



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB III

### METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

#### 3.1 Metode Penelitian

Adapun tahap-tahap metode penelitian sistem pendukung keputusan pemilihan oli mesin sepeda motor adalah sebagai berikut.

a. Wawancara dan observasi

Pada tahap wawancara dan observasi dilakukan untuk mempelajari proses pelanggan dalam membeli oli dan mengumpulkan data informasi tentang oli mesin sepeda motor di bengkel Wijaya Motor. Proses wawancara dilakukan kepada Bapak Andi Wiharja selaku pemilik bengkel dan beberapa pelanggan bengkel. Wawancara bertujuan untuk mencari informasi ketika pelanggan memilih oli dan spesifikasi oli yang dijual di bengkel Wijaya Motor. Transkrip wawancara dapat dilihat pada lampiran.

b. Studi Literatur

Pada tahap studi literatur akan dilakukan dengan pembelajaran pada teori-teori terkait penelitian. Teori-teori tersebut adalah sistem pendukung keputusan, metode MOORA, oli, bengkel Wijaya Motor, serta teori lain yang dibutuhkan selama proses penelitian. Referensi yang digunakan pada studi literatur bersumber dari jurnal ilmiah, buku, artikel, dan sumber lainnya.

c. Perancangan

Pada tahap perancangan aplikasi akan menghasilkan *flowchart*, *Data Flow Diagram* (DFD), *Entity Relationship Diagram* (ERD), struktur tabel, *database schema*, dan *User Interface* (UI). *Flowchart* digunakan untuk menunjukkan alur kerja sistem. DFD digunakan untuk menjelaskan alur data yang diproses di dalam

sistem. ERD digunakan untuk menggambarkan hubungan antara suatu entitas dengan entitas lain. Struktur tabel dan *database schema* digunakan untuk menjelaskan basis data yang digunakan sistem. UI digunakan untuk menggambarkan desain antarmuka sistem.

d. Implementasi

Hasil perancangan yang telah dibuat akan diimplementasikan ke dalam pembangunan aplikasi web. Aplikasi web akan dibangun dengan *framework* CodeIgniter menggunakan bahasa pemrograman PHP: *Hypertext Preprocessor* (PHP) dan *HyperText Markup Language* (HTML). Basis datanya akan menggunakan MySQL. Kemudian, untuk mendukung tampilan aplikasi akan menggunakan *library* Bootstrap, Data table, dan JQuery.

e. Uji coba

Setelah aplikasi selesai dirancang dan dibangun, akan dilaksanakan tahap uji coba dengan cara uji coba skenario. Uji coba skenario bertujuan untuk memberikan pemaparan perhitungan metode MOORA. Pemaparan perhitungan tersebut untuk mendapatkan hasil tingkat keakuratan sistem dan meminimalisir kesalahan yang ada.

f. Evaluasi

Pada tahap evaluasi akan dilakukan untuk mengetahui tingkat keberhasilan aplikasi yang telah dirancang dan dibangun dengan penyebaran kuisisioner EUCS kepada *user*. Hasil evaluasi tersebut menjadi tolak ukur keberhasilan sistem. Kemudian, hasil evaluasi dapat dibuat kesimpulan dari penelitian yang telah dilakukan.

g. Konsultasi dan penulisan laporan

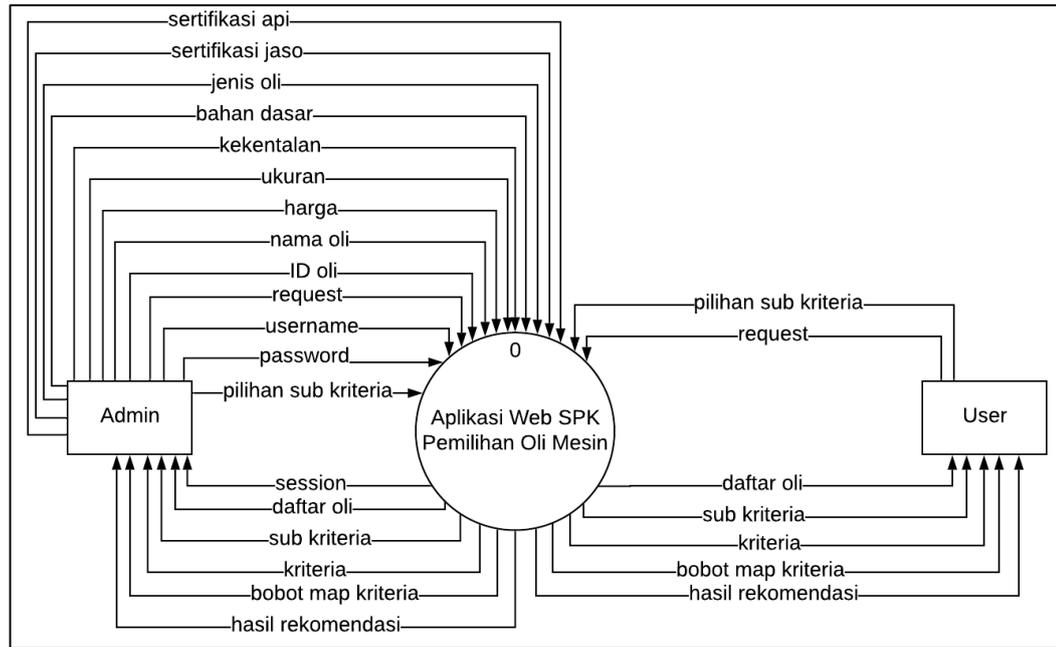
Pada bagian terakhir dari metode penelitian ini akan dilakukan konsultasi dan penulisan laporan. Hasil penelitian yang telah dibuat dan didapatkan, kemudian dituangkan ke dalam bentuk laporan. Selama penelitian dan penulisan laporan dilakukan konsultasi. Konsultasi bertujuan untuk berdiskusi kepada dosen pembimbing sehingga meminimalisir kesalahan dalam proses penelitian dan penulisan laporan. Konsultasi dan bimbingan penulisan laporan dengan dosen pembimbing dilakukan sebanyak minimal delapan kali pertemuan.

### **3.2 Perancangan Sistem**

Proses perancangan sistem dimulai dengan membuat *Data Flow Diagram* (DFD), *Flowchart*, *Entity Relationship Diagram* (ERD), skema *database*, struktur tabel, dan tampilan antarmuka.

#### **3.2.1 Data Flow Diagram**

*Data Flow Diagram* (DFD) digunakan untuk menggambarkan data aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*) untuk membantu memahami logika sistem secara terstruktur dan jelas. DFD yang dirancang terdiri dari satu buah diagram konteks dan satu buah DFD Level 0. Dalam diagram ini terdapat 2 entitas, yaitu entitas *admin* dan entitas *user*. Pembuatan DFD mengacu pada simbol DFD menurut Yourdan dan De Marco (Sukamto dan Shalahudin, 2014). Diagram konteks dari sistem pendukung keputusan pemilihan oli mesin sepeda motor ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Konteks

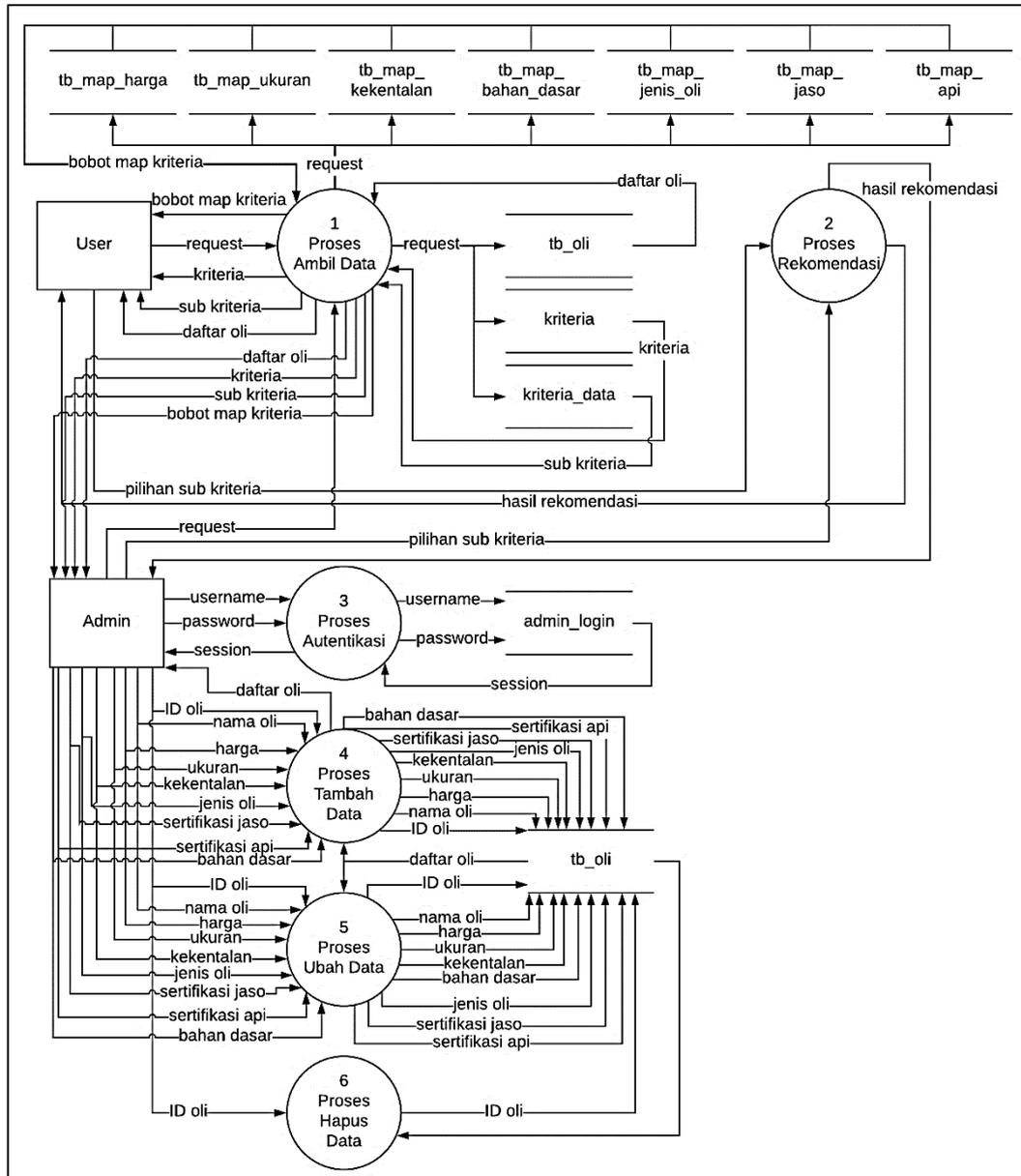
Aplikasi ini memiliki dua entitas, yaitu *user* dan *admin*. Entitas *user* melakukan proses ambil data dengan mengirimkan *request* ke aplikasi. Saat *request* berhasil, aplikasi akan mengirimkan daftar oli, sub kriteria, kriteria, dan bobot *map* kriteria ke *user*. Daftar oli merupakan alternatif yang berisi daftar oli mesin sepeda motor. Kriteria merupakan data kriteria yang berisi kriteria-kriteria beserta atributnya.

