



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, 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 DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Studi Literatur

Pada tahap ini, dicari referensi teori tentang kamus, kamus digital, obat, android, algoritma Levenshtein Distance, dan metode *Likert Scale* yang sesuai dengan kasus yang akan diteliti. Referensi dapat dicari melalui buku, jurnal, artikel, dan situs-situs di internet. Tahap ini dilakukan untuk memperkuat penyelesaian masalah dan sebagai dasar teori dalam melakukan penelitian.

2. Pengumpulan Data

Pada tahap ini, dikumpulkan data yang dibutuhkan dalam pembangunan aplikasi, salah satunya data obat tentang indikasi, dosis, komposisi, jenis obat, pemberian obat, kontra indikasi, efek samping, interaksi obat, peringatan, dan *brand* obat. Data yang digunakan pada penelitian ini berasal dari <http://www.apotikantar.com>.

3. Perancangan Aplikasi

Pada tahap ini, dirancang aplikasi yang dilakukan dengan pembuatan diagram *flowchart*, *data flow diagram*, *entity relationship diagram*, struktur tabel, dan rancangan antarmuka guna memahami dan mendesain alur kerja dari aplikasi yang akan dibangun, agar proses pembangunan aplikasi dapat berjalan sesuai dengan rencana.

4. Pembuatan Program

Pada tahap ini dilakukan proses pembuatan program yang berdasarkan tujuan dan kegunaan aplikasi. Pembuatan program meliputi tampilan antarmuka, *database*, dan *coding* program secara keseluruhan.

5. Testing dan Perbaikan Program

Pada tahap ini, dilakukan proses percobaan pada program yang telah dibuat pada tahap sebelumnya. Tahap ini dilakukan untuk mengetahui *error* atau *bug* dan evaluasi pada tahap selanjutnya.

6. Evaluasi

Evaluasi dilakukan berdasarkan hasil dari kuesioner yang telah dibagikan. Hasil evaluasi diperoleh berdasarkan faktor *functionality*, *user satification*, dan *user experience* yang telah dijelaskan pada studi literatur.

7. Penulisan Skripsi

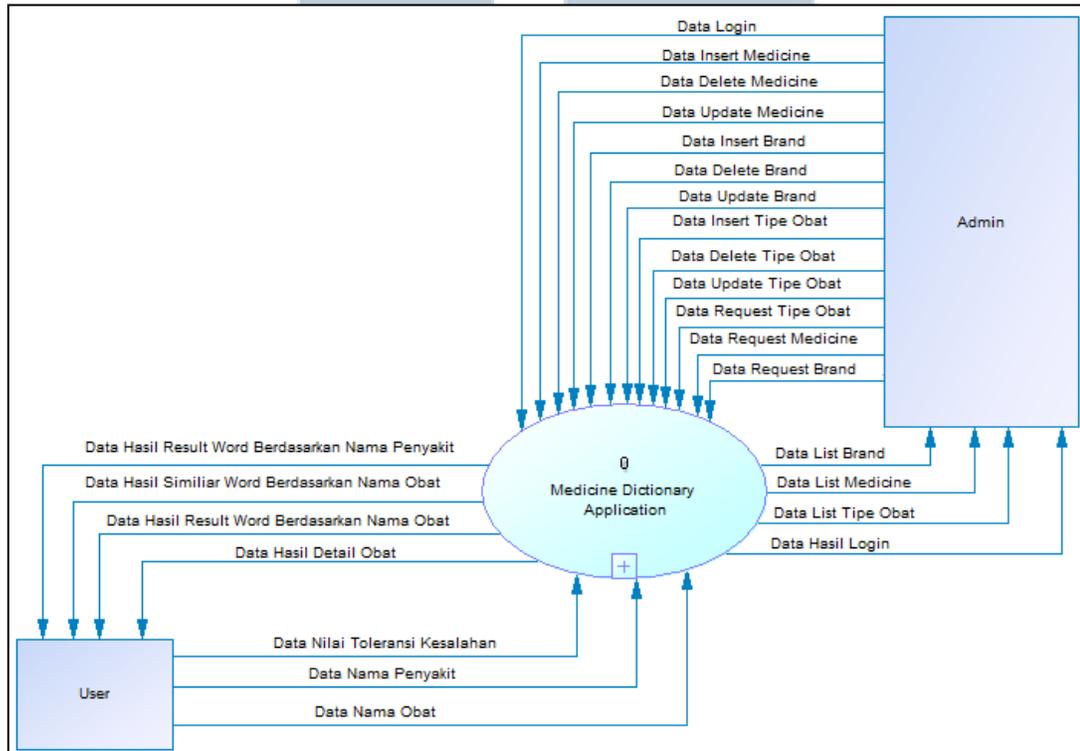
Pada tahap ini, dilakukan penulisan laporan skripsi sebagai dokumentasi.

3.2 Perancangan Aplikasi

Perancangan aplikasi yang dibuat dalam penelitian ini adalah *flowchart diagram* yang menggambarkan alur proses aplikasi, *Data Flow Diagram* yang menggambarkan aliran data dari suatu proses ke proses lain dalam aplikasi, *Entity Relationship Diagram* yang menggambarkan hubungan antar tabel yang dibutuhkan dalam aplikasi, struktur tabel yang menjelaskan detail dari setiap tabel yang

digunakan dalam aplikasi, dan rancangan antarmuka yang menggambarkan secara kasar tampilan dari aplikasi yang dibangun.

3.2.1 Data Flow Diagram

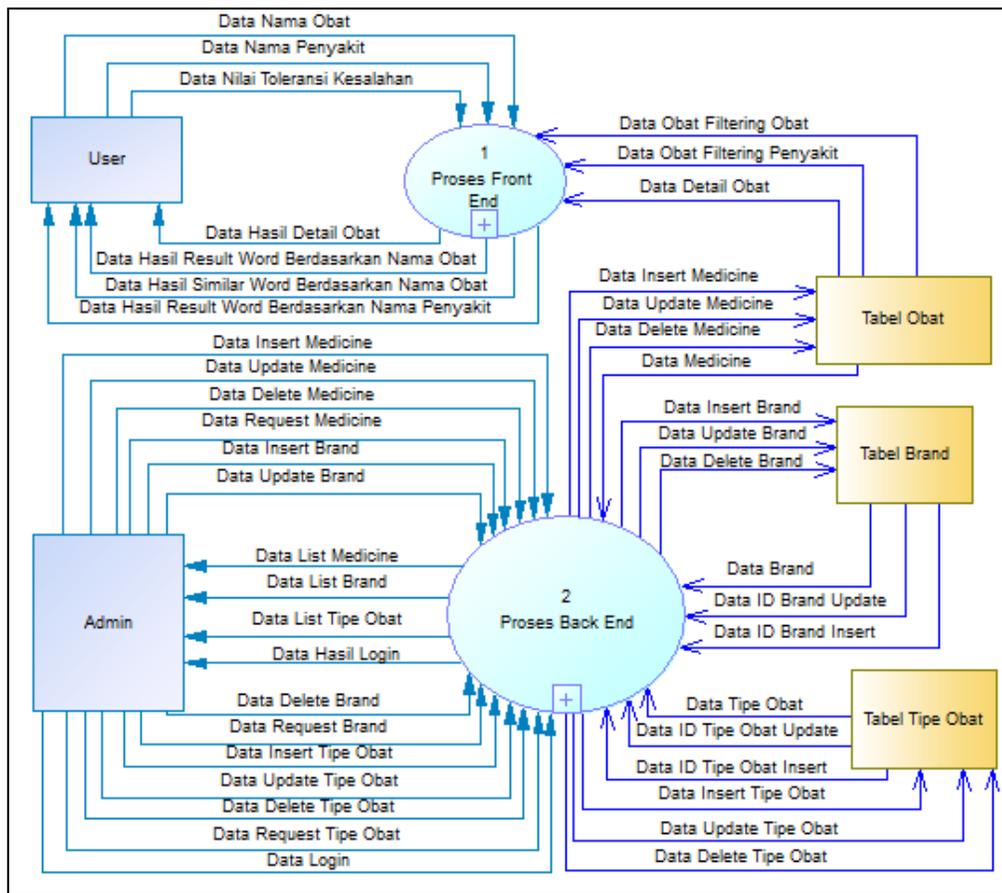


Gambar 3.1 Context Diagram

Gambar 3.1 merupakan *context diagram* untuk *medicine dictionary application*. Terdapat dua entitas yang menggambarkan pengguna sistem, yaitu *user* dan *admin*. Entitas *user* dapat memberikan data nama obat, data nama penyakit, dan data nilai toleransi kesalahan. Data nama obat dan nama penyakit berupa *string* yang akan digunakan untuk mencari nama obat ke dalam *database* sedangkan data nilai toleransi kesalahan berupa *float* yang akan dihitung pada proses *filtering* nama obat, dan kemudian akan digunakan untuk melengkapi perhitungan pada algoritma Levenshtein Distance. Entitas *user* akan menerima data hasil *result word* berdasarkan

nama obat, data hasil *similar word* berdasarkan nama obat, data hasil *result word* berdasarkan nama penyakit, dan data hasil detail obat. Data hasil *result word* berdasarkan nama obat dan data hasil *similar word* didapatkan dengan menggunakan algoritma Levenshtein Distance sedangkan data hasil detail obat berupa penjelasan obat, yaitu nama obat, indikasi, dosis, komposisi, jenis obat, pemberian obat, kontra indikasi, efek samping, interaksi obat, *brand*, peringatan, dan gambar obat tersebut. Entitas admin dapat memberikan data *login*, data *insert medicine*, data *update medicine*, data *delete medicine*, data *insert brand*, data *update brand*, data *delete brand*, data *insert tipe obat*, data *update tipe obat*, data *delete tipe obat*, data *request medicine*, data *request brand*, dan data *request request tipe obat*. Data *login* berupa string *username* dan *password* yang akan di-*response* oleh sistem dalam bentuk data hasil *login*. Data *request medicine*, data *request brand*, dan data *request tipe obat* digunakan untuk menampilkan semua data obat, *brand*, dan tipe obat sehingga sistem akan memberikan *response* berupa data *list medicine*, data *list brand*, dan data *list tipe obat*.

