

BAB III

METODOLOGI DAN PERANCANGAN APLIKASI

3.1 Metodologi Penelitian

Metodologi penelitian dalam penyusunan tugas akhir ini dibagi menjadi tujuh tahap, yaitu tahap wawancara, pengumpulan data, tahap perancangan, tahap pembangunan *software*, tahap uji coba beserta perbaikan, evaluasi, dan penulisan laporan.

a. Wawancara dan Observasi

Wawancara dan observasi dilakukan untuk mempelajari proses penyebaran informasi di bengkel SUM Hexaputra. Proses wawancara dilakukan dengan Ibu Eva Meilinda selaku anak dari pemilik SUM Hexaputra dan Pak Eko selaku *staff admin* dan gudang dari Bengkel SUM Hexaputra. Wawancara bertujuan untuk mencari informasi apa yang sering dicari oleh pelanggan SUM Hexaputra dan bentuk komunikasi yang terjadi ketika pelanggan menanyakan informasi di Bengkel SUM Hexaputra. Informasi tersebut akan dijadikan sebagai *knowledge base* pada program *chatbot* yang dirancang dan dibangun. Wawancara dengan Ibu Eva Meilinda dan Pak Eko dilakukan pada tanggal 05 Oktober 2017 dan 06 Oktober 2017.

b. Studi Literatur

Studi literatur dilakukan dengan mempelajari teori dengan mencari literatur, jurnal, *paper*, dan bacaan yang berhubungan dengan aplikasi yang akan dirancang dan dibangun. Teori-teori tersebut adalah *Chatbot*, *Natural Language Processing*,

Artificial Intelligence Markup Language (AIML), Text Preprocessing, Text Transformation, Algoritma Nazief & Adriani, Technology Acceptance Model (TAM), Skala Likert, Kateglo API, dan Program O.

c. Perancangan Aplikasi

Tahap awal perancangan aplikasi dimulai dengan analisis kebutuhan yang diperlukan dalam pembangunan aplikasi ini. Tahap berikutnya adalah merancang tampilan antarmuka untuk aplikasi ini. Dokumentasi perancangan aplikasi dilakukan dengan menggunakan model aplikasi, *Data Flow Diagram (DFD), sitemap, flowchart, Entity Relationship Diagram (ERD), database schema*, struktur tabel, dan perancangan dialog komunikasi *chatbot*.

d. Pembangunan Aplikasi

Setelah perancangan aplikasi, pembangunan aplikasi *chatbot* ini dilakukan dengan membuat antarmuka ke dalam bentuk HTML dengan *framework* Ionic. *Chatbot* yang telah dirancang diimplementasikan pada *Artificial Intelligence Markup Language (AIML)*. Aplikasi *chatbot* ini dibuat dengan menggunakan Typescript dari *framework* Ionic. Halaman admin akan dibuat dengan menggunakan bahasa PHP dan *database* MySQL. Library Program O digunakan sebagai *interpreter* untuk melakukan *parsing* dari bentuk AIML menjadi *row data* pada *database* MySQL.

e. *Testing dan Debugging*

Testing dilakukan untuk menguji apakah aplikasi sudah berfungsi sesuai dengan kebutuhan yang telah didefinisikan. Apabila ditemukan *error, bug*, atau *defect* akan langsung dilakukan *kode debugging* untuk memperbaiki kesalahan tersebut.

f. Evaluasi

Evaluasi dilakukan untuk mengetahui tingkat keberhasilan aplikasi yang telah dirancang dan dibangun dengan penyebaran kuesioner *Technology Acceptance Model* (TAM) dengan dua variabel *perceived ease of use* dan *perceived usefulness*. Penyebaran kuesioner dilakukan kepada pihak penjual dan pembeli.

g. Penulisan Laporan

Dari hasil yang dibuat dan didapatkan, semua dituangkan ke dalam laporan sebagai bukti bahwa telah melakukan penelitian.

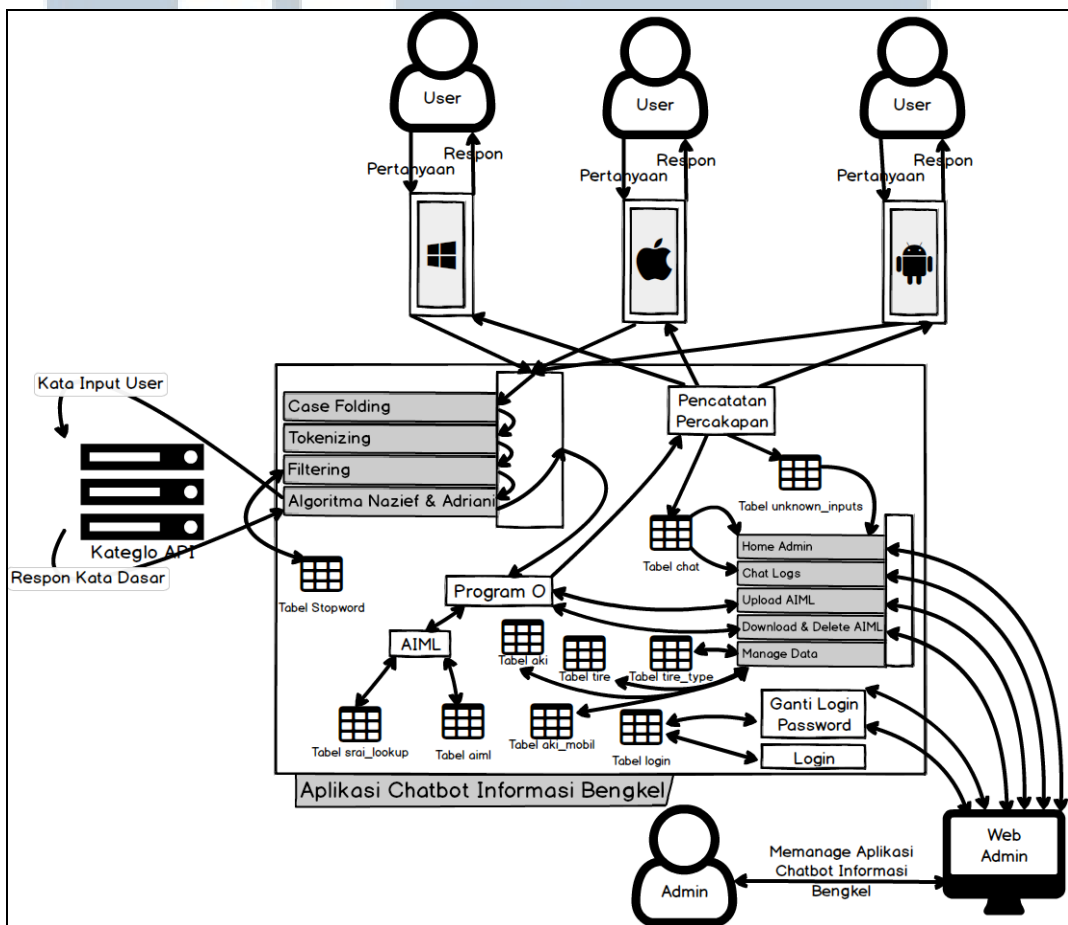
3.2 Perancangan Aplikasi

Perancangan aplikasi yang dilakukan menghasilkan model berupa model aplikasi, *Data Flow Diagram* (DFD), *sitemap*, *flowchart*, *Entity Relationship Diagram* (ERD), *database schema*, struktur tabel, dan perancangan dialog komunikasi *chatbot*.