Sub kriteria berisi sub kriteria dari masing-masing kriteria beserta bobotnya. Bobot *map* kriteria merupakan bilangan *fuzzy* kriteria yang berisi bobot bilangan *fuzzy* dari setiap kriteria. Kemudian, *user* dapat melakukan proses rekomendasi dengan memberikan masukan pilihan sub kriteria ke aplikasi. Data tersebut akan diproses oleh aplikasi dan akan memberikan keluaran ke *user* berupa hasil rekomendasi. Hasil rekomendasi tersebut berisi daftar oli beserta hasil perhitungan metode MOORA.

Entitas *admin* dapat melakukan proses autentikasi dengan memberikan masukkan *username* dan *password*. Ketika proses autentikasi berhasil maka aplikasi akan memberikan *session* ke *admin*. Kemudian, *admin* meminta *request* data ke aplikasi. Setelah *request* berhasil maka aplikasi akan mengirimkan daftar oli, sub kriteria, kriteria, dan bobot *map* kriteria. *Admin* dapat melakukan proses rekomendasi seperti *user* dan dapat melakukan pengolahan data alternatif, seperti menambah, mengubah, dan menghapus data alternatif oli.

*Admin* memberikan masukkan ID oli, nama oli, harga, ukuran, kekentalan, bahan dasar, jenis oli, sertifikasi jaso, dan sertifikasi api untuk menambah data alternatif oli. *Admin* memberikan masukkan ID oli, nama oli, harga, ukuran, kekentalan, bahan dasar, jenis oli, sertifikasi jaso, dan sertifikasi api untuk mengubah data alternatif oli. *Admin* memberikan masukkan berupa ID Oli untuk menghapus data. *Admin* dapat mengetahui hasil rekomendasi dari metode MOORA dengan memberikan masukkan pilihan sub kriteria beserta bobotnya. *Admin* dapat melihat daftar oli yang akan ditampilkan ke dalam bentuk tabel.

*Admin* dapat melihat kriteria dan bobotnya yang akan ditampilkan ke dalam bentuk tabel. *Admin* dapat melihat bobot *map* kriteria yang akan ditampilkan ke dalam bentuk tabel. Setelah diagram konteks selesai, terdapat DFD Level 0 yang ditunjukkan pada Gambar 3.2.



Gambar 3.2 DFD Level 0

Pada DFD Level 0 terdapat enam proses, yaitu proses ambil data, proses rekomendasi, proses autentikasi, proses tambah data, proses ubah data, dan proses hapus data. Pada proses Ambil Data, terdapat aliran data yang mengalir dari entitas *user* dan entitas *admin*, yaitu *request* yang mengalir ke tabel *tb\_oli*, tabel *kriteria*, tabel *kriteria\_data*, *tb\_map\_harga*, tabel *tb\_map\_ukuran*, tabel *tb\_map\_kekentalan*, tabel *tb\_map\_bahan\_dasar*, tabel *tb\_map\_jenis\_oli*, tabel *tb\_map\_jaso*, dan tabel

tb\_map\_api. Kemudian, terdapat aliran data dari tabel tb\_oli ke entitas *user* dan entitas *admin*, yaitu daftar oli. Terdapat aliran data dari tabel kriteria ke entitas *user* dan entitas *admin*, yaitu kriteria. Terdapat aliran data dari tabel kriteria\_data ke entitas *user* dan entitas *admin*, yaitu sub kriteria.

Terdapat aliran data dari tabel tb\_map\_harga, tabel tb\_map\_ukuran, tabel tb\_map\_kekentalan, tabel tb\_map\_bahan\_dasar, tabel tb\_map\_jenis\_oli, tabel tb\_map\_jaso, dan tabel tb\_map\_api ke entitas *user* dan entitas *admin*, yaitu bobot *map* kriteria. Semua data dari tabel tersebut akan digunakan untuk proses Rekomendasi. Pada proses Rekomendasi, terdapat aliran data yang mengalir dari entitas *user* dan entitas *admin*, yaitu pilihan sub kriteria. Kemudian, terdapat aliran data kembali ke entitas *user* dan entitas *admin*, yaitu hasil rekomendasi. Proses Ambil Data dan proses Rekomendasi untuk entitas *admin* hanya dapat dilakukan setelah proses Autentikasi berhasil.

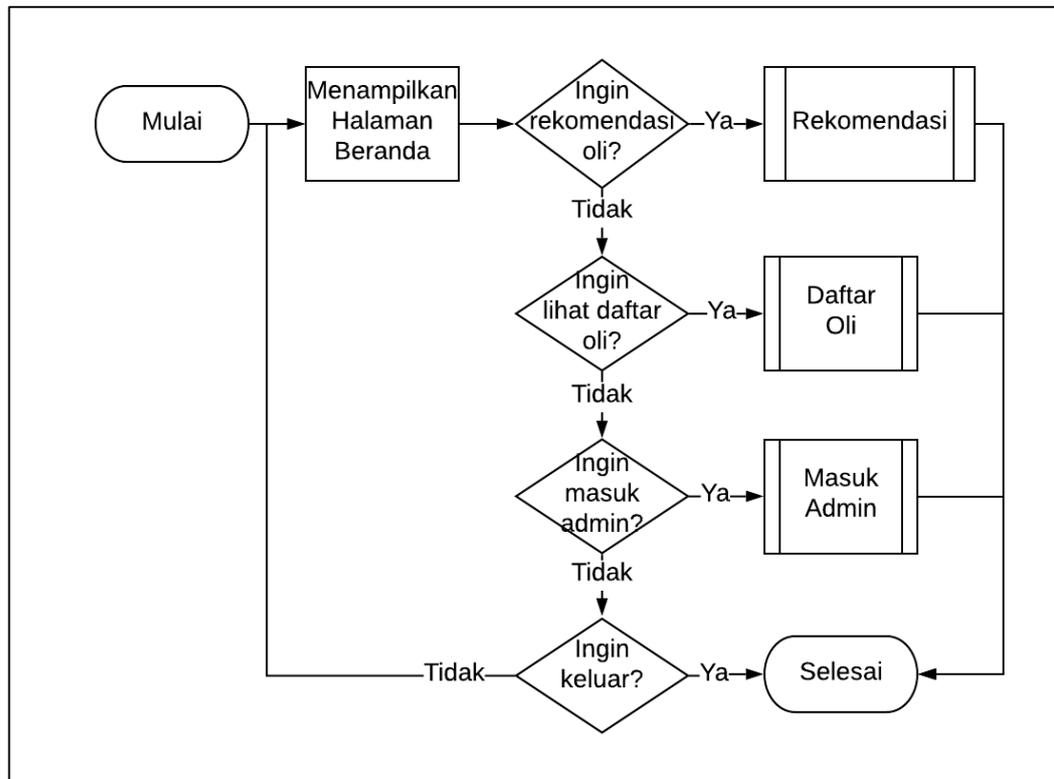
Pada proses Autentikasi, terdapat aliran data yang mengalir dari entitas *admin* ke tabel tabel admin\_login, yaitu *username* dan *password*. Kemudian, terdapat aliran data dari tabel admin\_login ke entitas *admin*, yaitu *session*. Setelah proses Autentikasi selesai maka entitas *admin* dapat melakukan proses Ambil Data dan proses Rekomendasi. Pada proses Tambah Data, terdapat beberapa aliran data yang mengalir dari entitas *admin* ke tabel tb\_oli, yaitu ID oli, nama oli, harga, ukuran, kekentalan, bahan dasar, jenis oli, sertifikasi jaso, dan sertifikasi api. Kemudian, terdapat aliran data yang mengalir dari tabel tb\_oli ke entitas *admin*, yaitu daftar oli.

Pada proses Ubah Data, terdapat beberapa aliran data yang mengalir dari entitas *admin* ke tabel tb\_oli, yaitu ID oli, nama oli, harga, ukuran, kekentalan,

bahan dasar, jenis oli, sertifikasi jaso, dan sertifikasi api. Kemudian, terdapat aliran data yang mengalir dari tabel *tb\_oli* ke entitas *admin*, yaitu daftar oli. Pada proses Hapus Data, terdapat aliran data yang mengalir dari entitas *admin* ke tabel *tb\_oli*, yaitu ID oli. Kemudian, terdapat aliran data yang mengalir dari tabel *tb\_oli* ke entitas *admin*, yaitu daftar oli.

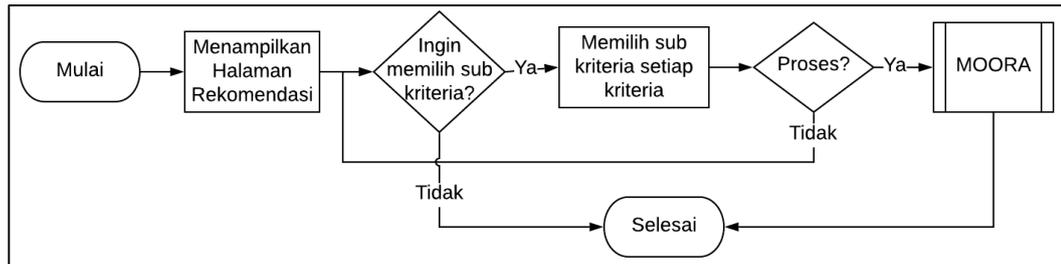
### 3.2.2 Flowchart

Pada bagian ini akan dijabarkan *flowchart* untuk menggambarkan alur aplikasi. *Flowchart* aplikasi ini terdiri dari dua bagian, yaitu *flowchart* untuk *user* dan *admin*. Namun terdapat beberapa *flowchart* yang sama untuk *user* dan *admin*, yaitu *flowchart* Beranda, *flowchart* Get Fuzzy, *flowchart* Normalisasi, *flowchart* Optimasi, *flowchart* MOORA. *Flowchart* Beranda dapat dilihat pada Gambar 3.3.