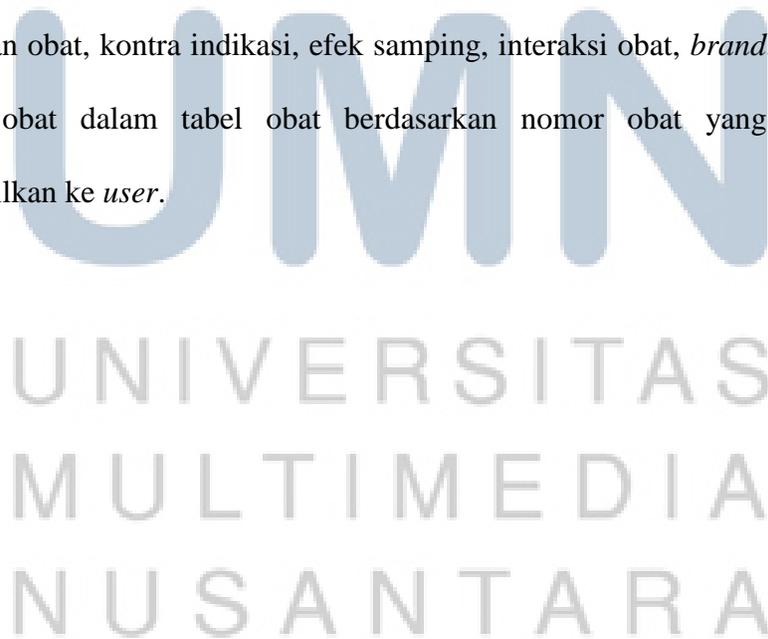


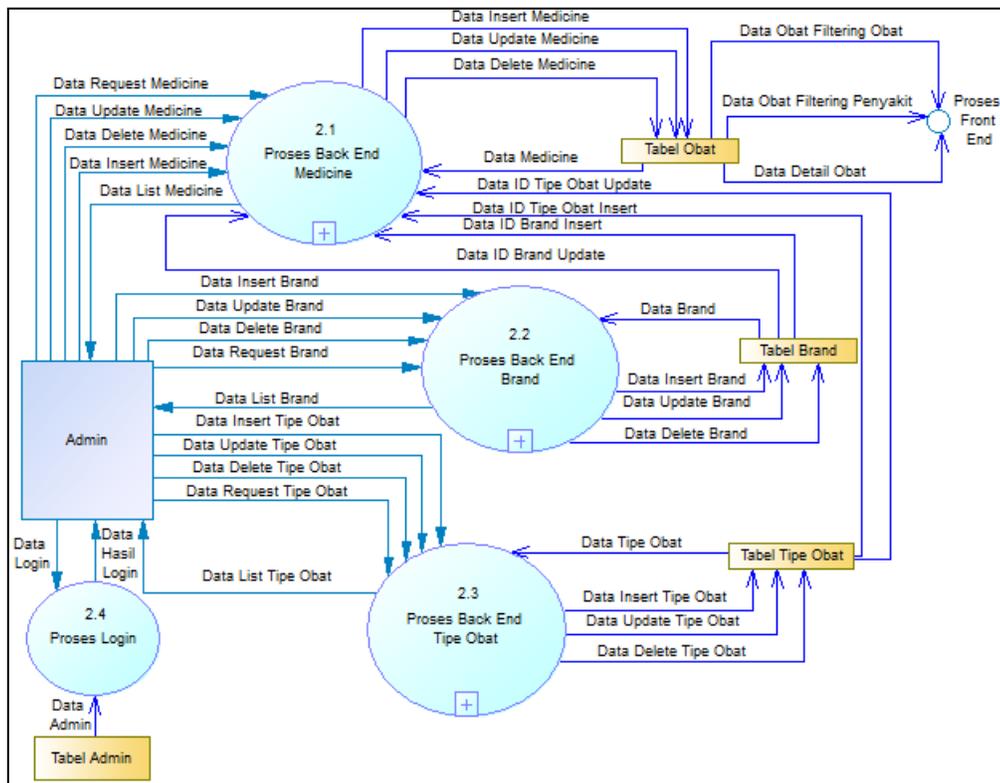


Gambar 3.2 Data Flow Diagram level 1

Data flow diagram level 1 memiliki dua buah proses, yaitu proses *front end* dan proses *back end*. Proses *front end* merupakan proses yang berinteraksi langsung dengan *user* dan menghasilkan *output* berupa hasil pencarian nama obat dan detail nama obat yang akan ditampilkan ke *user* sehingga proses *front end* hanya berhubungan dengan entitas *user*. Namun, proses *back end* merupakan proses yang mengatur dan memodifikasi data dalam *database*, seperti *insert*, *update*, *delete*. Data flow diagram level 1 menggunakan tiga tabel, yaitu tabel obat, tabel brand, dan tabel tipe obat. Setiap tabel dapat dilakukan proses modifikasi yang dilakukan oleh admin melalui proses *back end*.

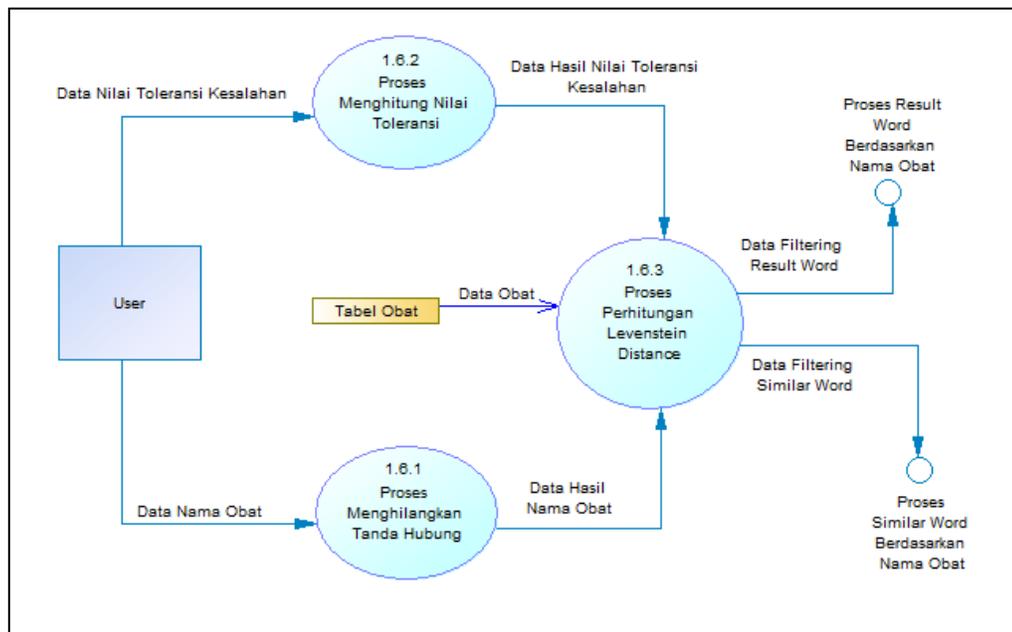
berasal dari proses *filtering* nama obat atau hasil perhitungan algoritma Levenshtein Distance kemudian memberikan daftar hasil pencarian yang sesuai dengan data nama obat yang diberikan oleh *user*. Proses *similar word* berdasarkan nama obat menerima data *filtering similar word* yang berasal dari proses *filtering* nama obat atau hasil perhitungan algoritma Levenshtein Distance, kemudian memberikan daftar hasil pencarian yang berada dalam batas nilai toleransi dari hasil perhitungan algoritma Levenshtein Distance. Proses *result word* berdasarkan nama penyakit menerima data *filtering* nama penyakit dari proses *filtering* nama penyakit dan memberikan daftar hasil pencarian yang sesuai dengan nama penyakit dalam indikasi obat. Proses *result word* berdasarkan nama obat, proses *similar word* berdasarkan nama obat, dan proses *result word* berdasarkan nama penyakit juga memberikan nomor obat yang digunakan dalam proses detail obat. Kemudian, proses detail obat akan menerima nomor obat dan mengambil data obat, seperti nama, indikasi, dosis, komposisi, jenis obat, pemberian obat, kontra indikasi, efek samping, interaksi obat, *brand*, peringatan, dan gambar obat dalam tabel obat berdasarkan nomor obat yang diberikan lalu menampilkan ke *user*.





Gambar 3.4 Data Flow Diagram level 2 Back End

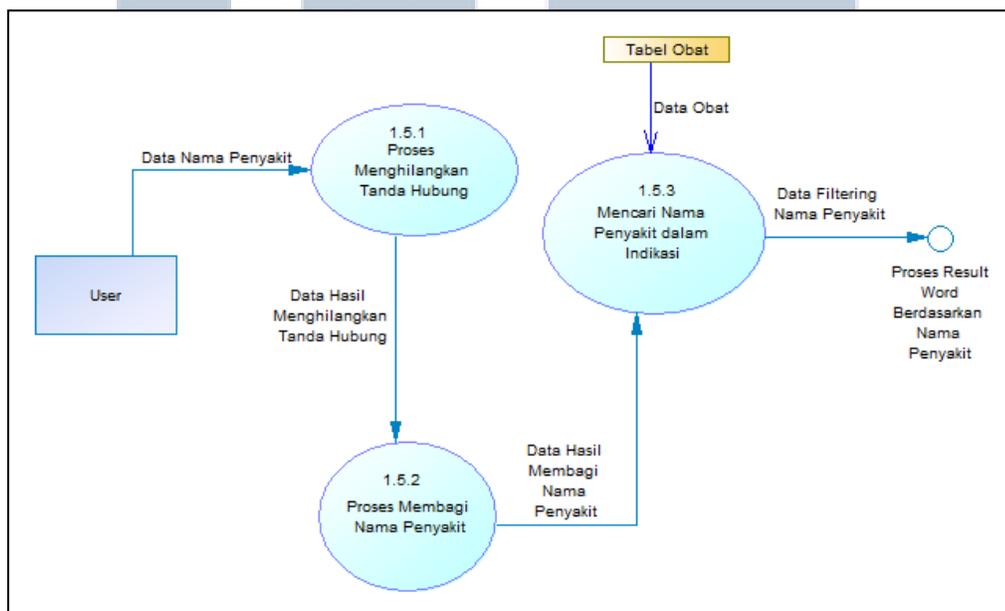
Gambar 3.4 merupakan *data flow diagram level 2 back end* yang digunakan untuk melakukan modifikasi data obat, *brand*, dan tipe obat. Terdapat empat proses, yaitu proses *login*, proses *back end medicine*, proses *back end brand*, dan proses *back end tipe obat*. Proses *back end medicine* merupakan proses yang melakukan modifikasi terhadap data obat. Proses *back end brand* merupakan proses yang melakukan modifikasi terhadap data *brand*. Proses *back end tipe obat* merupakan proses yang melakukan modifikasi terhadap data tipe obat. Selain itu, proses *login* digunakan untuk melakukan *login* sebagai *admin*.



Gambar 3.5 *Data Flow Diagram level 3* Proses *Filtering* Nama Obat

Gambar 3.5 merupakan *data flow diagram level 3* proses *filtering* nama obat yang memiliki tiga buah proses, yaitu proses menghilangkan tanda hubung, proses menghitung nilai toleransi, dan proses perhitungan Levenshtein Distance. Proses menghilangkan tanda hubung digunakan untuk meningkatkan perhitungan Levenshtein Distance karena nama obat tidak mengandung tanda hubung. Kemudian, data hasil nama obat akan diterima oleh proses perhitungan Levenshtein Distance. Proses menghitung nilai toleransi digunakan untuk mendapatkan data hasil nilai toleransi kesalahan yang bisa diterima oleh algoritma dan akan diterima oleh proses perhitungan Levenshtein Distance. Proses perhitungan Levenshtein Distance akan mengambil semua nama obat dalam tabel obat kemudian dibandingkan dengan data hasil nama obat menggunakan algoritma Levenshtein Distance dan berada dalam data hasil nilai toleransi kesalahan yang telah diterima sebelumnya. Selanjutnya proses

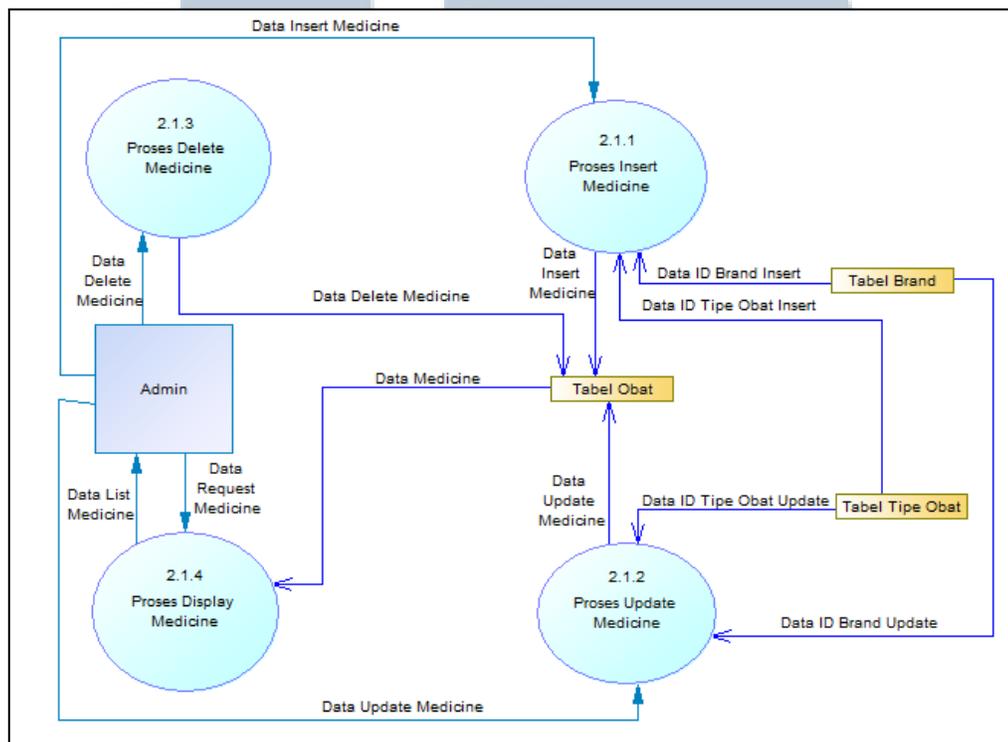
perhitungan Levenshtein Distance akan membagi hasil data obat menjadi dua jenis, yaitu data *filtering result word* dan data *filtering similar word*. Data *filtering result word* merupakan data yang sesuai atau memiliki tingkat kemiripan 100% dengan data nama obat yang dimasukkan oleh *user*, sedangkan data *filtering similar word* merupakan data yang berada dalam nilai toleransi kesalahan yang didapatkan dalam proses menghitung nilai toleransi.