3.2.1 Model Aplikasi

Gambar 3.1 menunjukkan model aplikasi chatbot penyedia informasi bengkel. Aplikasi Chatbot Penyedia Informasi Bengkel akan terpasang pada sebuah *web service*. User dapat berinteraksi dengan cara menggunakan *smartphone* bersistem operasi Android dengan memasang/melakukan *install* Aplikasi Chatbot Penyedia Informasi Bengkel. *User* dapat mengajukan pertanyaan dan aplikasi akan menjawab dengan respon dari pertanyaan yang *user* berikan. Proses dari pertanyaan sampai respon melalui proses *case folding*, *tokenizing*, *filtering*, dan pemotongan untuk mendapatkan kata dasar untuk setiap kata pada masukan *user* menggunakan algoritma Nazief & Adriani dengan bantuan Kateglo API untuk mencari kata dasar.

Proses *filtering* melakukan pembuangan kata *stopwords* dengan mencari kata *stopwords* pada tabel *stopwords*. Setelah menjadi kalimat dengan kata dasar, maka kalimat tersebut akan melakukan pencocokan pola menggunakan bantuan Program O dalam membaca AIML dari tabel *srai_lookup* dan tabel *aiml*. Hasil respon AIML dari Program O akan dicatat pada tabel *chat* dan *unknown_inputs* (jika ada input yang tidak dapat diterjemahkan *knowledge base chatbot*). Setelah pencatatan maka respon akan diteruskan kepada *user*.



Gambar 3.1 Model Aplikasi

Aplikasi ini menerapkan *Natural Language Processing* (NLP) dalam mengolah *input user*. Tahap-tahap yang digunakan adalah sebagai berikut.

1. *Morphological and Lexical Analysis*

Tahap ini dilakukan pada proses *text preprocessing* dan *text transformation*

2. *Discourse Integration*

Tahap ini dimiliki oleh *tag* pada AIML yaitu *tag* <that>, <set>, <get>, dan <topic>.

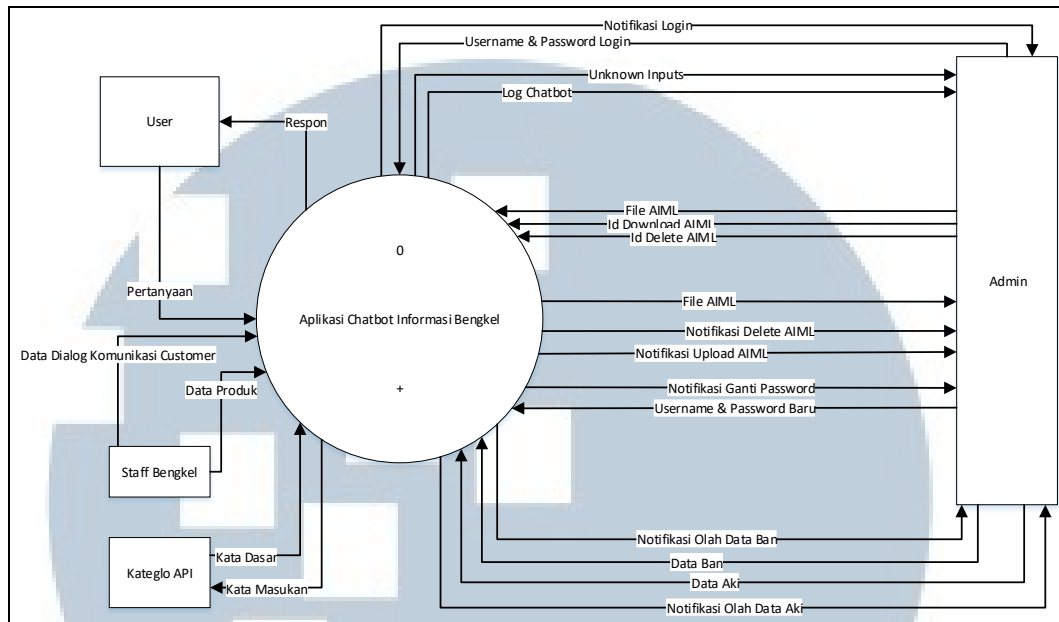
3. *Pragmatic Analysis*

Tahap ini dimiliki oleh *tag* pada AIML yaitu *tag* <condition> untuk memberikan respon dari berbagai kondisi dan <topic> untuk mengetahui topik pembicaraan yang sedang berjalan.

Admin dalam Aplikasi Chatbot Penyedia Informasi Bengkel berperan untuk melakukan pengolahan aplikasi *chatbot*. Admin dapat melakukan ganti *password login*, *download & delete AIML*, *upload AIML*, melihat *chat log*, melakukan olah data aki dan ban, serta melihat *home admin* itu sendiri yang berisi kumpulan *unknown input*. Proses yang berhubungan dengan AIML akan menggunakan bantuan Program O.

3.2.2 Data Flow Diagram (DFD)

Data Flow Diagram menggambarkan alur data yang mengalir dalam aplikasi *chatbot*. Gambar 3.2 menggambar *context diagram* Aplikasi Chatbot Penyedia Informasi Bengkel. Level ini terdiri dari 4 entitas dan 1 proses. Empat entitas tersebut adalah *user*, Kateglo API, *staff* bengkel, dan admin. Proses pada level ini adalah Aplikasi Chatbot Penyedia Informasi Bengkel.



Gambar 3.2 Context Diagram

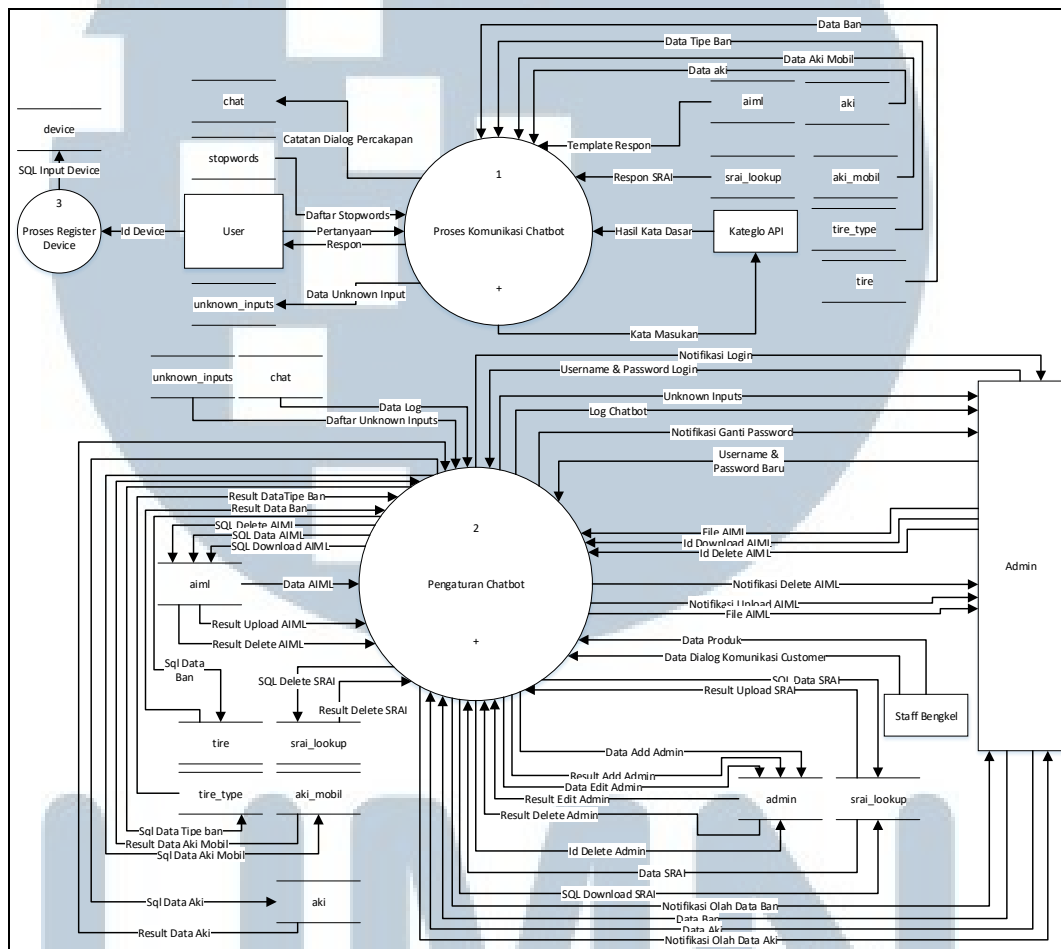
Entitas *user* memberikan masukan data ke dalam aplikasi berupa kalimat atau pertanyaan. Kalimat akan diproses oleh aplikasi. Aplikasi akan memberikan *output* kepada *user* berupa kalimat respon atas pertanyaan yang diajukan.