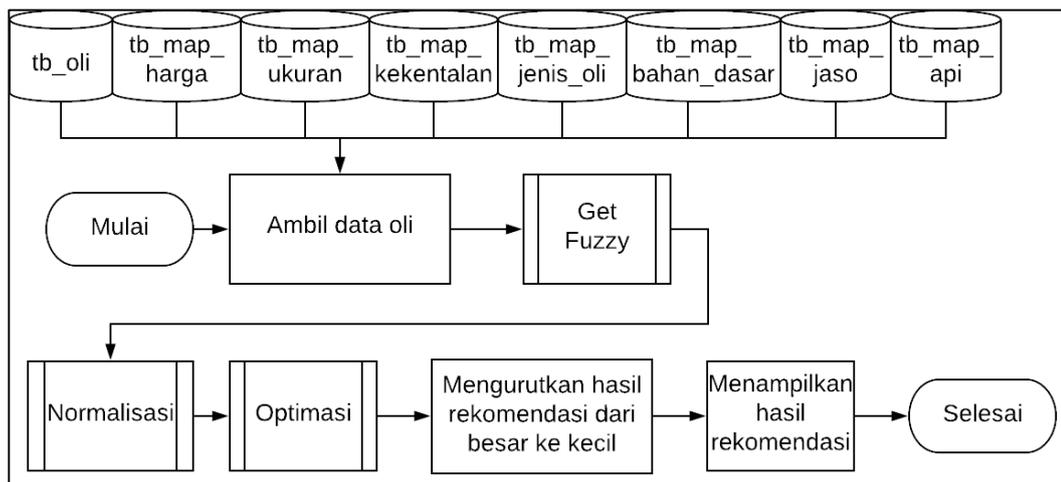
Gambar 3.3 *Flowchart* Beranda

Gambar 3.3 menunjukkan alur aplikasi secara umum. Halaman Beranda akan muncul ketika *user* maupun *admin* pertama kali membuka aplikasi. *User* dan *admin* dapat melakukan pemilihan rekomendasi oli dan melihat daftar oli. Hanya *admin* yang dapat melakukan masuk *admin* karena mengetahui aksesnya. Kemudian, *flowchart* Rekomendasi dapat dilihat pada Gambar 3.4.



Gambar 3.4 *Flowchart* Rekomendasi

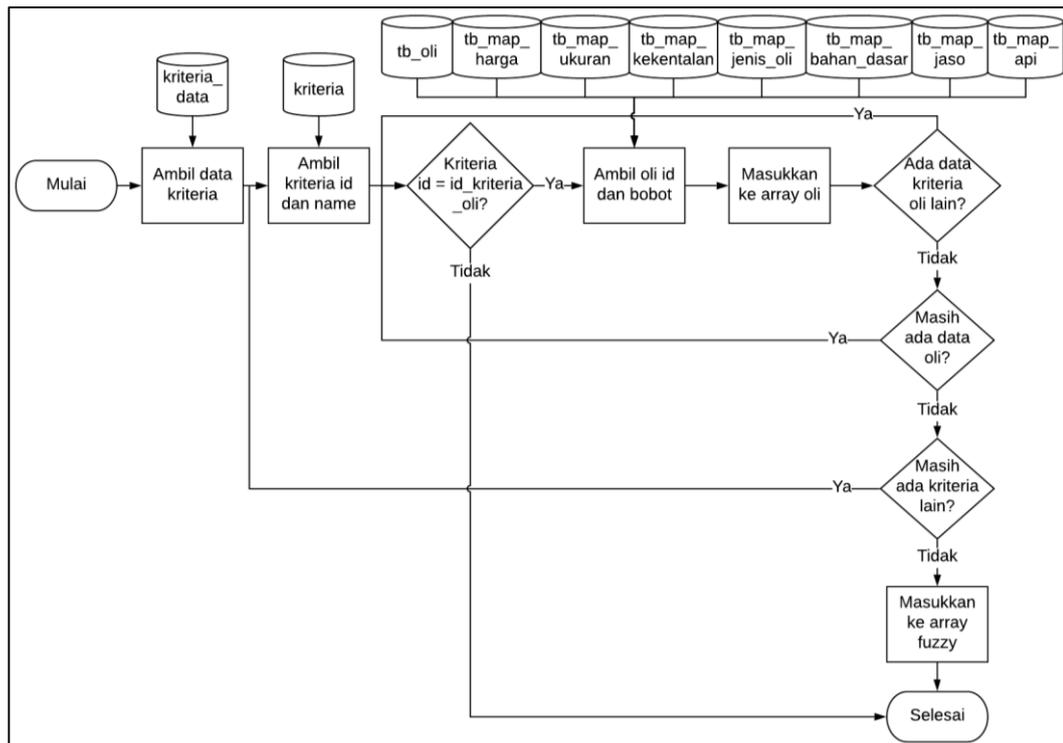
Gambar 3.4 menunjukkan alur bagian Rekomendasi. Setelah halaman Rekomendasi selesai ditampilkan maka akan muncul pilihan sub kriteria dari setiap kriteria. *User* dan *admin* dapat memilih sub kriteria yang sesuai kemudian melanjutkan proses untuk memperoleh hasil rekomendasinya. Hasil rekomendasi didapat dari proses MOORA. *Flowchart* MOORA dapat dilihat pada Gambar 3.5.



Gambar 3.5 *Flowchart* MOORA

Gambar 3.5 menunjukkan alur perhitungan metode MOORA. Data oli akan diambil dari tabel *tb\_oli*, *tb\_map\_harga*, *tb\_map\_ukuran*, *tb\_map\_kekentalan*,

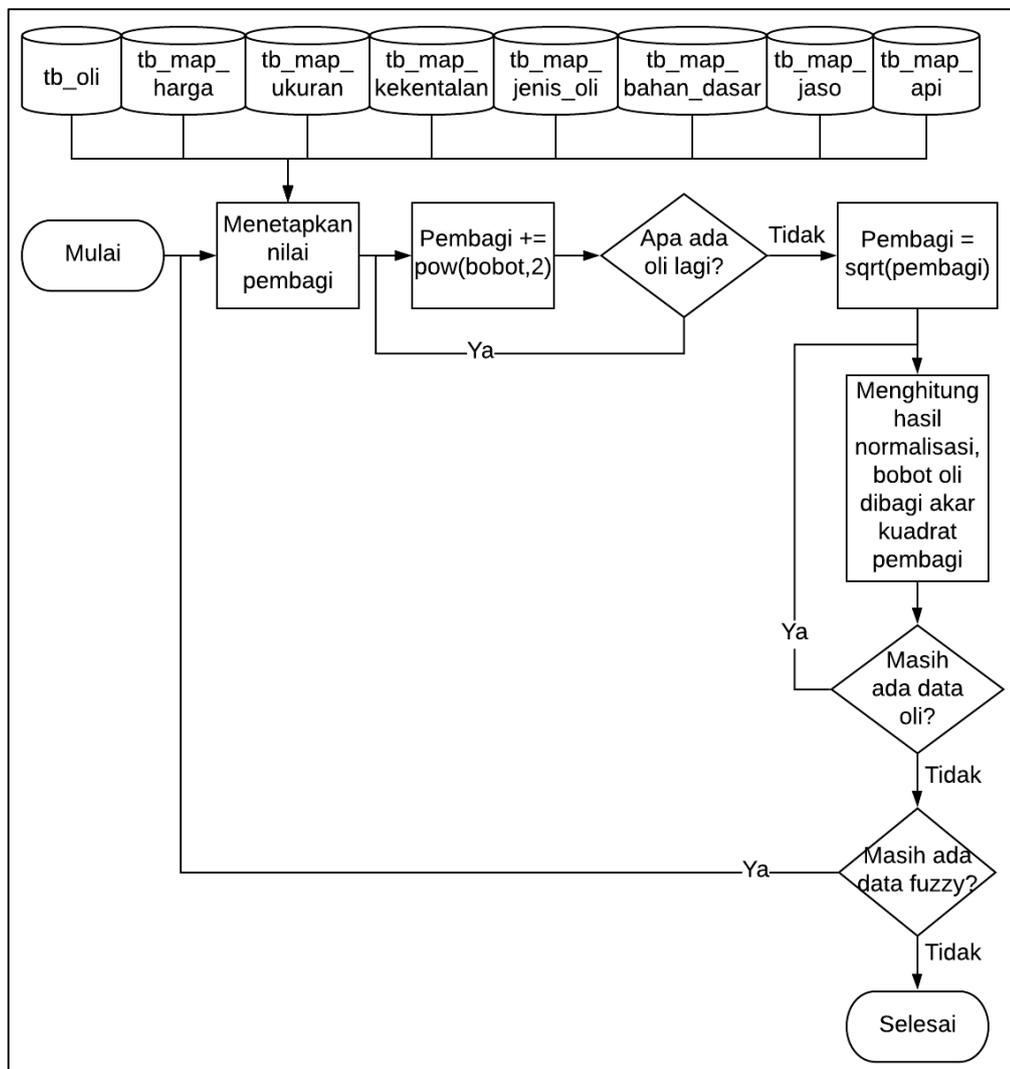
tb\_map\_bahan\_dasar, tb\_map\_jenis\_oli, tb\_map\_jaso, dan tb\_map\_api. Kemudian, dilakukan proses *Get Fuzzy*, proses Normalisasi, dan proses Optimasi. Hasil rekomendasi diperoleh dari proses Optimasi dan dilakukan pengurutan dari besar ke kecil. Setelah itu, hasil rekomendasi tersebut ditampilkan. *Flowchart Get Fuzzy* dapat dilihat pada Gambar 3.6.



Gambar 3.6 *Flowchart Get Fuzzy*

Gambar 3.6 menunjukkan alur proses *Fuzzy*. Proses awal dimulai dengan pengambilan data kriteria dari tabel kriteria\_data. Kemudian, dilakukan pengambilan id kriteria dan nama kriteria dari tabel kriteria. Setelah itu, dilakukan pengecekan kriteria id sama dengan id\_kriteria\_oli. Jika ya, dilakukan pengambilan id oli dan bobot *mapping (fuzzy)* dari tabel tb\_oli, tb\_map\_harga, tb\_map\_ukuran, tb\_map\_kekentalan, tb\_map\_bahan\_dasar, tb\_map\_jenis\_oli, tb\_map\_jaso, dan tb\_map\_api. Lalu, dimasukkan ke dalam *array* oli.

Dilakukan pengecekan kembali apakah masih ada data kriteria oli lainnya. Jika ya, maka proses pengambilan id oli dan bobotnya akan diulang lagi sampai tidak ada data kriteria oli lain yang tersisa. Kemudian, dilakukan pengecekan apakah masih ada data oli lain. Jika ya, maka akan dilakukan pengecekan kriteria id sama dengan id\_kriteria\_oli lagi. Jika tidak, maka akan dimasukkan ke *array fuzzy*. Setelah *flowchart Fuzzy* selesai maka dilanjutkan dengan *flowchart Normalisasi* yang dapat dilihat pada Gambar 3.7.