Gambar 3.6 Data Flow Diagram level 3 Proses Filtering Nama Penyakit

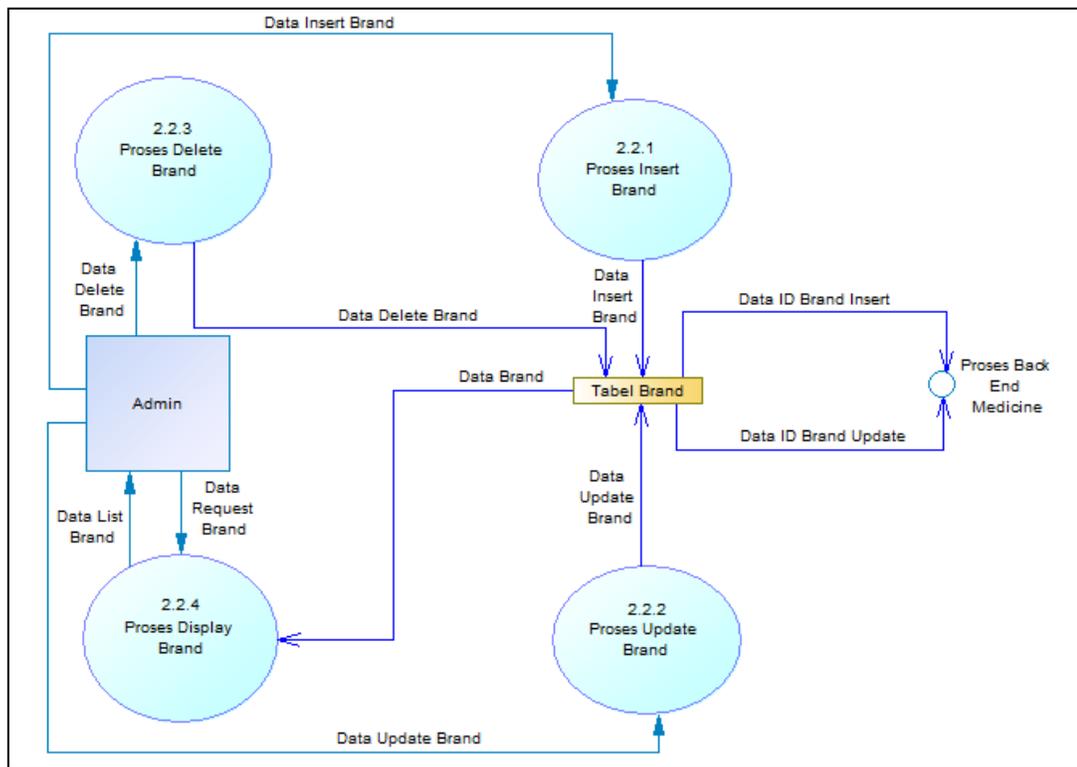
Gambar 3.6 merupakan *data flow diagram level 3* proses *filtering* nama penyakit yang memiliki tiga buah proses, yaitu proses menghilangkan tanda hubung, proses membagi nama penyakit, dan proses mencari nama penyakit dalam indikasi. Proses menghilangkan tanda hubung digunakan untuk menghilangkan tanda hubung sehingga dapat membagi nama penyakit yang dimasukkan oleh *user*. Kemudian, hasilnya akan diterima oleh proses membagi nama penyakit. Proses membagi nama penyakit digunakan untuk membuat sebuah *string query* dari hasil proses

menghilangkan tanda hubung sehingga dapat mencari satu atau lebih nama penyakit. Proses mencari nama penyakit dalam indikasi akan mengambil semua data indikasi obat dalam tabel obat dan dibandingkan dengan *string query* yang telah dibentuk dalam proses sebelumnya. Selanjutnya hasil tersebut diterima oleh proses *result word* berdasarkan nama penyakit.



Gambar 3.7 Data Flow Diagram level 3 Proses Back End Medicine

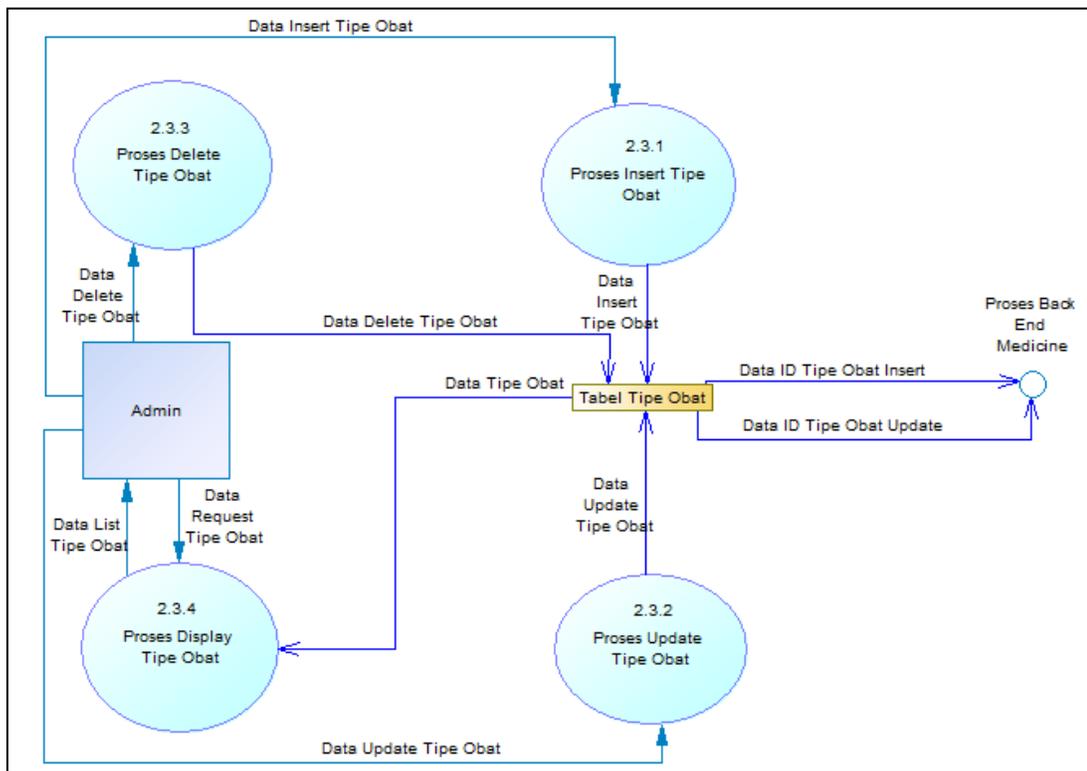
Gambar 3.7 merupakan *data flow diagram level 3* proses *back end medicine* yang digunakan *admin* untuk melakukan modifikasi data obat. Terdapat empat proses, yaitu proses *delete medicine*, proses *insert medicine*, proses *update medicine*, dan proses *display medicine*. Pada proses *insert medicine* dan proses *update medicine* akan menerima data *id brand* dan data *id tipe obat* karena akan menampilkan *dropdown list* data *brand* dan tipe obat.



Gambar 3.8 Data Flow Diagram level 3 Proses Back End Brand

Gambar 3.8 merupakan *data flow diagram level 3* proses *back end brand* yang digunakan *admin* untuk melakukan modifikasi data *brand*. Terdapat empat proses, yaitu proses *delete medicine*, proses *insert brand*, proses *update brand*, dan proses *display brand*. Tabel *brand* akan mengirimkan data *id brand insert* dan data *id brand update* ke proses *back end medicine* yang telah dijelaskan pada Gambar 3.7.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

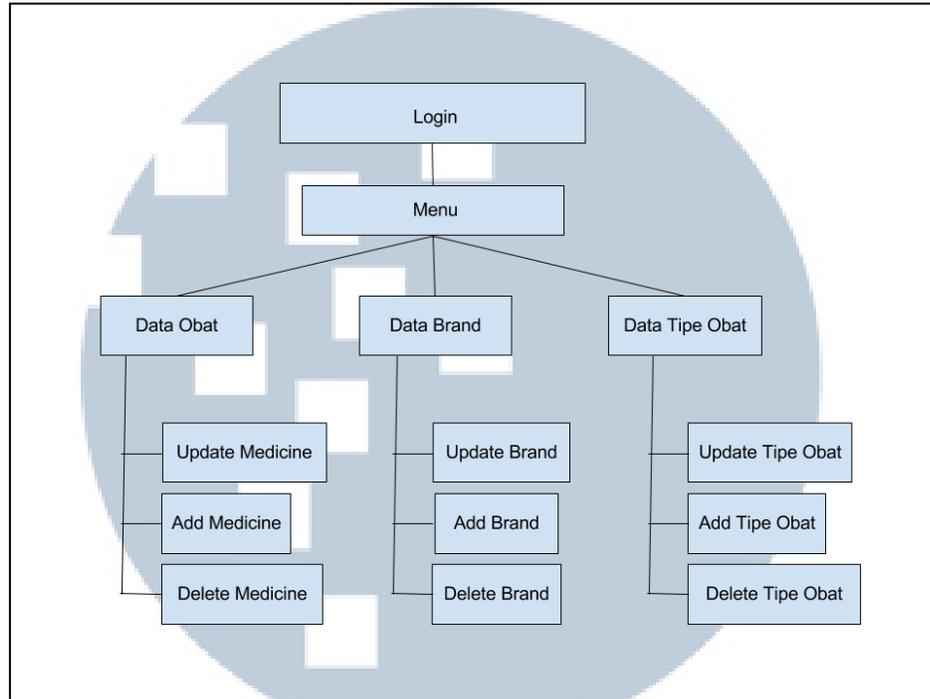


Gambar 3.9 Data Flow Diagram level 3 Proses Back End Tipe Obat

Gambar 3.9 merupakan *data flow diagram level 3* proses *back end* tipe obat yang digunakan *admin* untuk melakukan modifikasi data tipe obat. Terdapat empat proses, yaitu proses *delete* tipe obat, proses *insert* tipe obat, proses *update* tipe obat, dan proses *display* tipe obat. Tabel tipe obat akan mengirimkan data *id* tipe obat *insert* dan data *id* tipe obat *update* ke proses *back end medicine* yang telah dijelaskan pada Gambar 3.7.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2.2 Sitemap

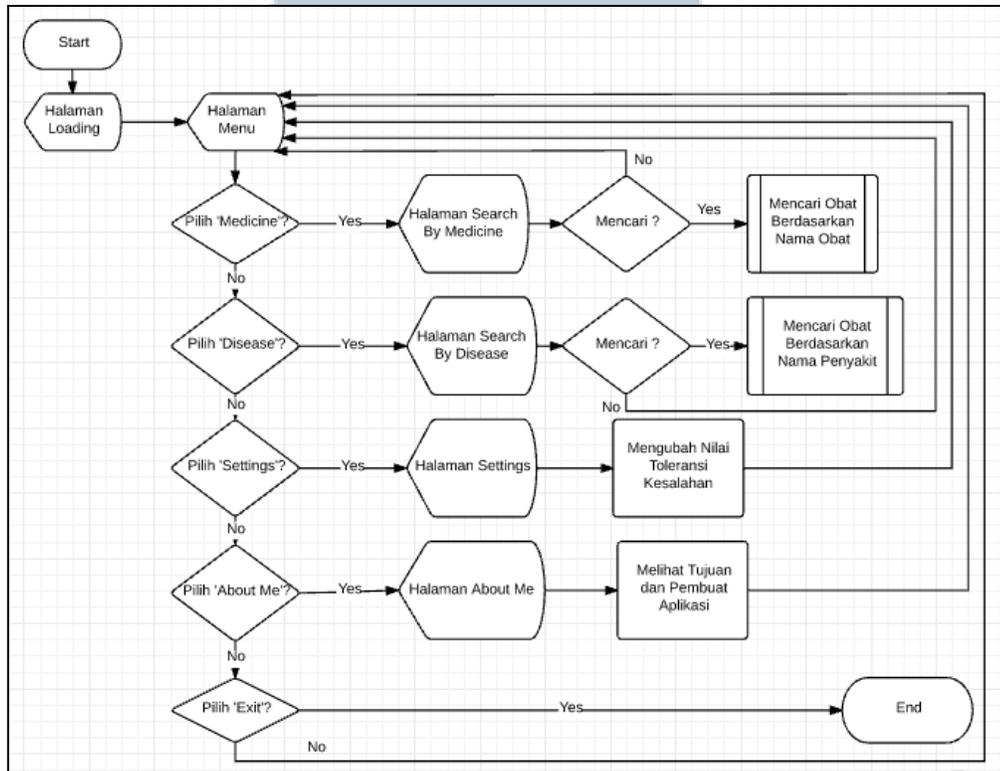


Gambar 3.10 Sitemap Backend

Gambar 3.15 merupakan *sitemap backend* yang digunakan untuk menjelaskan halaman-halaman dalam *backend*. Halaman pertama dalam *backend* adalah halaman *login* yang berfungsi untuk mengetahui *admin* yang *login*. Selanjutnya terdapat halaman menu yang menampilkan tiga tombol untuk melakukan *link* ke halaman data obat, halaman data *brand*, dan halaman data tipe obat. Setiap halaman tersebut mampu dilakukan proses *update*, *add*, dan *delete* yang akan dijelaskan pada *flowchart*.

