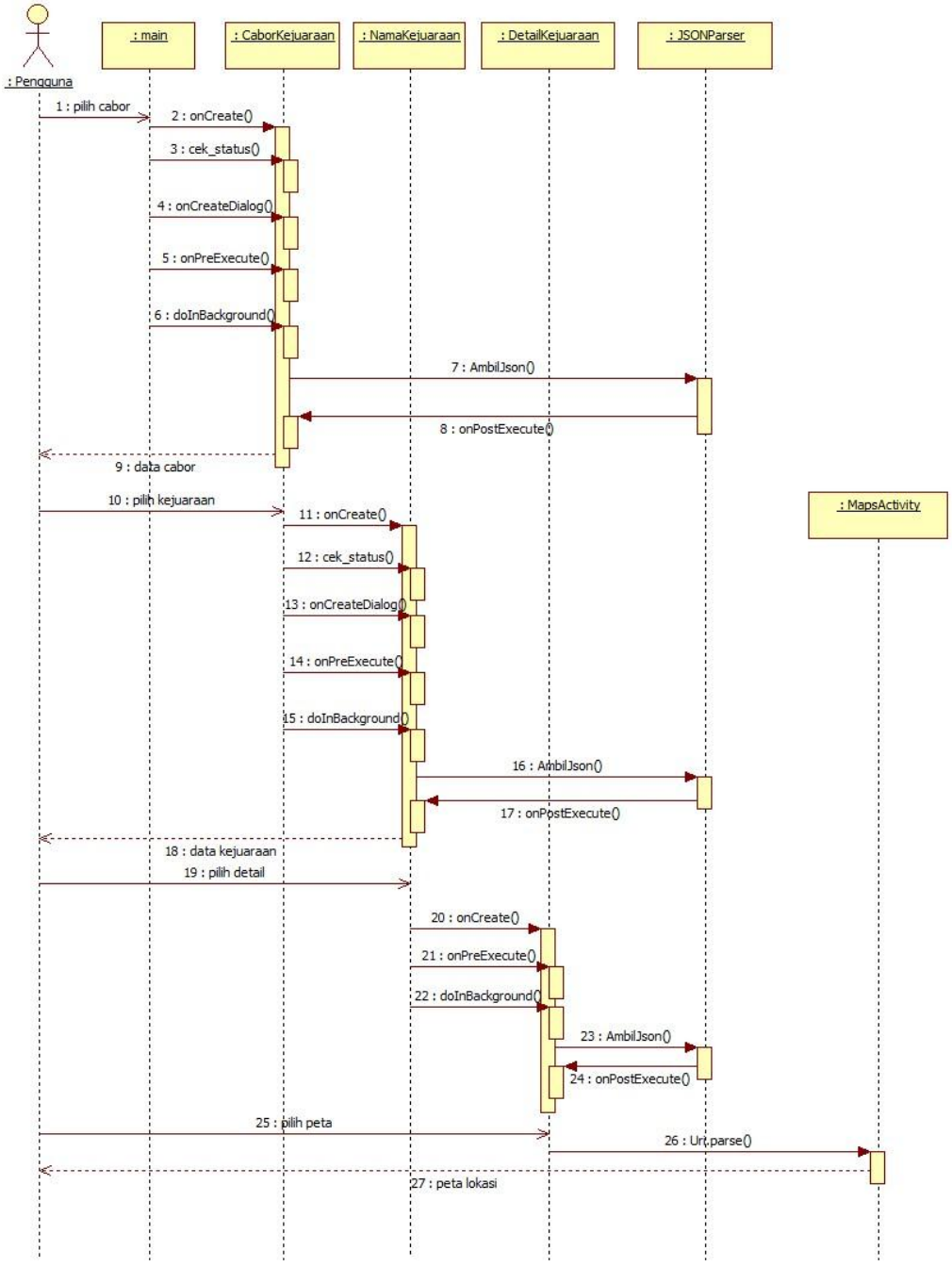
Gambar 3. 27 *Sequence* Diagram Melihat Info Cabor (*backend)*

b. *Sequence* Diagram Melihat Info Klub (*frontend)*



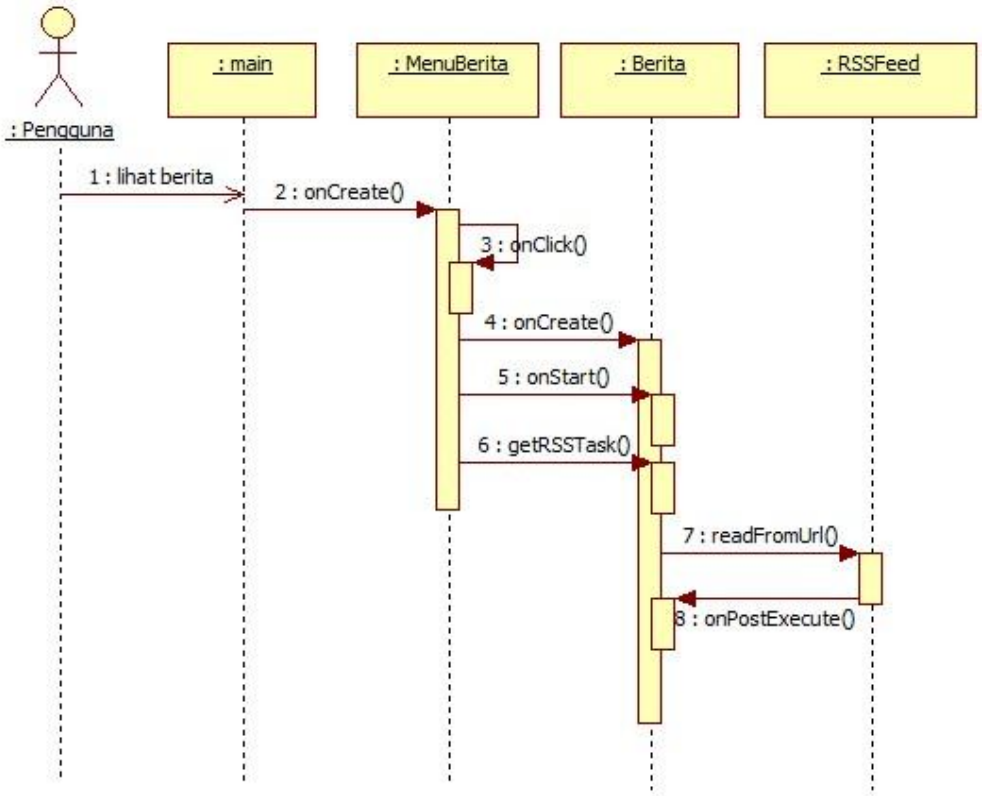
Gambar 3. 28 *Sequence* Diagram Melihat Info Klub (*backend*)

c. *Sequence* Diagram Melihat Info Kejuaraan (*frontend)*



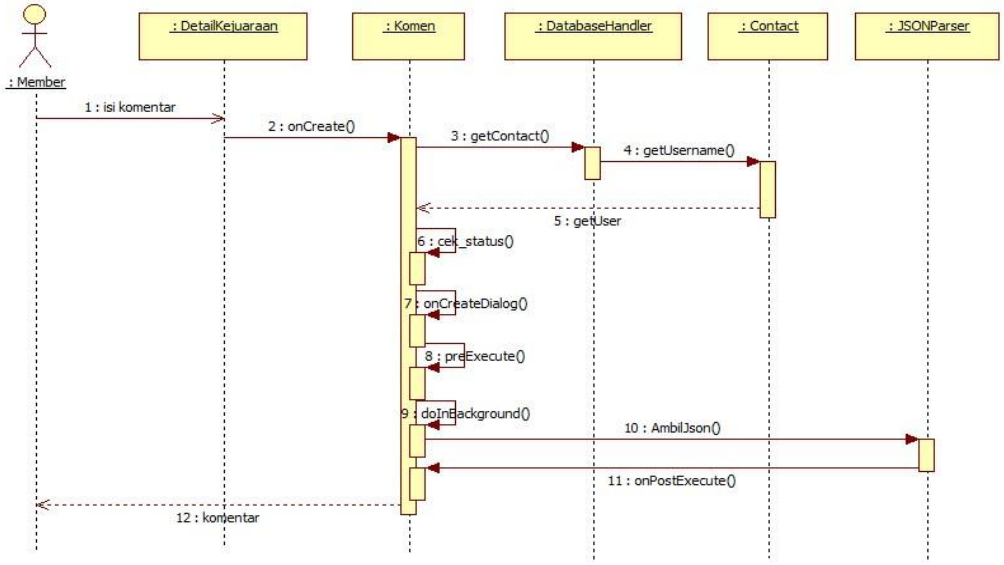
Gambar 3. 29 *Sequence* Diagram Melihat Info Kejuaraan (*backend)*

d. *Sequence* Diagram Melihat Berita Olahraga



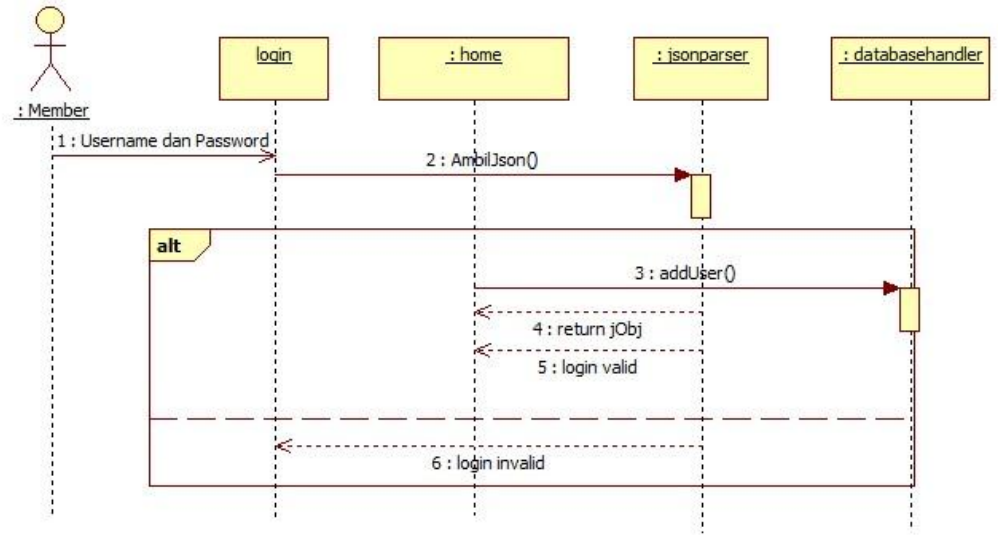
Gambar 3. 30 *Sequence* Diagram Pilih Berita Olahraga

e. *Sequence* Diagram Kirim Komentar



Gambar 3. 31 *Sequence* Diagram Kirim Komentar

f. *Sequence* Diagram Login Member

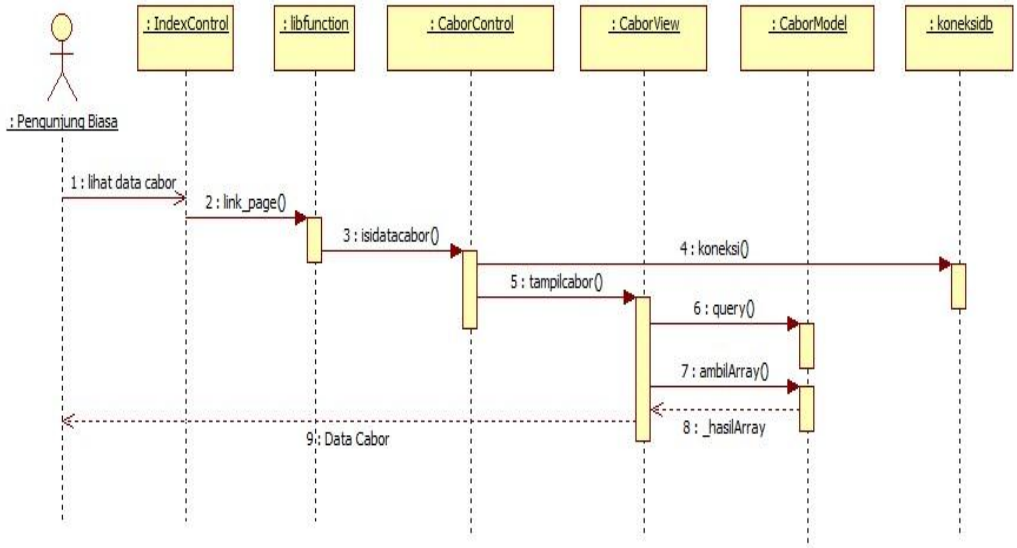


Gambar 3. 32 *Sequence* Diagram Login Member

2. *Sequence* Diagram Aplikasi *Backend*

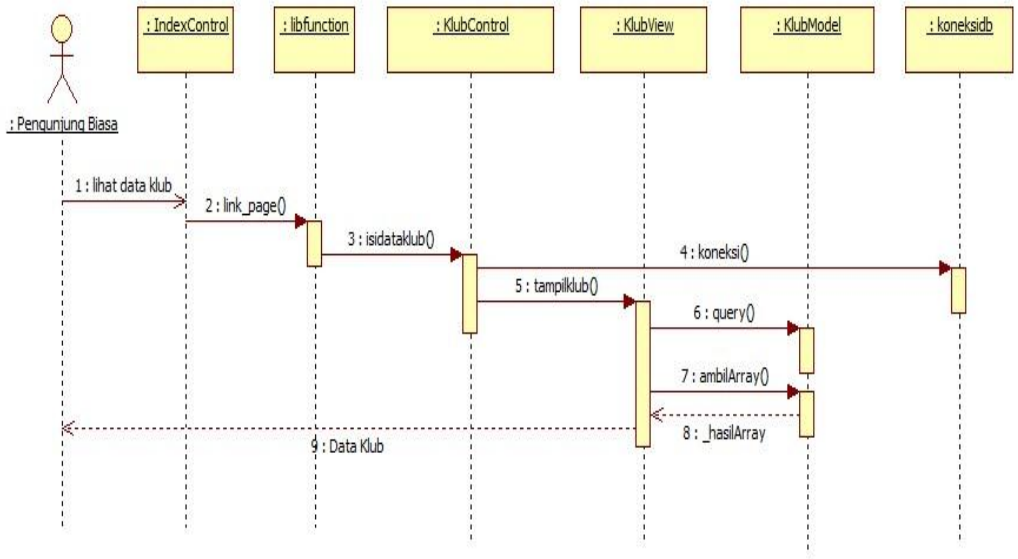
A Sequence Diagram Aplikasi *Backend Halaman Depan*

a. Sequence Diagram Melihat Cabor (*backend)*



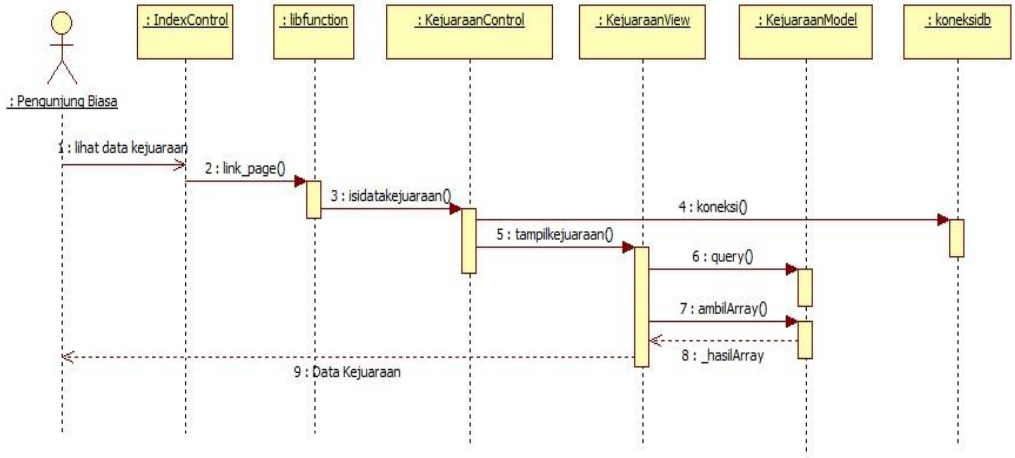
Gambar 3. 33 *Sequence* Diagram Melihat Cabor (*backend)*

b. *Sequence* Diagram Melihat Klub (*backend)*



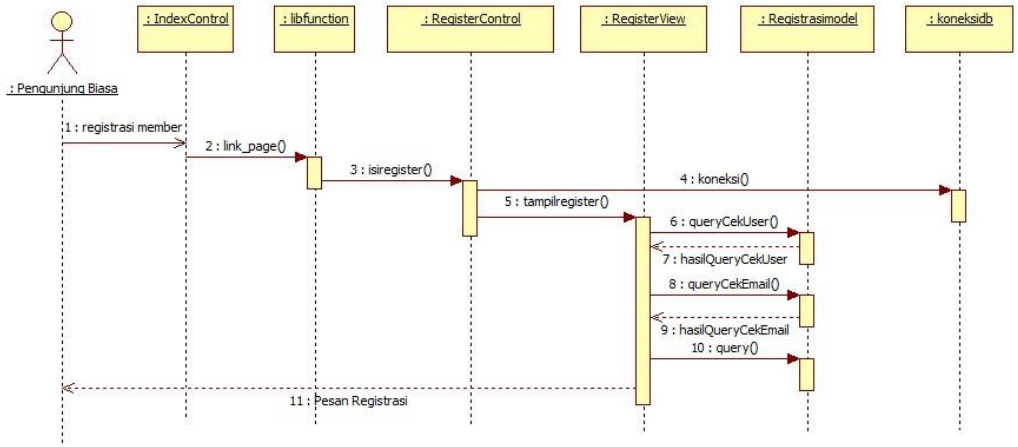
Gambar 3. 34 *Sequence* Diagram Melihat Klub (*backend)*

c. *Sequence* Diagram Melihat Kejuaraan (*backend)*



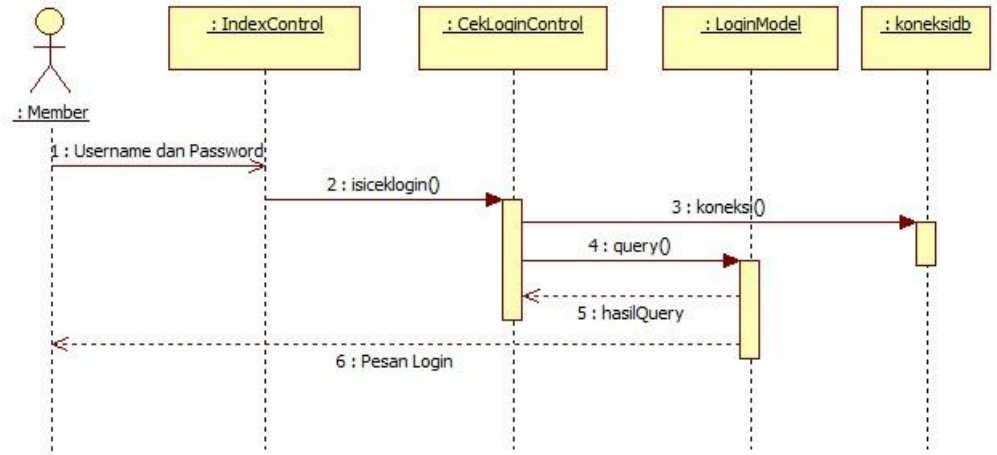
Gambar 3. 35 *Sequence* Diagram Melihat Kejuaraan (*backend)*

d. *Sequence* Diagram Registrasi



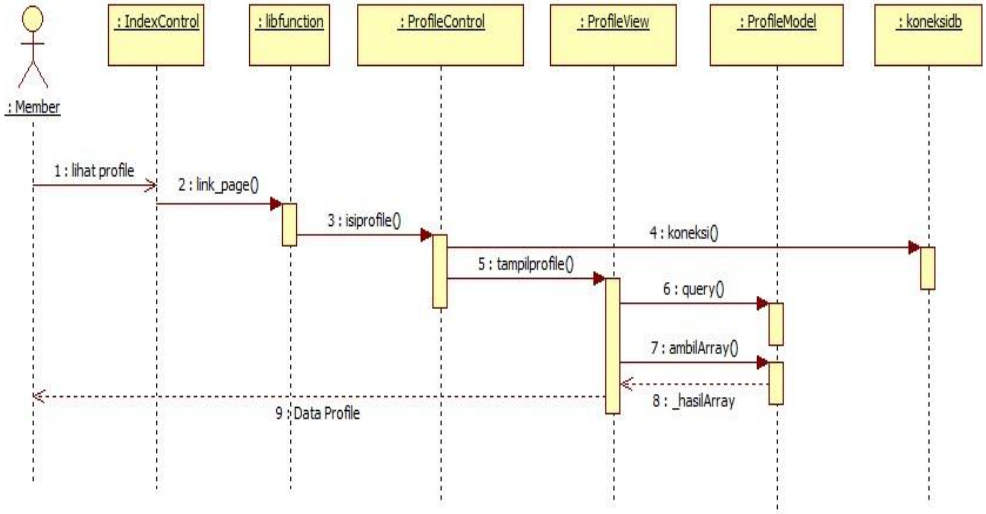
Gambar 3. 36 *Sequence* Diagram Registrasi