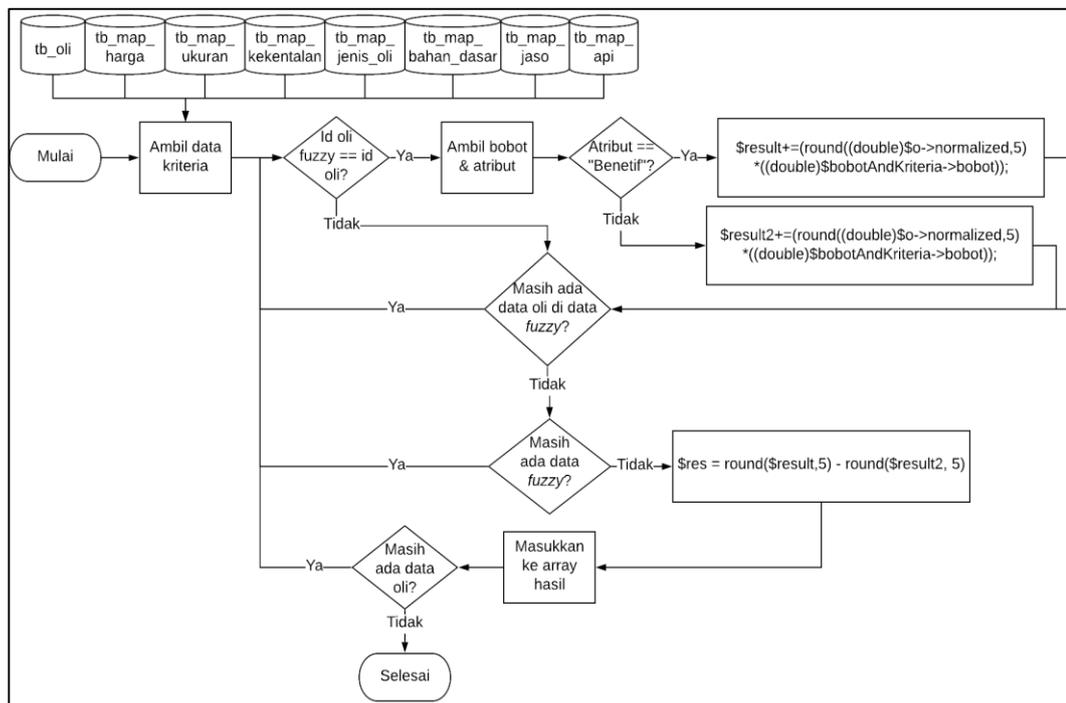


Gambar 3.7 *Flowchart* Normalisasi

Gambar 3.7 menunjukkan alur proses Normalisasi. Proses awal dimulai dengan menetapkan nilai pembagi yang didapatkan dari tabel *tb\_oli*, *tb\_map\_harga*,

tb\_map\_ukuran, tb\_map\_kekentalan, tb\_map\_bahan\_dasar, tb\_map\_jenis\_oli, tb\_map\_jaso, dan tb\_map\_api. Kemudian, pembagi sama dengan bobot pangkat dua. Jika ada oli lain maka pembagi yang baru akan dihitung lagi dengan bobot baru pangkat dua. Jika sudah tidak ada oli lain lagi, maka dilanjutkan dengan pembagi sama dengan akar pangkat dua dari pembagi.

Kemudian, menghitung nilai normalisasi dengan bobot dikali akar pangkat dua pembagi. Jika masih ada data oli maka dihitung lagi nilai normalisasi yang baru sampai tidak ada lagi data oli. Jika masih ada data *fuzzy* maka kembali ke menetapkan nilai pembagi lagi sampai tidak ada data *fuzzy*. Setelah *flowchart* Normalisasi selesai maka dilanjutkan dengan *flowchart* Optimasi yang dapat dilihat pada Gambar 3.8.



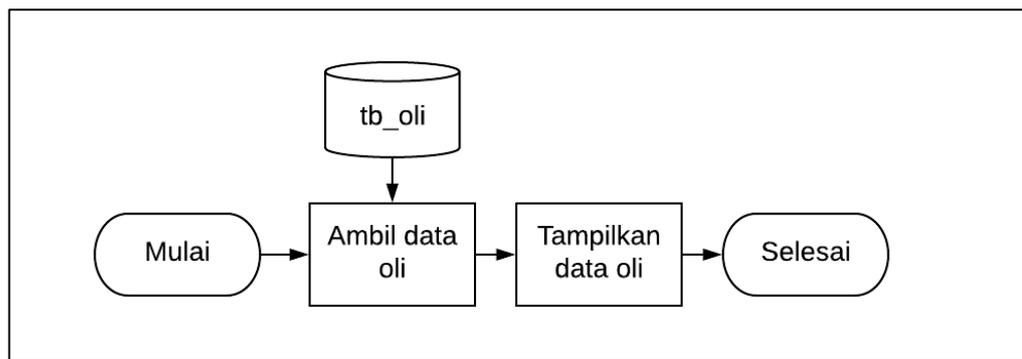
Gambar 3.8 *Flowchart* Optimasi

Gambar 3.8 menunjukkan alur proses Optimasi. Proses awal dimulai dengan mengambil data kriteria dari tabel tb\_oli, tb\_map\_harga, tb\_map\_ukuran,

tb\_map\_kekentalan, tb\_map\_bahan\_dasar, tb\_map\_jenis\_oli, tb\_map\_jaso, dan tb\_map\_api. Jika id oli *fuzzy* sama dengan id oli maka diambil bobot dan atributnya dari setiap alternatif. Setelah itu, dilakukan perhitungan optimasi masing-masing kriteria dari setiap alternatif, yaitu hasil normalisasi dari proses sebelumnya dikalikan dengan bobot kriteria. Jika kriteria dari alternatif memiliki atribut *benefit*, maka hasilnya disimpan pada variabel \$result dan yang memiliki atribut *cost* akan disimpan pada variabel \$result2.

Proses tersebut diulang sampai semua data alternatif habis. Kemudian dilakukan perhitungan optimasi akhir, yaitu \$result dikurang \$result2 dari masing-masing alternatif sampai semua alternatif habis. Hasil perhitungan optimasi akhir disimpan ke dalam *array* hasil. Selanjutnya, kembali ke *flowchart* MOORA dimana masih terdapat proses mengurutkan *array* hasil yang akan menjadi hasil rekomendasi dan ditampilkan hasil rekomendasi tersebut ke *user* atau *admin*.

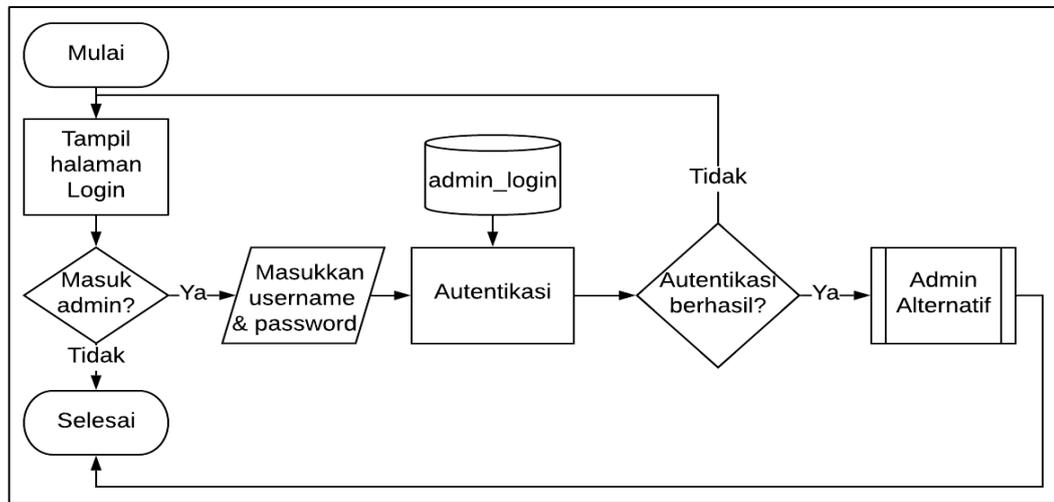
*User* dan *admin* dapat melihat daftar oli yang dijual oleh bengkel. *Flowchart* Daftar Oli dapat dilihat pada Gambar 3.9.



Gambar 3.9 *Flowchart* Daftar Oli

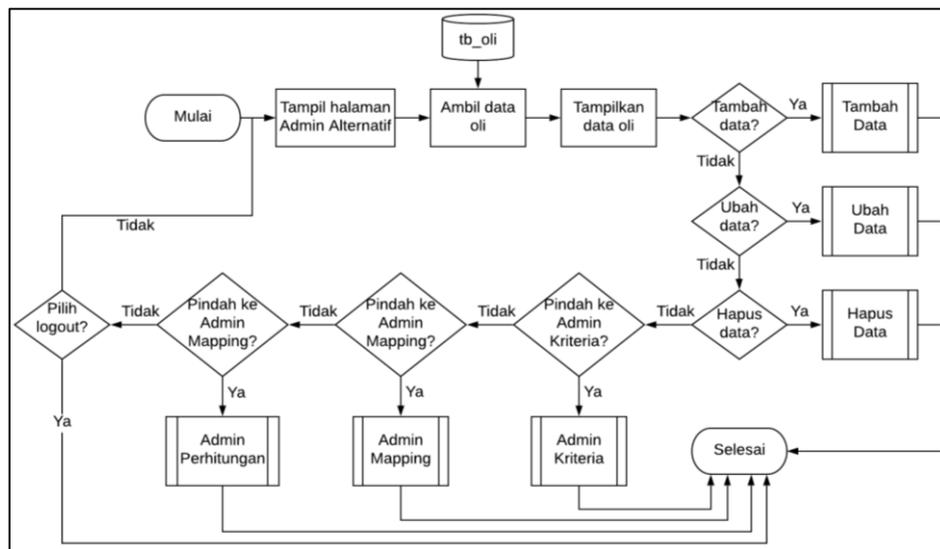
Proses awal dimulai dengan mengambil data oli dari tabel tb\_oli. Kemudian, data oli tersebut ditampilkan pada halaman Daftar Oli. Selanjutnya, *admin* dapat

masuk ke dalam halaman *Admin*. *Flowchart* Masuk Admin dapat dilihat pada Gambar 3.10.