Entitas *admin* dapat melakukan *login* ke Halaman *admin* dengan memberikan *username* dan *password*. Setelah berhasil melakukan proses *login*, *admin* dapat membuat dan merubah *knowledge base*. *Admin* juga dapat melakukan pemeriksaan *unknown input*, *log chatbot*, dan mengolah data ban serta aki.

Entitas *Katego API* akan menerima masukan dari aplikasi berupa kata. Kata tersebut akan diproses di dalam *API*. *API* akan memberikan respon apakah kata yang masuk sudah merupakan kata dasar atau tidak. *Katego API* sendiri merupakan aplikasi dan layanan *web* sumber dan isi terbuka untuk kamus, tesaurus, dan glosarium bahasa Indonesia (Katego, 2009).

Entitas *staff bengkel* memberikan data awal kepada *admin* untuk membuat *brain file* atau *knowledge base chatbot* (*AIML file*). Data yang diberikan adalah

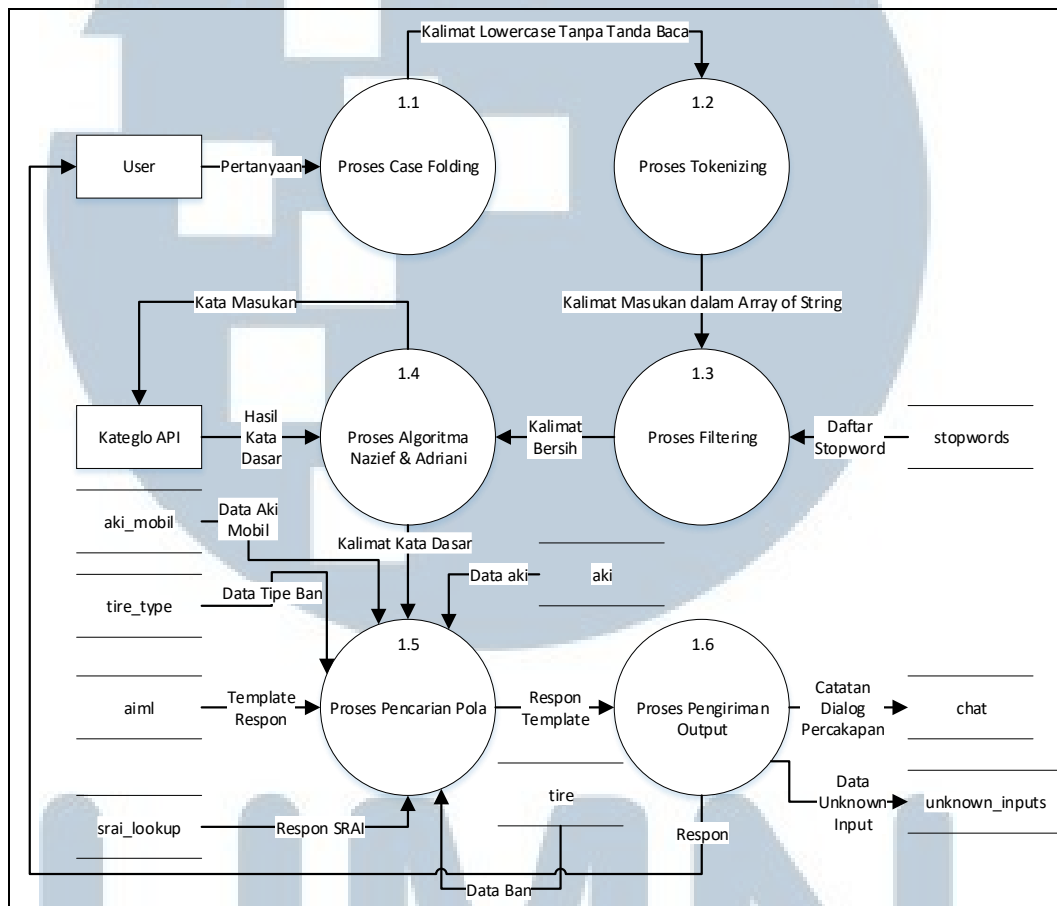
data produk dan jasa yang tersedia pada bengkel SUM Hexaputra serta data kalimat-kalimat percakapan yang dilakukan dengan *customer*. Berdasarkan data tersebut, *admin* membuat pola dialog dalam bentuk AIML yang digunakan *chatbot* untuk pencarian pola kalimat dan respon yang sesuai.



Gambar 3.3 DFD Level 0

Gambar 3.3 menggambarkan DFD *level 0* dari Aplikasi Chatbot Penyedia Informasi Bengkel. Pada level ini terdapat 3 proses, yaitu proses komunikasi *chatbot*, pengaturan *chatbot*, dan proses *register device*. Proses komunikasi *chatbot* merupakan proses yang menerima kalimat *input* dari *user* lalu memproses kalimat tersebut sehingga mendapatkan kalimat respon yang tepat untuk diberikan kembali kepada *user*. Proses pengaturan *chatbot* merupakan proses yang dilakukan oleh