e. *Sequence* Diagram Login Member (*backend)*



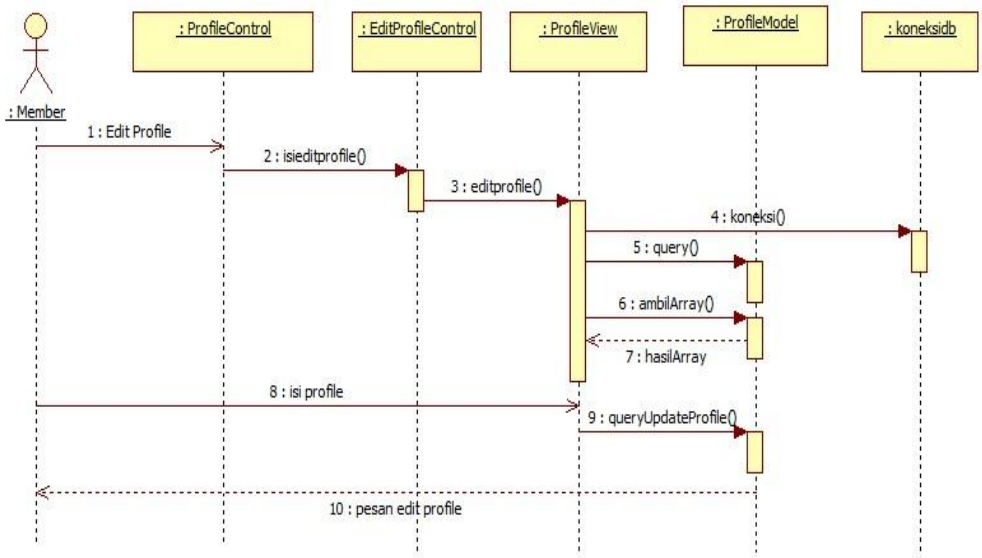
Gambar 3. 37 *Sequence* Diagram Login Member (*backend)*

f. *Sequence* Diagram Melihat Profile



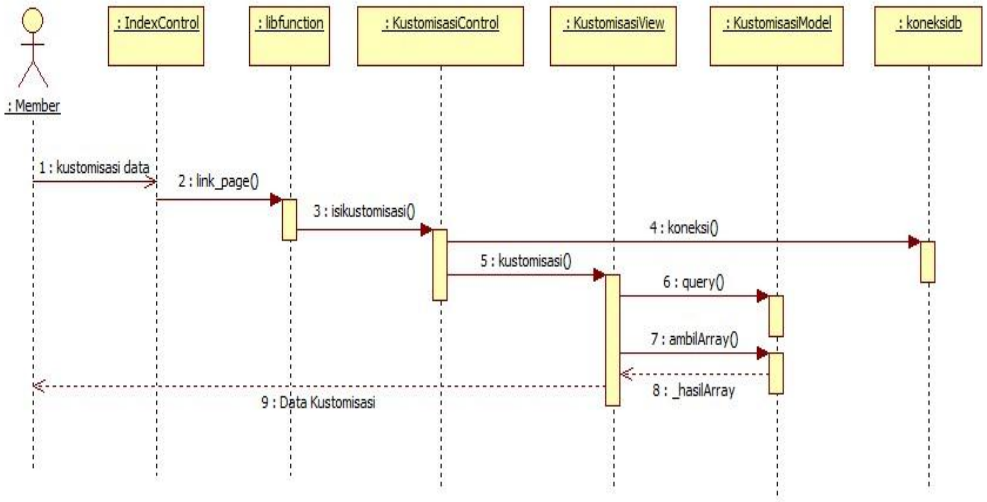
Gambar 3. 38 *Sequence* Diagram Melihat Profile

g. *Sequence* Diagram Ubah Profile



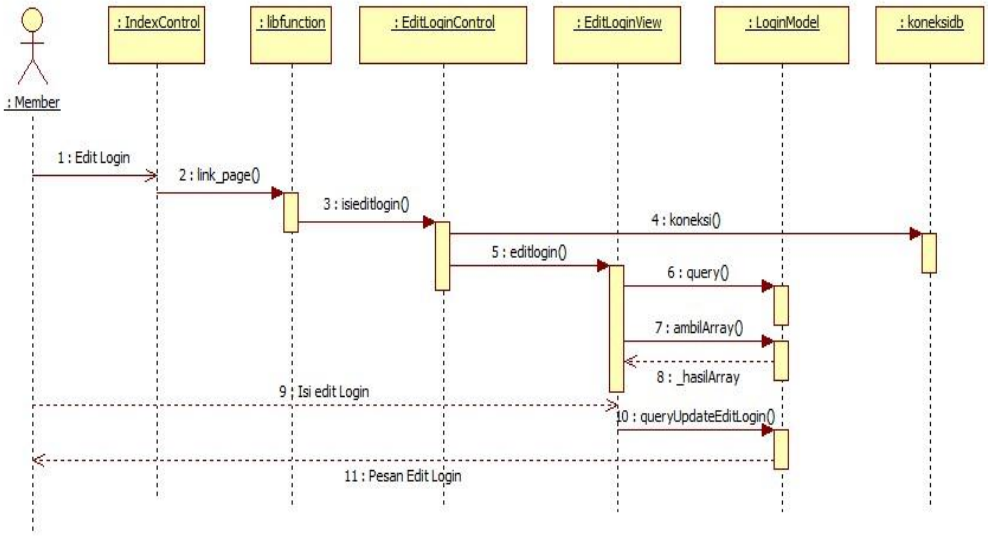
Gambar 3. 39 *Sequence* Diagram Ubah Profile

h. Sequence Diagram Kustomisasi Data



Gambar 3. 40 *Sequence* Diagram Kustomisasi Data

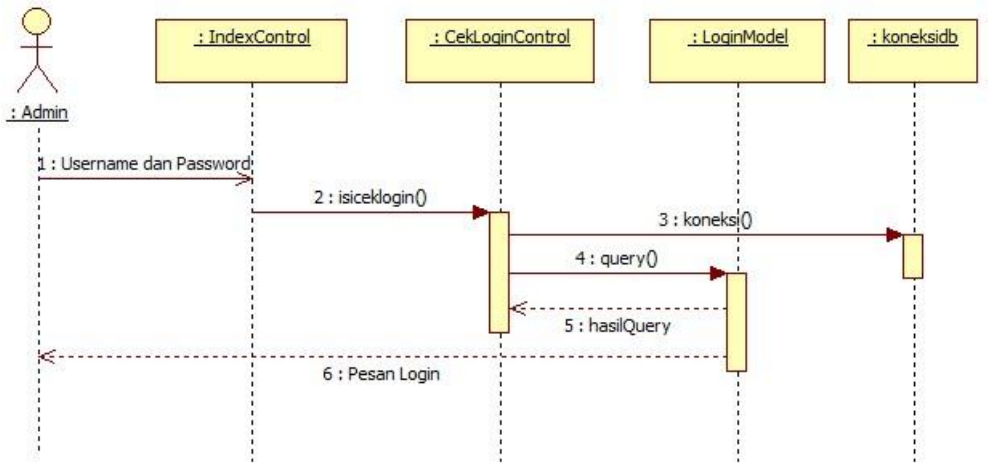
i. *Sequence* Diagram Edit Login



Gambar 3. 41 *Sequence* Diagram Edit Login

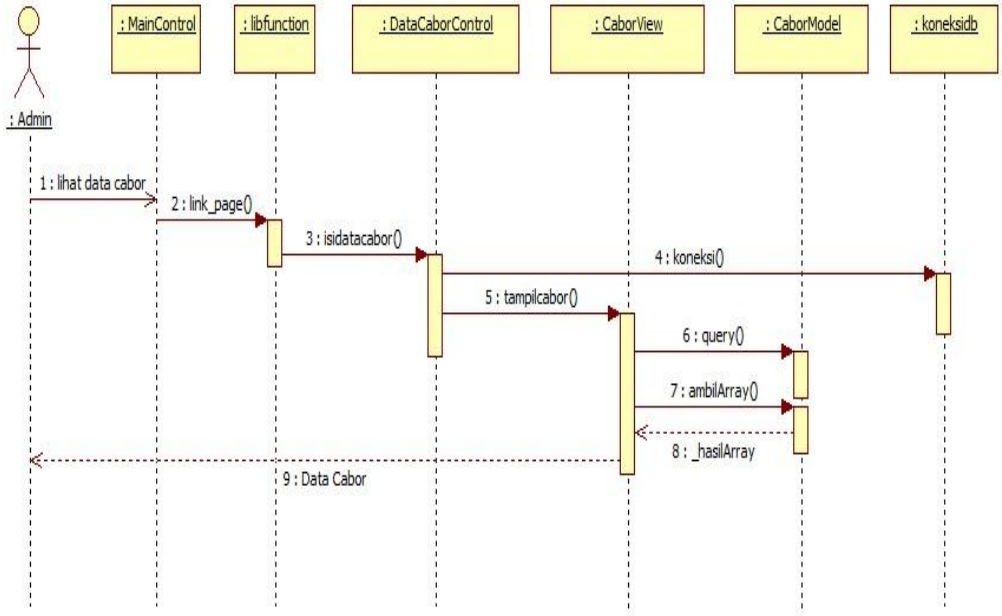
B. *Sequence* Diagram Aplikasi *Backend* Halaman Admin

a. *Sequence* Diagram Login Admin



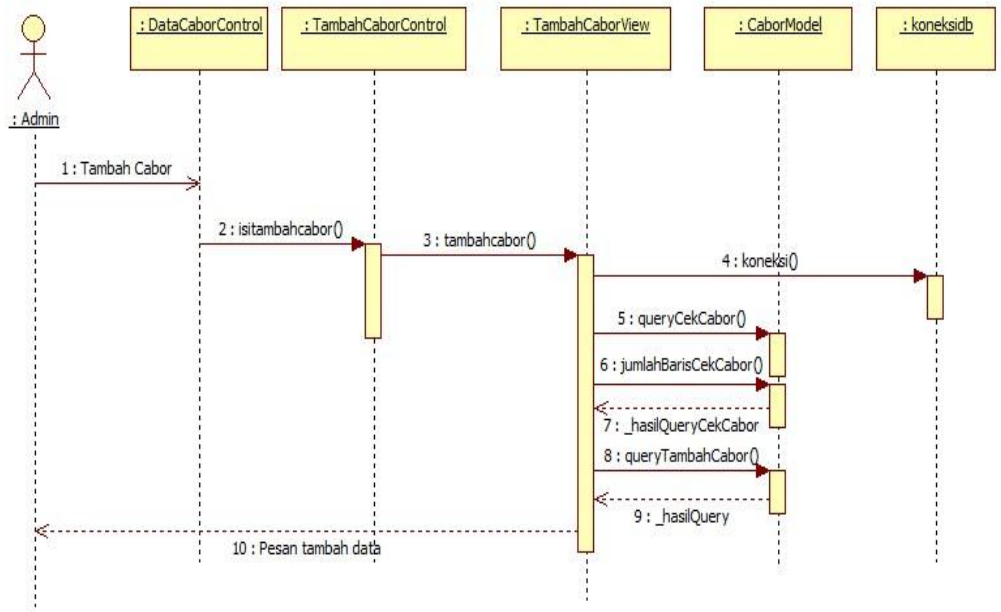
Gambar 3. 42 *Sequence* Diagram Login Admin

b. *Sequence* Diagram Pengolahan Data Cabor



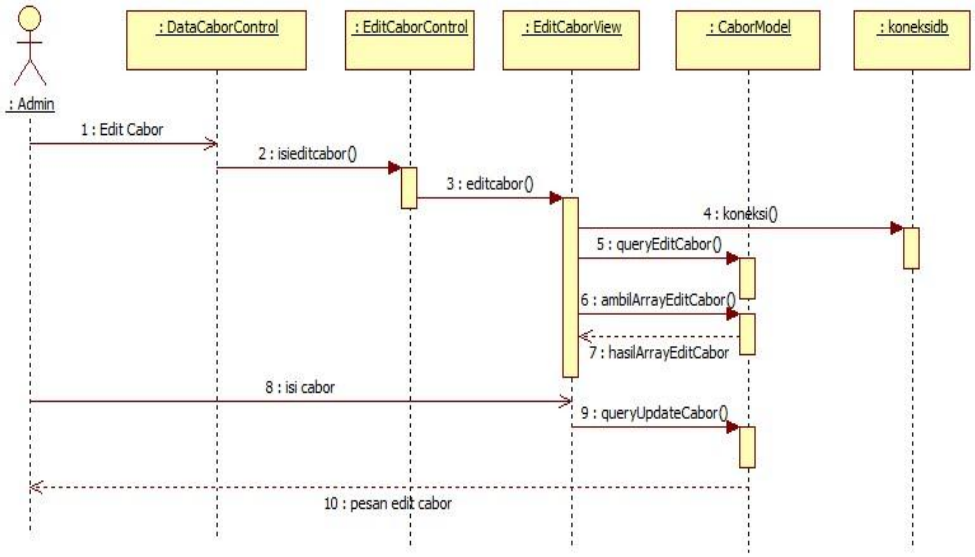
Gambar 3. 43 *Sequence* Diagram Pengolahan Data Cabor

c. *Sequence* Diagram Tambah Cabor



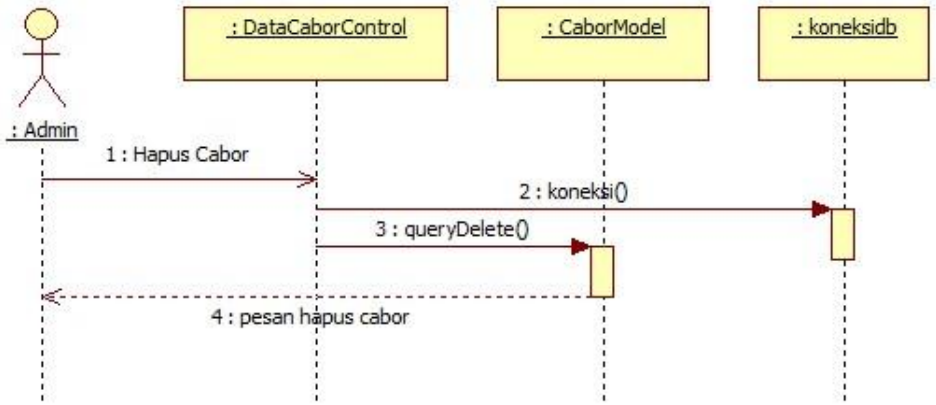
Gambar 3. 44 *Sequence* Diagram Tambah Cabor

d. *Sequence* Diagram Ubah Cabor



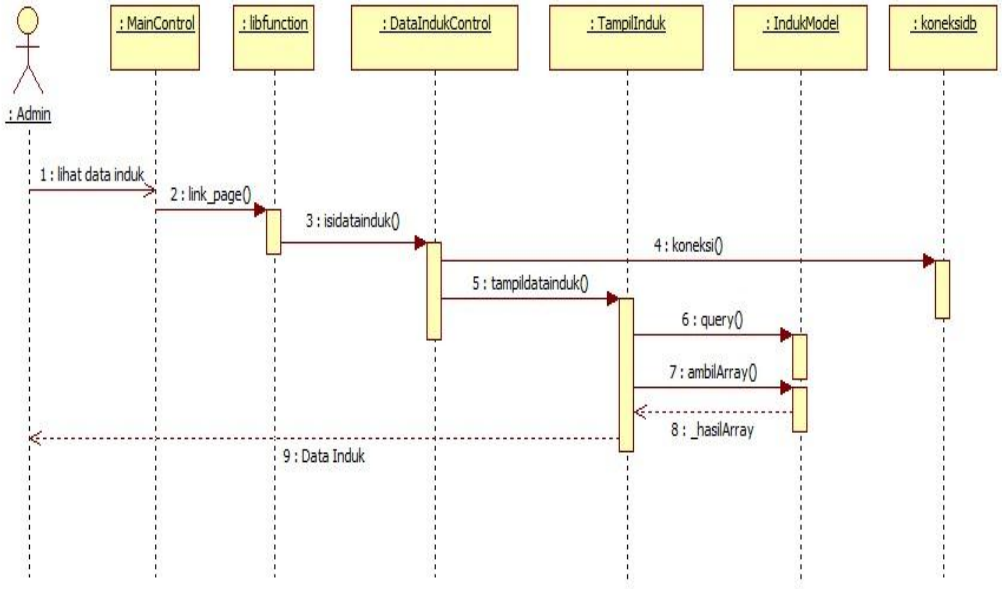
Gambar 3. 45 *Sequence* Diagram Ubah Cabor

e. *Sequence* Diagram Hapus Cabor



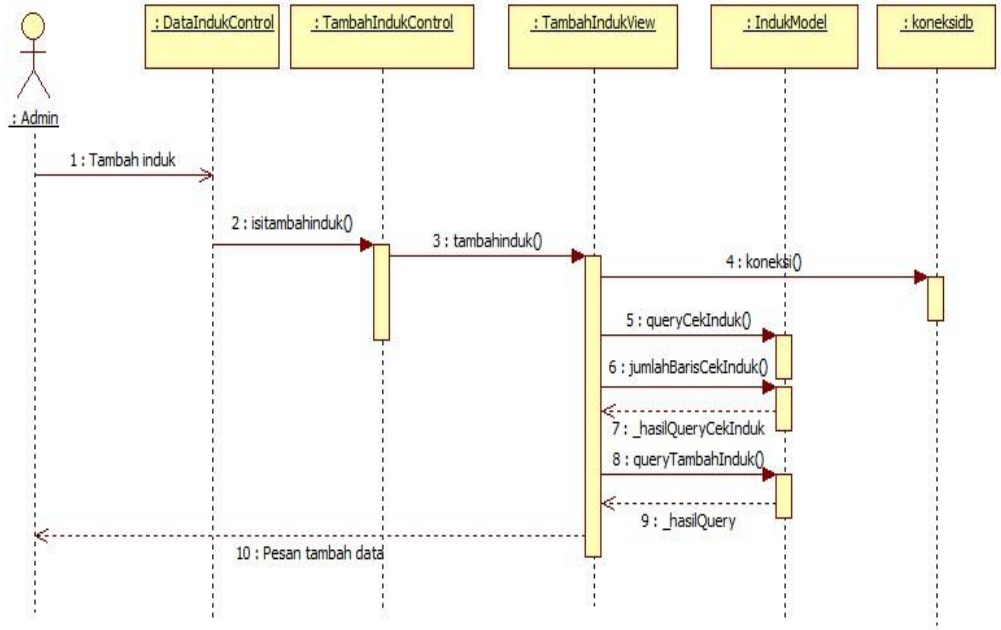
Gambar 3. 46 *Sequence* Diagram Hapus Cabor

f. *Sequence* Diagram Pengolahan Data Induk Organisasi



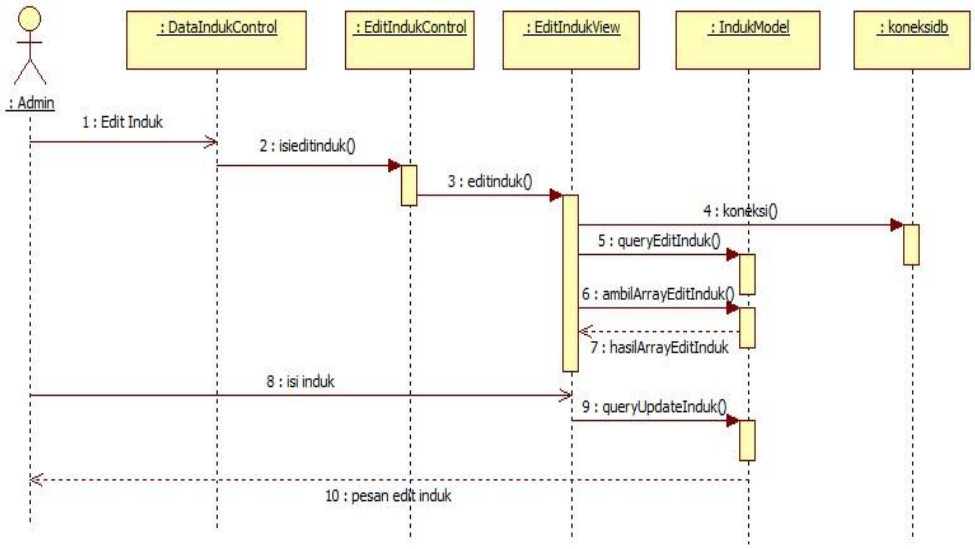
Gambar 3. 47 *Sequence* Diagram Pengolahan Data Induk Organisasi

g. *Sequence* Diagram Tambah Induk Organisasi



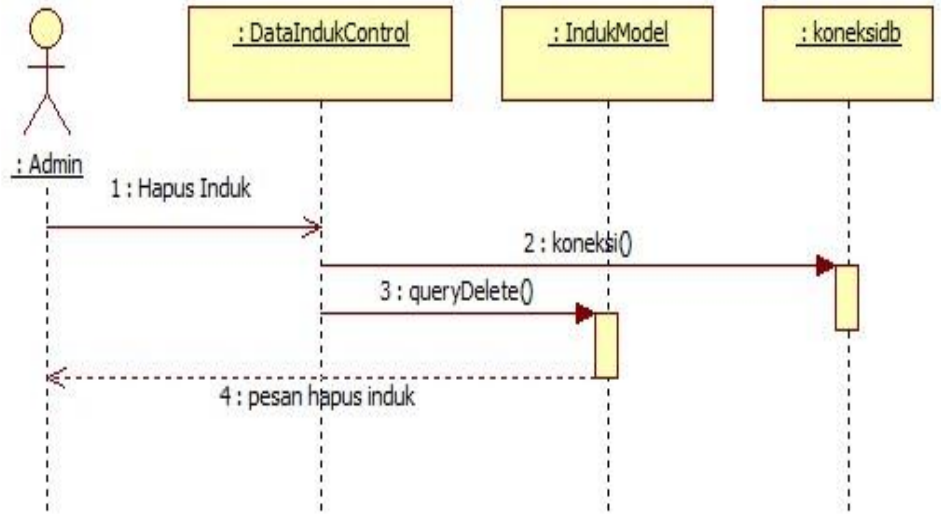
Gambar 3. 48 *Sequence* Diagram Tambah Induk Organisasi