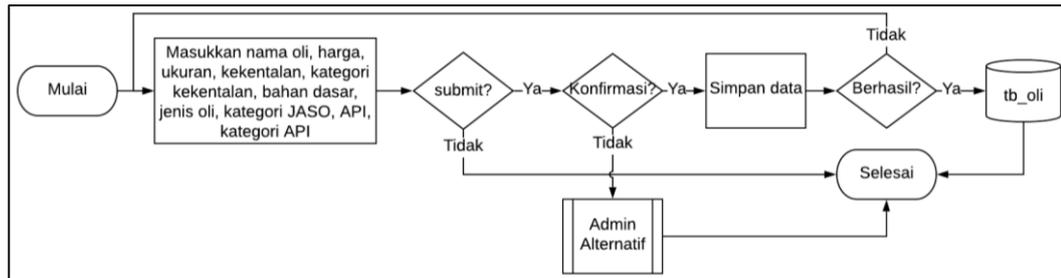
Gambar 3.10 *Flowchart* Masuk Admin

Proses awal dimulai dengan menampilkan halaman *Login*. Kemudian, *admin* melakukan masukkan berupa *username* dan *password*. Masukkan tersebut diautentikasi dari data di dalam tabel *admin\_login*. Setelah proses autentikasi berhasil maka akan masuk ke halaman *Admin*. Halaman *Admin* dibagi menjadi 4, yaitu *Admin Alternatif*, *Admin Kriteria*, *Admin Mapping*, dan *Admin Perhitungan*. *Flowchart* *Admin Alternatif* dapat dilihat pada Gambar 3.11.



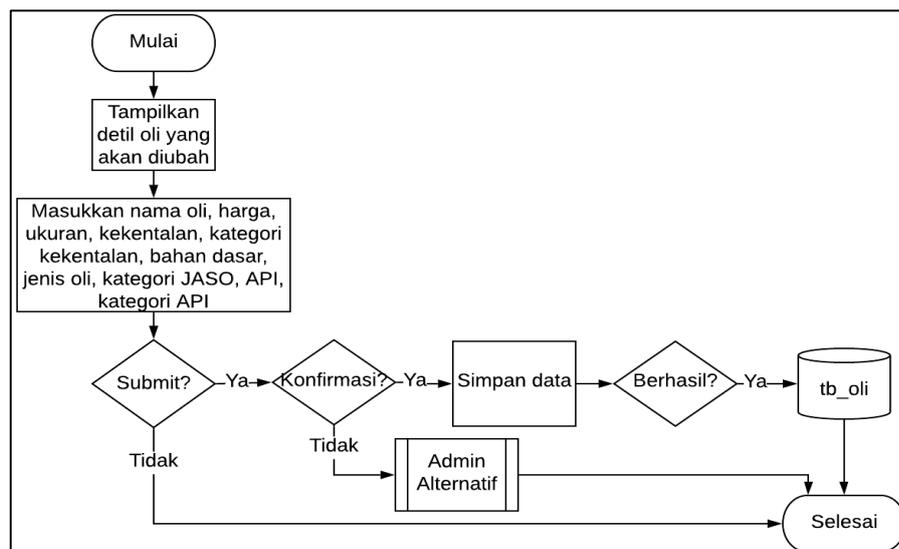
Gambar 3.11 *Flowchart* Admin Alternatif

Proses awal dimulai dengan menampilkan halaman *Admin Alternatif*. Halaman *Admin Alternatif* merupakan halaman utama untuk *admin* yang akan muncul setelah proses autentikasi berhasil. *Admin* dapat mengelola data alternatif, seperti tambah data, ubah data, dan hapus data. *Flowchart* Tambah Data dapat dilihat pada Gambar 3.12.



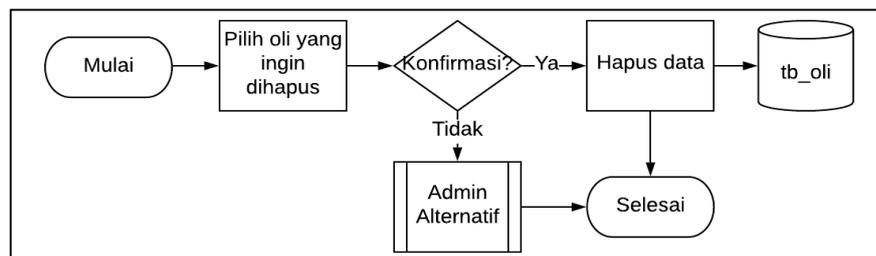
Gambar 3.12 *Flowchart* Tambah Data

Proses awal dimulai dengan memasukkan data-data yang diperlukan, yaitu nama oli, harga, ukuran, kekentalan, kategori kekentalan, bahan dasar, jenis oli, kategori JASO, API, dan kategori API. Kemudian, dilakukan proses penyimpanan data setelah *admin* melakukan *submit* dan konformasi. Jika proses simpan data berhasil maka data baru akan masuk ke dalam tabel *tb\_oli*. Berikut ini adalah *flowchart* Ubah Data yang dapat dilihat pada Gambar 3.13.



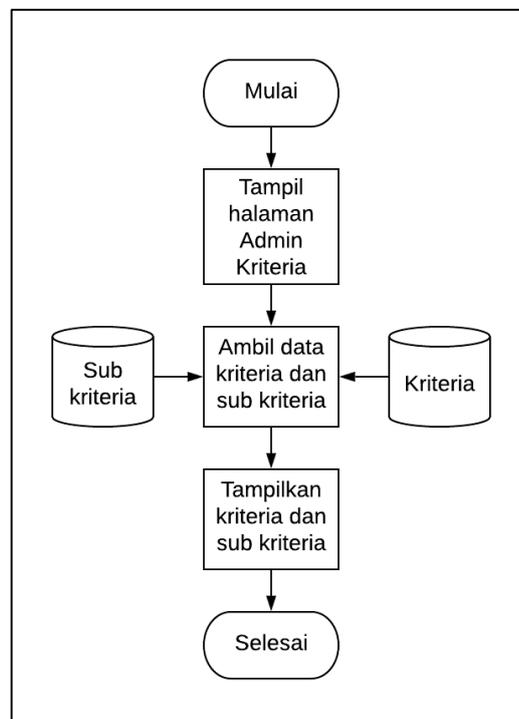
Gambar 3.13 *Flowchart* Ubah Data

Proses awal dimulai dengan menampilkan detail data oli yang akan diubah. *Admin* diminta untuk memasukkan data baru sesuai yang ingin diubah. Kemudian, dilakukan proses penyimpanan data setelah *admin* melakukan *submit* dan konfirmasi. Jika proses simpan data berhasil maka data baru akan masuk ke dalam tabel *tb\_oli*. Berikut ini adalah *flowchart* Hapus Data yang dapat dilihat pada Gambar 3.14.



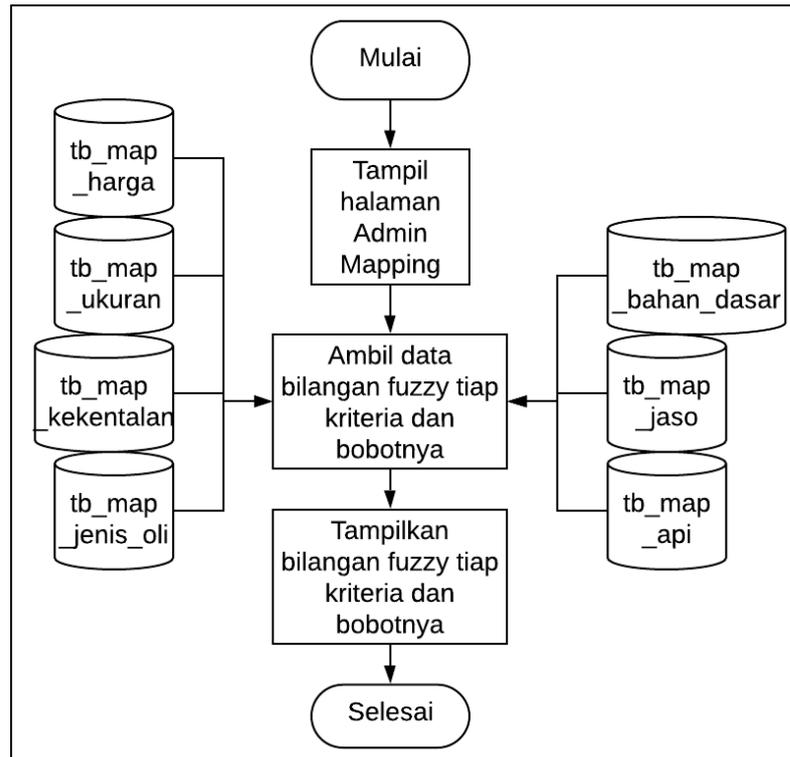
Gambar 3.14 *Flowchart* Hapus Data

Proses awal dimulai dengan pemilihan oli yang akan dihapus. Kemudian, setelah konfirmasi dilakukan maka data akan dihapus dari tabel *tb\_oli*. Berikut ini adalah *flowchart* Admin Kriteria yang dapat dilihat pada Gambar 3.15.



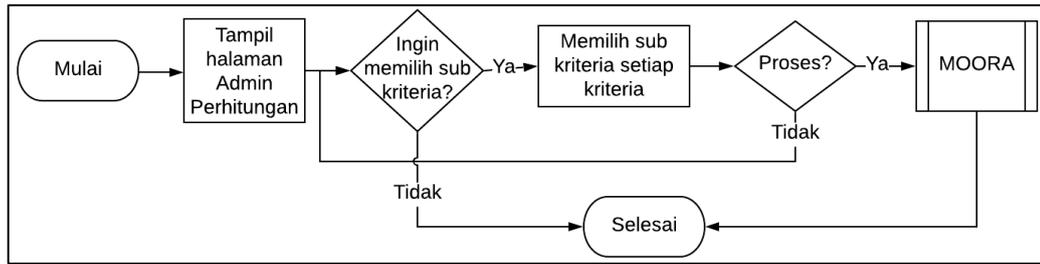
Gambar 3.15 *Flowchart* Admin Kriteria

Proses awal dimulai dengan menampilkan halaman *Admin Kriteria*. Kemudian, diambil data kriteria dari tabel kriteria dan data sub kriteria dari tabel sub kriteria. Setelah itu, data kriteria dan sub kriteria ditampilkan ke dalam halaman *Admin Kriteria*. Berikut ini adalah *flowchart Admin Mapping* yang dapat dilihat pada Gambar 3.16.



Gambar 3.16 *Flowchart Admin Mapping*

Proses awal dimulai dengan menampilkan halaman *Admin Mapping*. Kemudian, data bilangan *fuzzy* tiap kriteria dan bobotnya diambil dari tabel *tb\_map\_harga*, *tb\_map\_ukuran*, *tb\_map\_kekentalan*, *tb\_map\_bahan\_dasar*, *tb\_map\_jenis\_oli*, *tb\_map\_jaso*, dan *tb\_map\_api*. Setelah itu, data bilangan *fuzzy* tiap kriteria dan bobotnya ditampilkan ke dalam halaman *Admin Mapping*. Berikut ini adalah *flowchart Admin Perhitungan* yang dapat dilihat pada Gambar 3.17.