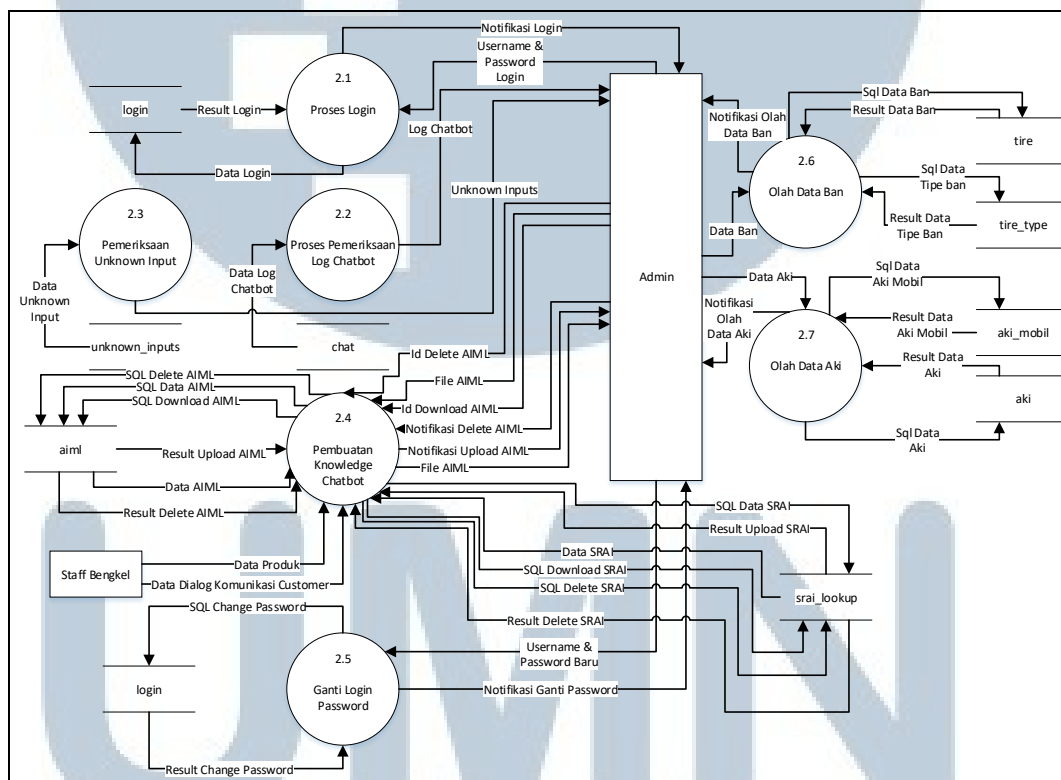
entitas admin dan *staff* bengkel dalam membuat dan merubah *file* AIML untuk *knowledge base* pada *chatbot*. Proses *register device* adalah proses mendaftarkan/memberikan id kepada perangkat yang digunakan dalam mengakses aplikasi *chatbot*.



Gambar 3.4 DFD Level 1 Proses Komunikasi Chatbot

Proses komunikasi *chatbot* memiliki 6 proses yang digambarkan pada DFD level 1 pada Gambar 3.4. Proses *case folding* dilakukan dengan mengubah kalimat masukan atau pertanyaan ke dalam bentuk *lowercase* (huruf kecil) dan membuang semua tanda baca yang ada pada kalimat tersebut. Proses *tokenizing* mengubah kalimat menjadi *array of string* yang akan digunakan pada proses *filtering*. Proses *filtering* sendiri akan menghilangkan kata-kata tertentu jika kata tersebut terdapat dalam tabel stopwords. Tabel stopwords merupakan kumpulan *stopwords* yang

diambil dari referensi penelitian Bahasa Indonesia oleh Fadillah Z Tala (2003). Kemudian melakukan proses *stemming* menggunakan algoritma Nazief & Adriani. Hasil proses pertanyaan menghasilkan kalimat dalam kata dasar yang selanjutnya digunakan pada proses pencarian pola. Proses pencarian menggunakan data pada tabel *aiml* dan *srai_lookup*. Hasil pencarian merupakan kalimat respon yang akan dikirim ke *user*. Kalimat respon dan pertanyaan dari *user* akan dicatat ke dalam tabel *chat* dan kalimat yang tidak bisa dijawab akan di catat pada tabel *unknown_inputs*.

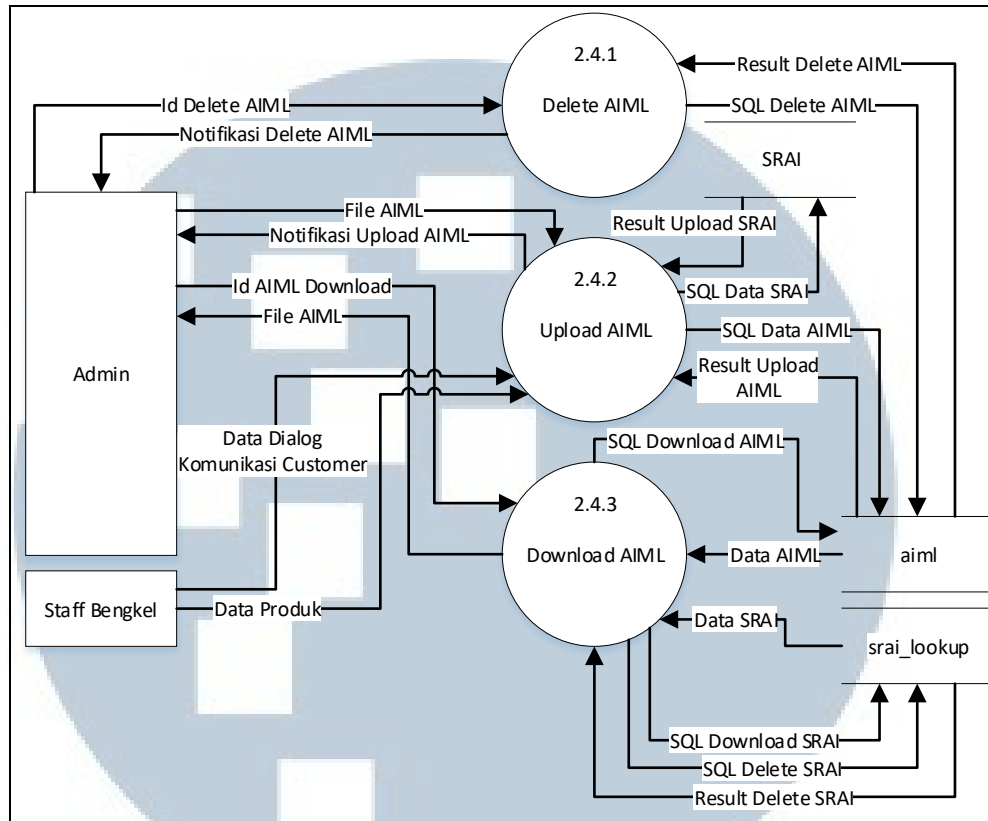


Gambar 3.5 DFD Level 1 Proses Pengaturan Chatbot

Proses pengaturan *chatbot* tergambar pada DFD level 1 pada Gambar 3.5. Level ini memiliki 7 proses, yaitu proses login, pemeriksaan *unknown input*, proses pemeriksaan *log chatbot*, pembuatan *knowledge chatbot*, olah data aki, olah data ban, dan ganti *login password*. Sebelum dapat melakukan proses lain, admin

harus melakukan *login* terlebih dahulu dengan memasukkan *username* dan *password*. Aplikasi akan melakukan proses pencarian *username* dan *password* pada tabel login. Jika gagal/berhasil melakukan *login*, maka aplikasi akan memberikan notifikasi konfirmasi keberhasilan *login*. Semua percakapan yang telah dilakukan oleh *chatbot* dapat dilihat oleh admin termasuk *input user* yang tidak dapat dikenali atau tidak dapat direspon oleh *chatbot*. Informasi percakapan tercatat pada tabel chat dan *input* yang tidak dikenali tercatat pada tabel *unknown_inputs*. Admin dapat melihat kesalahan atau *error* yang terjadi saat percakapan berlangsung. Kesalahan tersebut dapat diperbaiki oleh admin dengan melakukan *download file* AIML lalu merubahnya di luar aplikasi dan melakukan *upload* untuk mengganti *file* dan data yang lama pada aplikasi. Data aki dapat diolah dengan proses olah data aki. Data ban dapat diolah dengan proses olah data ban. Proses olah data itu terdiri dari buat data baru, *update* data yang sudah ada, dan dapat aktivasi / non-aktivasi data.