3.2.3 Flowchart

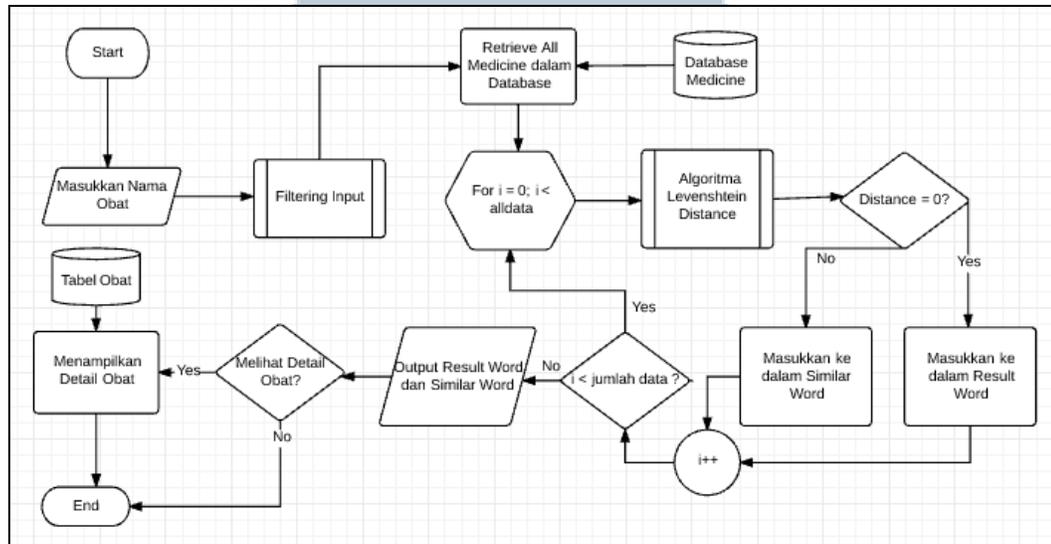
A. Flowchart Front End



Gambar 3.11 *Flowchart* Aplikasi Secara Umum

Gambar 3.11 merupakan *flowchart* aplikasi secara umum, dimulai dengan tampilan halaman *loading* lalu tampilan halaman menu yang terdapat lima buah *imagebutton*, yaitu *medicine*, *disease*, *settings*, *about me*, dan *exit*. Pada setiap *imagebutton* terdapat tampilan *interface* dan fungsi yang berbeda – beda. Pada halaman *search by medicine*, dilakukan proses mencari obat berdasarkan nama obat yang dapat dilihat pada Gambar 3.12. Pada halaman *search by disease*, dilakukan proses mencari obat berdasarkan nama penyakit yang dapat dilihat pada Gambar 3.13. Pada halaman *settings*, *user* dapat mengubah nilai toleransi kesalahan yang digunakan

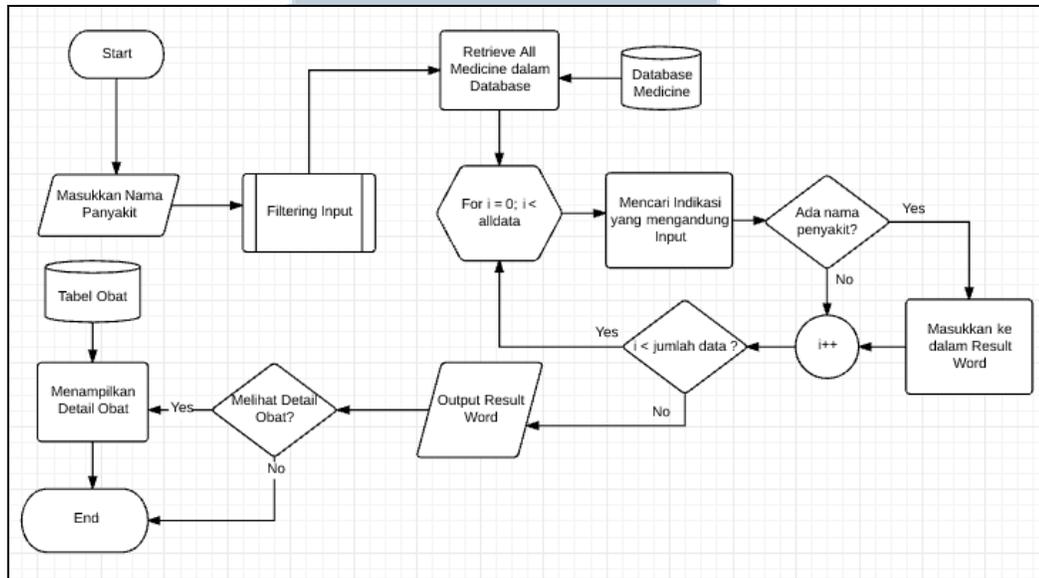
dalam perhitungan algoritma Levenshtein Distance, sedangkan halaman *about me* digunakan untuk melihat pembuat dan tujuan pembuatan aplikasi.



Gambar 3.12 *Flowchart* Mencari Obat Berdasarkan Nama Obat

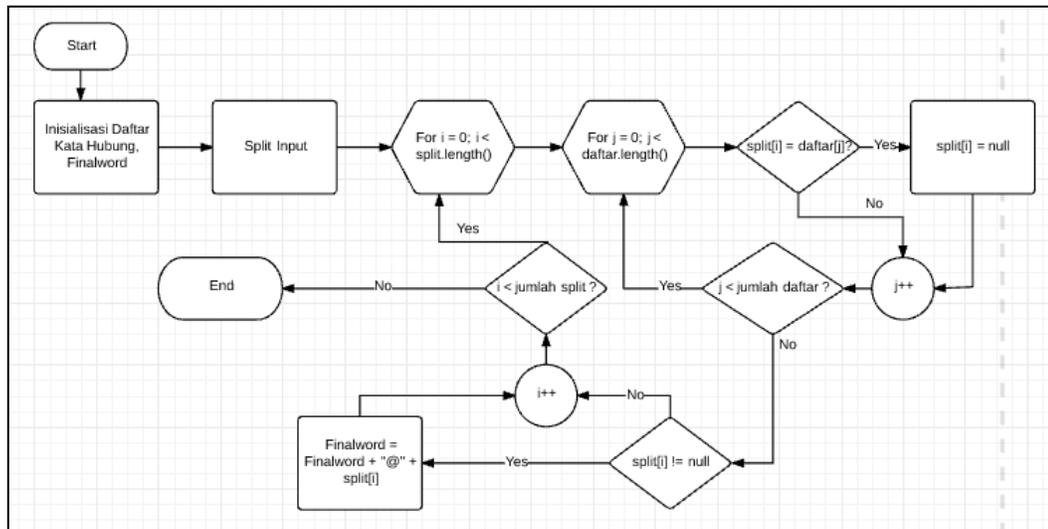
Gambar 3.12 merupakan *flowchart* mencari obat berdasarkan nama obat yang dimulai dengan memasukkan nama obat kemudian nama obat tersebut akan diproses dalam proses *filtering* input yang dapat dilihat pada Gambar 3.14. Selanjutnya aplikasi akan mengambil semua data dalam tabel obat dan membandingkan masing – masing nama obat dengan menggunakan algoritma Levenshtein Distance. Jika *distance* yang didapatkan adalah nol atau memiliki tingkat kemiripan 100% maka akan dimasukkan ke dalam daftar *result word*, sedangkan jika tidak didapatkan *distance* nol maka akan dimasukkan ke dalam daftar *similar word* yang artinya nama obat berada di bawah nilai toleransi kesalahan dan dapat diterima namun tidak mirip atau *distance* di atas nol. Kemudian setiap daftar *result word* dan *similar word* akan ditampilkan ke *user* dilanjutkan dengan melihat detail obat berupa nama, indikasi,

dosis, komposisi, jenis obat, pemberian obat, kontra indikasi, efek samping, interaksi obat, *brand*, peringatan, dan gambar obat tersebut.



Gambar 3.13 *Flowchart* Mencari Obat Berdasarkan Nama Penyakit

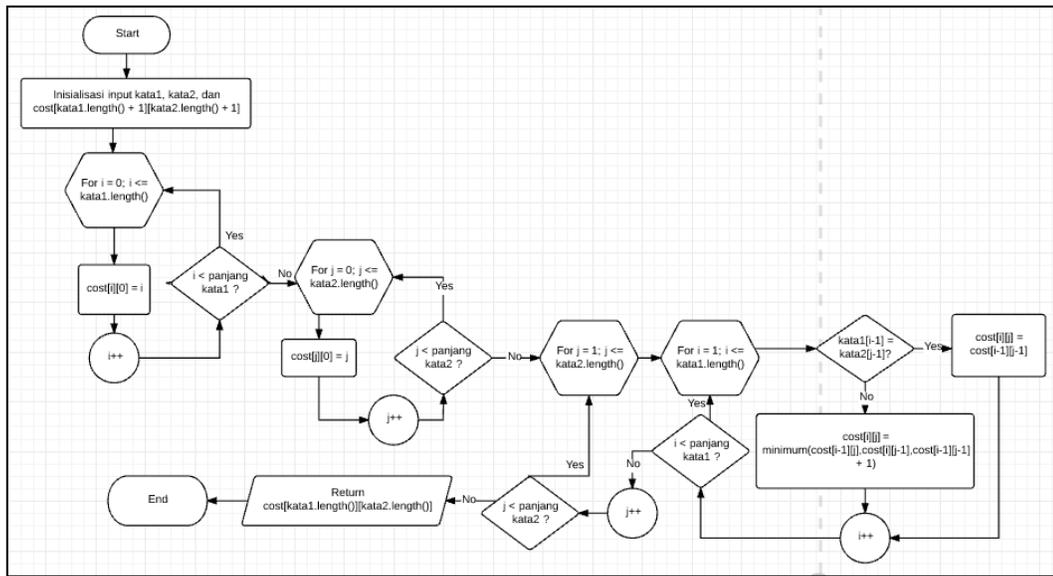
Gambar 3.13 merupakan *flowchart* mencari obat berdasarkan nama penyakit yang dimulai dengan memasukkan nama penyakit kemudian nama penyakit tersebut akan dilakukan *filtering* input yang akan dijelaskan pada Gambar 3.14. Selanjutnya proses *retrieve all medicine* dalam *database* untuk melakukan pencarian nama penyakit dalam indikasi pada setiap obat. Jika terdapat nama penyakit dalam indikasi obat tersebut maka akan dimasukkan ke dalam daftar *result word* kemudian menghasilkan *output* berupa daftar *result word*. Selanjutnya, *user* dapat melihat detail dari obat yang ditampilkan dalam daftar *result word* tersebut berupa nama, indikasi, dosis, komposisi, jenis obat, pemberian obat, kontra indikasi, efek samping, interaksi obat, *brand*, peringatan, dan gambar obat tersebut.



Gambar 3.14 *Flowchart Filtering Input*

Flowchart filtering input digunakan untuk melakukan penghapusan tanda hubung, dimulai dengan melakukan inisialisasi daftar kata hubung dan *finalword*. Daftar kata hubung tersebut berupa tanda hubung, seperti ‘-’, ‘.’, ‘ ‘, dan ‘,’ serta kata ‘dan’. Setelah melakukan inisialisasi daftar kata hubung akan dilakukan *split* input untuk mendapatkan nama-nama penyakit. Jika hasil *split* adalah kata hubung yang telah terinisialisasi sebelumnya maka *split* adalah *null*. Setelah melakukan pengecekan terhadap semua *input* nama penyakit, maka dilakukan proses seleksi untuk mengumpulkan kembali nama-nama penyakit yang telah dilakukan proses *split* sebelumnya.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

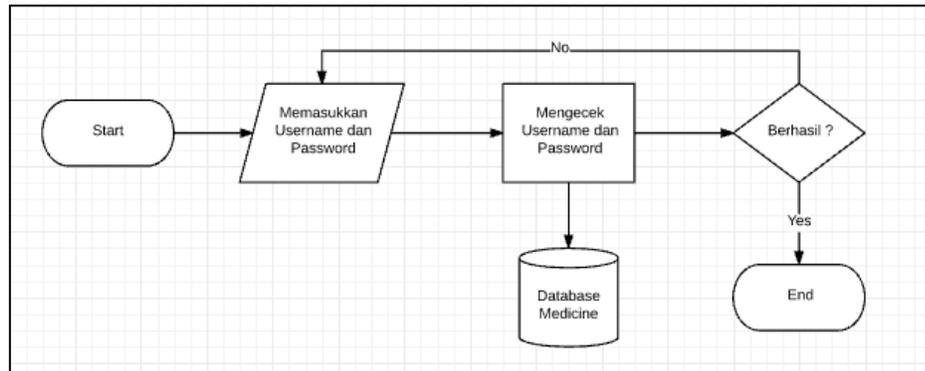


Gambar 3.15 *Flowchart* Algoritma Levenshtein Distance

Gambar 3.15 merupakan *flowchart* algoritma Levenshtein Distance yang digunakan untuk mendapatkan nilai kemiripan kata-kata yang dibandingkan berupa nilai *distance*. Algoritma Levenshtein Distance dimulai dengan melakukan inisialisasi *index* kata-kata yang akan saling dibandingkan kemudian akan membentuk kolom dan baris sesuai dengan panjang masing-masing kata yang dibandingkan. Pada setiap kolom dan baris diisi dengan nilai terkecil dari kolom dan baris sebelumnya, nilai terkecil menandakan bahwa huruf dalam kata tersebut sama yang ditulis dalam nilai *cost*. Selanjutnya hasil akhir berupa nilai *distance* yang berada dalam kolom dan baris terakhir.