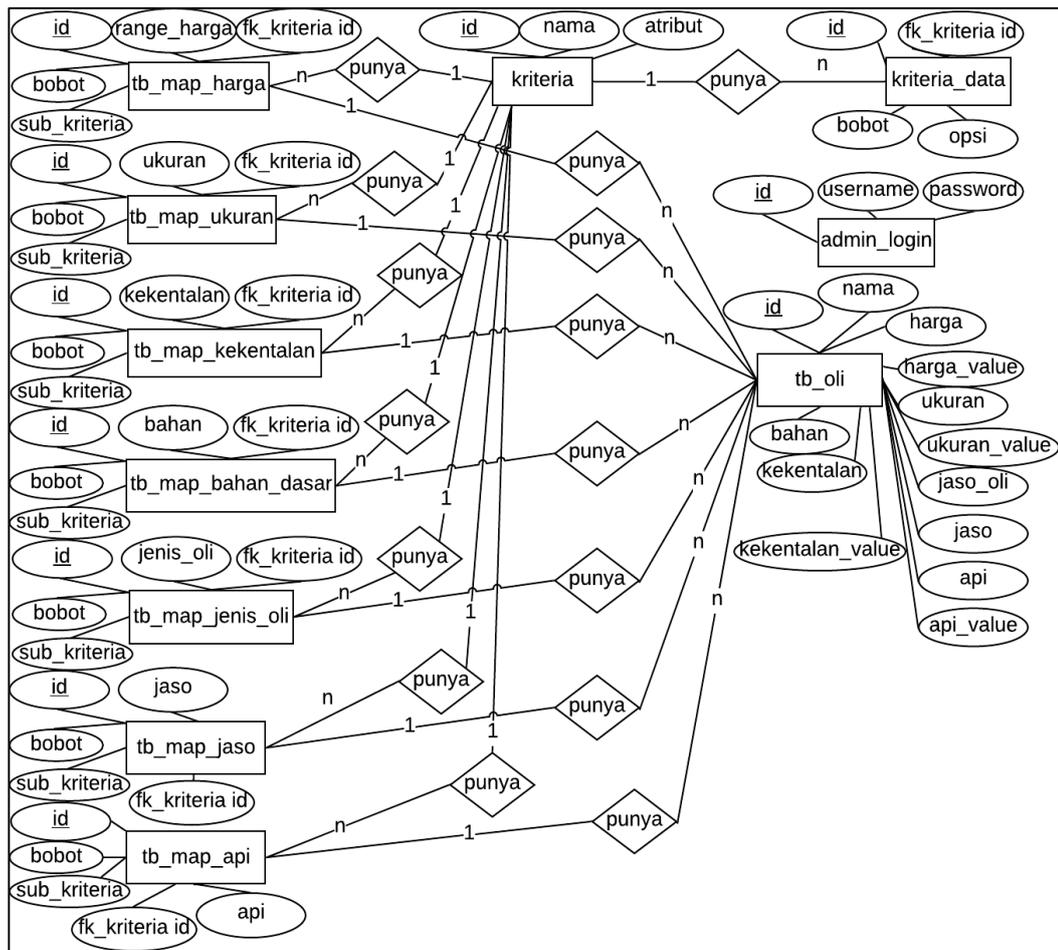


Gambar 3.17 *Flowchart Admin Perhitungan*

Pada bagian *Admin Perhitungan* mirip seperti bagian *Rekomendasi* untuk melihat hasil rekomendasi oli. Proses awal dimulai dengan menampilkan halaman *Admin Perhitungan*. Jika *admin* ingin memilih sub kriteria, maka *admin* memberikan masukan berupa pilihan sub kriteria dari setiap kriteria dan diproses ke bagian MOORA. Jika *admin* tidak ingin memilih sub kriteria maka menuju ke selesai. Setelah memilih sub kriteria, maka *admin* diminta untuk melanjutkan proses. Jika *admin* melakukan proses, maka data pilihan *admin* akan menuju ke modul MOORA untuk diolah. Setelah proses MOORA selesai maka hasilnya akan ditampilkan ke halaman *Admin Perhitungan*.

### 3.2.3 Entity Relationship Diagram

Pada bagian ini akan dijabarkan perancangan *Entity Relationship Diagram* (ERD) yang memiliki tujuan untuk mengetahui hubungan antar entitas dan atribut-atribut milik entitas. Pada ERD ini akan dijabarkan ke dalam bentuk *logical Entity Relationship Diagram* dari basis data yang digunakan pada aplikasi ini. *Logical ERD* aplikasi ini dapat dilihat pada Gambar 3.18.



Gambar 3.18 Logical Entity Relationship Diagram

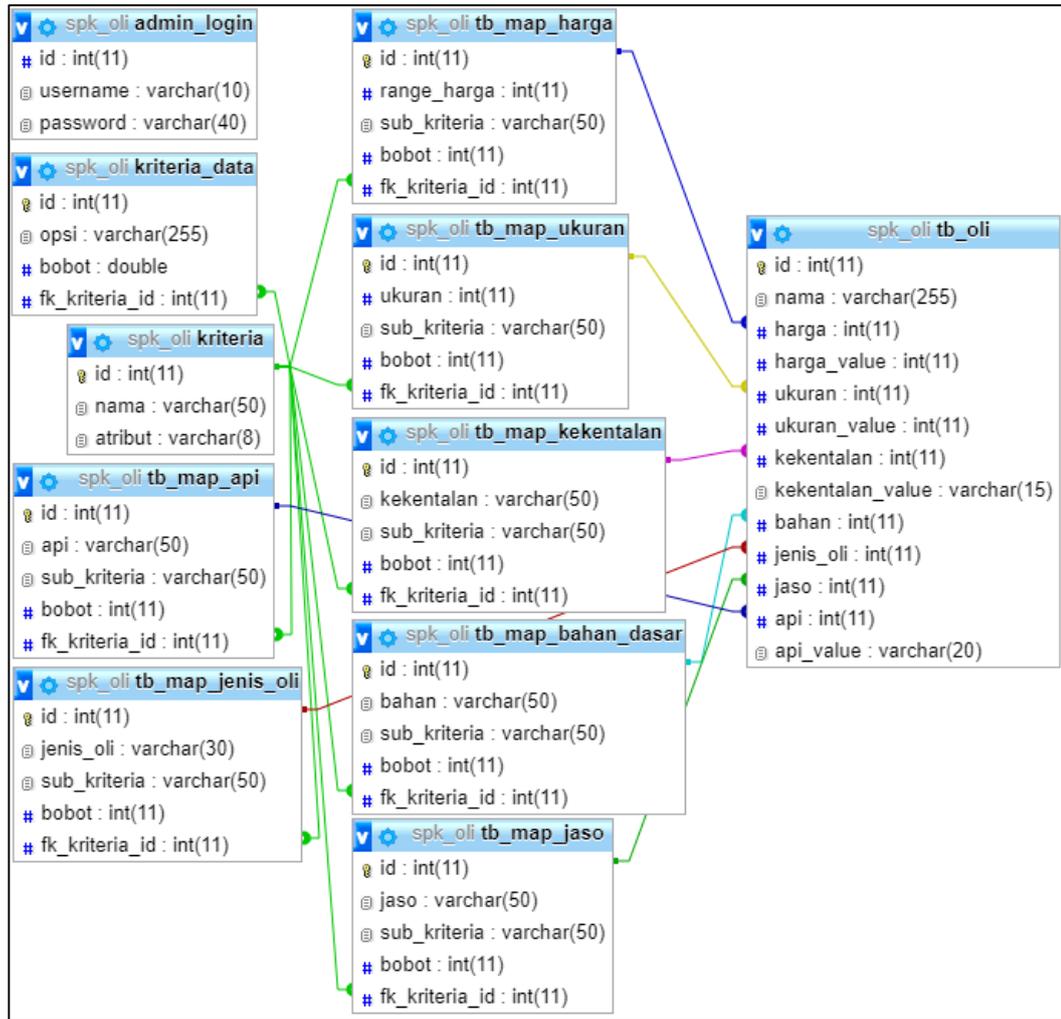
Relasi antara tabel kriteria dengan tabel kriteria\_data adalah kriteria memiliki banyak kriteria\_data dan kriteria\_data memiliki satu kriteria. Relasi antara tabel kriteria dengan tabel tb\_map\_harga adalah kriteria memiliki banyak tb\_map\_harga dan tb\_map\_harga memiliki satu kriteria. Relasi antara tabel kriteria dengan tabel tb\_map\_ukuran adalah kriteria memiliki banyak tb\_map\_ukuran dan tb\_map\_ukuran memiliki satu kriteria. Relasi antara tabel kriteria dengan tabel tb\_map\_kekentalan adalah kriteria memiliki banyak tb\_map\_kekentalan dan tb\_map\_kekentalan memiliki satu kriteria. Relasi antara tabel kriteria dengan tabel tb\_map\_bahan\_dasar adalah kriteria memiliki banyak tb\_map\_bahan\_dasar dan tb\_map\_bahan\_dasar memiliki satu kriteria.

Relasi antara tabel kriteria dengan tabel tb\_map\_jenis\_oli adalah kriteria memiliki banyak tb\_map\_jenis\_oli dan tb\_map\_jenis\_oli memiliki satu kriteria. Relasi antara tabel kriteria dengan tabel tb\_map\_jaso adalah kriteria memiliki banyak tb\_map\_jaso dan tb\_map\_jaso memiliki satu kriteria. Relasi antara tabel kriteria dengan tabel tb\_map\_api adalah kriteria memiliki banyak tb\_map\_api dan tb\_map\_api memiliki satu kriteria. Relasi antara tabel tb\_oli dengan tabel tb\_map\_harga adalah tb\_oli memiliki satu tb\_map\_harga dan tb\_map\_harga memiliki banyak tb\_oli. Relasi antara tabel tb\_oli dengan tabel tb\_map\_ukuran adalah tb\_oli memiliki satu tb\_map\_ukuran dan tb\_map\_ukuran memiliki banyak tb\_oli.