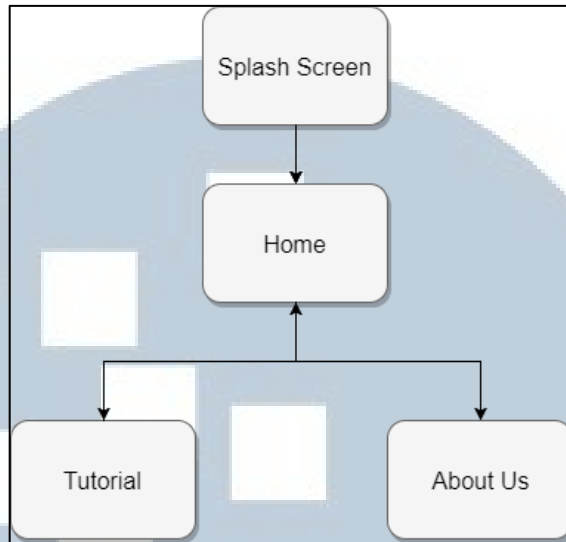


Gambar 3.6 DFD Level 2 Proses Pengaturan Knowledge Chatbot

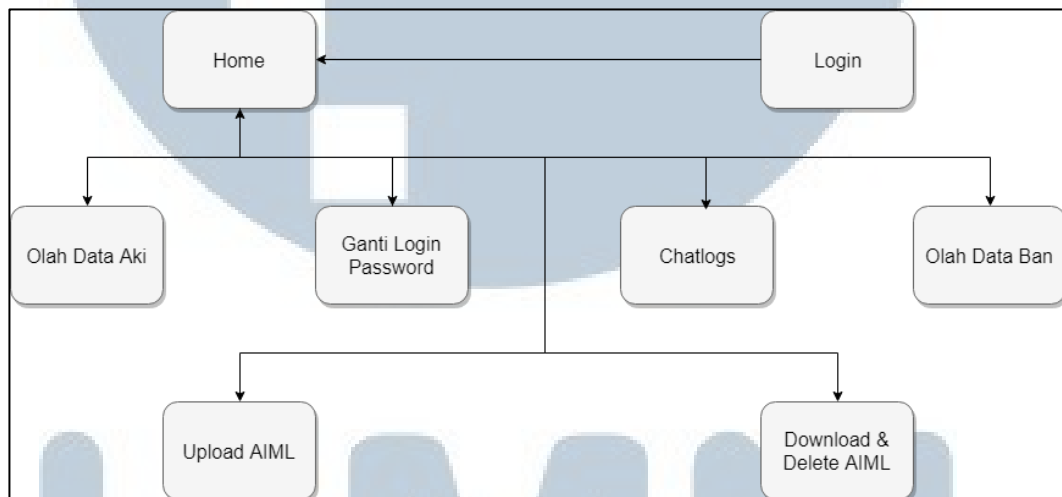
Pada Gambar 3.6 menggambarkan DFD *level 2* proses pengaturan knowledge *chatbot*. Entitas admin dapat melakukan *upload* dan *download* AIML. Tabel yang digunakan pada level ini adalah tabel *aiml* dan *srai_lookup*.

3.2.3 Sitemap

Perancangan aplikasi *chatbot* pada suatu aplikasi berbasis *mobile* (Android) dilakukan dengan membuat rancangan struktur halaman aplikasi. Gambar 3.7 menggambarkan *sitemap* aplikasi *chatbot*. Aplikasi *chatbot* memiliki Halaman Home yang merupakan halaman melakukan proses komunikasi *chatbot*. Dari Halaman Home, *user* dapat bernavigasi ke Halaman Tutorial atau About Us.