h. *Sequence* Diagram ubah Induk Organisasi



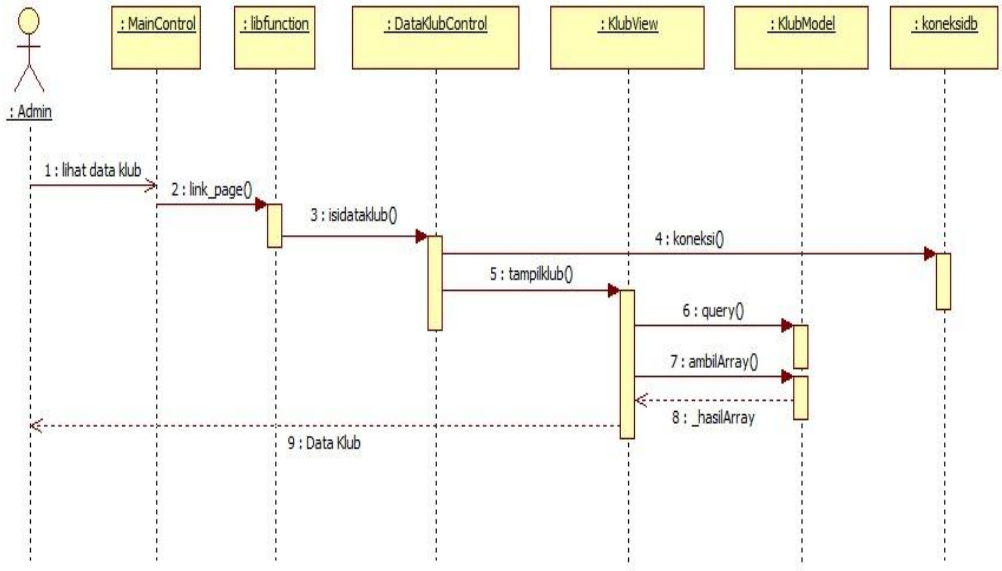
Gambar 3. 49 *Sequence* Diagram Ubah Induk Organisasi

i. *Sequence* Diagram Hapus Induk Organisasi



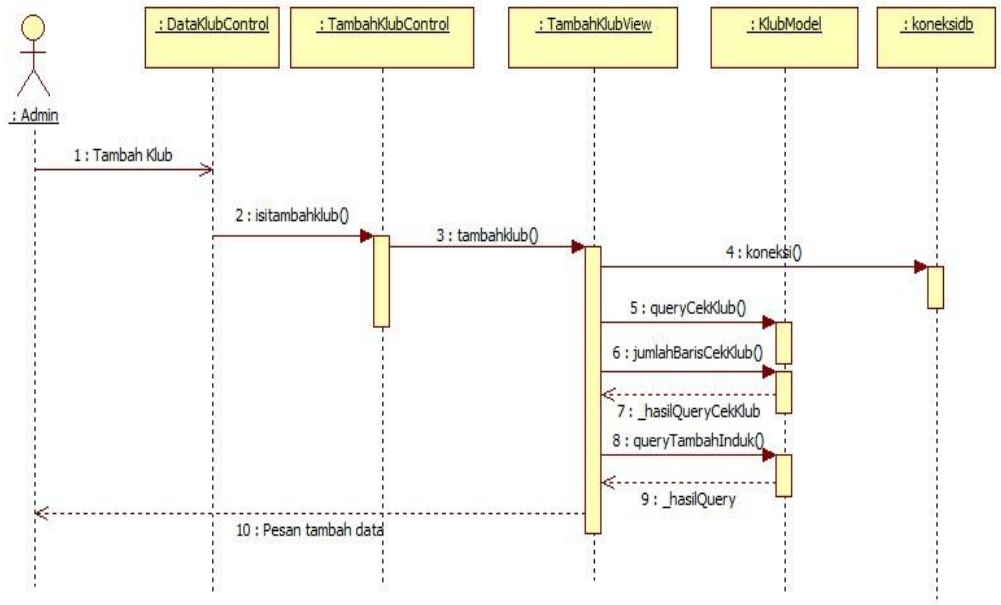
Gambar 3. 50 *Sequence* Diagram Hapus Induk Organisasi

j. *Sequence* Diagram Pengolahan Data Klub



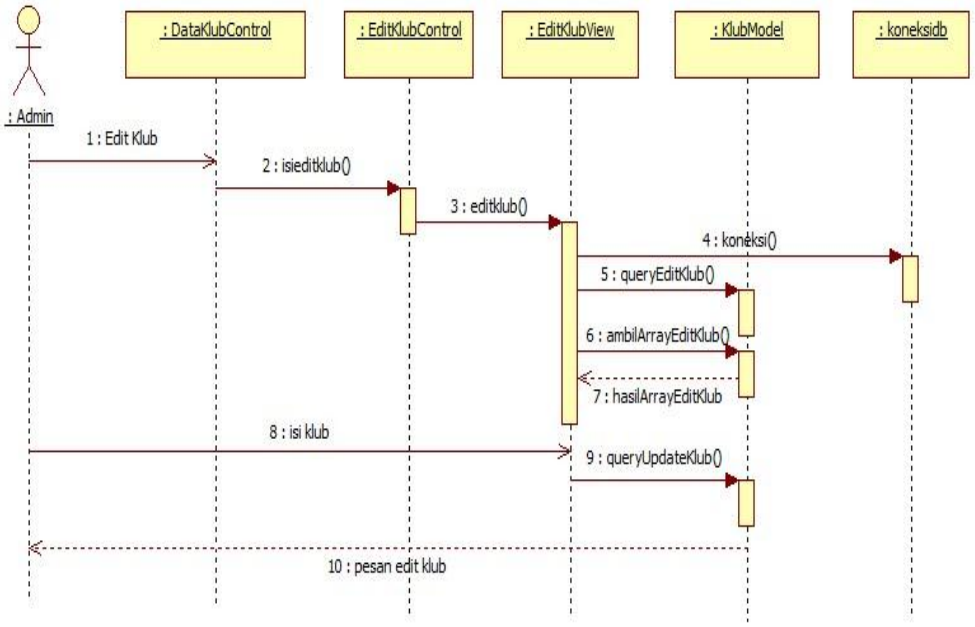
Gambar 3. 51 *Sequence* Diagram Pengolahan Data Klub

k. *Sequence* Diagram Tambah Klub



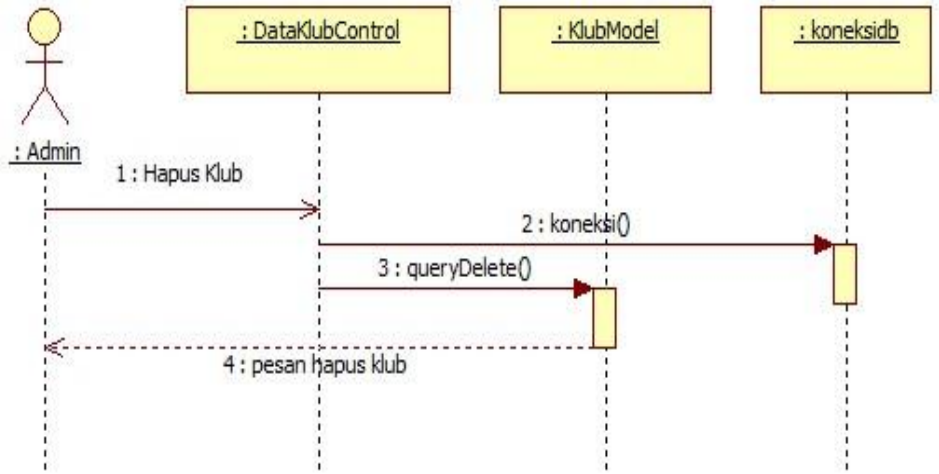
Gambar 3. 52 *Sequence* Diagram Tambah Klub

l. *Sequence* Diagram Ubah Klub



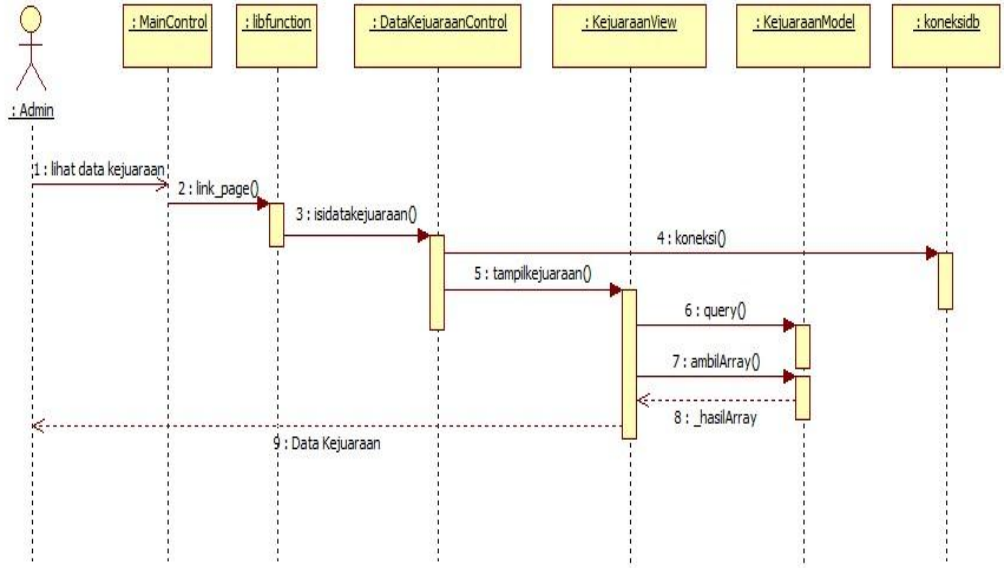
Gambar 3. 53 *Sequence* Diagram Ubah Klub

m. *Sequence* Diagram Hapus Klub



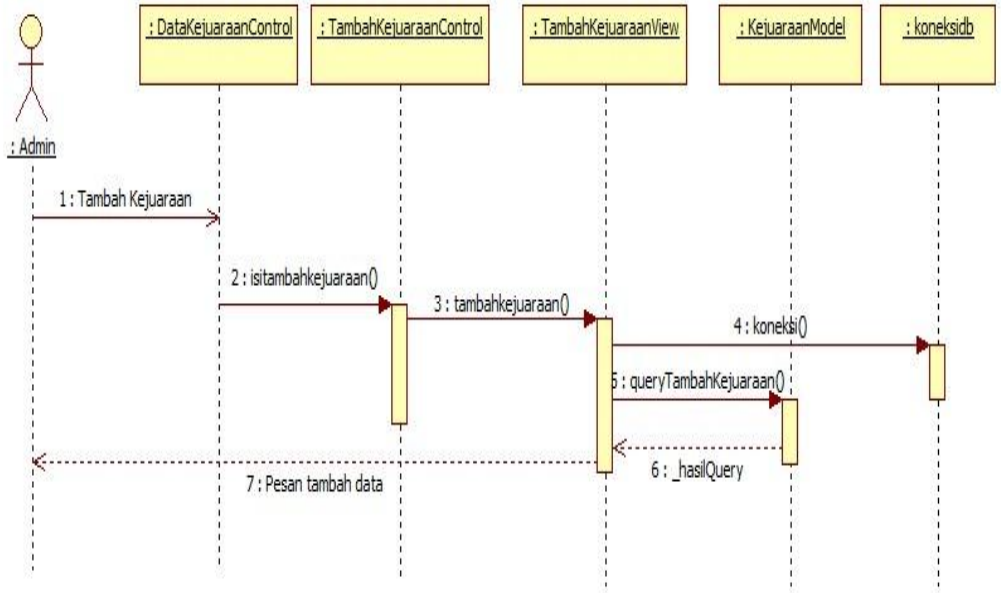
Gambar 3. 54 *Sequence* Diagram Hapus Klub

n. *Sequence* Diagram Pengolahan Data Kejuaraan



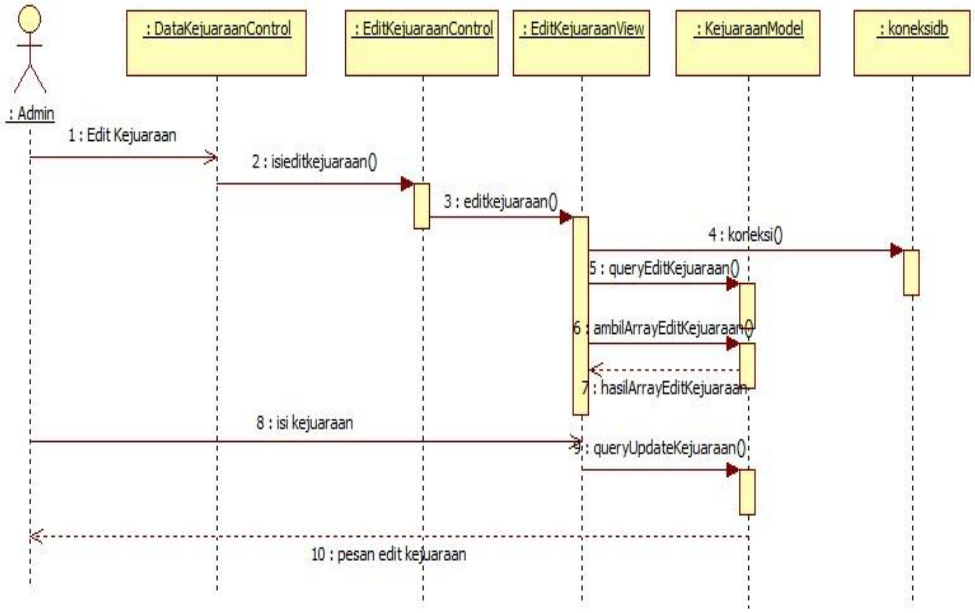
Gambar 3. 55 *Sequence* Diagram Pengolahan Data Kejuaraan

o. *Sequence* Diagram Tambah Kejuaraan



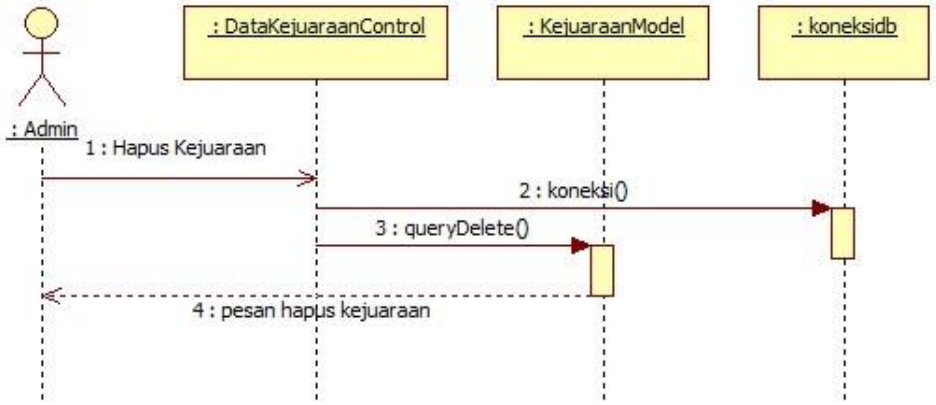
Gambar 3. 56 *Sequence* Diagram Tambah Kejuaraan

p. *Sequence* Diagram Ubah Kejuaraan



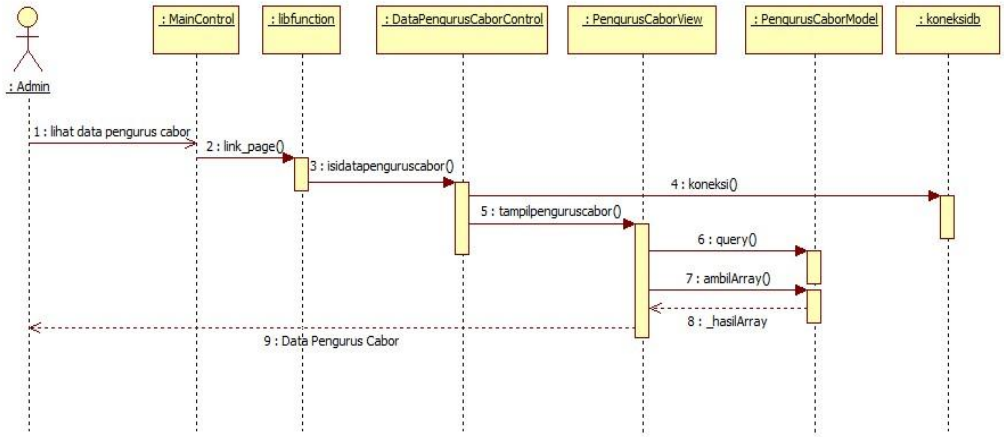
Gambar 3. 57 *Sequence* Diagram Ubah Kejuaraan

q. *Sequence* Diagram Hapus Kejuaraan



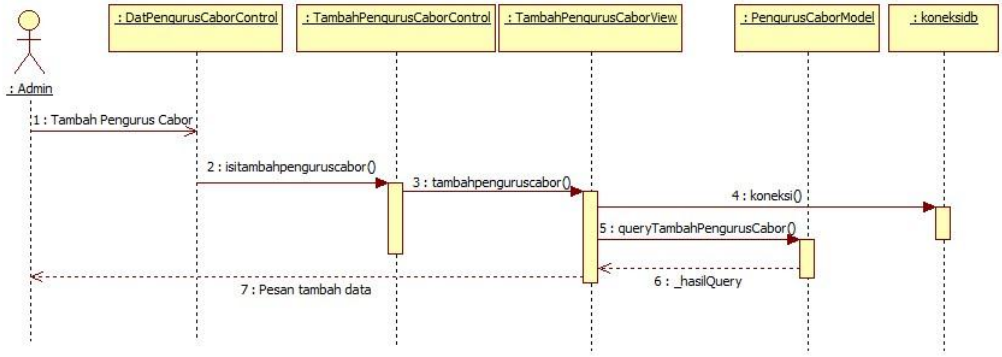
Gambar 3. 58 *Sequence* Diagram Hapus Kejuaraan

r. *Sequence* Diagram Pengolahan Data Pengurus Cabor



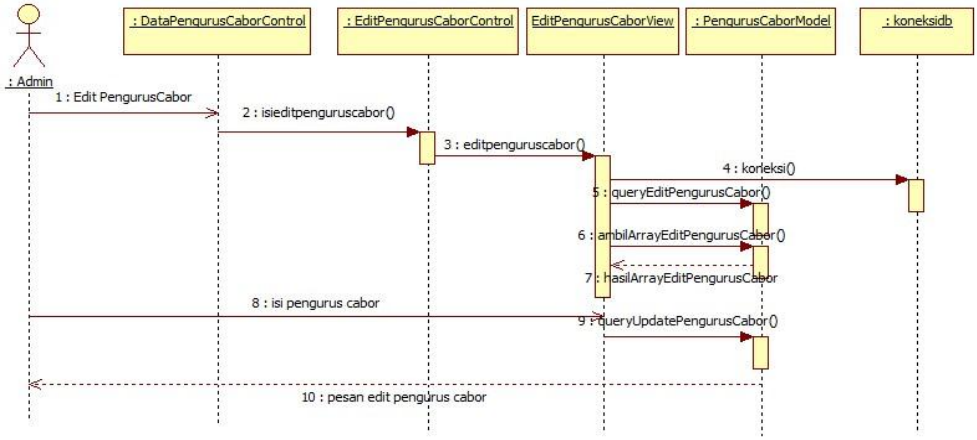
Gambar 3. 59 *Sequence* Diagram Pengolahan Data Pengurus Cabor

s. *Sequence* Diagram Tambah Pengurus cabor



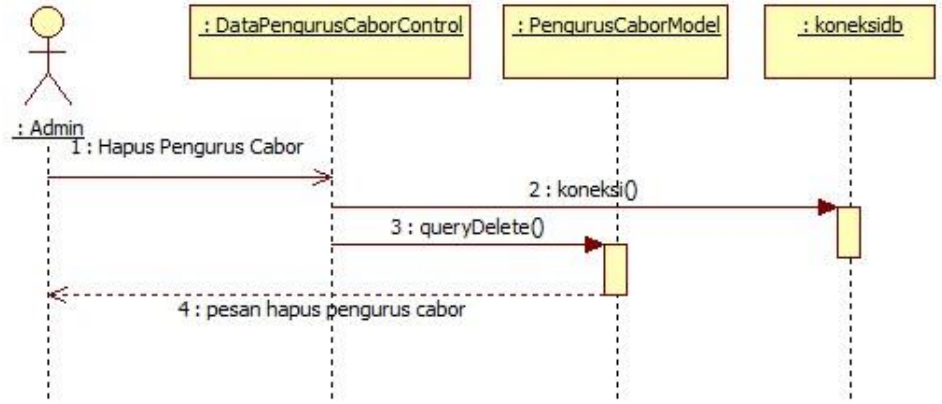
Gambar 3. 60 *Sequence* Diagram Tambah Pengurus Cabor