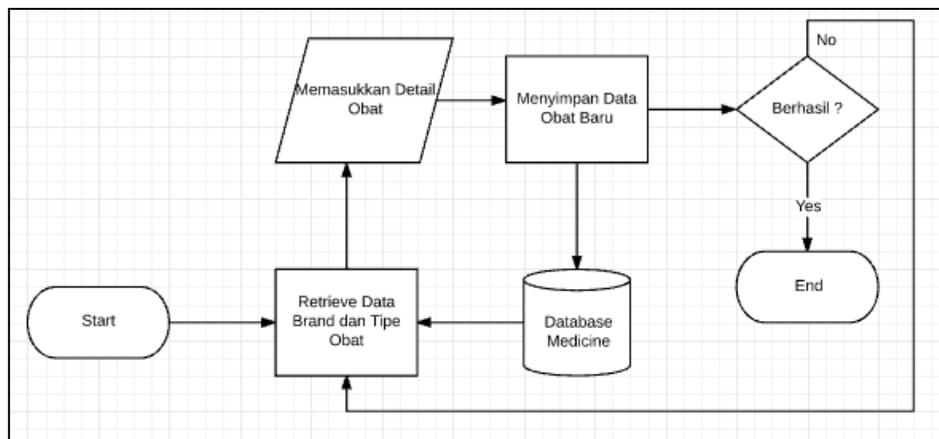
UNIVERSITAS
MULTIMEDIA
NUSANTARA

B. Flowchart Back End



Gambar 3.16 Flowchart Login Admin

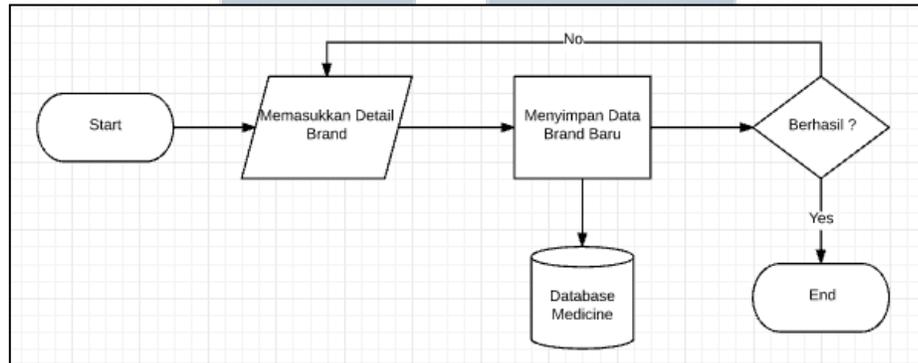
Gambar 3.16 merupakan *flowchart login admin* yang digunakan oleh *admin* untuk memasukkan *username* dan *password*. Kemudian, sistem akan melakukan pengecekan *username* dan *password*. Jika *username* dan *password* telah terdaftar dalam *database* maka *admin* dapat melakukan modifikasi data dalam *database*.



Gambar 3.17 Flowchart Insert Obat

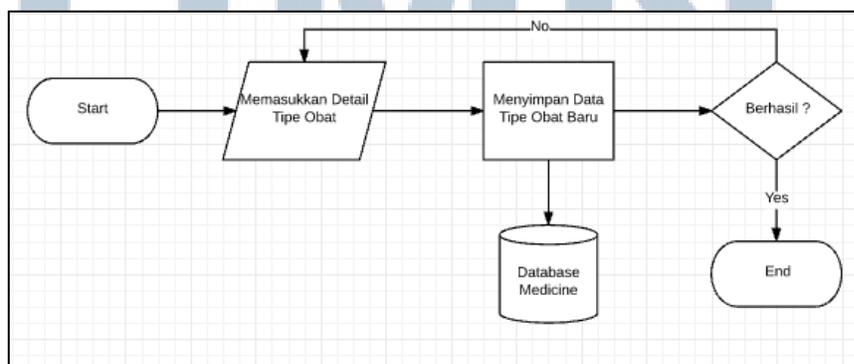
Gambar 3.17 merupakan *flowchart insert obat* yang digunakan untuk memasukkan data obat baru ke dalam *database*. Sistem akan melakukan *retrieve data brand* dan tipe obat yang akan ditampilkan dalam bentuk *dropdown list*. Kemudian,

admin akan memasukkan detail obat dan sistem akan menyimpan data obat yang baru ke dalam *database*. Jika gagal maka sistem akan melakukan *retrieve brand* dan tipe obat sehingga *admin* harus memasukkan data detail obat kembali.



Gambar 3.18 *Flowchart Insert Brand*

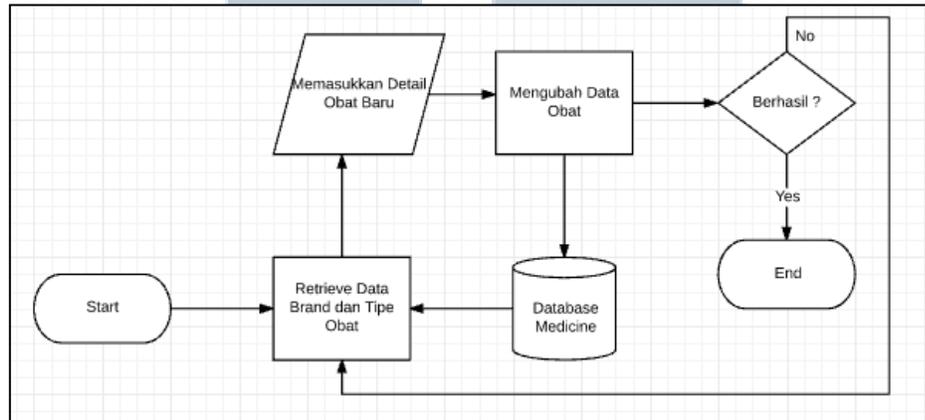
Gambar 3.18 merupakan *flowchart insert brand* yang digunakan untuk memasukkan data *brand* baru ke dalam *database*. *Admin* akan memasukkan detail *brand* berupa *brandcode* dan *brandname*. Kemudian, sistem akan menyimpan data *brand* yang baru ke dalam *database*. Jika gagal maka *admin* harus memasukkan data detail *brand* kembali.



Gambar 3.19 *Flowchart Insert Tipe Obat*

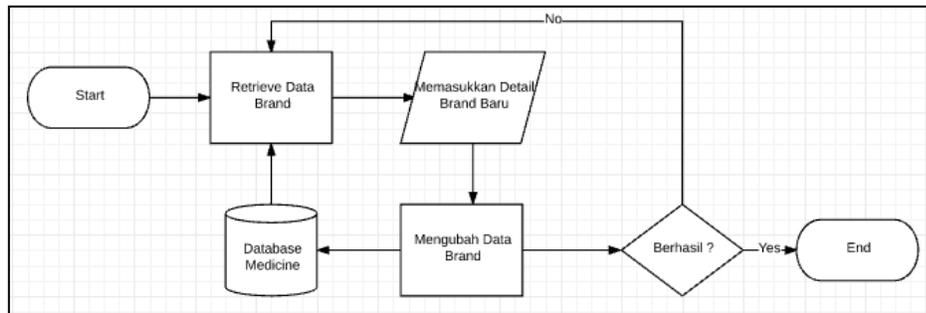
Gambar 3.19 merupakan *flowchart insert tipe obat* yang digunakan untuk memasukkan data tipe obat baru ke dalam *database*. *Admin* akan memasukkan detail

tipe obat berupa *tipeobatcode* dan *tipeobatname*. Kemudian, sistem akan menyimpan data tipe obat yang baru ke dalam *database*. Jika gagal maka *admin* harus memasukkan data detail tipe obat kembali.



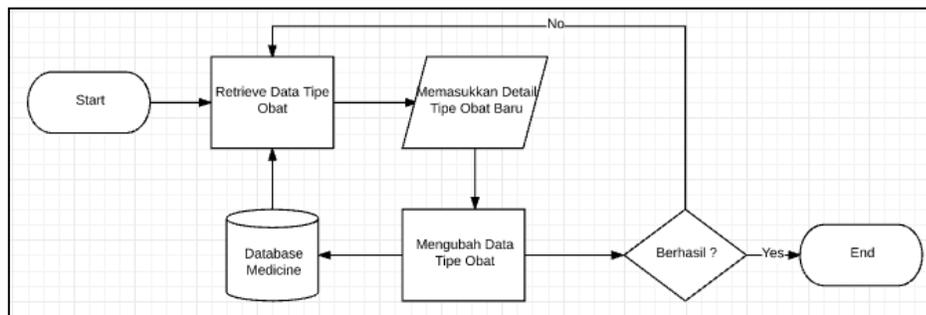
Gambar 3.20 *Flowchart Update Obat*

Gambar 3.20 merupakan *flowchart update* obat yang digunakan untuk melakukan modifikasi pada data obat dan disimpan ke dalam *database*. Sistem akan melakukan *retrieve* data detail obat, data *brand*, dan data tipe obat yang akan ditampilkan dalam bentuk *text* dan *dropdown list*. Terdapat sembilan detail obat yang akan ditampilkan dalam bentuk *text*, yaitu nama, indikasi, dosis, komposisi, pemberian obat, kontraindikasi, efek samping, interaksi obat, dan peringatan. Kemudian, *admin* akan memasukkan detail obat yang baru dan sistem akan mengubah data obat yang dipilih ke dalam *database*. Jika gagal maka sistem akan melakukan *retrieve* detail obat, *brand* dan tipe obat sehingga *admin* harus memasukkan kembali data detail obat yang baru.



Gambar 3.21 *Flowchart Update Brand*

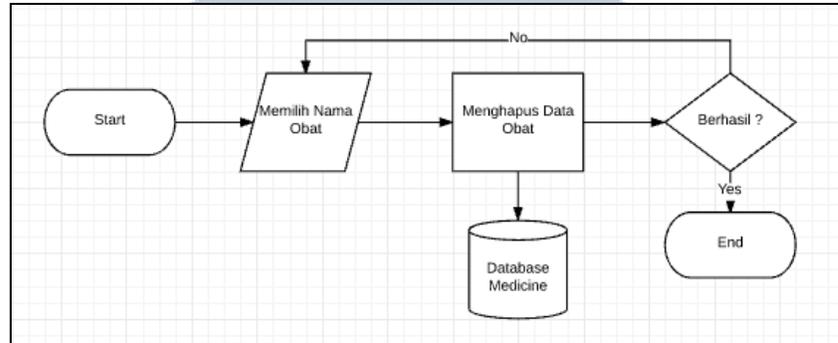
Gambar 3.21 merupakan *flowchart update brand* yang digunakan untuk melakukan modifikasi pada data *brand* dan disimpan ke dalam *database*. Sistem akan melakukan *retrieve data brand* yang akan ditampilkan dalam bentuk *text*. Kemudian, *admin* akan memasukkan detail *brand* yang baru dan sistem akan melakukan proses *update*. Jika gagal maka sistem akan melakukan *retrieve data brand* sehingga *admin* harus memasukkan kembali data detail *brand* yang baru.



Gambar 3.22 *Flowchart Update Tipe Obat*

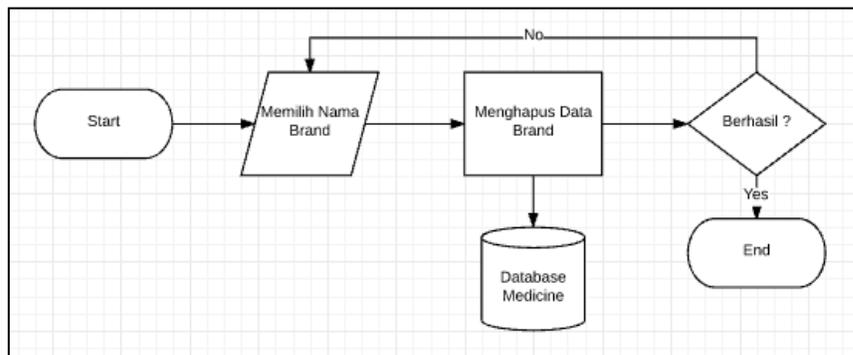
Gambar 3.22 merupakan *flowchart update tipe obat* yang digunakan untuk melakukan modifikasi pada data tipe obat dan disimpan ke dalam *database*. Sistem akan melakukan *retrieve data tipe obat* yang akan ditampilkan dalam bentuk *text*. Kemudian, *admin* akan memasukkan detail tipe obat yang baru dan sistem akan

melakukan proses *update*. Jika gagal maka sistem akan melakukan *retrieve data* tipe obat sehingga *admin* harus memasukkan kembali data detail tipe obat yang baru.