Gambar 3.7 Sitemap Aplikasi Chatbot

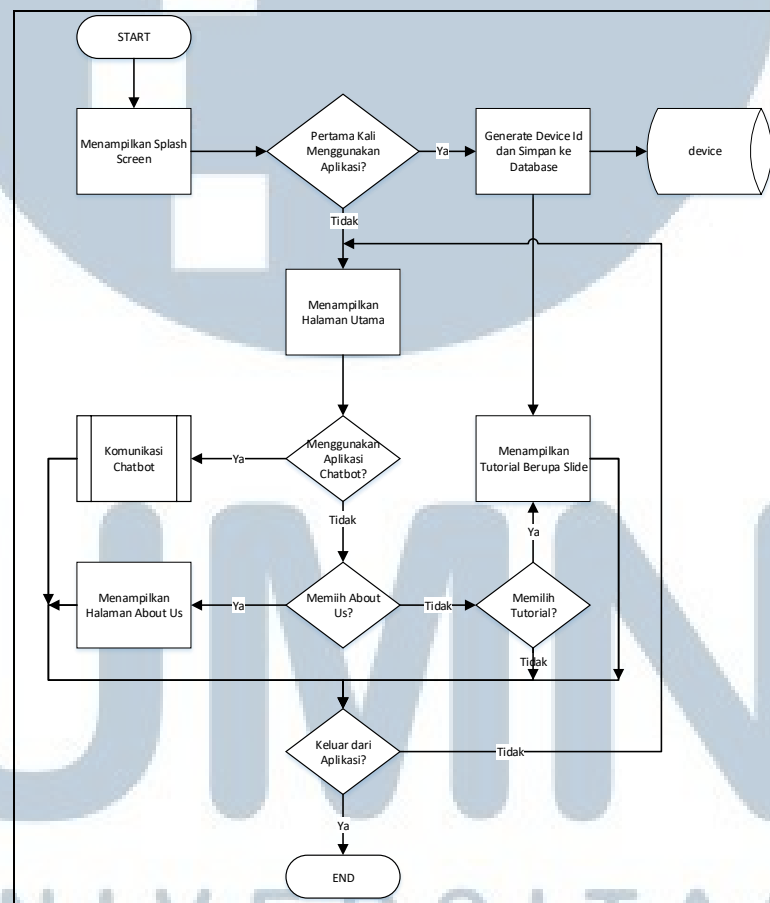


Gambar 3.8 Sitemap Web Admin

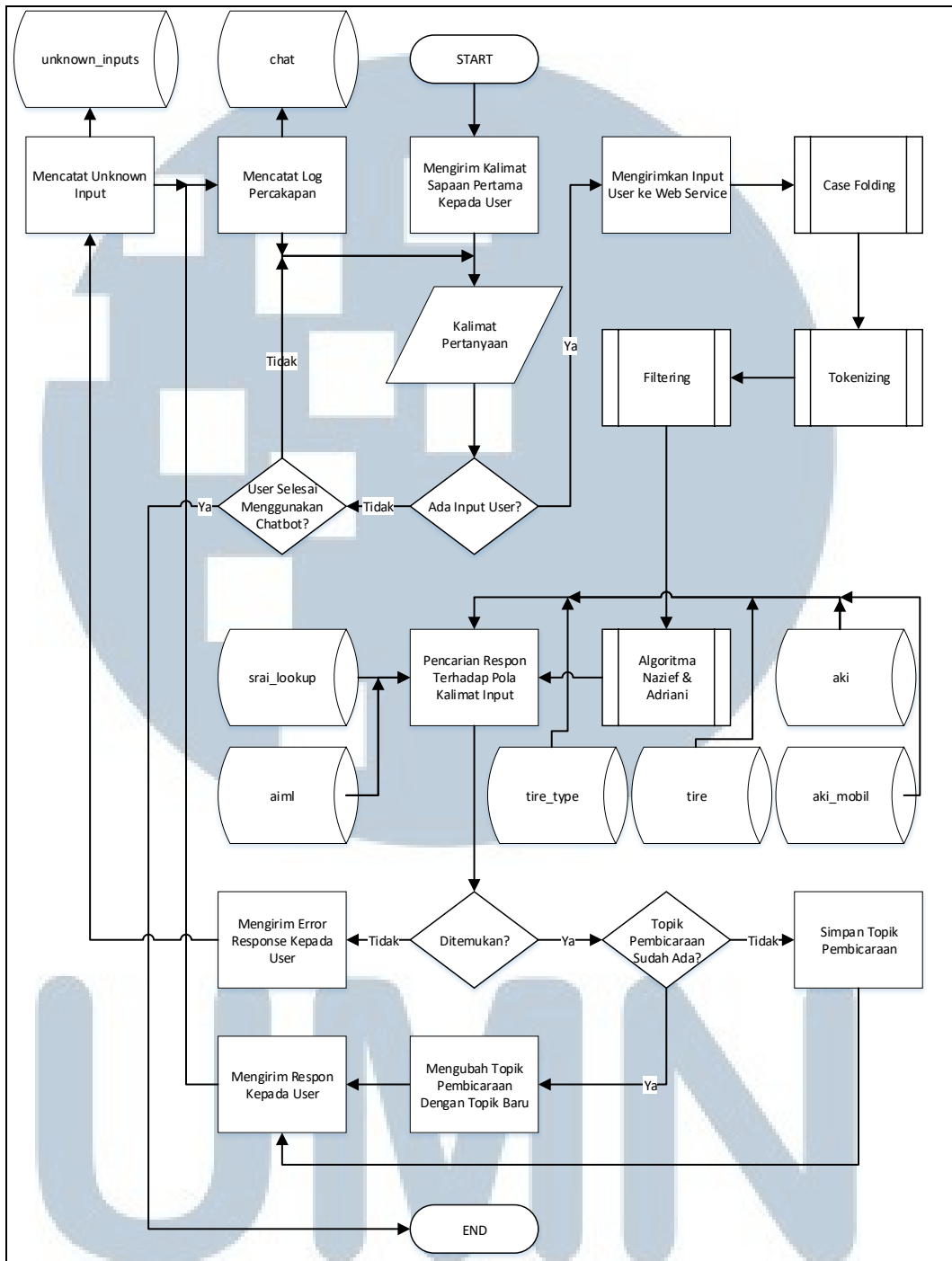
Perancangan aplikasi admin pada suatu *website* dilakukan dengan membuat rancangan struktur halaman *web*. Gambar 3.8 menggambarkan *sitemap web* Admin. *Web* admin memiliki 8 halaman, yaitu Login, Home, Chatlogs, Upload AIML, Download & Delete AIML, Ganti Login Password, Olah Data Ban, dan Olah Data Aki.

3.2.4 Flowchart

Flowchart merupakan diagram yang menunjukkan alur kerja program. Gambar 3.9 merupakan *flowchart main* aplikasi *chatbot* penyedia informasi bengkel yang dirancang dan dibangun pada penelitian ini. Aplikasi untuk *chatbot* berada pada *platform mobile* (Android). Pertama kali *user* membuka aplikasi, *user* akan diberikan *tutorial* singkat dan panduan penggunaan *chat* dengan *bot* dari menggunakan aplikasi *chatbot* dan aplikasi secara otomatis akan melakukan *generate id* untuk disimpan ke dalam tabel *device*.



Gambar 3.9 Flowchart Main Aplikasi Chatbot

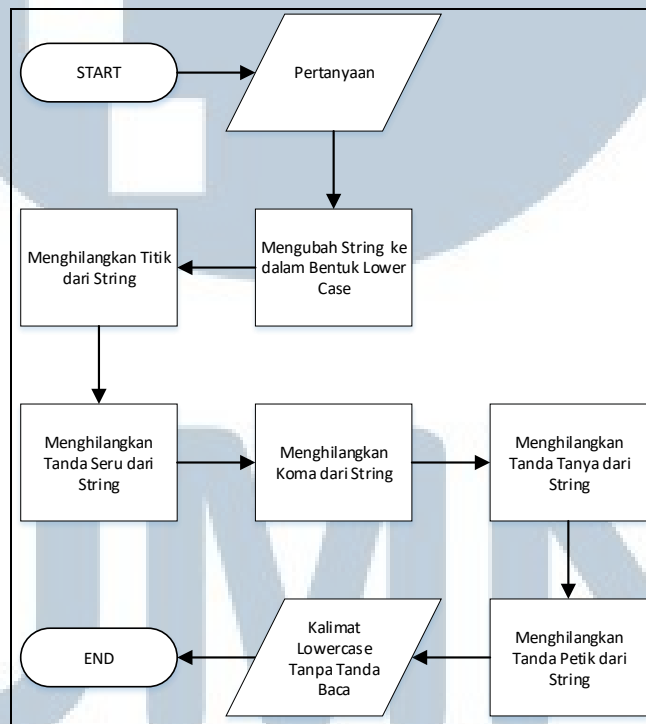


Gambar 3.10 Flowchart Komunikasi Chatbot

Gambar 3.10 menggambarkan *flowchart* komunikasi *chatbot*. Proses ini akan memproses masukan yang diberikan oleh *user* dan mengolah menjadi respon yang akan diterima kembali oleh *user*. Dalam proses ini melibatkan proses *case folding*, *tokenizing*, *fltering*, algoritma Nazief & Adriani yang digunakan untuk

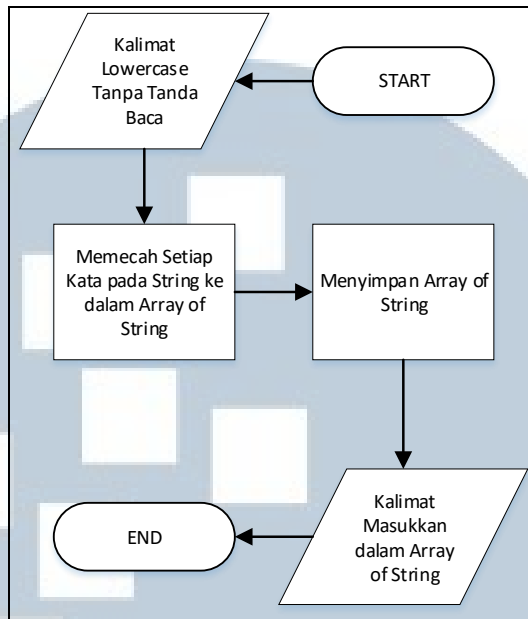
memperoleh kalimat dalam kata dasar. Kalimat dengan kata dasar akan digunakan dalam pencocokan pola dengan data AIML dan SRAI dari tabel aiml dan srai_lookup. Kalimat masukan dari *user* dan respon dari pencocokan AIML dan SRAI akan dicatat ke dalam tabel chat dan apabila ada input yang tidak dapat diterjemahkan maka akan dicatat pada tabel unknown_inputs.