t. *Sequence* Diagram Ubah Pengurus Cabor



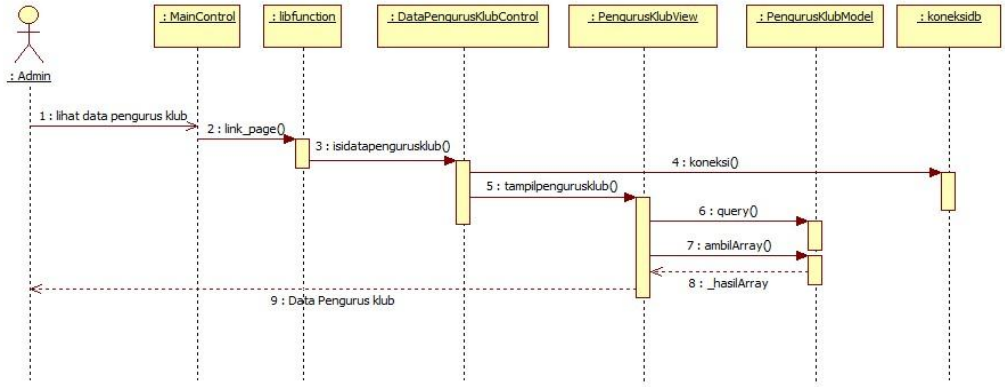
Gambar 3. 61 *Sequence* Diagram Ubah Pengurus Cabor

u. *Sequence* Diagram Hapus Pengurus Cabor



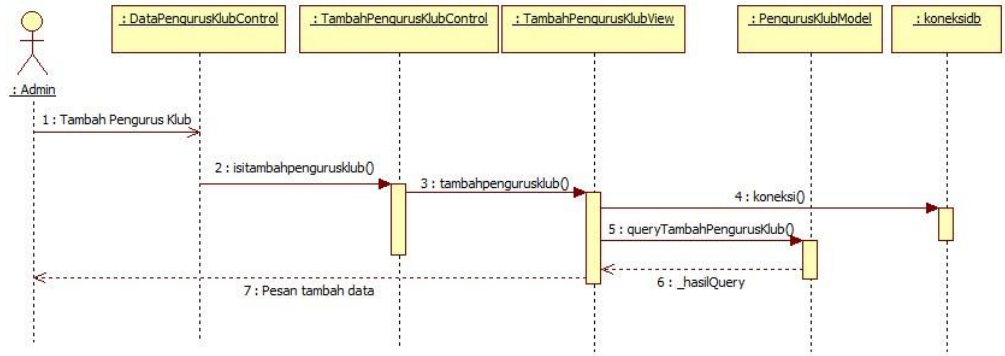
Gambar 3. 62 *Sequence* Diagram Hapus Pengurus Klub

v. *Sequence* Diagram Pengorlahan Data Pengurus klub



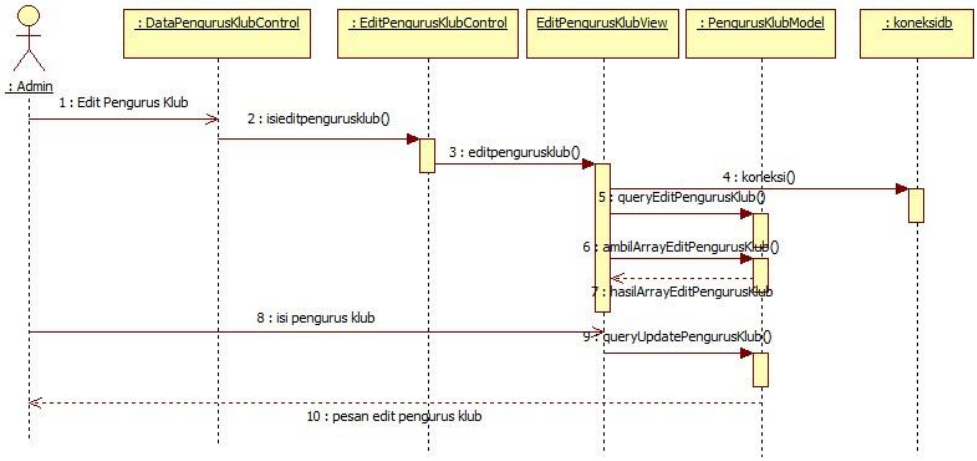
Gambar 3. 63 *Sequence* Diagram Pengolahan Data Pengurus Klub

w. *Sequence* Diagram Tambah Pengurus Klub



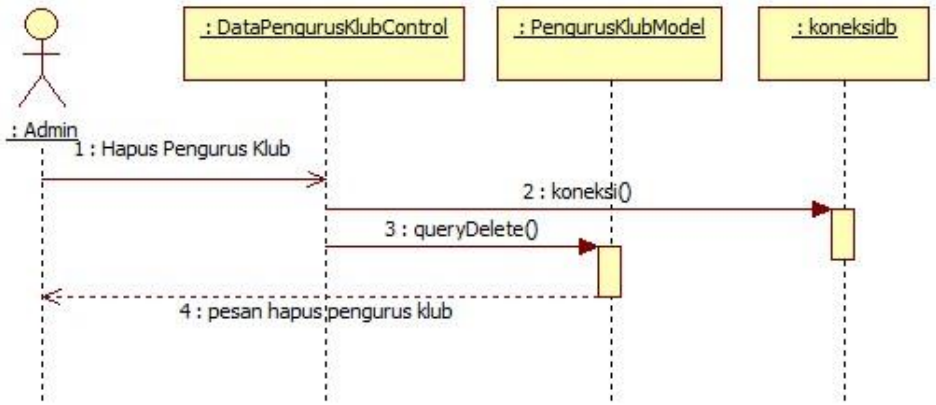
Gambar 3. 64 *Sequence* Diagram Tambah Pengurus Klub

x. *Sequence* Diagram Ubah Pengurus Klub



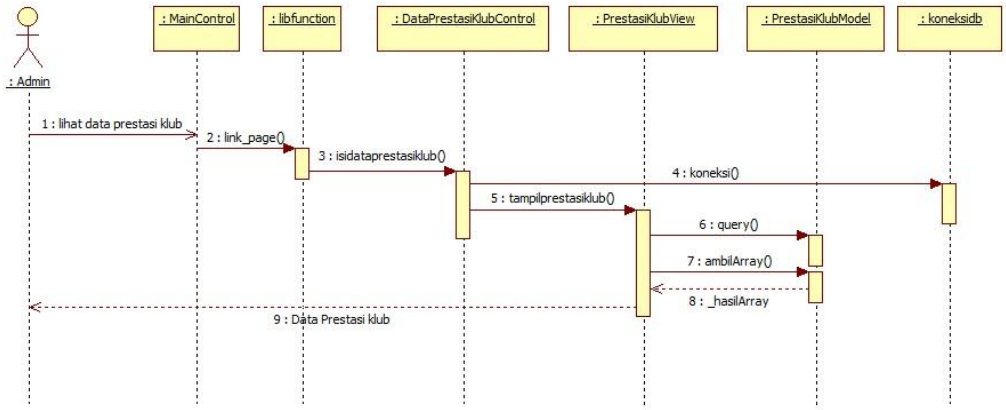
Gambar 3. 65 *Sequence* Diagram Ubah Pengurus Klub

y. *Sequence* Diagram Hapus Pengurus Klub



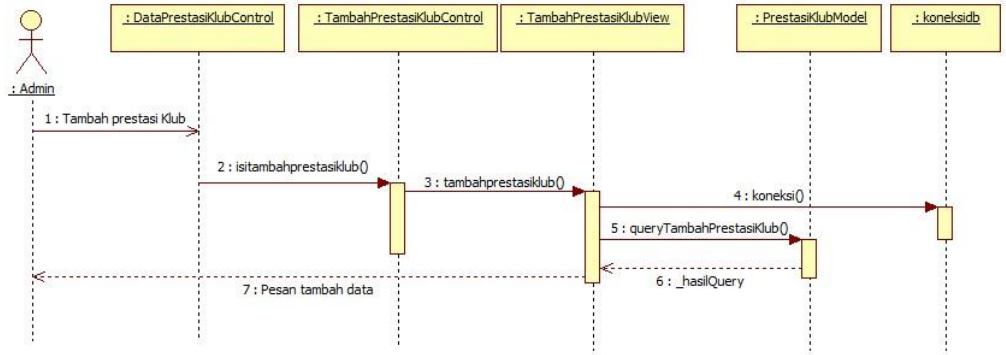
Gambar 3. 66 *Sequence* Diagram Hapus Pengurus Klub

- *Sequence* Diagram Pengolahan Data Prestasi Klub



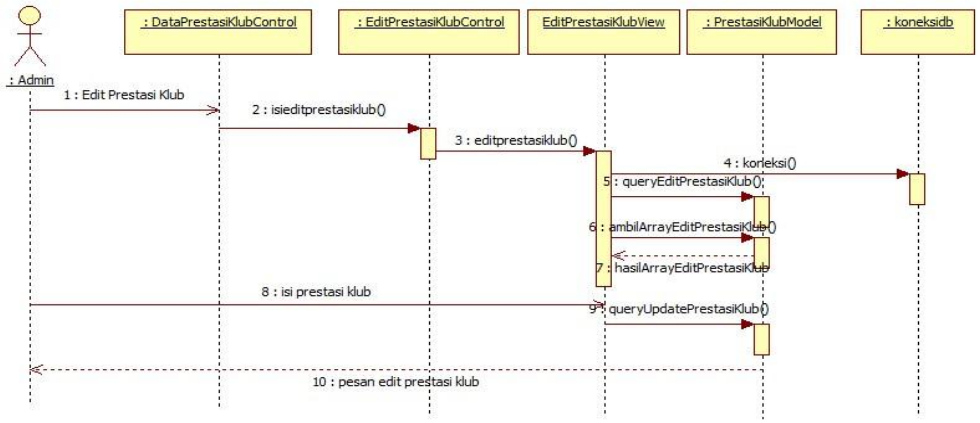
Gambar 3. 67 *Sequence* Diagram Pengolahan Data Prestasi Klub

- *Sequence* Diagram Tambah Prestasi Klub



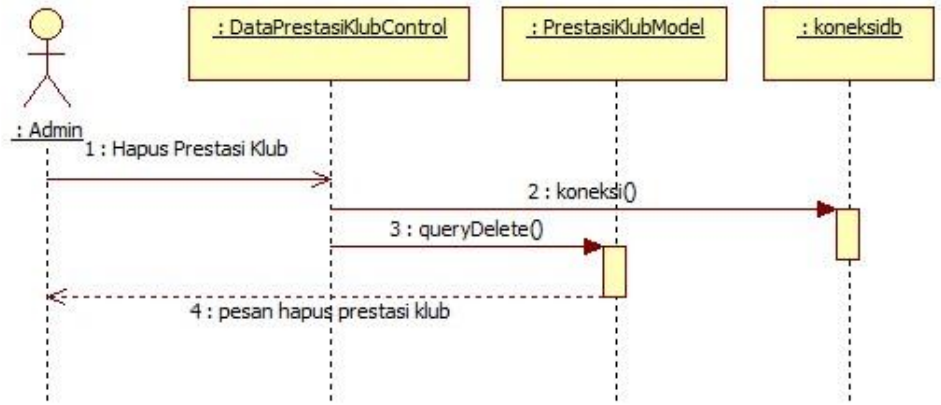
Gambar 3. 68 *Sequence* Diagram Tambah Prestasi Klub

- *Sequence* Diagram Ubah prestasi Klub



Gambar 3. 69 *Sequence* Diagram Ubah Prestasi Klub

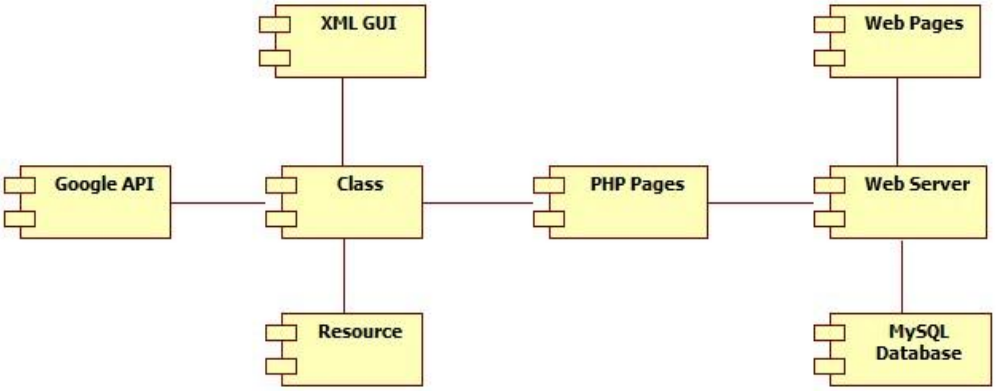
- *Sequence* Diagram Hapus Prestasi Klub



Gambar 3. 70 *Sequence* Diagram Hapus Prestasi Klub

e. *Component* Diagram

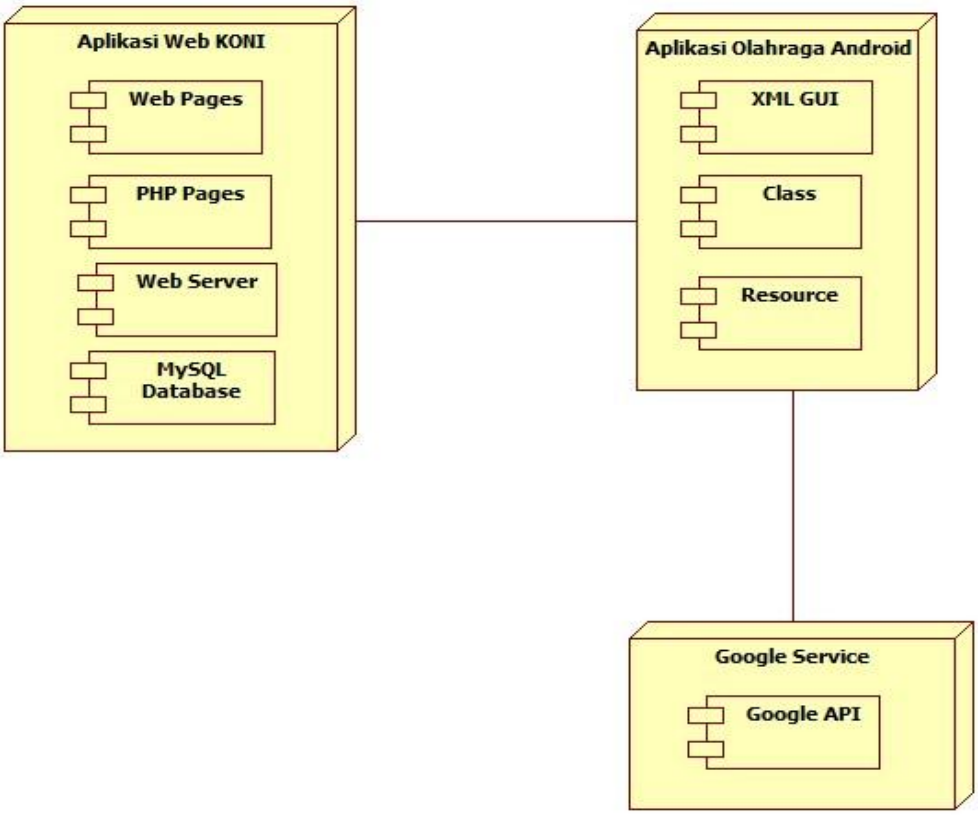
Pada tahap ini, *Component Diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak termasuk ketergantungan (*dependency*). Komponen piranti lunak yang yang dimaksud adalah modul yang berisi *source code* atau *binary code*, yang ada *library* atau *excutetable* yang muncul pada *compile time*, *link time* ataupunpada *runtime*. *Component Diagram* ditunjukan pada gambar dibawah ini.



Gambar 3. 71 *Component* Diagram

f. *Deployment* Diagram

Diagram Deployment menggambarkan rinci bagaimana komponen dideploy dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin *Server* atau perangkat keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *Server* dan hal-hal yang lain bersifat fisikal. Gambaran arsitektur fisik dari perangkat keras dan perangkat lunak dari aplikasi olahraga seperti ditunjuk dalam gambar dibawah ini :



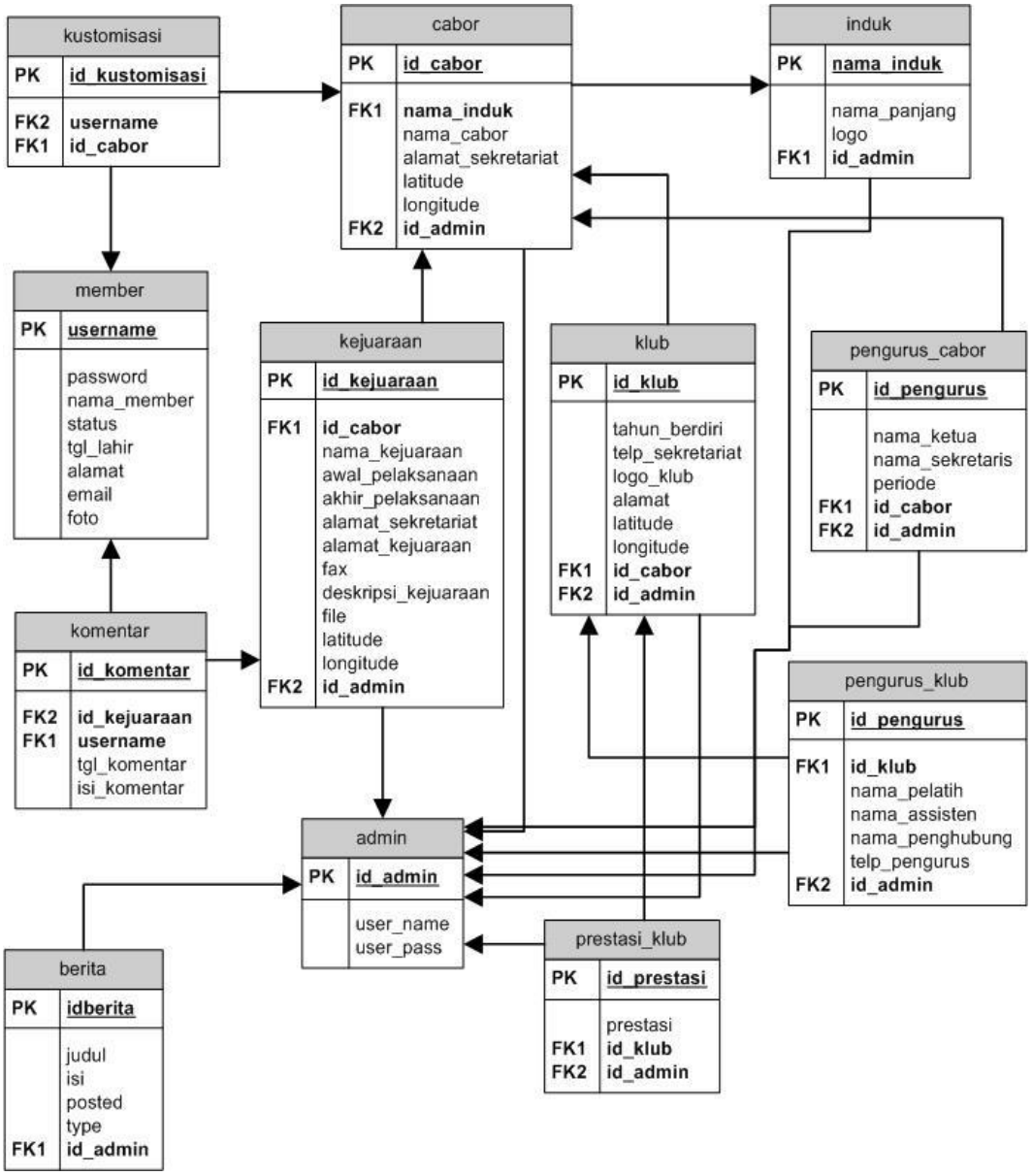
Gambar 3. 72 *Deployment* Diagram

3.2 Perancangan Sistem

Perancangan sistem bertujuan untuk menspesifikasikan aspek-aspek teknik yang menjadi solusi dalam perencanaan. Pada tahap ini perancangan akan didefinisikan secara detail untuk mengatasi masalah-masalah yang lebih teknis, berkaitan dengan kegiatan implementasi seperti perancangan database, dan perancangan antarmuka.

3.2.1 Skema Relasi

Relasi antar tabel merupakan gabungan antar *file* yang mempunyai kunci utama yang sama, sehingga *file-file* tersebut menjadi satu kesatuan yang dihubungkan oleh *field* kunci (*Primary Key*). Pada proses ini elemen-elemen data dikelompokkan menjadi satu *file database* beserta *entitas* dan hubungannya. Skema relasi ini dapat dilihat pada gambar 3.54.