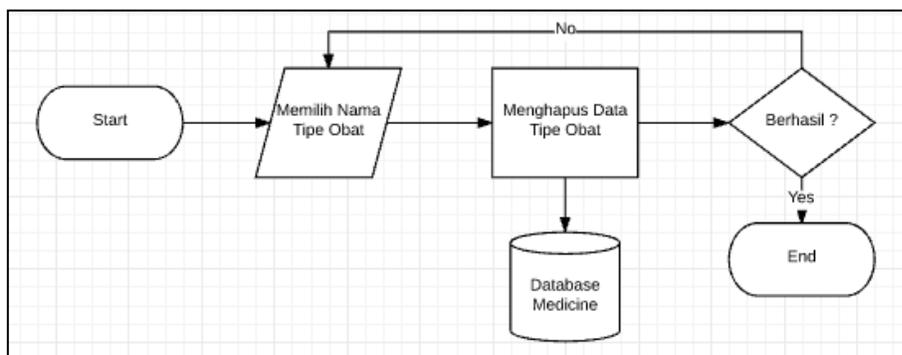
Gambar 3.23 Flowchart Delete Obat

Gambar 3.23 merupakan *flowchart delete* obat yang digunakan untuk menghapus data obat dari *database*. *Admin* akan memilih nama obat yang akan dihapus dan sistem akan melakukan proses *delete* untuk disimpan ke dalam *database*. Jika gagal maka *admin* harus memilih kembali nama obat yang akan dihapus.



Gambar 3.24 Flowchart Delete Brand

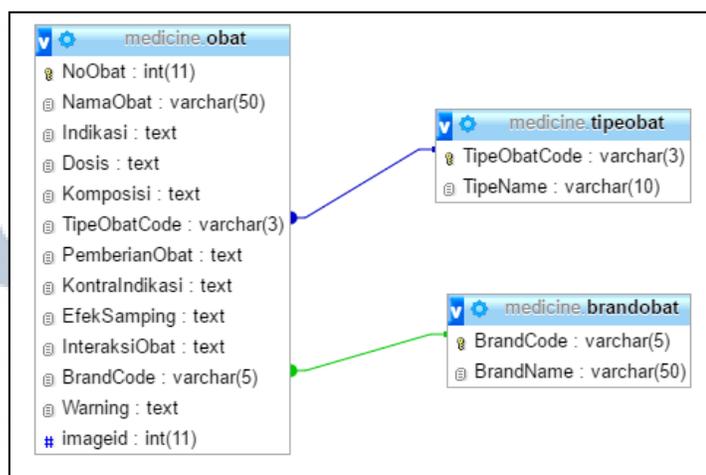
Gambar 3.24 merupakan *flowchart delete brand* yang digunakan untuk menghapus data *brand* dari *database*. *Admin* akan memilih nama *brand* yang akan dihapus dan sistem akan melakukan proses *delete* untuk disimpan ke dalam *database*. Jika gagal maka *admin* harus memilih kembali nama *brand* yang akan dihapus.



Gambar 3.25 *Flowchart Delete Tipe Obat*

Gambar 3.25 merupakan *flowchart delete* tipe obat yang digunakan untuk menghapus data tipe obat dari *database*. *Admin* akan memilih nama tipe obat yang akan dihapus dan sistem akan melakukan proses *delete* untuk disimpan ke dalam *database*. Jika gagal maka *admin* harus memilih kembali nama tipe obat yang akan dihapus.

3.2.4 Entity Relationship Diagram



Gambar 3.26 *Entity Relationship Diagram*

Entity relationship diagram dalam aplikasi ini memiliki hubungan antara tabel obat dan tabel tipeobat adalah *one to many*. Hal ini berarti satu data tipe obat dalam

tabel tipeobat bisa memiliki banyak data obat dalam tabel obat dan satu data obat dalam tabel obat hanya memiliki satu data tipe obat dalam tabel tipeobat. Hubungan antara tabel obat dan tabel brandobat adalah *one to many* sehingga satu tipe data dalam tabel brandobat bisa memiliki banyak data obat dalam tabel obat dan satu data obat dalam tabel obat hanya memiliki satu data tipe obat dalam tabel brandobat.

3.2.5 Struktur Tabel

A. Tabel brandobat

Primary key : BrandCode

Tabel 3.1 Struktur Tabel brandobat

| Attribute | Tipe | Ukuran | Keterangan |
|-----------|---------|--------|-----------------|
| BrandCode | Varchar | 5 | Kode unik brand |
| BrandName | Varchar | 50 | Nama brand |

B. Tabel tipeobat

Primary key : TipeObatCode

Tabel 3.2 Struktur Tabel tipeobat

| Attribute | Tipe | Ukuran | Keterangan |
|--------------|---------|--------|---------------------|
| TipeObatCode | Varchar | 3 | Kode unik tipe obat |
| TipeName | Varchar | 10 | Nama tipe obat |

C. Tabel obat

Primary key : NoObat

Foreign key : TipeObatCode dan BrandCode

Tabel 3.3 Struktur Tabel obat

| Attribute | Tipe | Ukuran | Keterangan |
|-----------|------|--------|----------------|
| NoObat | Int | 11 | Kode unik obat |

Tabel 3.3 Struktur Tabel obat (Lanjutan)

| | | | |
|----------------|---------|----|--|
| NamaObat | Varchar | 50 | Nama lengkap obat |
| Indikasi | Text | - | Penjelasan kegunaan obat |
| Dosis | Text | - | Penjelasan dosis yang harus diterapkan |
| Komposisi | Text | - | Penjelasan bahan yang digunakan dalam obat |
| TipeObatCode | Varchar | 3 | Kode unik tipe obat |
| PemberianObat | Text | - | Penjelasan cara menggunakan obat |
| KontraIndikasi | Text | - | Penjelasan dari malfungsi obat |
| EfekSamping | Text | - | Penjelasan efek samping setelah menggunakan obat |
| InteraksiObat | Text | - | Penjelasan cara kerja obat |
| BrandCode | Varchar | 5 | Kode unik brand obat |
| Warning | Text | - | Penjelasan penggunaan yang tidak diperbolehkan |
| imageid | Int | 11 | Mendefinisikan gambar |

D. Tabel admin

Primary key : id

Tabel 3.4 Struktur Tabel admin

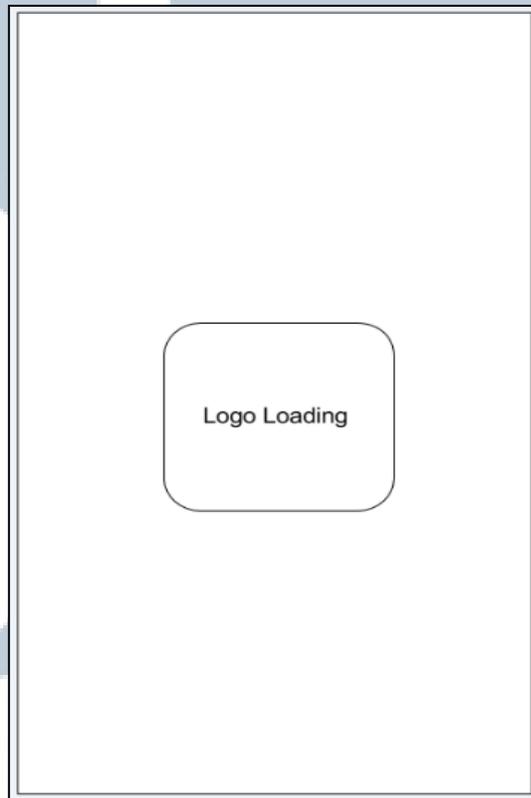
| Attribute | Tipe | Ukuran | Keterangan |
|-----------|---------|--------|-------------------------------------|
| id | Varchar | 11 | Kode unik admin |
| name | Varchar | 255 | Nama admin |
| username | Varchar | 255 | Username yang digunakan untuk login |
| password | Varchar | 255 | Password yang digunakan untuk login |

3.2.6 Rancangan Antarmuka

Rancangan antarmuka dari *medicine dictionary application* ini terdiri dari rancangan antarmuka halaman *front end* dan rancangan antarmuka halaman *back end*.

A. Rancangan Antarmuka Front End

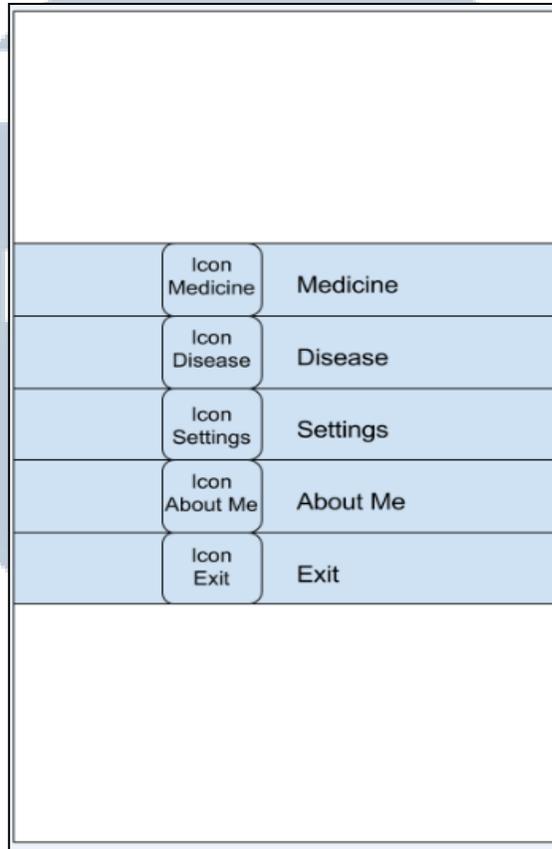
Rancangan antarmuka *medicine dictionary application* adalah tampilan aplikasi yang ditampilkan atau digunakan kepada *user*. Tampilan ini diawali oleh sebuah logo *loading* seperti pada Gambar 3.27.



Gambar 3.27 Rancangan Antarmuka Halaman *Loading*

Gambar 3.27 merupakan rancangan antarmuka halaman *loading* yang menampilkan logo *loading* selama dua detik. Logo *loading* telah diberikan *animate rotation* dengan banyak putaran dua kali atau setara dengan 720° sehingga setiap

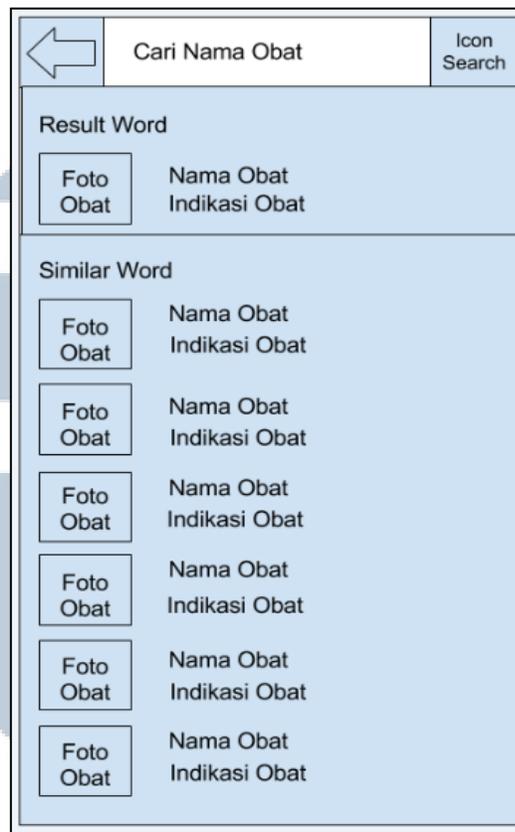
putaran akan menggunakan waktu selama satu detik. Selanjutnya aplikasi akan menampilkan halaman *menu* seperti pada Gambar 3.28.



| | |
|------------------|----------|
| Icon Medicine | Medicine |
| Icon Disease | Disease |
| Icon Settings | Settings |
| Icon About Me | About Me |
| Icon Exit | Exit |

Gambar 3.28 Rancangan Antarmuka Halaman *Menu*

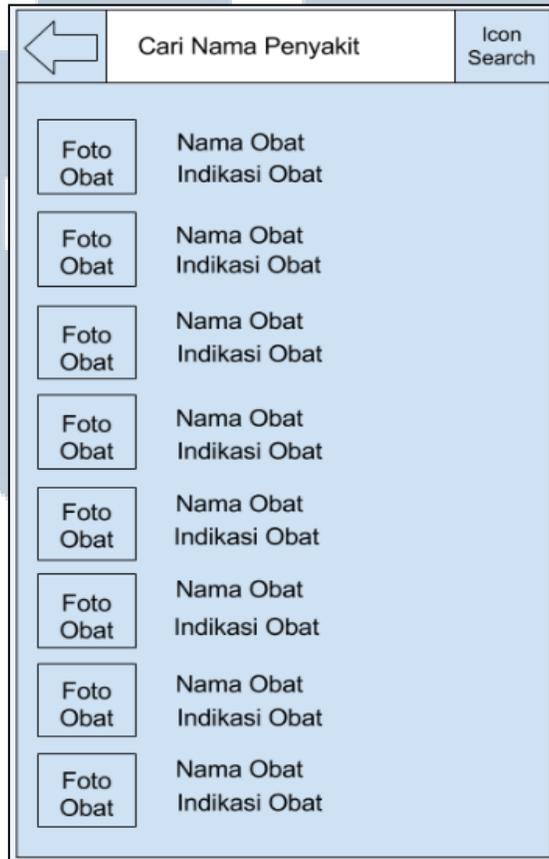
Gambar 3.28 merupakan rancangan antarmuka halaman menu yang berisi lima buah *icon*, yaitu *medicine*, *disease*, *settings*, *about me*, dan *exit*. Setiap *icon* memiliki tampilan yang berbeda-beda, kecuali *icon exit* akan memberikan *alert dialog* yang berisi pertanyaan untuk keluar dari aplikasi.