Relasi antara tabel tb\_oli dengan tabel tb\_map\_kekentalan adalah tb\_oli memiliki satu tb\_map\_kekentalan dan tb\_map\_kekentalan memiliki banyak tb\_oli. Relasi antara tabel tb\_oli dengan tabel tb\_map\_bahan\_dasar adalah tb\_oli memiliki satu tb\_map\_bahan\_dasar dan tb\_map\_bahan\_dasar memiliki banyak tb\_oli. Relasi antara tabel tb\_oli dengan tabel tb\_map\_jenis\_oli adalah tb\_oli memiliki satu tb\_map\_jenis\_oli dan tb\_map\_jenis\_oli memiliki banyak tb\_oli. Relasi antara tabel tb\_oli dengan tabel tb\_map\_jaso adalah tb\_oli memiliki satu tb\_map\_jaso dan tb\_map\_jaso memiliki banyak tb\_oli. Relasi antara tabel tb\_oli dengan tabel tb\_map\_api adalah tb\_oli memiliki satu tb\_map\_api dan tb\_map\_api memiliki banyak tb\_oli.

#### **3.2.4 Database Schema**

Pada bagian ini akan dijabarkan skema basis data yang digunakan dalam aplikasi ini. Skema basis data dapat dilihat pada Gambar 3.19.



Gambar 3.19 Database Schema

Terdapat 11 tabel pada skema basis data yang digunakan pada aplikasi ini, dimana 1 tabel tidak memiliki relasi dan 10 lainnya memiliki relasi satu sama lain. Tabel yang tidak memiliki relasi, yaitu tabel admin\_login. Tabel yang memiliki relasi, yaitu tabel kriteria, kriteria\_data, tb\_map\_harga, tb\_map\_ukuran, tb\_map\_kekentalan, tb\_map\_bahan\_dasar, tb\_map\_jenis\_oli, tb\_map\_jaso, tb\_map\_api, dan tb\_oli.

### 3.2.5 Struktur Tabel

Aplikasi ini menggunakan basis data MySQL. Terdapat 11 tabel, yaitu tabel admin\_login, kriteria, kriteria\_data, tb\_map\_harga, tb\_map\_ukuran,

tb\_map\_kekentalan, tb\_map\_bahan\_dasar, tb\_map\_jenis\_oli, tb\_map\_jaso, tb\_map\_api, dan tb\_oli. Struktur tabel Admin\_login dapat dilihat pada Tabel 3.1.

Tabel 3.1 Struktur Tabel *Admin\_login*

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
Id	Int(11)	Primary key
Username	Varchar(10)	Username untuk masuk admin
Password	Varchar(20)	Password untuk admin

Tabel *admin\_login* digunakan untuk menyimpan data *username* dan *password admin*. Struktur tabel kriteria dapat dilihat pada Tabel 3.2.

Tabel 3.2 Struktur Tabel Kriteria

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
Id	Int(11)	Primary key
Nama	Varchar(25)	Nama kriteria
Atribut	Varchar(8)	Atribut <i>benefit / cost</i>

Tabel kriteria digunakan untuk menyimpan data kriteria dan atributnya. Struktur tabel kriteria\_data dapat dilihat pada Tabel 3.3.

Tabel 3.3 Struktur Tabel Kriteria\_data

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
Id	Int(11)	Primary key
Opsi	Varchar(25)	Nama Sub kriteria
Bobot	Double	Bobot sub kriteria
Fk_kriteria_id	Int(11)	Foreign key ke field id tabel kriteria

Tabel kriteria\_data digunakan untuk menyimpan data sub kriteria dan bobotnya dari masing-masing kriteria. Struktur tabel tb\_map\_harga dapat dilihat pada Tabel 3.4.

Tabel 3.4 Struktur Tabel Tb\_map\_harga

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
Id	Int(11)	Primary key
Range_harga	Int(11)	Jarak harga oli
Sub_kriteria	Varchar(25)	Nama bilangan fuzzy kriteria harga
Bobot	Int(11)	Bobot bilangan fuzzy
Fk_kriteria_id	Int(11)	Foreign key ke field id tabel kriteria

Tabel *tb\_map\_harga* digunakan untuk menyimpan data nama bilangan *fuzzy* dan bobotnya dari kriteria harga sesuai jarak harga. Struktur tabel *tb\_map\_ukuran* dapat dilihat pada Tabel 3.5.

Tabel 3.5 Struktur Tabel *Tb\_map\_ukuran*

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
Id	Int(11)	Primary key
Ukuran	Int(11)	Ukuran volume oli (dalam milliliter)
Sub_kriteria	Varchar(25)	Bilangan fuzzy kriteria ukuran
Bobot	Int(11)	Bobot bilangan fuzzy
Fk_kriteria_id	Int(11)	Foreign key ke field id tabel kriteria

Tabel *tb\_map\_ukuran* digunakan untuk menyimpan data nama bilangan *fuzzy* dan bobotnya dari kriteria ukuran sesuai volume oli. Struktur tabel *tb\_map\_kekentalan* dapat dilihat pada Tabel 3.6.

Tabel 3.6 Struktur Tabel *Tb\_map\_kekentalan*

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
Id	Int(11)	Primary key
Kekentalan	Varchar(25)	Tingkat kekentalan oli
Sub_kriteria	Varchar(25)	Bilangan fuzzy kriteria kekentalan
Bobot	Int(11)	Bobot bilangan fuzzy
Fk_kriteria_id	Int(11)	Foreign key ke field id tabel kriteria

Tabel *tb\_map\_kekentalan* digunakan untuk menyimpan data nama bilangan *fuzzy* dan bobotnya dari kriteria kekentalan sesuai tingkat kekentalan oli. Struktur tabel *tb\_map\_bahan\_dasar* dapat dilihat pada Tabel 3.7.

Tabel 3.7 Struktur Tabel *Tb\_map\_bahan\_dasar*

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
Id	Int(11)	Primary key
Bahan	Varchar(25)	Bahan dasar (base oil) oli
Sub_kriteria	Varchar(25)	Bilangan fuzzy kriteria bahan dasar
Bobot	Int(11)	Bobot bilangan fuzzy
Fk_kriteria_id	Int(11)	Foreign key ke field id tabel kriteria

Tabel *tb\_map\_bahan\_dasar* digunakan untuk menyimpan data nama bilangan *fuzzy* dan bobotnya dari kriteria bahan dasar sesuai bahan dasar oli. Struktur tabel *tb\_map\_jenis\_oli* dapat dilihat pada Tabel 3.8.

Tabel 3.8 Struktur Tabel *Tb\_map\_jenis\_oli*

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
Id	Int(11)	Primary key
Jenis_oli	Varchar(25)	Nama jenis oli
Sub_kriteria	Varchar(25)	Bilangan fuzzy kriteria jenis oli
Bobot	Int(11)	Bobot bilangan fuzzy
Fk_kriteria_id	Int(11)	Foreign key ke field id tabel kriteria

Tabel *tb\_map\_jenis\_oli* digunakan untuk menyimpan data nama bilangan *fuzzy* dan bobotnya dari kriteria jenis oli sesuai jenis oli. Struktur tabel *tb\_map\_jaso* dapat dilihat pada Tabel 3.9.

Tabel 3.9 Struktur Tabel *Tb\_map\_jaso*

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
Id	Int(11)	Primary key
Jaso	Varchar(25)	Nama sertifikasi jaso oli
Sub_kriteria	Varchar(25)	Bilangan fuzzy kriteria ukuran
Bobot	Int(11)	Bobot bilangan fuzzy
Fk_kriteria_id	Int(11)	Foreign key ke field id tabel kriteria

Tabel *tb\_map\_jaso* digunakan untuk menyimpan data nama bilangan *fuzzy* dan bobotnya dari kriteria sertifikasi JASO sesuai nama sertifikasi JASO oli. Struktur tabel *tb\_map\_api* dapat dilihat pada Tabel 3.10.

Tabel 3.10 Struktur Tabel *Tb\_map\_api*

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
Id	Int(11)	Primary key
Jaso	Varchar(25)	Nama sertifikasi jaso oli
Sub_kriteria	Varchar(25)	Bilangan fuzzy kriteria ukuran
Bobot	Int(11)	Bobot bilangan fuzzy
Fk_kriteria_id	Int(11)	Foreign key ke field id tabel kriteria

Tabel *tb\_map\_api* digunakan untuk menyimpan data nama bilangan *fuzzy* dan bobotnya dari kriteria sertifikasi API sesuai nama sertifikasi API oli. Struktur tabel *tb\_oli* dapat dilihat pada Tabel 3.11.

Tabel 3.11 Struktur Tabel *Tb\_oli*

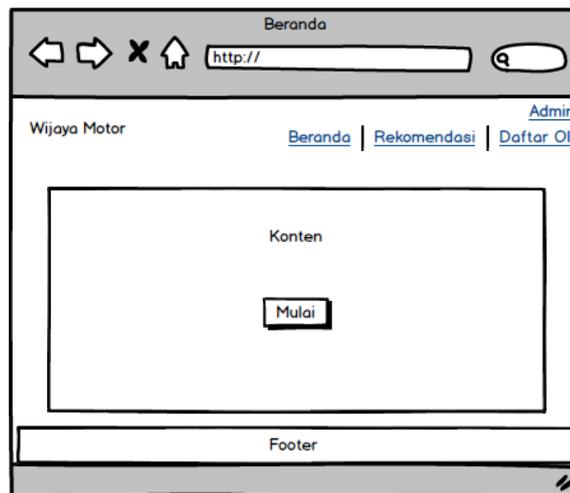
<b>Nama Field</b>	<b>Tipe Data</b>	<b>Keterangan</b>
Id	Int(11)	Primary key
Nama	Varchar(255)	Nama oli
Harga	Int(11)	Foreign key ke field id tabel <i>tb_map_harga</i>
Harga_value	Int(11)	Harga oli
Ukuran	Int(11)	Foreign key ke field id tabel <i>tb_map_ukuran</i>
Ukuran_value	Int(11)	Ukuran volume oli (dalam milliliter)
Kekentalan	Int(11)	Foreign key ke field id tabel <i>tb_map_kekentalan</i>
Kekentalan_value	Varchar(25)	Tingkat kekentalan oli
Bahan	Int(11)	Foreign key ke field id tabel <i>tb_map_bahan_dasar</i>
Jenis_oli	Int(11)	Foreign key ke field id tabel <i>tb_map_jenis_oli</i>
Jaso	Int(11)	Foreign key ke field id tabel <i>tb_map_jaso</i>
Api	Int(11)	Foreign key ke field id tabel <i>tb_map_api</i>
Api_value	Varchar(25)	Sertifikasi API oli

Tabel *tb\_oli* digunakan untuk menyimpan data alternatif/ oli mesin sepeda motor yang dijual oleh bengkel Wijaya Motor.