Gambar 3. 73 Skema Relasi

Struktur Tabel

Perancangan struktur tabel adalah perancangan tabel-tabel yang akan digunakan pada database. Tabel-tabel yang terdapat dalam database yang digunakan dalam aplikasi mobile android ini adalah :

Tabel 3. 52 Struktur Tabel admin

Nama Field	Tipe	Panjang	Kunci
id_admin	Int	11	*Primary key*
user_name	Varchar	50	
user_pass	Varchar	200	

Tabel 3. 53 Struktur Tabel cabor

Nama Field	Tipe	Panjang	Kunci
id_cabor	Int	11	*Primary key*
nama_induk	varchar	50	*Foreign Key reference* induk (nama_induk)
nama_cabor	varchar	50	*Index*
alamat_sekretariat	varchar	50	
latitude	double		
longitude	double		
id_admin	int	11	*Foreign Key reference* admin(id_admin)

Tabel 3. 54 Struktur Tabel induk

Nama Field	Tipe	Panjang	Kunci
nama_induk	varchar	50	*Primary key*
nama_panjang	varchar	100	
logo	Varchar	50	
id_admin	int	11	*Foreign Key reference* admin(id_admin)

Tabel 3. 55 Struktur Tabel kejuaraan

Nama Field	Tipe	Panjang	Kunci
id_kejuaraan	Int	11	*Primary key*
id_cabor	Int	11	*Foreign Key reference* cabor(id_cabor)
nama_kejuaraan	Varchar	200	*Index*

awal_pelaksanaan	Date		
akhir_pelaksanaan	Date		
alamat_sekretariat	Text		
fax	varchar	10	
deskripsi_kejuaraan	Text		
file	Varchar	50	
alamat_kejuaraan	Text		
latitude	Double		
longitude	Double		
id_admin	int	11	*Foreign Key reference* admin(id_admin)

Tabel 3. 56 Struktur Tabel klub

Nama Field	Tipe	Panjang	Kunci
id_klub	Int	11	*Primary key*
id_cabor	Int	11	*Foreign Key reference* cabor(id_cabor)
nama_klub	Varchar	50	*Index*
tahun_berdiri	Date		
telp	Varchar	15	
logo_klub	Varchar	50	
alamat	Text		
latitude	Double		
longitude	Double		
id_admin	int	11	*Foreign Key reference* admin(id_admin)

Tabel 3. 57 Struktur Tabel kustomisasi

Nama Field	Tipe	Panjang	Kunci
id_kustomisasi	int	11	*Primary Key*
username	Varchar	40	*Foreign key reference* member (username)
id_cabor	Int	11	*Foreign Key reference* cabor(id_cabor)

Tabel 3. 58 Struktur Tabel member

Nama Field	Tipe	Panjang	Kunci
username	Varchar	40	*Primary key*
password	Varchar	400	
nama_member	Varchar	40	
status	Varchar	20	
tgl_lahir	Date		
alamat	Text		
email	Varchar	50	*Unique*
foto	Varchar	50	

Tabel 3. 59 Struktur Tabel komentar

Nama Field	Tipe	Panjang	Kunci
id_komentar	Int	11	*Primary key*
id_kejuaraan	Int	11	*Foreign Key reference* kejuaraan(id_kejuaraan)
username	Varchar	40	*Foreign key reference* member (username)
tgl_komentar	Date	20	
Isi_komentar	Text		

Tabel 3. 60 Struktur Tabel pengurus_cabor

Nama Field	Tipe	Panjang	Kunci
id_pengurus	Int	11	*Primary key*
id_cabor	Int	11	*Foreign Key reference* cabor(id_cabor)
nama_ketua	Varchar	50	
nama_sekretaris	Varchar	50	
periode	Varchar	50	
id_admin	int	11	*Foreign Key reference* admin(id_admin)

Tabel 3. 61 Struktur Tabel pengurus_klub

Nama Field	Tipe	Panjang	Kunci
id_pengurus	Int	11	*Primary key*
id_klub	Int	11	*Foreign Key reference* klub(id_klub)
nama_pelatih	Varchar	50	
nama_assisten	Varchar	50	
nama_penghubung	Varchar	50	
Telp_pengurus	Varchar	15	

id_admin	int	11	*Foreign Key reference* admin(id_admin)

Tabel 3. 62 Struktur Tabel prestasi_klub

Nama Field	Tipe	Panjang	Kunci
id_prestasi	Int	11	*Primary key*
id_klub	Int	11	*Foreign Key reference* klub(id_klub)
prestasi	Text		
id_admin	int	11	*Foreign Key reference* admin(id_admin)

Tabel 3. 63 Struktur Tabel berita

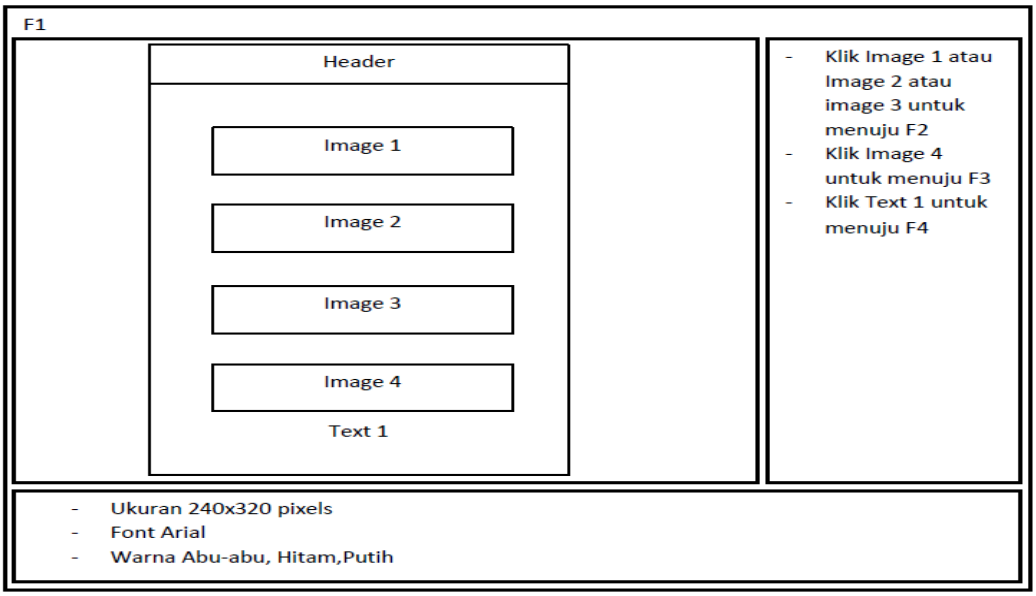
Nama Field	Tipe	Panjang	Kunci
id_berita	Int	11	*Primary key*
judul	varchar	50	
isi	Text		
posted	Date		
type	Enum	(‘Berita’,’informasi’)	
id_admin	int	11	*Foreign Key reference* admin(id_admin)

3.2.2 Perancangan Grafis dan Antar Muka

Tahap perancangan desain bertujuan untuk mencari bentuk yang optimal dari aplikasi yang akan dibangun dengan pertimbangan faktor-faktor permasalahan dan kebutuhan yang ada pada sistem seperti yang telah ditetapkan pada tahap analisis. Dalam tahap ini upaya yang dilakukan yaitu dengan cara mengkombinasikan penggunaan teknologi perangkat keras dan perangkat lunak yang tepat sehingga diperoleh yang optimal dan mudah diimplementasikan. Perancangan grafis dan antarmuka terdiri dari dua bagian utama yaitu disain grafis dan antarmuka *frontend application* dan *backend application*.

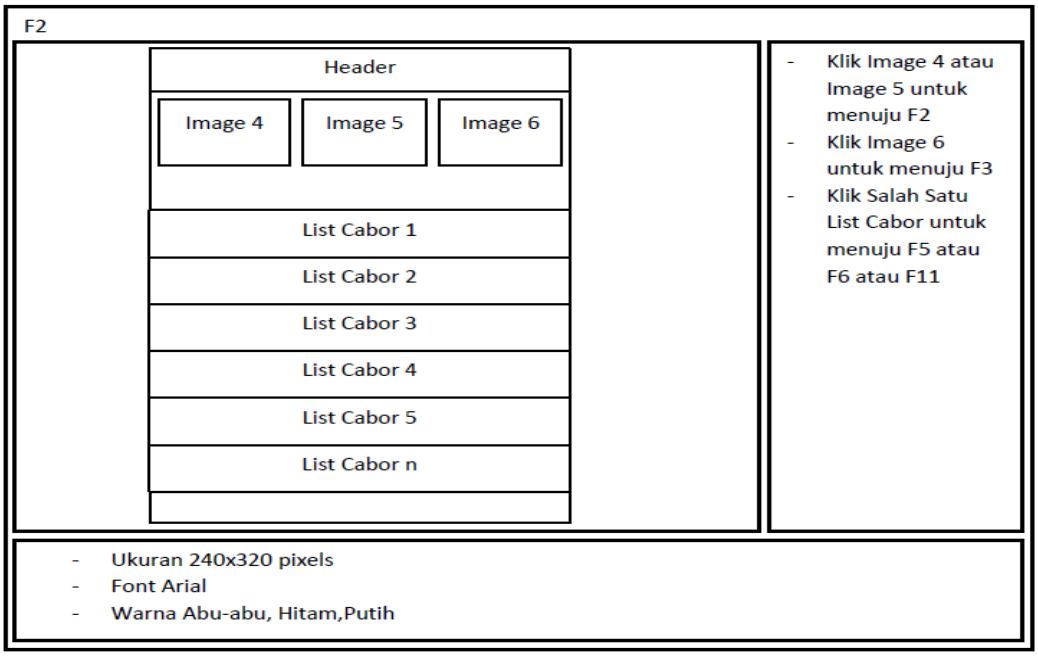
1. Perancangan Antarmuka Aplikasi *Frontend*

a. Perancangan Antarmuka Menu Utama



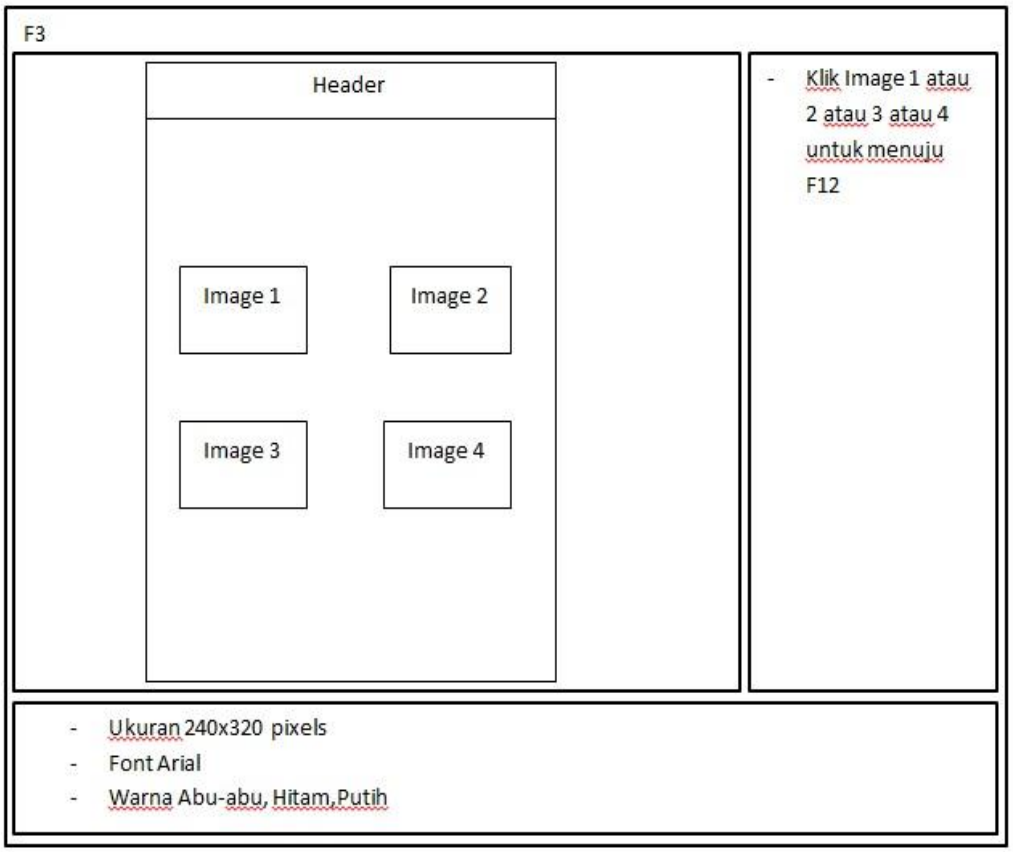
Gambar 3. 74 Antarmuka Menu Utama

b. Perancangan Antarmuka ListView Cabor



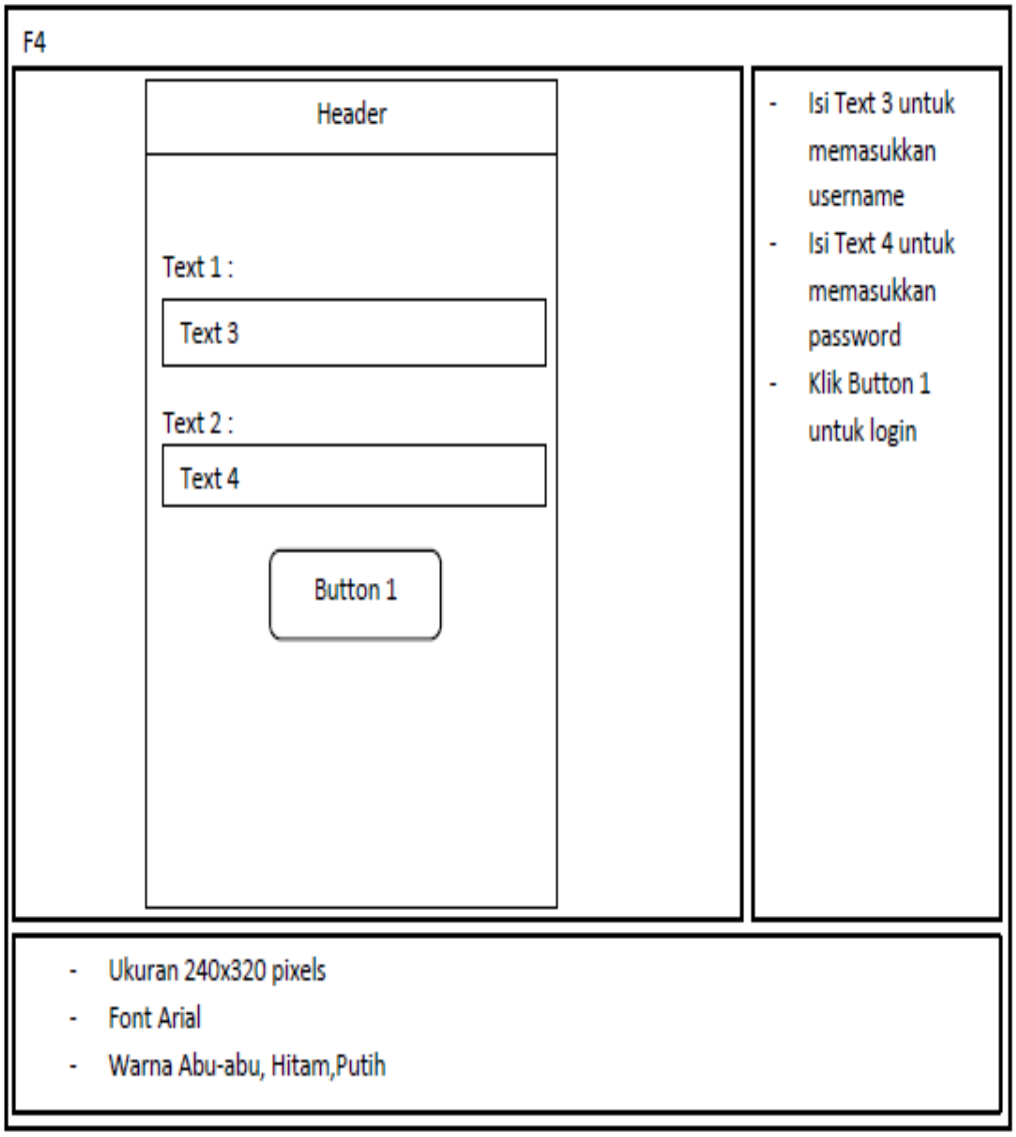
Gambar 3. 75 Antarmuka List View Cabo

c. Perancangan Antarmuka Menu Portal Berita



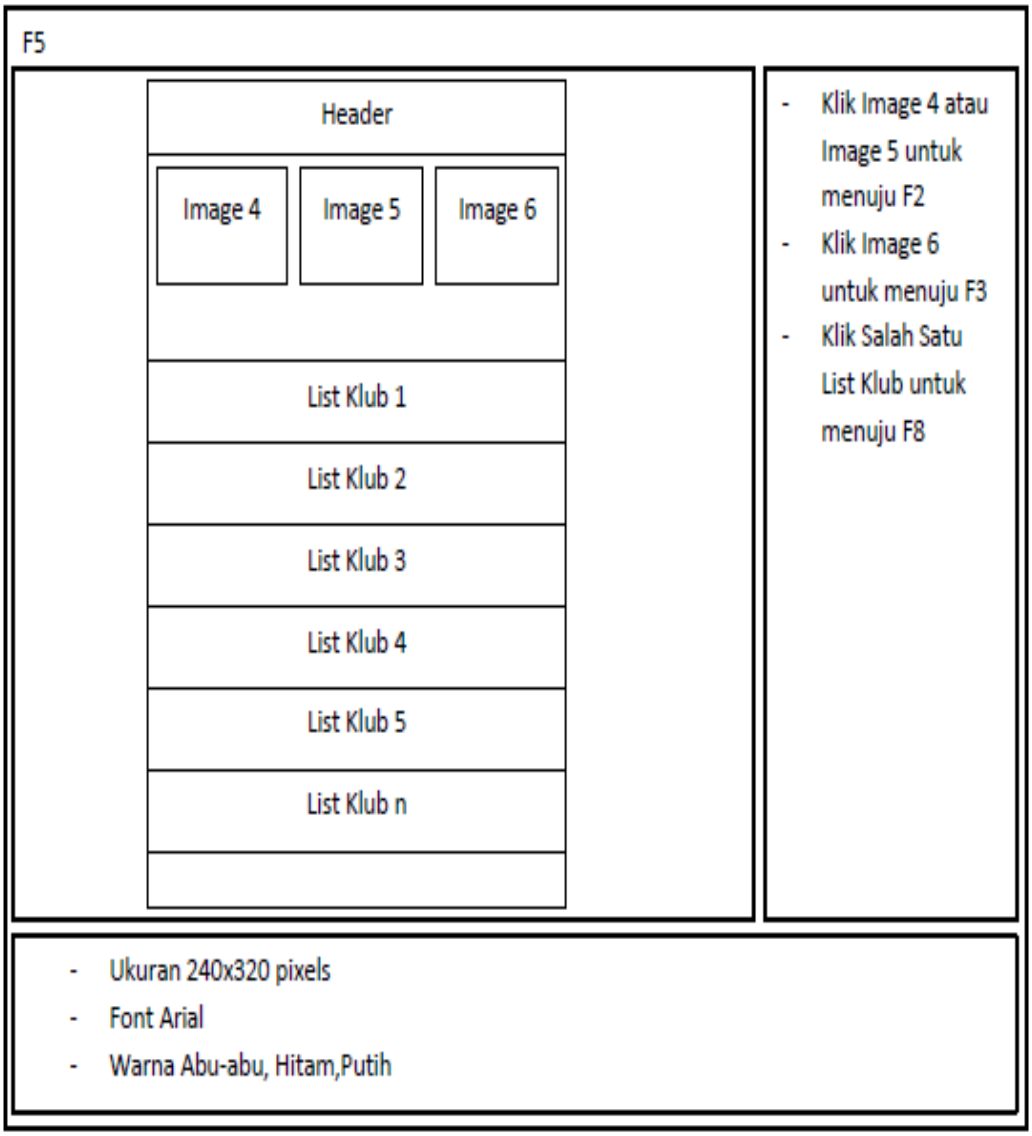
Gambar 3. 76 Antarmuka Menu Portal Berita