Gambar 3.11 menggambarkan *flowchart case folding*. Pada proses ini kalimat masukan akan dibuat menjadi huruf kecil (*lowercase*). Proses berikutnya adalah menghilangkan tanda baca pada kalimat masukan dan membuat kalimat hanya tersisa kalimat dengan huruf saja.



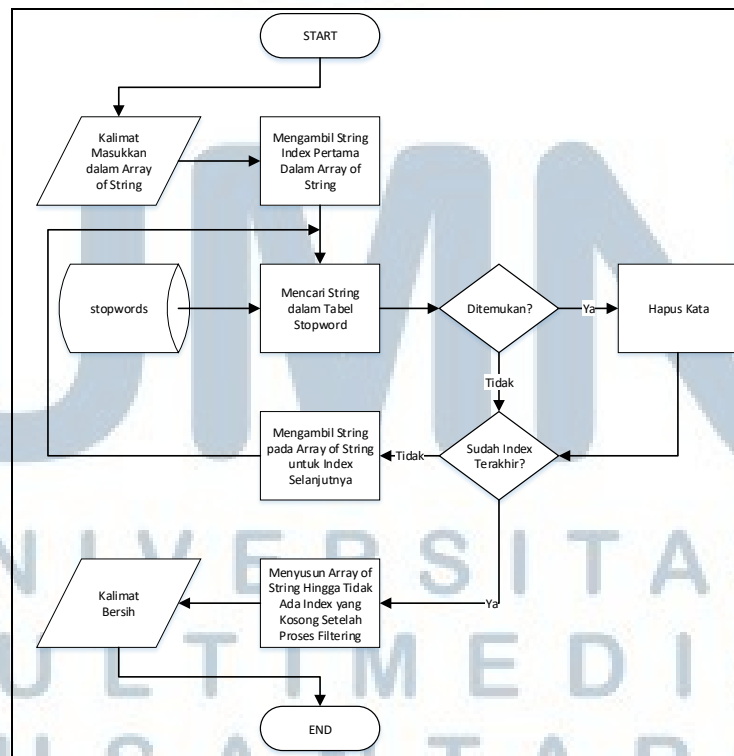
Gambar 3.11 Flowchart Case Folding

UNIVERSITAS
MULTIMEDIA
NUSANTARA



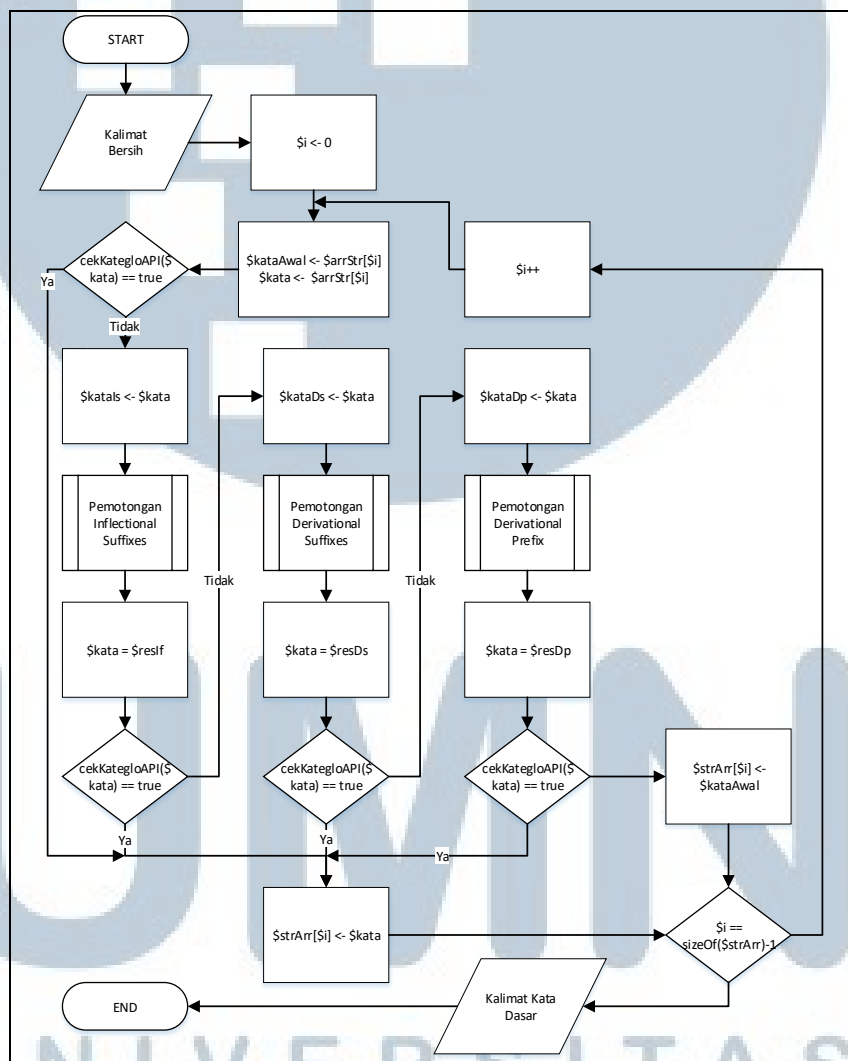
Gambar 3.12 Flowchart Tokenizing

Gambar 3.12 menggambarkan *flowchart tokenizing*. Pada proses ini akan memecah kalimat masukan menjadi *array of string*. *Array of string* akan disimpan untuk diproses pada proses selanjutnya, yaitu proses *filtering*.



Gambar 3.13 Flowchart Filtering

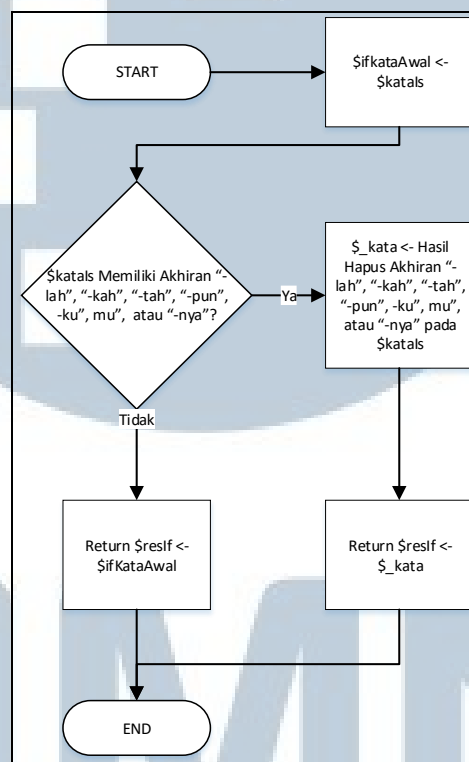
Gambar 3.13 menggambarkan *flowchart filtering*. Pada proses ini *array of string* akan dicek tiap indexnya apakah merupakan kata yang ada dalam tabel stopword atau tidak. Apa bisa memiliki kesamaan dengan data pada tabel stopword, maka *string* pada *index* tersebut akan dihapus. Setelah proses pengecekan kata *stopword*, *array of string* akan diperbaiki urutannya akibat ada *string* pada *index* tertentu yang hilang.