### 3.2.6 Rancangan Tampilan Antar Muka

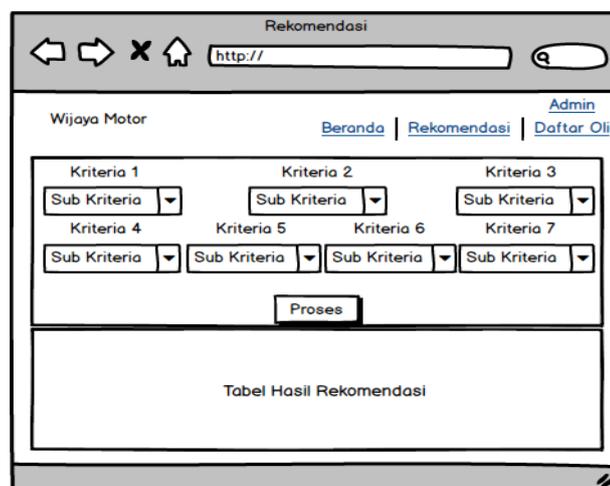
Perancangan tampilan antar muka aplikasi ini terdiri dari antar muka untuk *user* dan antar muka untuk *admin*. Rancangan antar muka untuk *user* terdiri dari beberapa rancangan, yaitu rancangan antar muka Beranda, rancangan antar muka Rekomendasi, dan rancangan antar muka Daftar Oli. Rancangan antar muka untuk *admin* sama seperti milik *user* dan ditambahkan beberapa rancangan antar muka

khusus *admin*, yaitu rancangan antar muka Masuk *Admin*, rancangan antar muka *Admin* Alternatif, rancangan antar muka *Admin* Kriteria, rancangan antar muka *Admin Mapping*, dan rancangan antar muka *Admin* Perhitungan. Rancangan antar muka Beranda dapat dilihat pada Gambar 3.20.



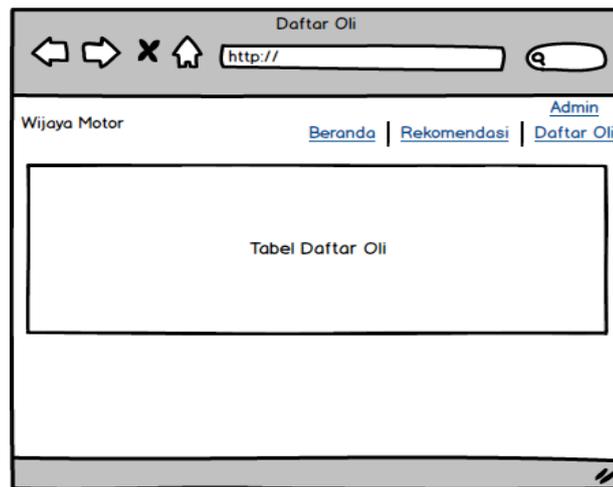
Gambar 3.20 Rancangan Antar Muka Beranda

Pada bagian atas terdapat *header* yang berisi logo bengkel dan navigasi untuk berpindah halaman. Pada bagian tengah terdapat konten halaman Beranda dan tombol Mulai untuk pindah ke halaman Rekomendasi. Pada bagian tengah terdapat *footer*. Rancangan antar muka Rekomendasi dapat dilihat pada Gambar 3.21.



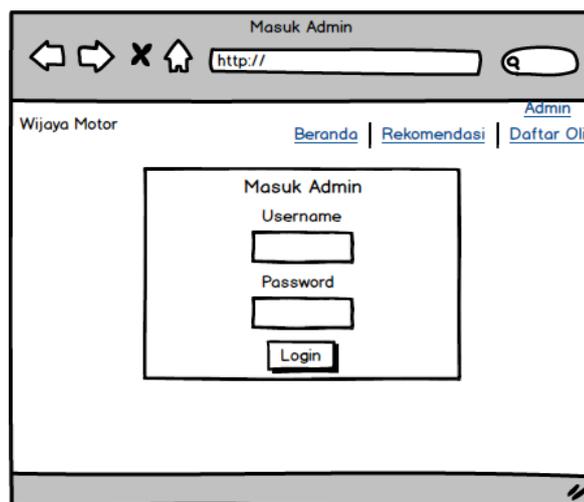
Gambar 3.21 Rancangan Antar Muka Rekomendasi

Pada bagian atas terdapat *header* yang berisi logo bengkel dan navigasi untuk berpindah halaman. Pada bagian tengah terdapat pilihan kriteria dan tombol Proses. Jika tombol proses ditekan maka akan muncul hasil rekomendasi oli berupa tabel. Rancangan antar muka Daftar Oli dapat dilihat pada Gambar 3.22.



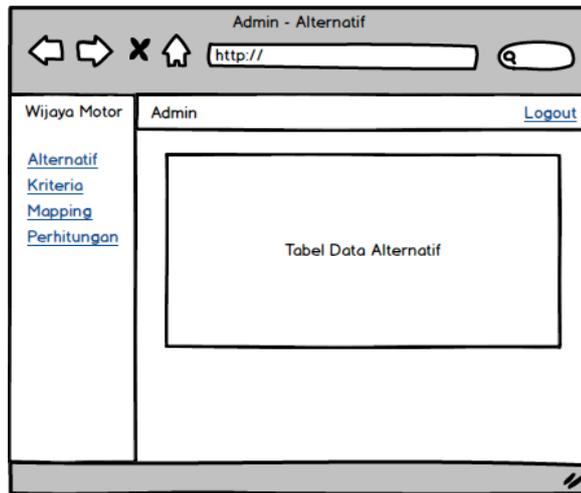
Gambar 3.22 Rancangan Antar Muka Daftar Oli

Pada bagian atas terdapat *header* yang berisi logo bengkel dan navigasi untuk berpindah halaman. Pada bagian tengah terdapat daftar oli mesin yang dijual bengkel Wijaya Motor berupa tabel. Rancangan antar muka Masuk Admin dapat dilihat pada Gambar 3.23.



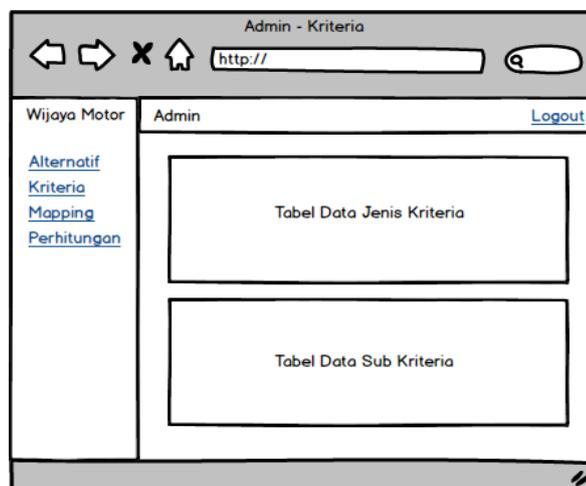
Gambar 3.23 Rancangan Antar Muka Masuk Admin

Pada bagian atas terdapat *header* yang berisi logo bengkel dan navigasi untuk berpindah halaman. Pada bagian tengah terdapat sebuah tombol *login* dan dua buah *input field* yang akan digunakan untuk memasukkan *username* dan *password*. Rancangan antar muka *Admin* Alternatif dapat dilihat pada Gambar 3.24.



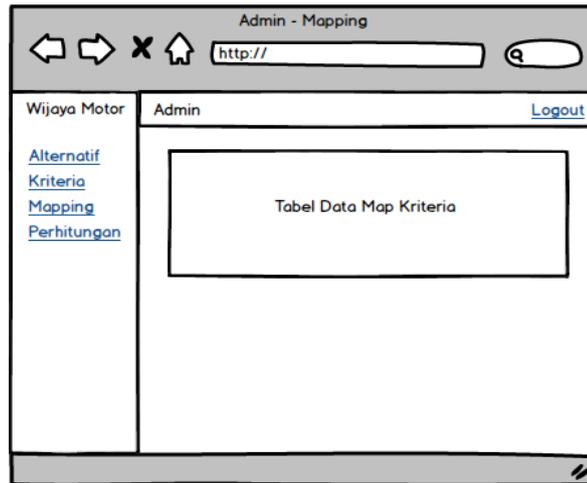
Gambar 3.24 Rancangan Antar Muka *Admin* Alternatif

Pada bagian atas terdapat *top bar* menu yang berisi tulisan *Admin* dan tombol *Logout*. Pada bagian tengah kiri terdapat *side bar* menu yang berisi navigasi untuk berpindah halaman. Pada bagian tengah kanan terdapat data daftar oli (alternatif) berupa tabel. Rancangan antar muka *Admin* Kriteria dapat dilihat pada Gambar 3.25.



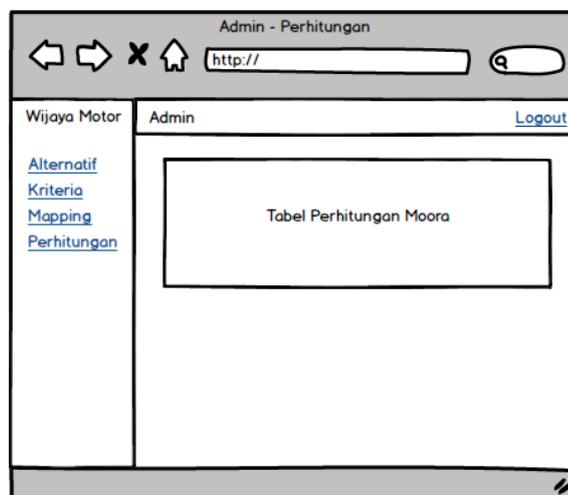
Gambar 3.25 Rancangan Antar Muka *Admin* Kriteria

Pada bagian atas terdapat *top bar* menu yang berisi tulisan *Admin* dan tombol *Logout*. Pada bagian tengah kiri terdapat *side bar* menu yang berisi navigasi untuk berpindah halaman. Pada bagian tengah kanan terdapat data kriteria dan sub kriteria berupa tabel. Rancangan antar muka *Admin Mapping* dapat dilihat pada Gambar 3.26.



Gambar 3.26 Rancangan Antar Muka *Admin Mapping*

Pada bagian atas terdapat *top bar* menu yang berisi tulisan *Admin* dan tombol *Logout*. Pada bagian tengah kiri terdapat *side bar* menu yang berisi navigasi untuk berpindah halaman. Pada bagian tengah kanan terdapat tabel data *map* kriteria. Rancangan antar muka *Admin Perhitungan* dapat dilihat pada Gambar 3.27.



Gambar 3.27 Rancangan Antar Muka *Admin Perhitungan*

Pada bagian atas terdapat *top bar* menu yang berisi tulisan *Admin* dan tombol *Logout*. Pada bagian tengah kiri terdapat *side bar* menu yang berisi navigasi untuk berpindah halaman. Pada bagian tengah kanan terdapat data perhitungan hasil rekomendasi berupa tabel.