d. Perancangan Antarmuka Login Member



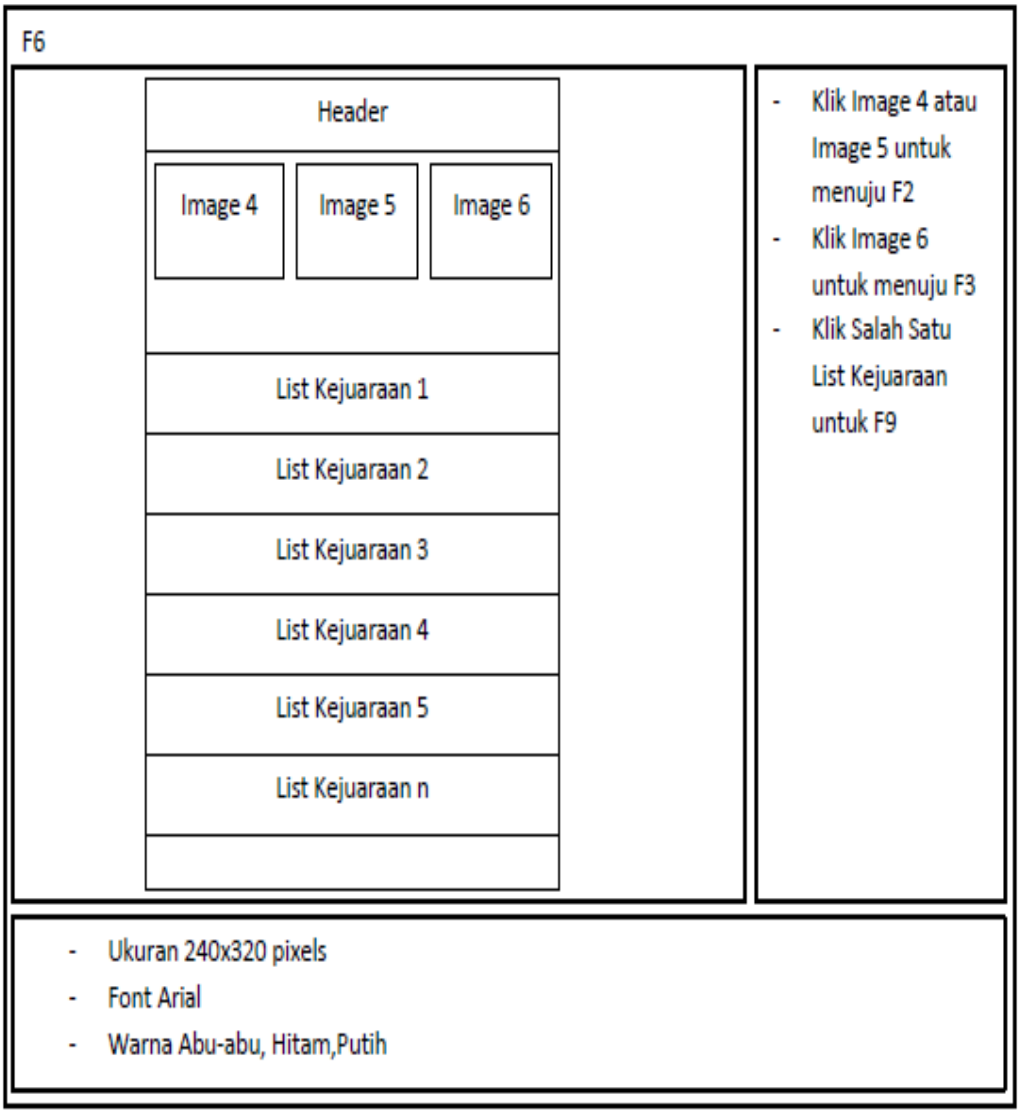
Gambar 3. 77 Antarmuka Login Member

e. Perancangan Antarmuka ListView Klub



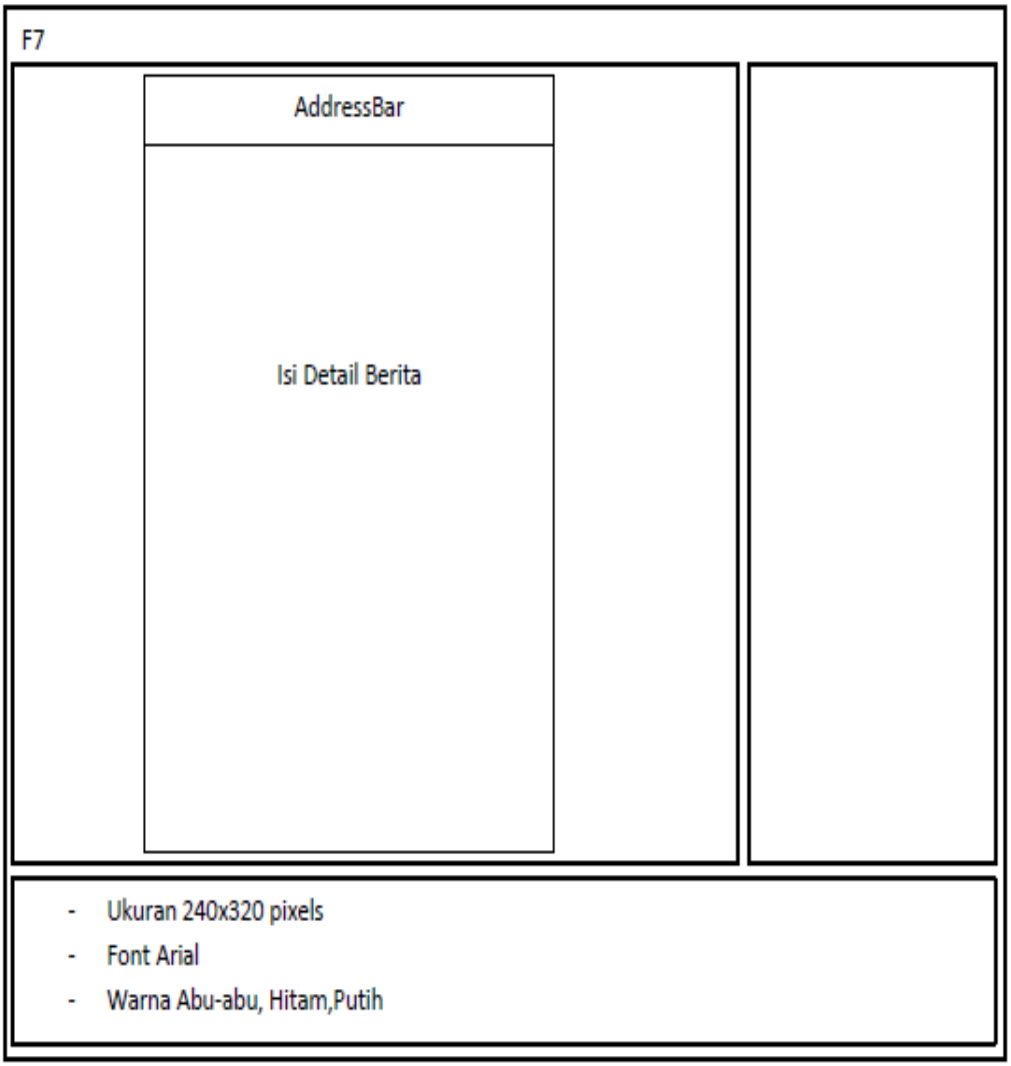
Gambar 3. 78 Antarmuka List View Klub

f. Perancangan Antarmuka ListView Kejuaraan



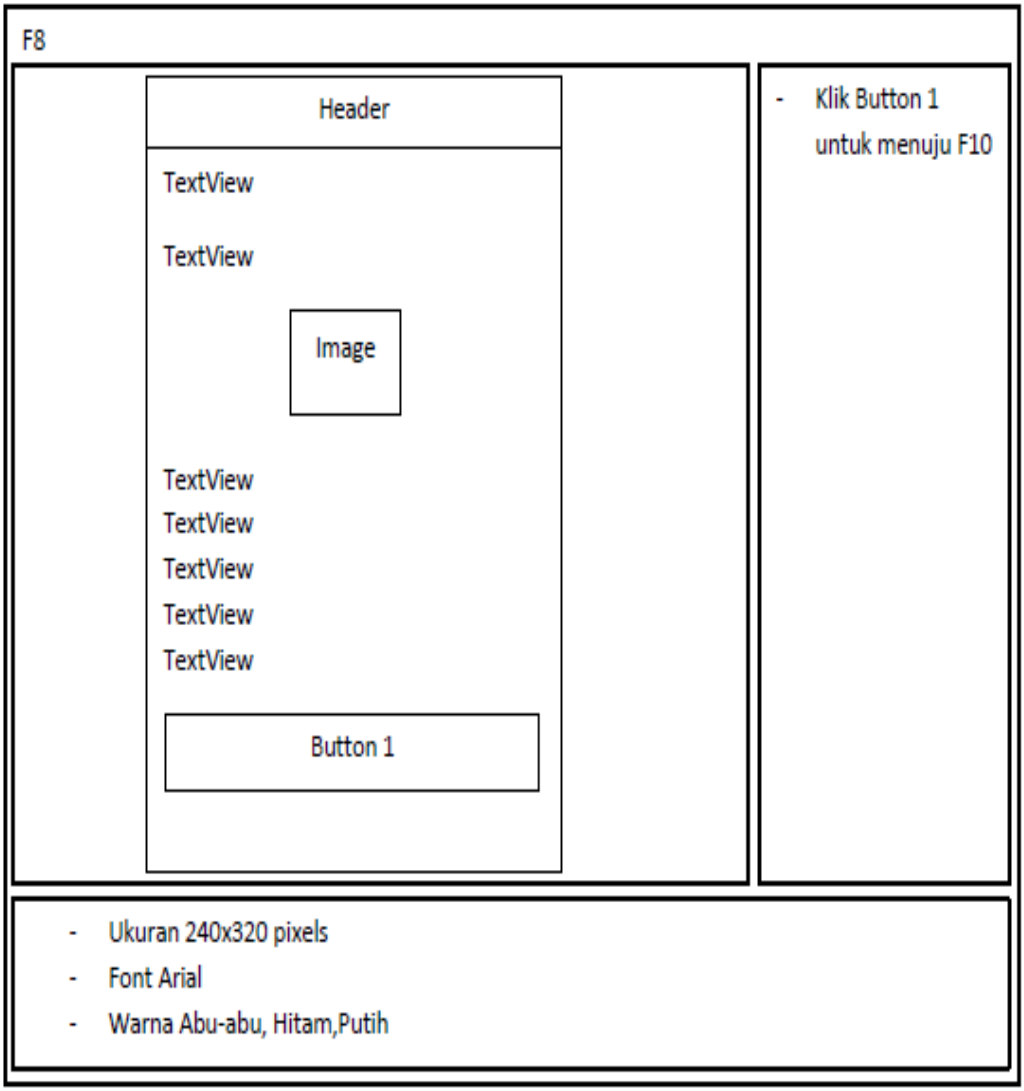
Gambar 3. 79 Antarmuka ListView Kejuaraan

g. Perancangan Antarmuka Detail Berita



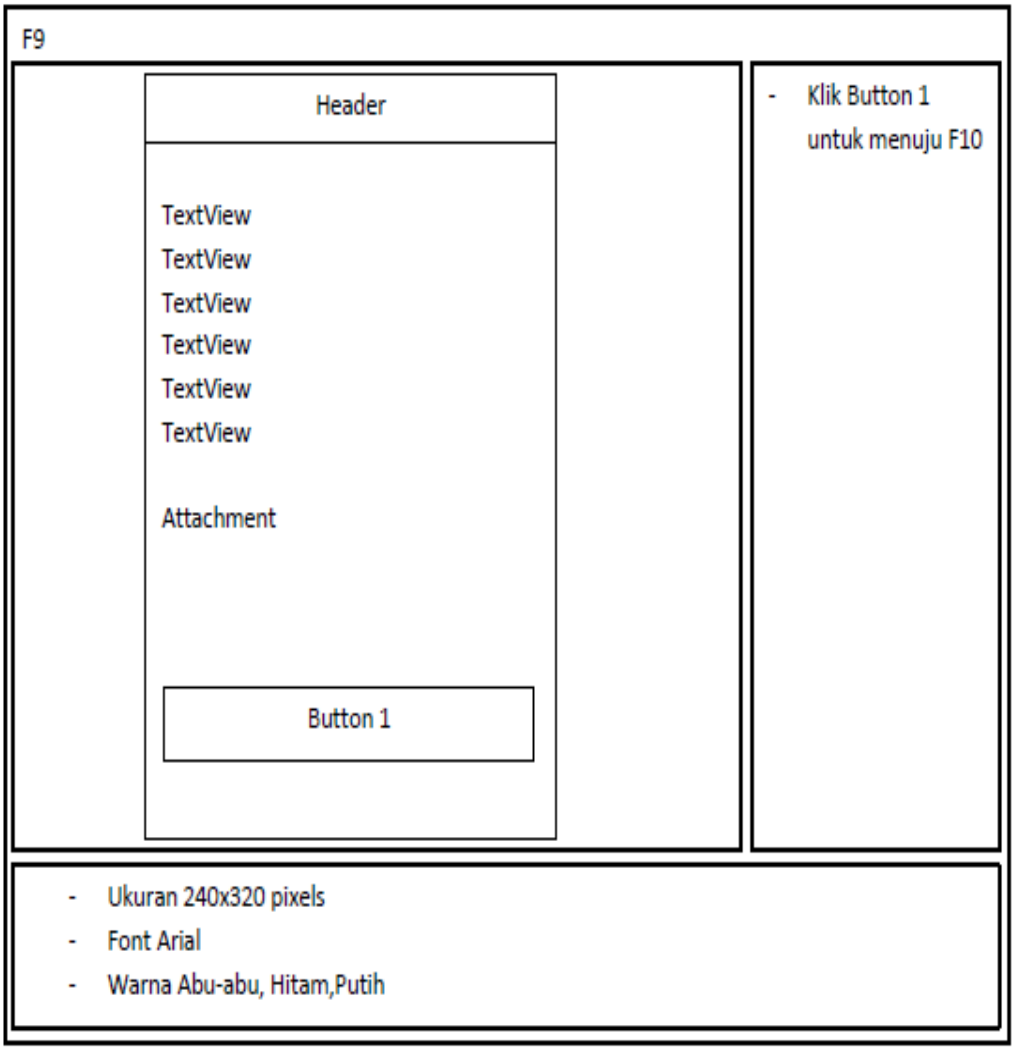
Gambar 3. 80 Antarmuka Detail Berita

h. Perancangan Antarmuka Detail Klub



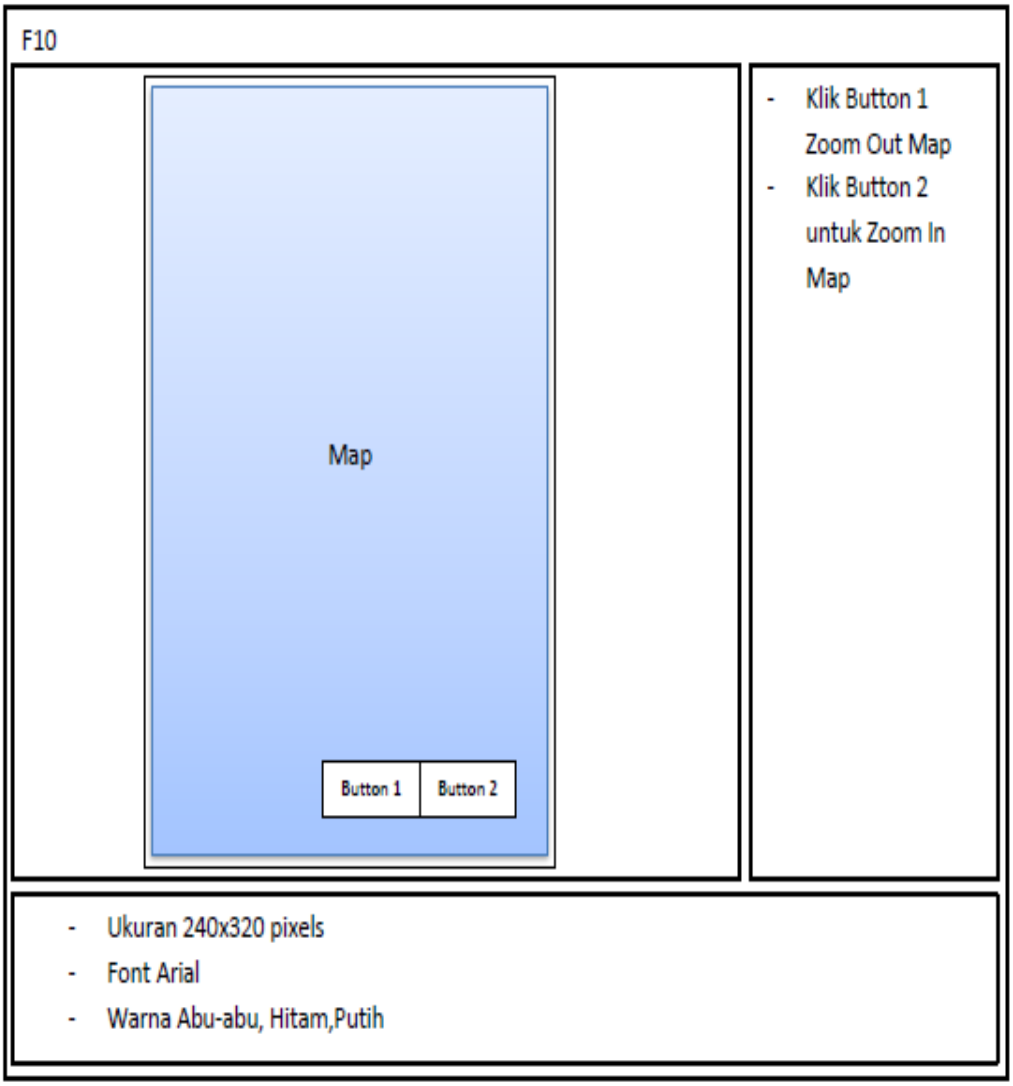
Gambar 3. 81 Antarmuka Detail klub

i. Perancangan Antarmuka Detail Kejuaraan



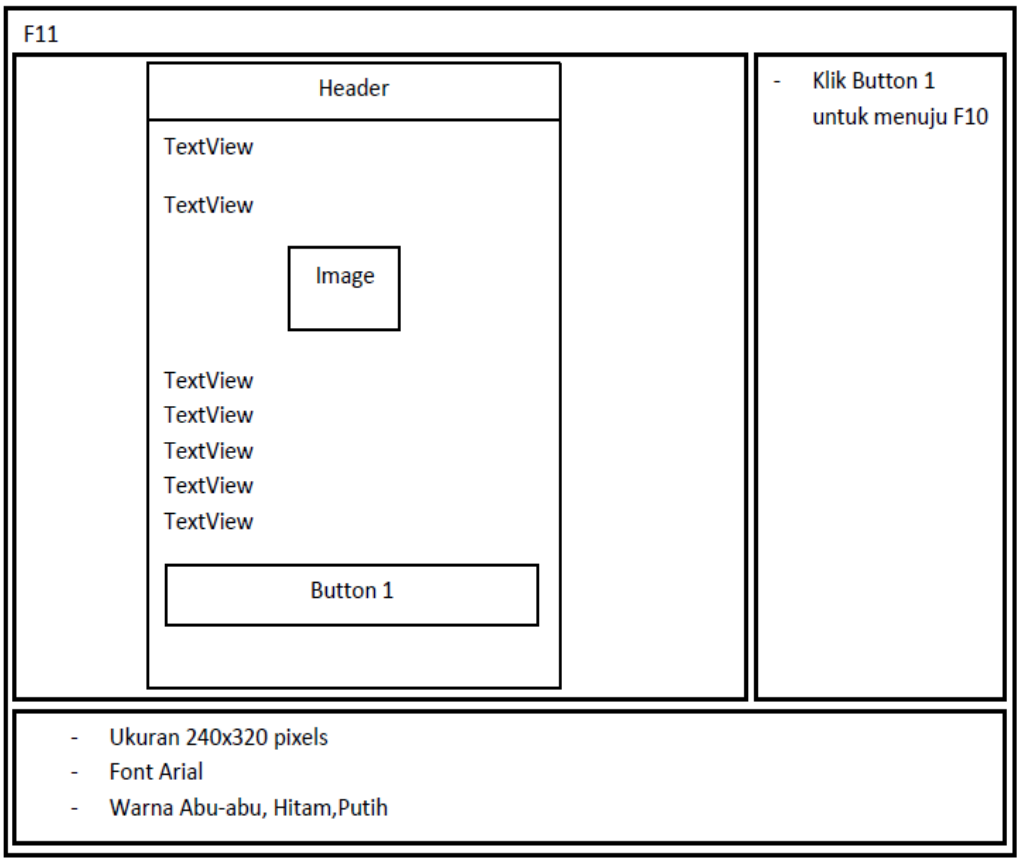
Gambar 3. 82 Antarmuka Detail Kejuaraan

j. Perancangan Antarmuka Peta Lokasi



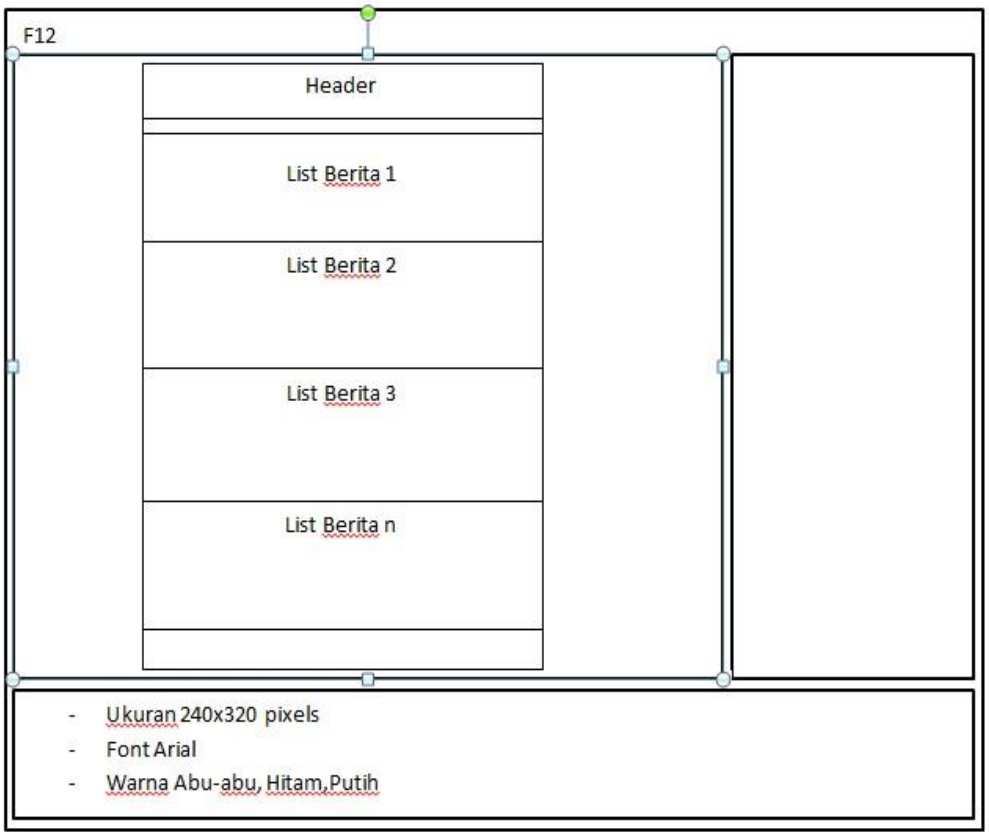
Gambar 3. 83 Antarmuka Peta Lokasi

k. Perancangan Antarmuka Detail Cabor



Gambar 3. 84 Antarmuka Detail Cabor

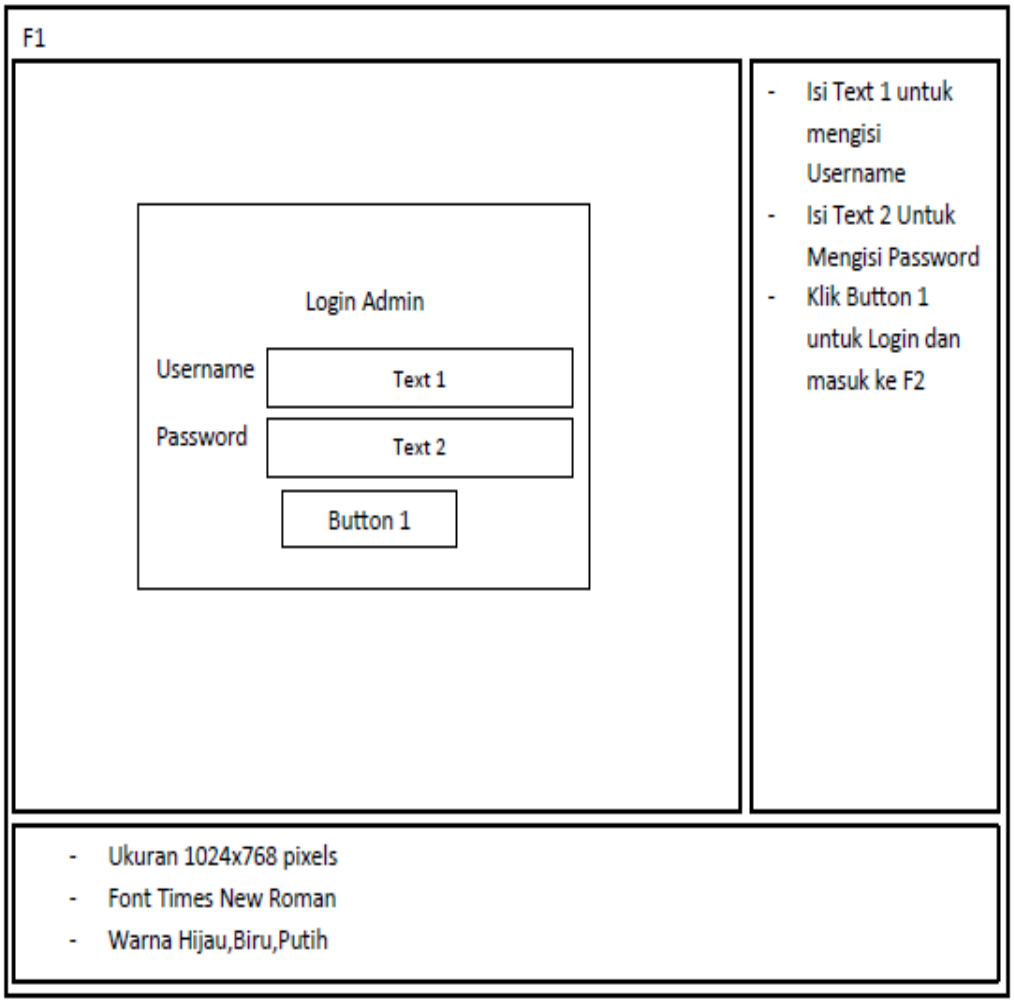
l. Perancangan Antarmuka ListView Berita