Gambar 3.29 Rancangan Antarmuka Halaman *Medicine*

Gambar 3.29 merupakan rancangan antarmuka halaman *medicine* yang memiliki dua buah tombol pada bagian atas, yaitu tombol *back* dan tombol *search* serta sebuah *edit text* yang akan diisikan dengan sebuah nama obat. Jika tombol *search* digunakan maka akan muncul daftar nama obat baik *result word* maupun *similar word*. Namun, tidak selalu hasil pencarian terdapat *result word* dan *similar word*. Hasil *result word* berisi nama obat yang sesuai dengan nama obat yang dicari atau yang memiliki tingkat kemiripan 100%, sedangkan hasil *similar word* berisi nama obat yang dianggap memiliki kemiripan dengan nama obat yang dicari. Hasil pencarian *result word* dan *similar word* menggunakan *listview* dan setiap *listview* terdapat dua buah *text view*, yaitu nama obat dan indikasi obat serta sebuah *image*

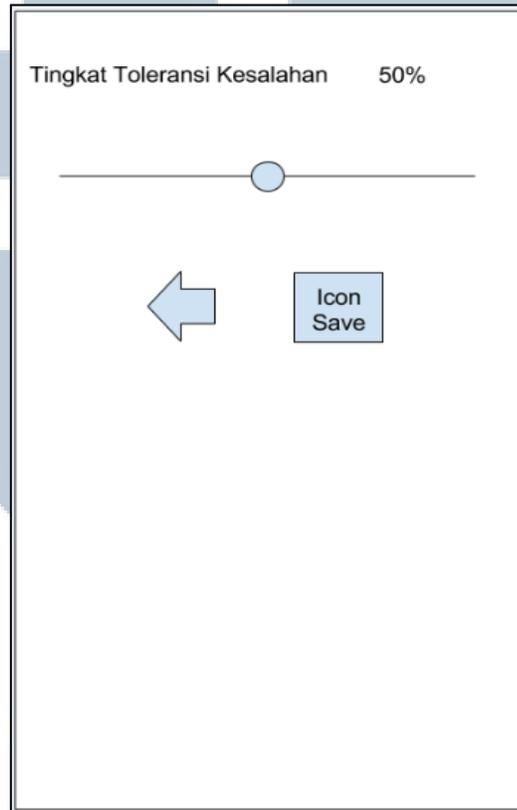
view untuk memberikan informasi berupa foto obat tersebut. Pada setiap data dalam *result word* dan *similar word* memiliki *event click* yang digunakan untuk melihat detail dari obat tertentu.



Gambar 3.30 Rancangan Antarmuka Halaman *Disease*

Gambar 3.30 merupakan rancangan antarmuka halaman *disease* yang memiliki dua buah tombol pada bagian atas, yaitu tombol *back* dan tombol *search* serta sebuah *edit text* yang akan diisi dengan sebuah nama penyakit. Jika tombol *search* digunakan maka akan menghasilkan daftar nama obat yang terdapat nama penyakit yang ingin dicari dalam indikasi obat. Hasil pencarian menggunakan *listview* dan setiap *listview* terdapat dua buah *text view*, yaitu nama obat dan indikasi obat

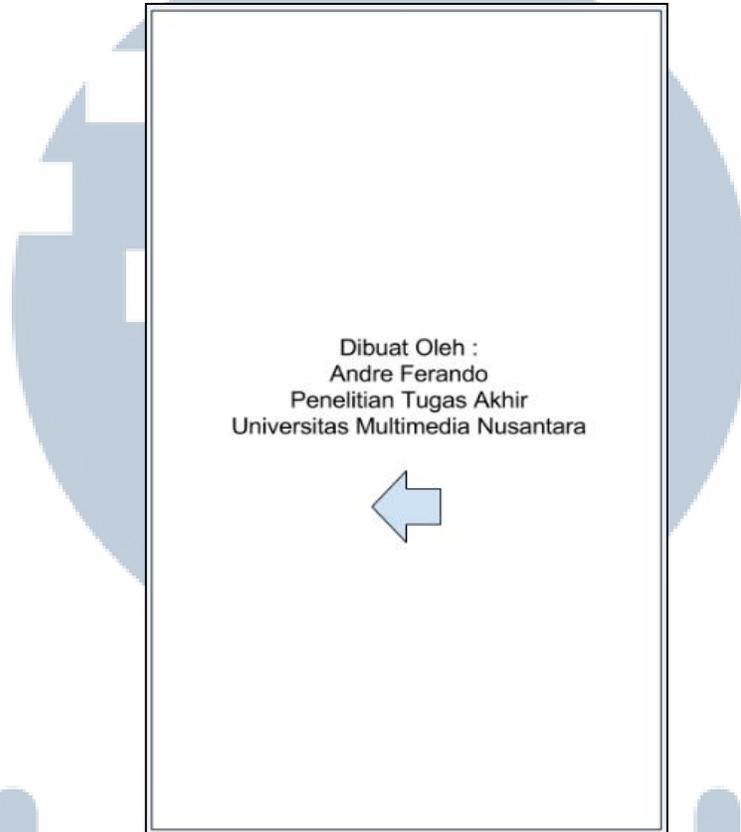
serta sebuah *image view* untuk memberikan informasi berupa foto obat tersebut. Pada setiap data dalam *list view* memiliki *event click* yang digunakan untuk melihat detail dari obat tertentu.



Gambar 3.31 Rancangan Antarmuka Halaman *Settings*

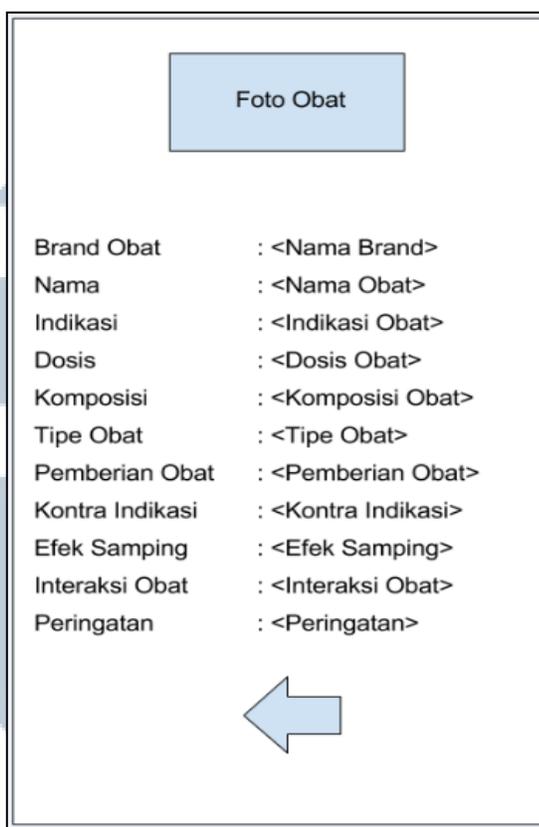
Gambar 3.31 merupakan rancangan antarmuka halaman *settings* yang memiliki sebuah *seek bar*, dan dua buah tombol, yaitu tombol *back* dan tombol *save*. *Seek bar* tersebut memiliki *range* 1% hingga 100% dan digunakan untuk memilih tingkat toleransi kesalahan nama obat yang akan digunakan dalam proses perhitungan algoritma Levenshtein Distance sehingga dapat memberikan daftar *similar word* pada halaman *medicine*. Tombol *save* digunakan untuk menyimpan nilai toleransi yang

akan digunakan dalam aplikasi sedangkan tombol *back* digunakan untuk kembali ke halaman *menu* seperti pada Gambar 3.28.



Gambar 3.32 Rancangan Antarmuka Halaman *About Me*

Gambar 3.32 merupakan rancangan antarmuka halaman *about me* yang digunakan untuk menampilkan informasi dari aplikasi yang telah dibangun. Informasi tersebut berada di tengah layar *android* dan berisi tentang pembuat aplikasi, tujuan pembuatan aplikasi, dan asal pembuat aplikasi. Halaman *about me* juga memiliki sebuah tombol *back* yang dapat digunakan untuk kembali ke halaman *menu* seperti pada Gambar 3.28.

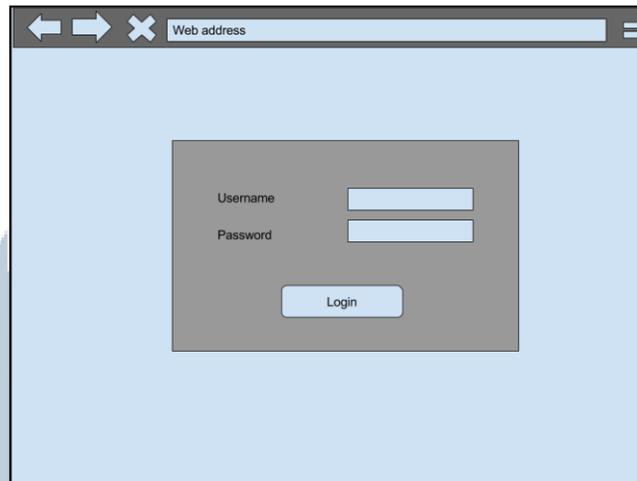


Gambar 3.33 Rancangan Antarmuka Halaman Detail *Medicine*

Gambar 3.33 merupakan rancangan antarmuka halaman detail *medicine* yang berisi informasi mengenai obat tertentu, seperti foto obat, *brand*, nama obat, indikasi obat, dosis obat, komposisi obat, tipe obat, pemberian obat, kontra indikasi, efek samping, interaksi obat, dan peringatan. Pada bagian bawah terdapat sebuah tombol *back* untuk kembali ke halaman sebelumnya.

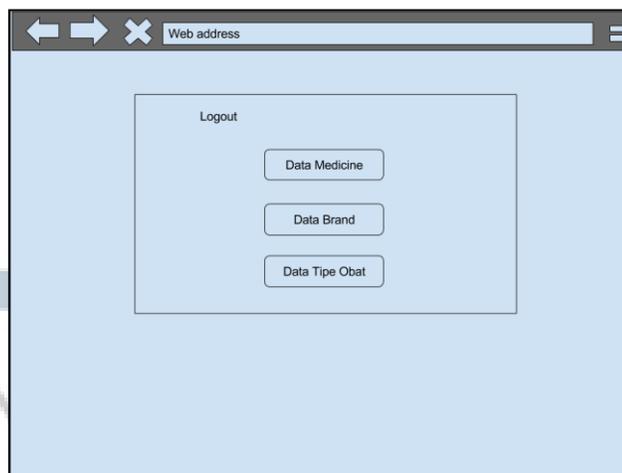
B. Rancangan Antarmuka Back End

Rancangan antarmuka halaman *back end* adalah tampilan halaman *website* yang ditampilkan atau digunakan kepada *admin*. Tampilan ini diawali oleh sebuah halaman *login* seperti pada Gambar 3.34.



Gambar 3.34 Rancangan Antarmuka Halaman *Login Admin*

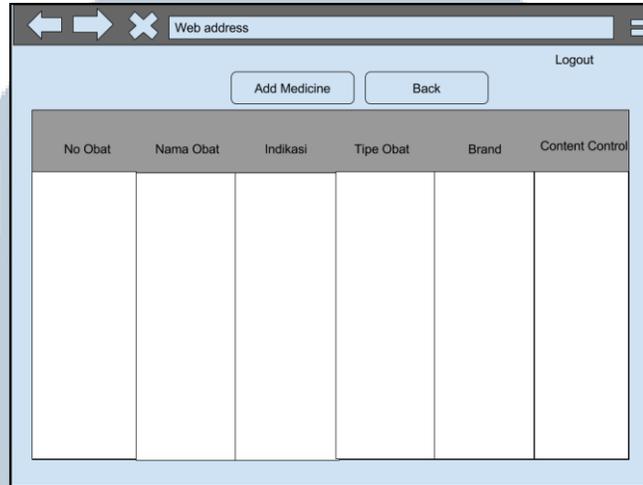
Gambar 3.34 merupakan rancangan antarmuka halaman *login* yang digunakan oleh admin yang telah terdaftar di dalam *database*. Terdapat dua buah *textview* dan sebuah tombol *login*. Jika admin berhasil *login* maka akan muncul tiga buah menu seperti pada Gambar 3.35.