Gambar 3. 85 Antarmuka ListView Berita

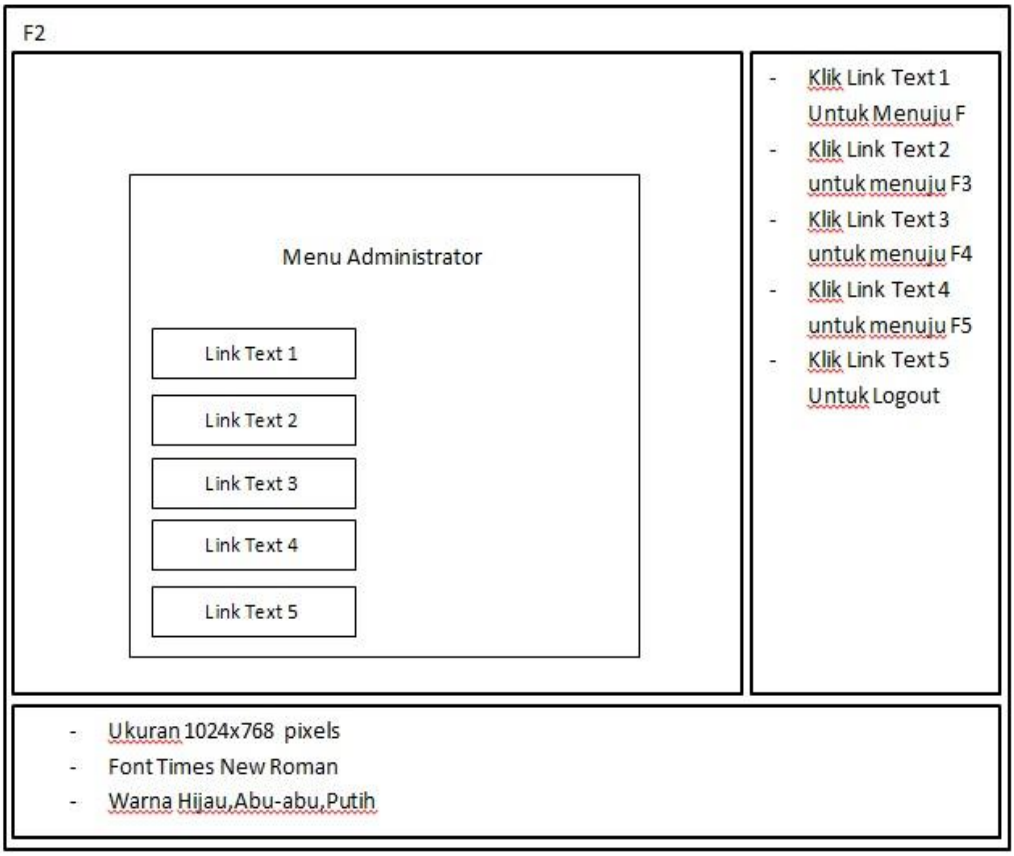
2. Perancangan Antarmuka Aplikasi *Backend*

a. Perancangan Antarmuka Login Admin



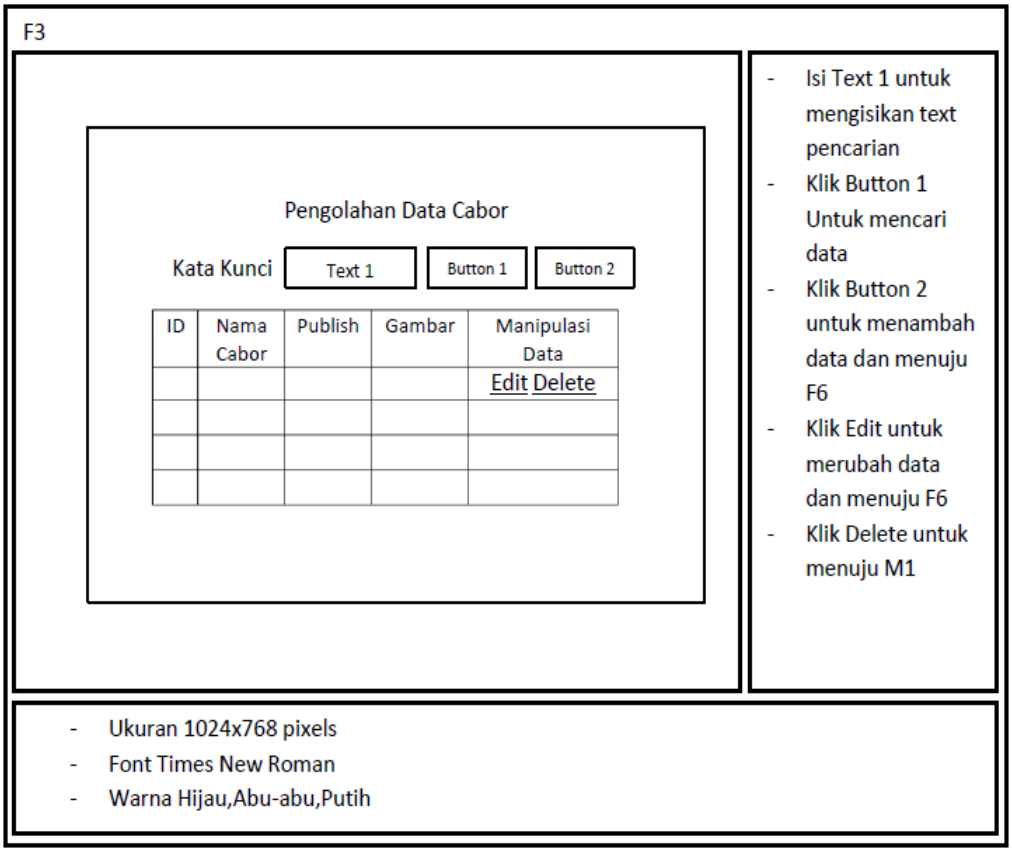
Gambar 3. 86 Antarmuka Login Admin

b. Perancangan Antarmuka Menu Utama Admin



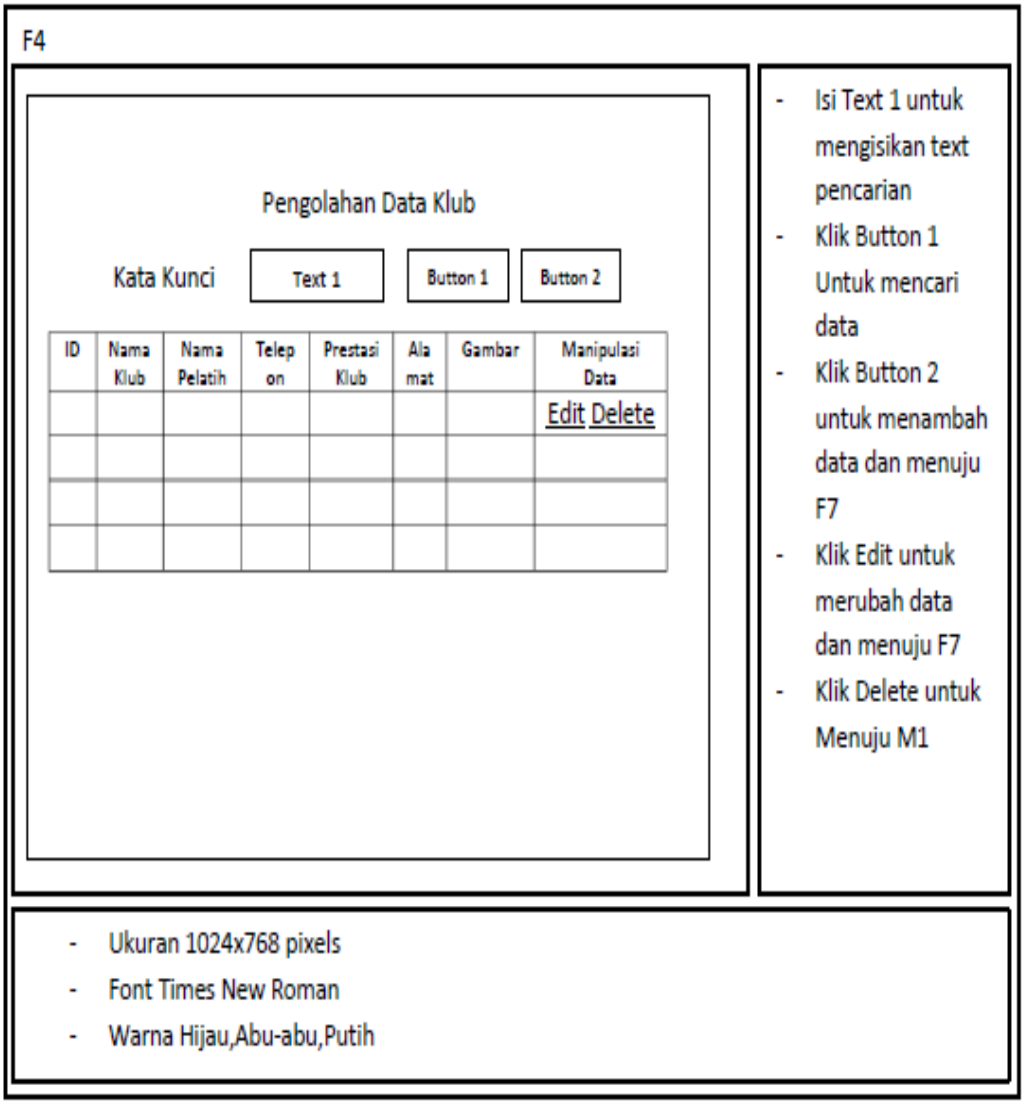
Gambar 3. 87 Antarmuka Menu Utama Admin

c. Perancangan Antarmuka Pengolahan Data Cabor



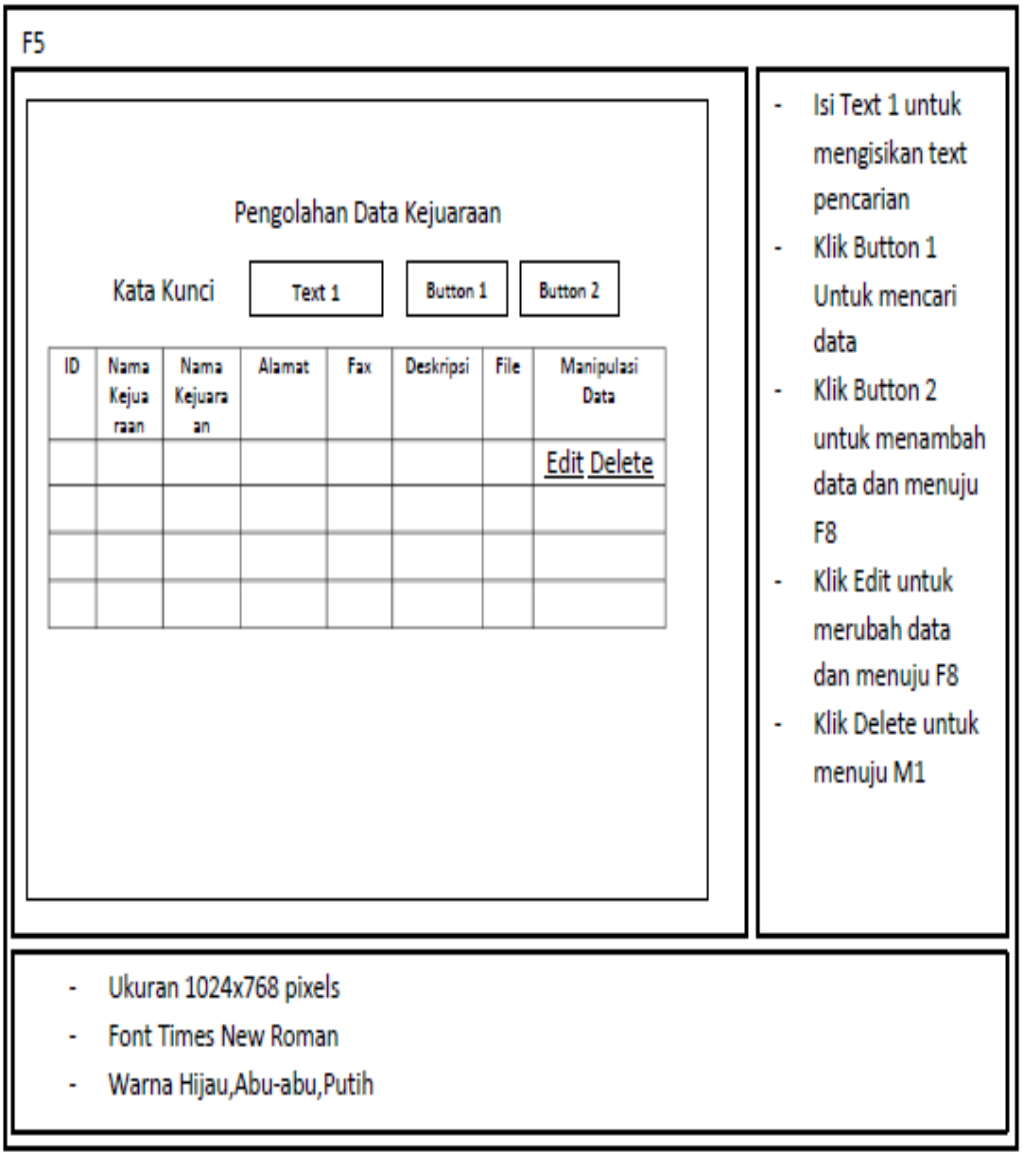
Gambar 3. 88 Antarmuka Pengolahan Data Cabor

d. Perancangan Antarmuka Pengolahan Data Klub



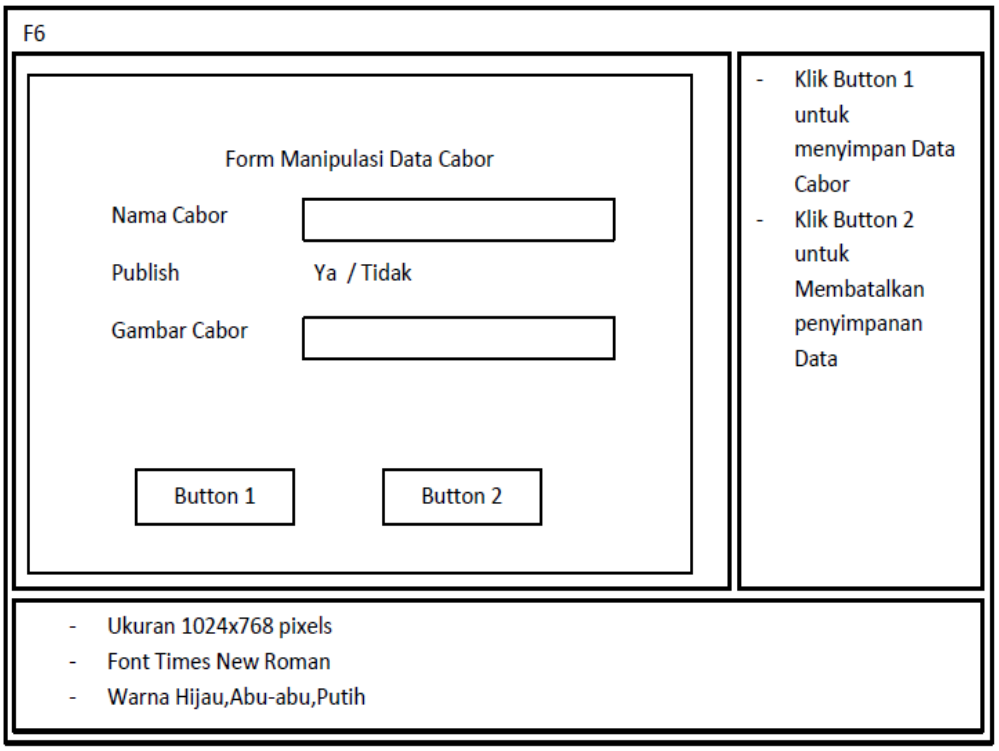
Gambar 3. 89 Antarmuka Pengolahan Data Klub

e. Perancangan Antarmuka Pengolahan Data Kejuaraan



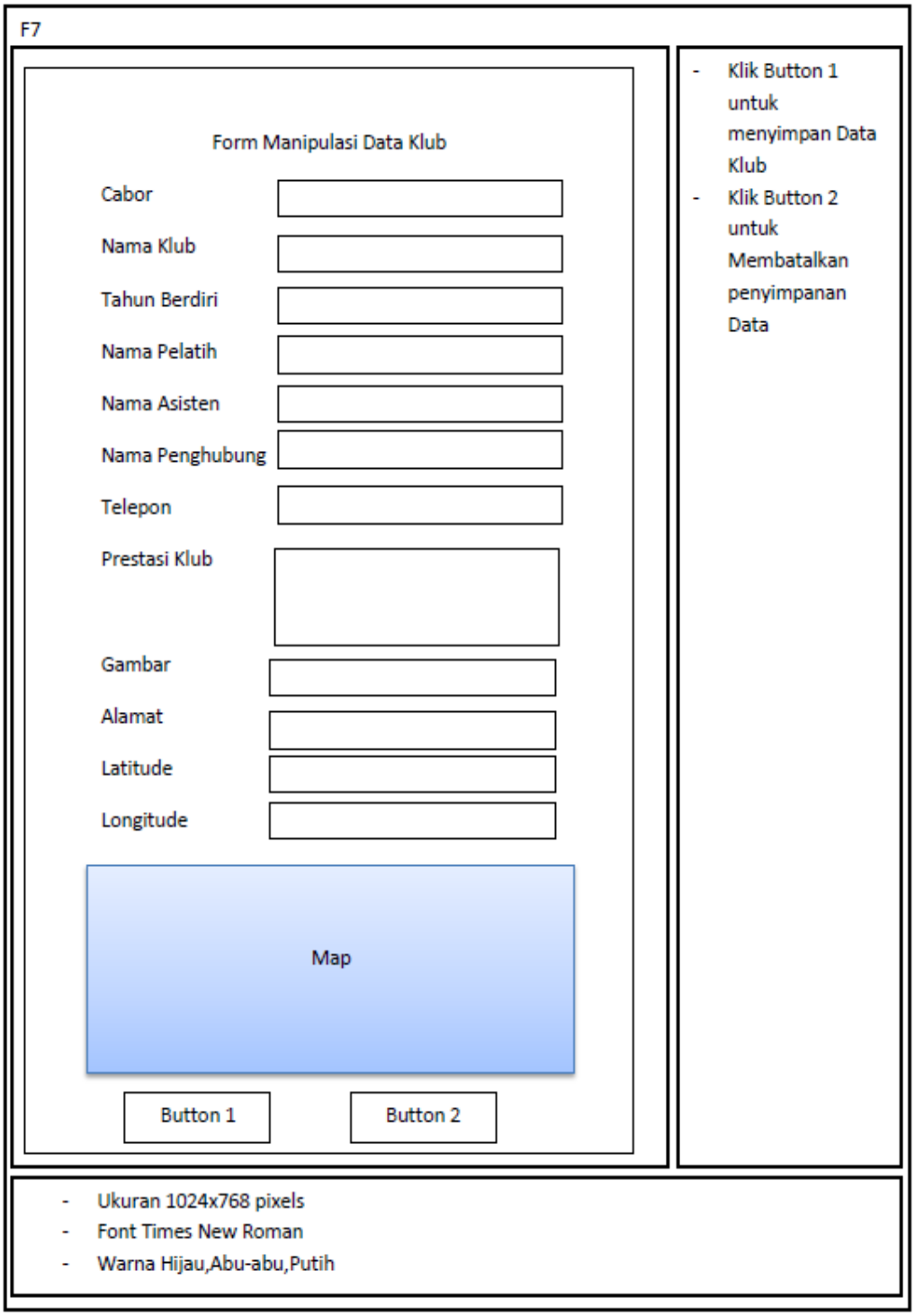
Gambar 3. 90 Antarmuka Pengolahan Data Kejuaraan