Gambar 3.35 Rancangan Antarmuka Halaman Menu Admin

Gambar 3.35 merupakan rancangan antarmuka halaman menu untuk admin yang berfungsi untuk melihat data obat, *brand*, dan tipe obat. Terdapat tiga buah

tombol, yaitu data *medicine*, data *brand*, dan data tipe obat. Setiap tombol akan bernavigasi ke halaman-halaman yang berbeda.



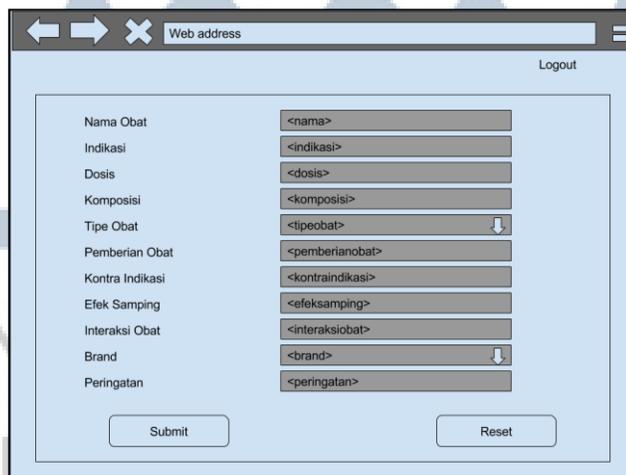
Gambar 3.36 Rancangan Antarmuka Halaman *List Medicine Admin*

Gambar 3.36 merupakan rancangan antarmuka halaman *list medicine* yang menampilkan semua data obat pada *database*. Terdapat sebuah tabel yang terdiri dari enam buah kolom, yaitu nomor obat, nama obat, indikasi, tipe obat, *brand*, dan *content control*. Pada bagian *header* terdapat dua buah tombol, yaitu tombol *add medicine* yang berfungsi untuk menambahkan obat baru seperti pada Gambar 3.37 dan tombol *back* yang berfungsi untuk kembali ke halaman menu. Pada bagian *top header* terdapat sebuah *link logout* untuk keluar dari admin. Kolom *content control* digunakan untuk melakukan *update* dan *delete* obat. *Update* obat digunakan untuk mengubah data detail obat seperti pada Gambar 3.38 sedangkan *delete* obat digunakan untuk menghapus data obat.



Gambar 3.37 Rancangan Antarmuka Halaman *Add Medicine Admin*

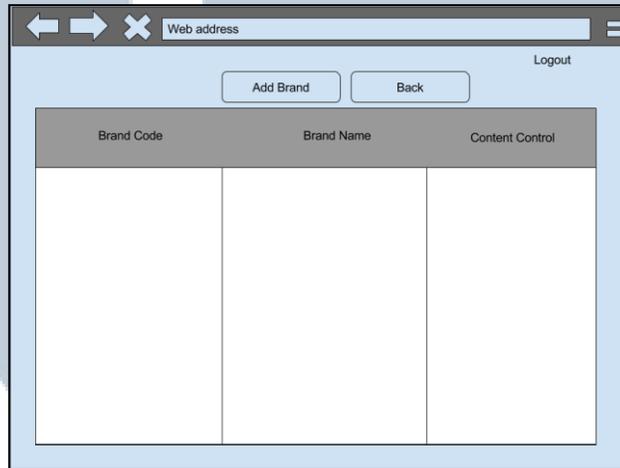
Gambar 3.37 merupakan rancangan antarmuka halaman *add medicine* yang berfungsi untuk menambahkan obat baru. Halaman ini terdapat sembilan buah *textview* dan dua buah *dropdown list*. Pada bagian bawah terdapat dua buah tombol, yaitu tombol *submit* untuk melakukan penambahan obat baru dan tombol *reset* untuk menghapus semua data detail obat yang telah diisi.



Gambar 3.38 Rancangan Antarmuka Halaman *Edit Medicine Admin*

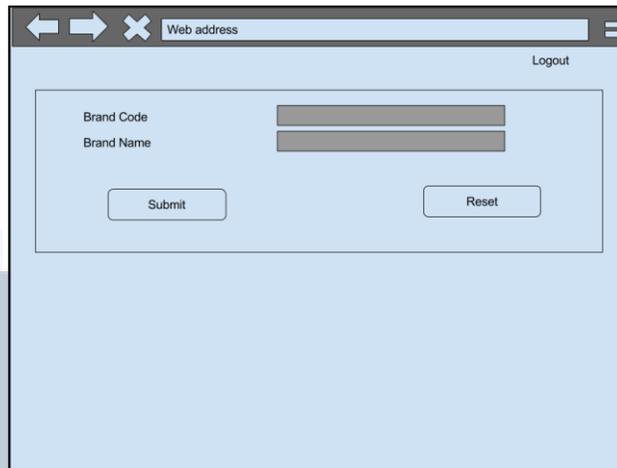
Gambar 3.38 merupakan rancangan antarmuka halaman *edit* obat yang berfungsi untuk melakukan modifikasi obat yang telah dipilih dari halaman *list* obat.

Halaman ini terdapat sembilan buah *textview* dan dua buah *dropdown list*. Pada bagian bawah terdapat dua buah tombol, yaitu tombol *submit* untuk melakukan modifikasi obat dan tombol *reset* untuk menghapus semua data detail obat yang telah diisi.



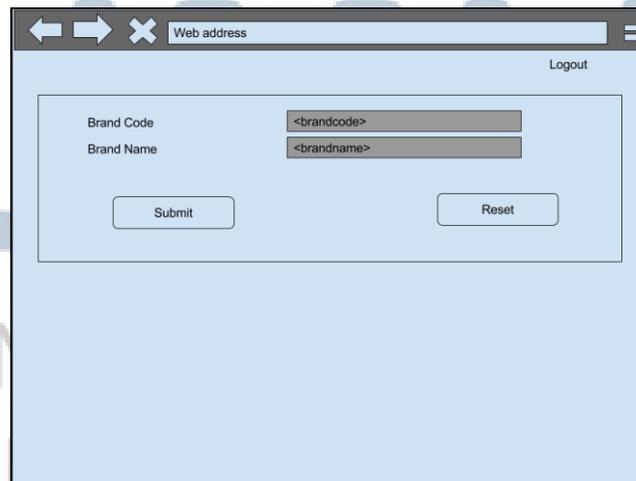
Gambar 3.39 Rancangan Antarmuka Halaman *List Brand Admin*

Gambar 3.39 merupakan rancangan antarmuka halaman *list brand* yang menampilkan semua data *brand* pada *database*. Terdapat sebuah tabel yang terdiri dari tiga buah kolom, yaitu *brand code*, *brand name*, dan *content control*. Pada bagian *header* terdapat dua buah tombol, yaitu tombol *add brand* yang berfungsi untuk menambahkan *brand* baru seperti pada Gambar 3.40 dan tombol *back* yang berfungsi untuk kembali ke halaman menu. Pada bagian *top header* terdapat sebuah *link logout* untuk keluar dari admin. Kolom *content control* digunakan untuk melakukan *update* dan *delete brand*. *Update brand* digunakan untuk mengubah data detail *brand* seperti pada Gambar 3.41 sedangkan *delete brand* digunakan untuk menghapus data *brand*.



Gambar 3.40 Rancangan Antarmuka Halaman *Add Brand Admin*

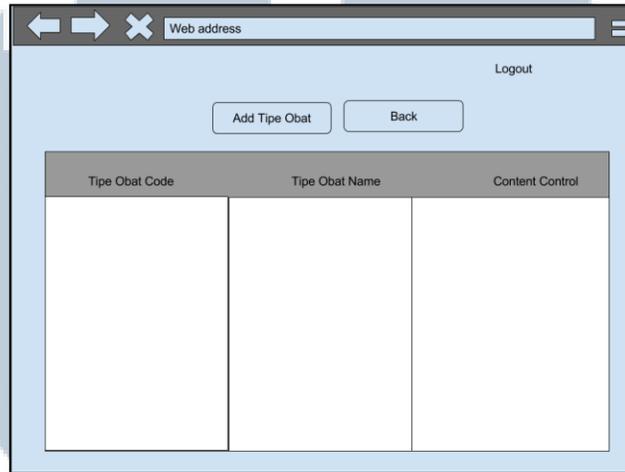
Gambar 3.40 merupakan rancangan antarmuka halaman *add brand* yang berfungsi untuk menambahkan *brand* baru. Halaman ini terdapat dua buah *textview*. Pada bagian bawah terdapat dua buah tombol, yaitu tombol *submit* untuk melakukan penambahan *brand* baru dan tombol *reset* untuk menghapus semua data detail *brand* yang telah diisi.



Gambar 3.41 Rancangan Antarmuka Halaman *Edit Brand*

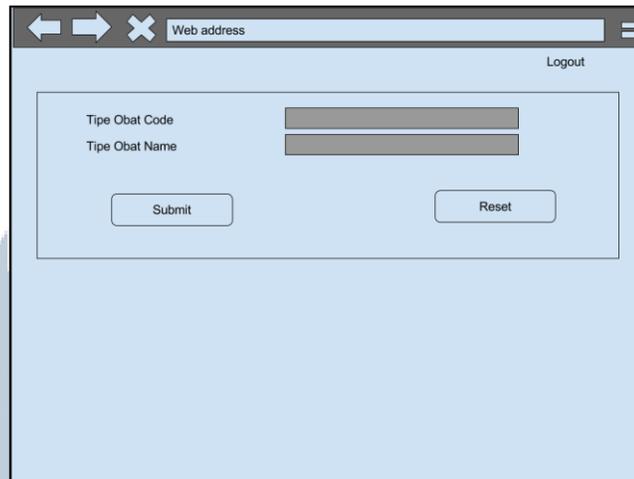
Gambar 3.41 merupakan rancangan antarmuka halaman *edit brand* yang berfungsi untuk melakukan modifikasi *brand* yang telah dipilih dari halaman *list*

brand. Halaman ini terdapat dua buah *textview*. Pada bagian bawah terdapat dua buah tombol, yaitu tombol *submit* untuk melakukan modifikasi *brand* dan tombol *reset* untuk menghapus semua data detail *brand* yang telah diisi kecuali *brand code*.



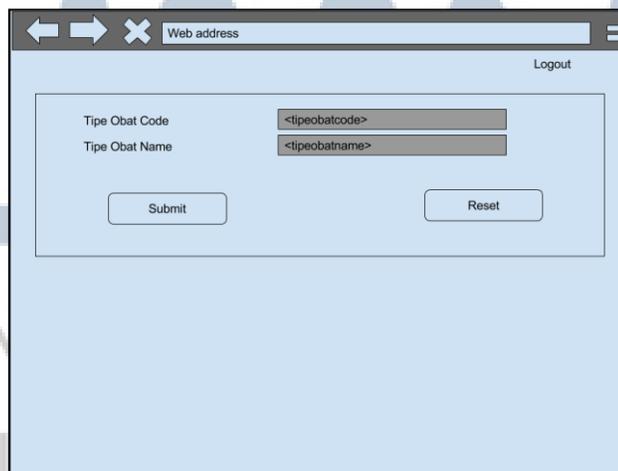
Gambar 3.42 Rancangan Antarmuka Halaman *List Tipe Obat Admin*

Gambar 3.42 merupakan rancangan antarmuka halaman *list* tipe obat yang menampilkan semua data tipe obat pada *database*. Terdapat sebuah tabel yang terdiri dari tiga buah kolom, yaitu tipe obat *code*, tipe obat *name*, dan *content control*. Pada bagian *header* terdapat dua buah tombol, yaitu tombol *add* tipe obat yang berfungsi untuk menambahkan tipe obat baru seperti pada Gambar 3.43 dan tombol *back* yang berfungsi untuk kembali ke halaman menu. Pada bagian *top header* terdapat sebuah *link logout* untuk keluar dari *admin*. Kolom *content control* digunakan untuk melakukan *update* dan *delete brand*. *Update* tipe obat digunakan untuk mengubah data detail tipe obat seperti pada Gambar 3.44 sedangkan *delete* tipe obat digunakan untuk menghapus data tipe obat.



Gambar 3.43 Rancangan Antarmuka Halaman *Add Tipe Obat Admin*

Gambar 3.43 merupakan rancangan antarmuka halaman *add* tipe obat yang berfungsi untuk menambahkan tipe obat baru. Halaman ini terdapat dua buah *textview*. Pada bagian bawah terdapat dua buah tombol, yaitu tombol *submit* untuk melakukan penambahan tipe obat baru dan tombol *reset* untuk menghapus semua data detail tipe obat yang telah diisi.



Gambar 3.44 Rancangan Antarmuka Halaman *Edit Tipe Obat Admin*

Gambar 3.44 merupakan rancangan antarmuka halaman *edit* tipe obat yang berfungsi untuk melakukan modifikasi tipe obat yang telah dipilih dari halaman *list*

tipe obat. Halaman ini terdapat dua buah *textview*. Pada bagian bawah terdapat dua buah tombol, yaitu tombol *submit* untuk melakukan modifikasi tipe obat dan tombol *reset* untuk menghapus semua data detail tipe obat yang telah diisi kecuali tipe obat *code*.