f. Perancangan Antarmuka Form Manipulasi Data Cabor



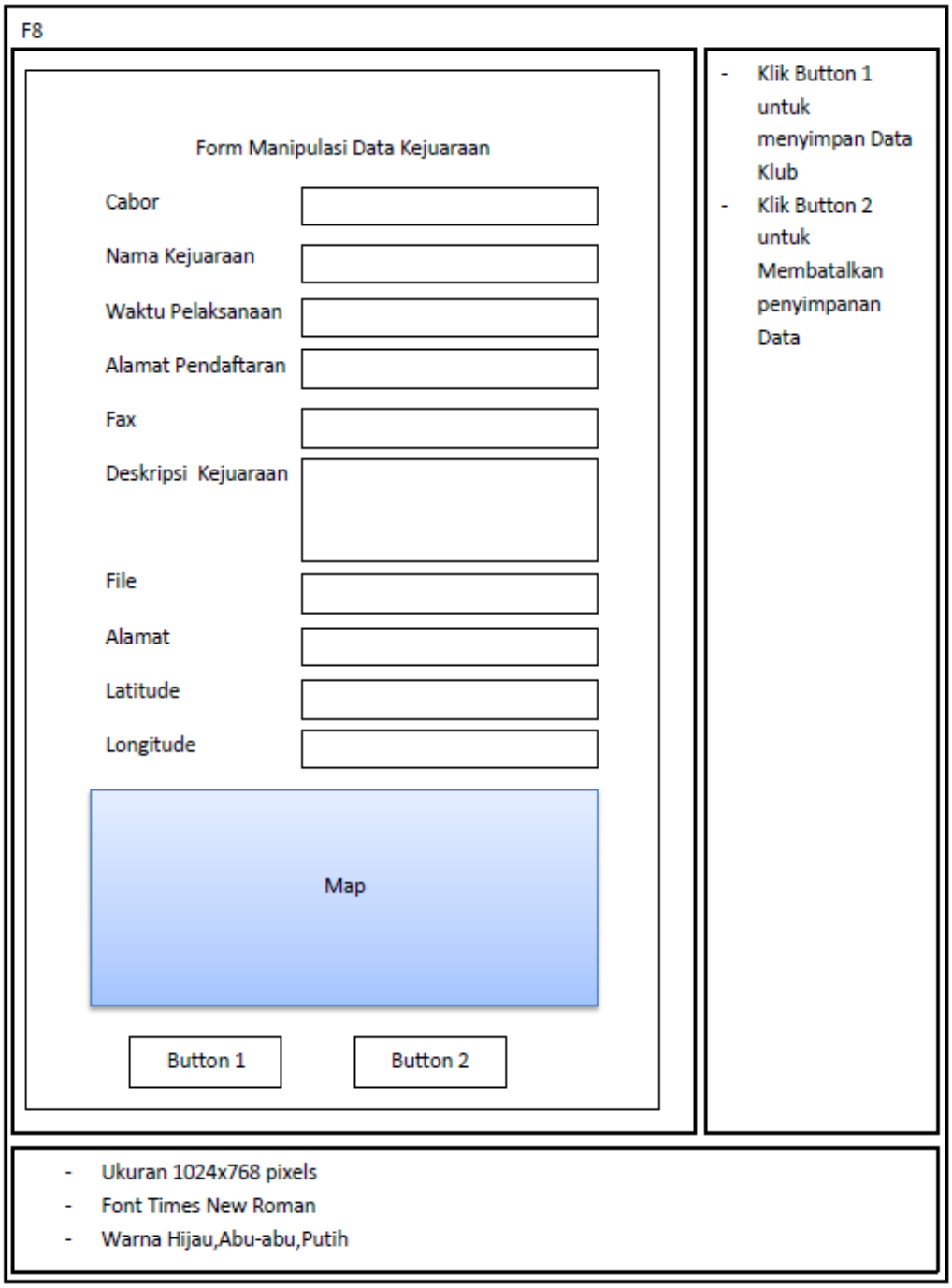
Gambar 3. 91 Antarmuka Form Manipulasi Data Cabor

g. Perancangan Antarmuka Form Manipulasi Data Klub



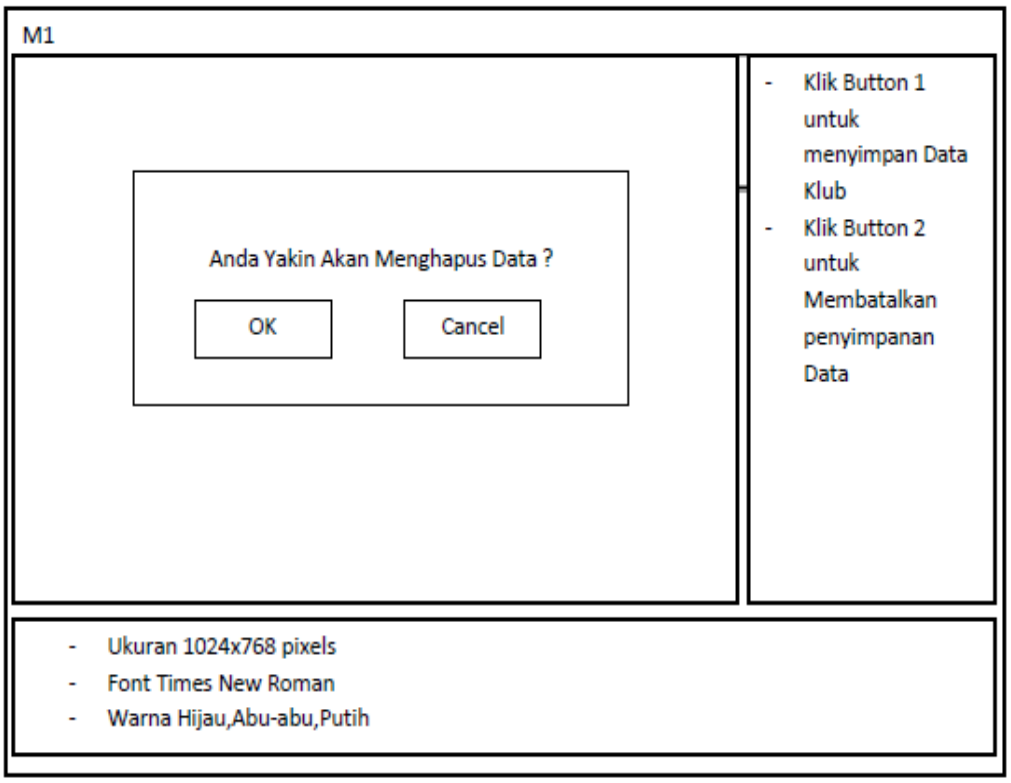
Gambar 3. 92 Antarmuka Form Manipulasi Data Klub

h. Perancangan Antarmuka Form Manipulasi Data Kejuaraan



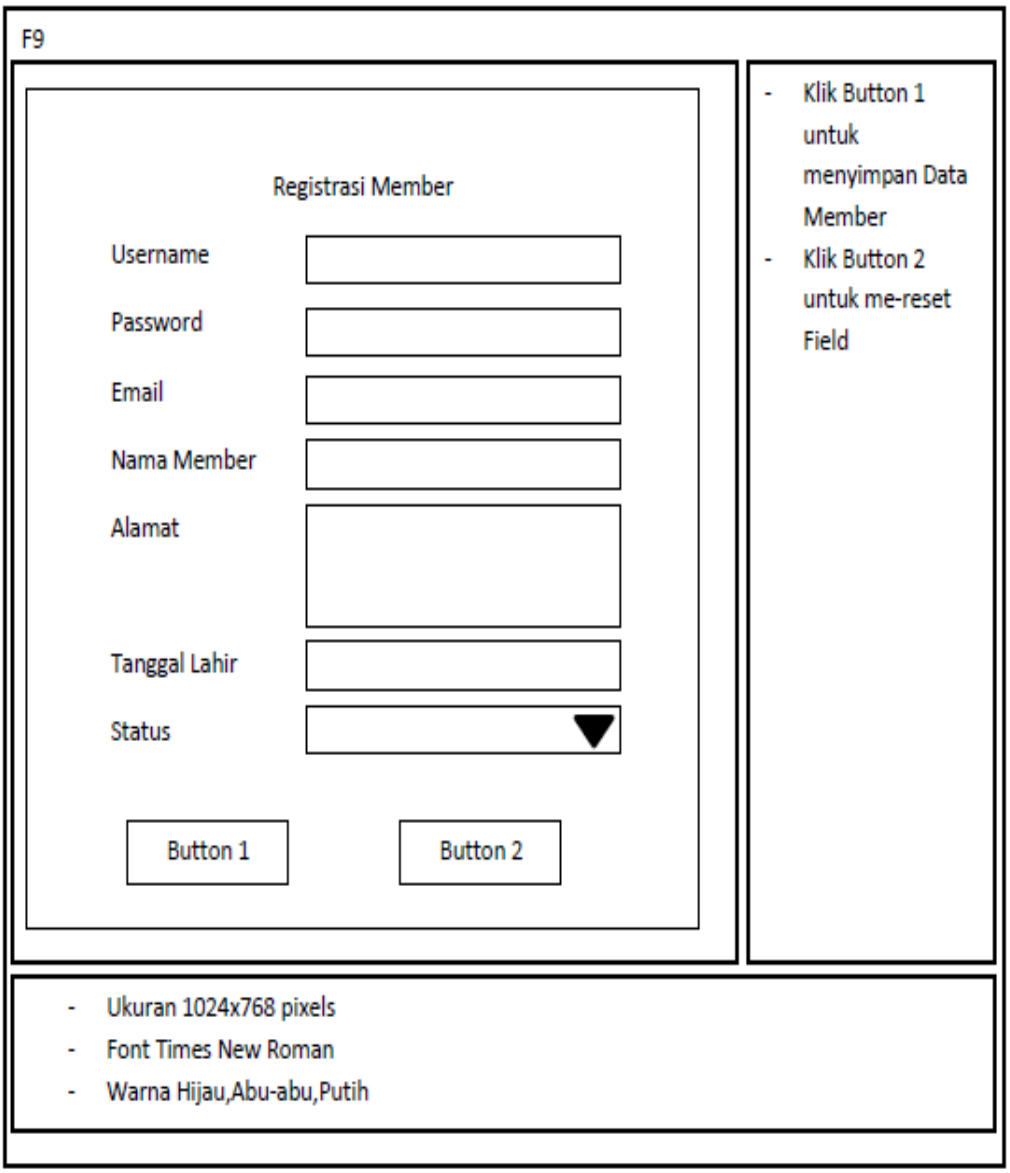
Gambar 3. 93 Antarmuka Form Manipulasi Data Kejuaraan

i. Perancangan Antarmuka Message Hapus Data



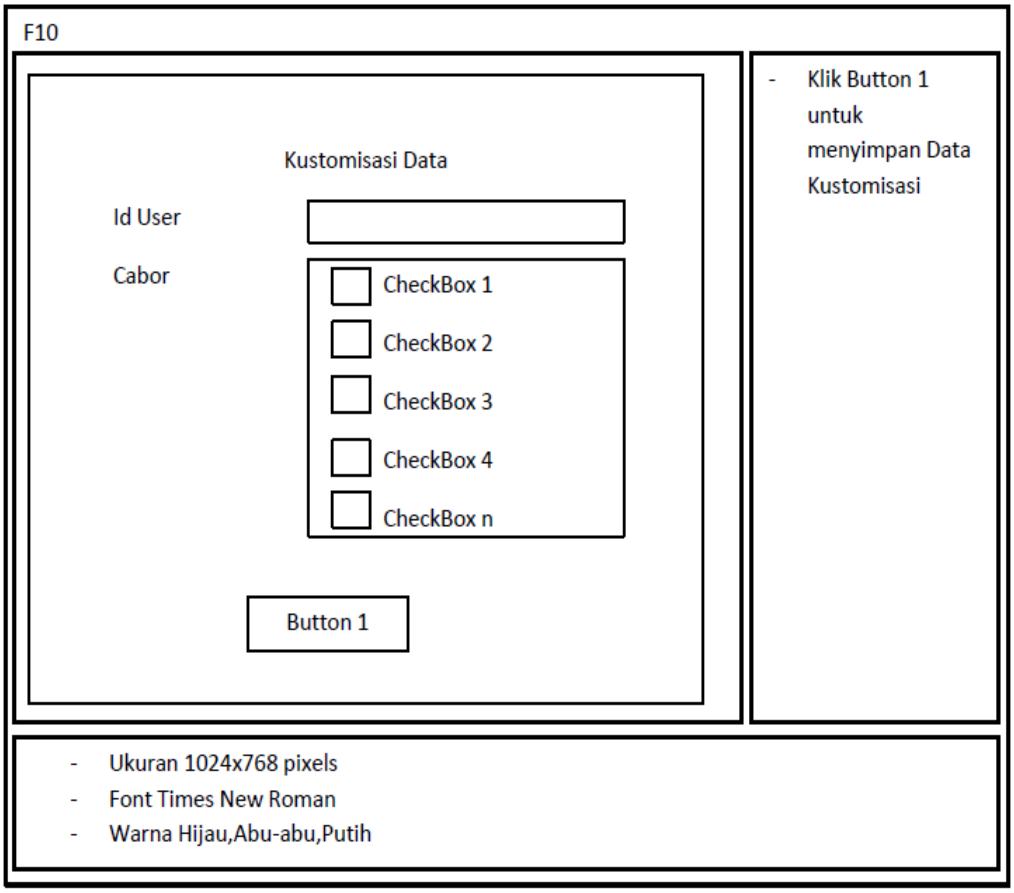
Gambar 3. 94 Antarmuka Message Hapus Data

j. Perancangan Antarmuka Registrasi



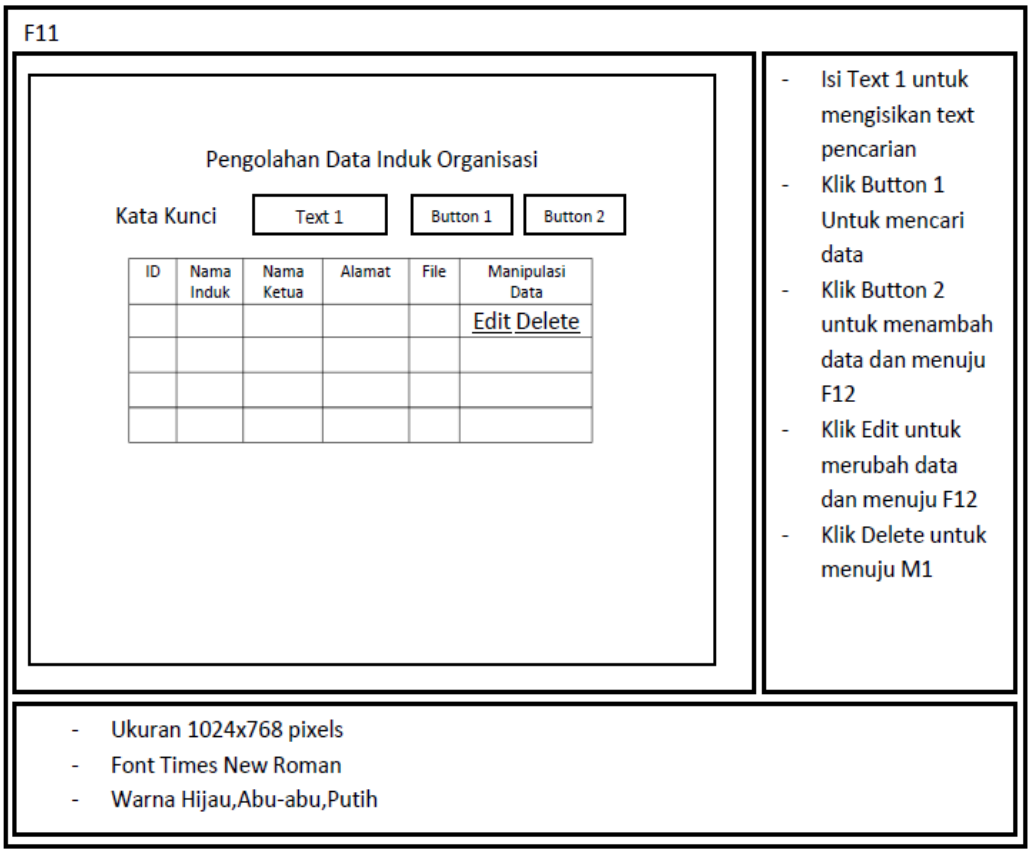
Gambar 3. 95 Antarmuka Registrasi

k. Perancangan Antarmuka Kustomisasi Data



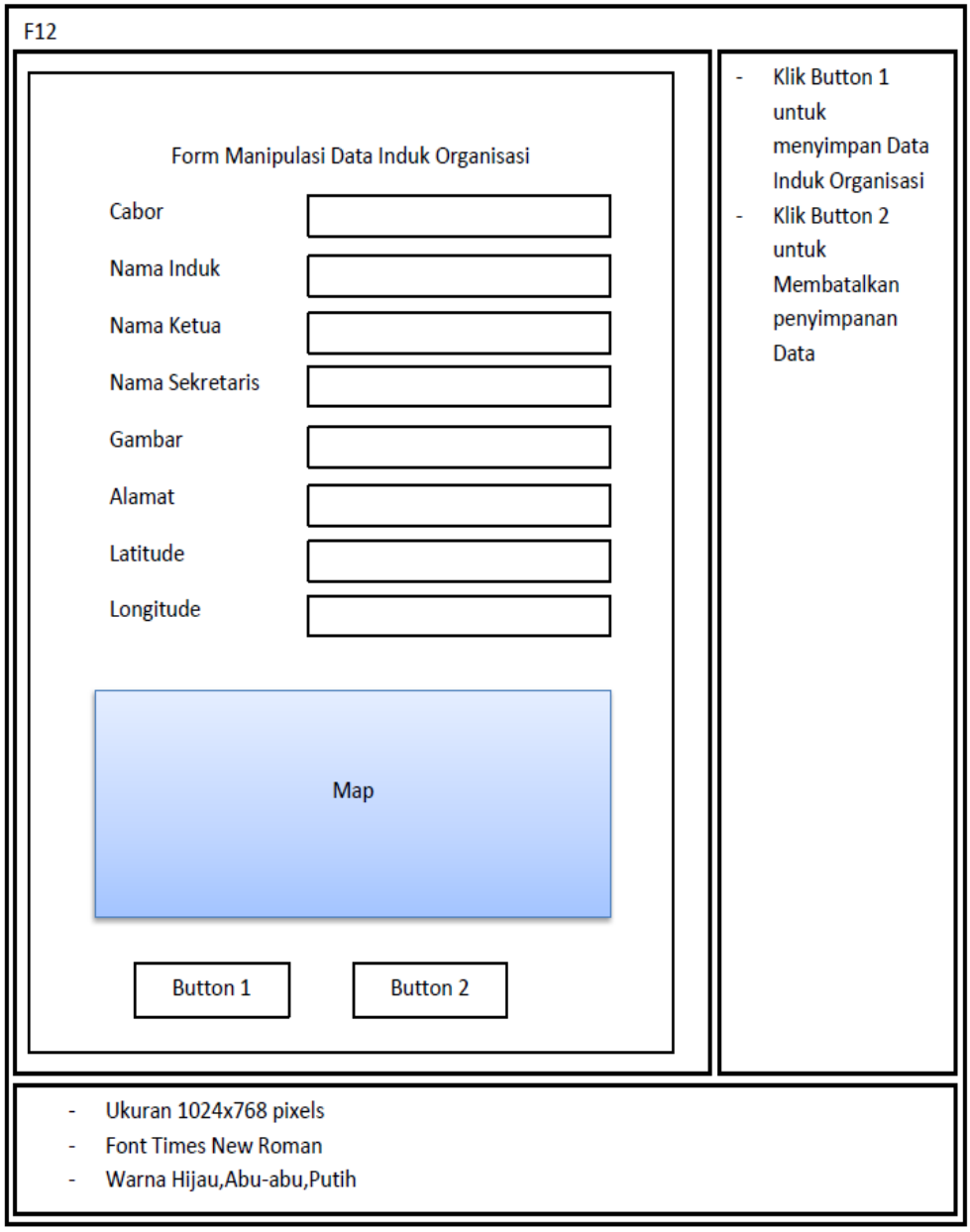
Gambar 3. 96 Antarmuka Kustomisasi Data

l. Perancangan Antarmuka Pengolahan Data Induk Organisasi



Gambar 3. 97 Antarmuka Pengolahan Data Induk Organisasi

m. Perancangan Antarmuka Form Manipulasi Data Induk Organisasi



Gambar 3. 98 Antarmuka Form Manipulasi Data Induk Organisasi