Gambar 3.14 Flowchart Algoritma Nazief & Adriani

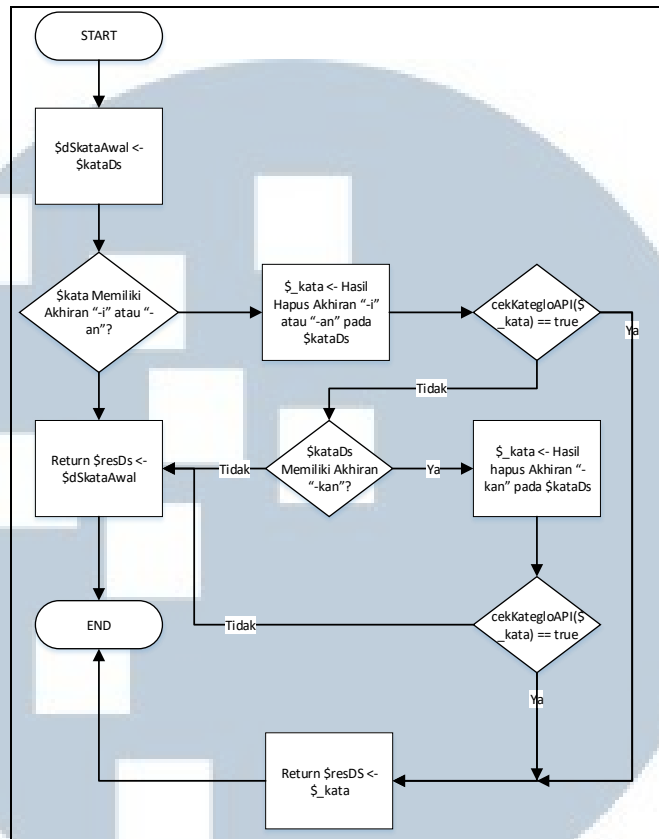
Gambar 3.14 menggambarkan *flowchart* Algoritma Nazief & Adriani. *Array of string* hasil dari proses *filtering* akan dilakukan pemotongan awalan dan akhiran. Proses pemotongan awalan yang pertama adalah *inflectional suffixes*. Apa

bila ditemukan maka akan disimpan ke strArr[\$i]. Apa bila tidak ditemukan maka akan lanjut ke proses pemotongan *derivational suffix*. Apabila ditemukan maka akan disimpan ke strArr[\$i]. Apa bila tidak ditemukan maka akan lanjut ke proses pemotongan *derivational prefix*. Hasil dari pemotongan ini akan selalu disimpan karena sudah merupakan pemotongan terakhir. Setelah penyimpanan ke strArr[\$i] maka akan \$i akan dicek apakah sudah sampai index kurang 1 dari seluruh jumlah \$strArr[].



Gambar 3.15 Flowchart Proses Pemotongan Inflectional Suffixes

Gambar 3.15 menggambarkan *flowchart* proses pemotongan *inflectional suffixes* pada algoritma Nazief & Adriani. Kata dengan akhiran “-lah”, “-kah”, “-tah”, “-pun”, “-ku”, “mu”, atau “-nya” akan dihapus awalnya dan dikembalikan (*return*) dari proses ini. Apa bila tidak ditemukan maka akan mengembalikan \$isKataAwal.

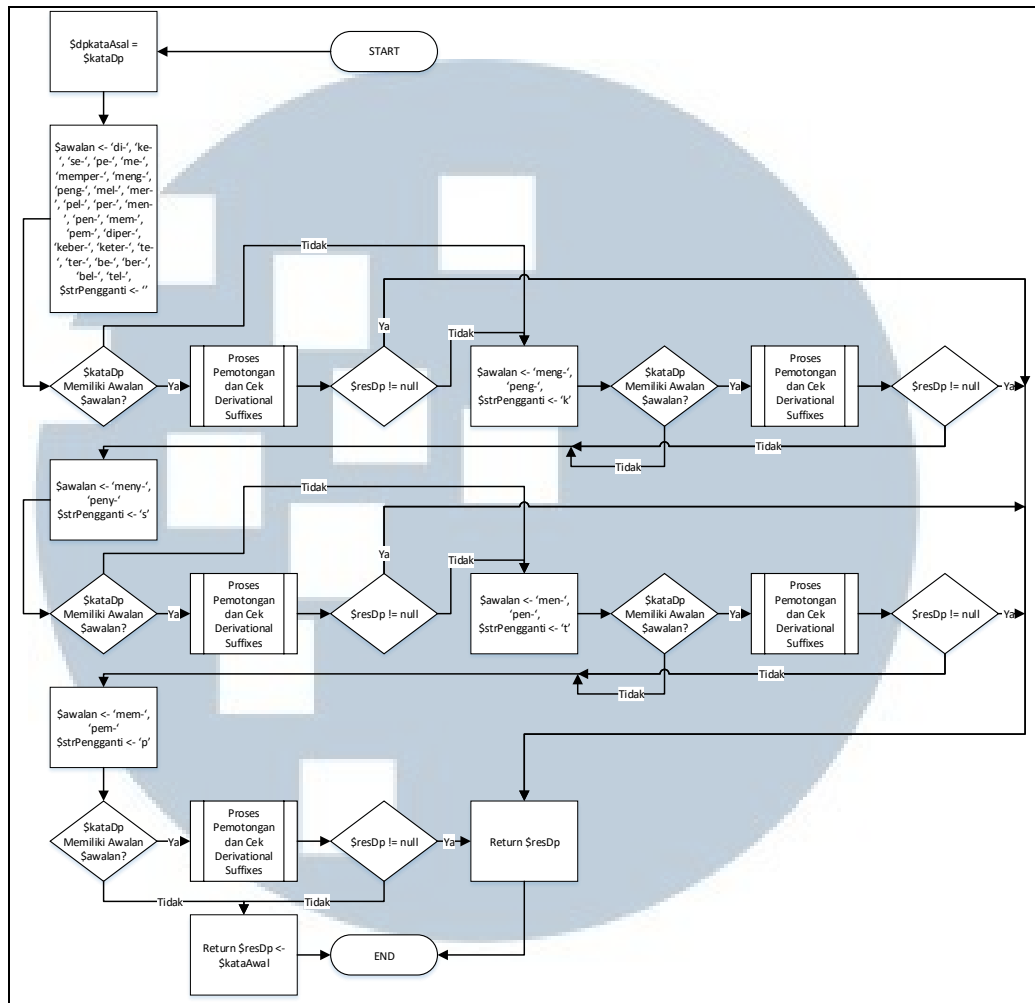


Gambar 3.16 Flowchart Proses Pemotongan Derivational Suffixes

Gambar 3.16 menggambarkan *flowchart* proses pemotongan *derivational suffixes* pada algoritma Nazief & Adriani. Kata dengan akhiran “-lah”, “-kah”, “-tah”, “-pun”, “-ku”, “mu”, atau “-nya” akan dihapus awalnya dan dikembalikan (*return*) dari proses ini. Jika kata tidak ditemukan maka mengembalikan *SifKataAwal*.

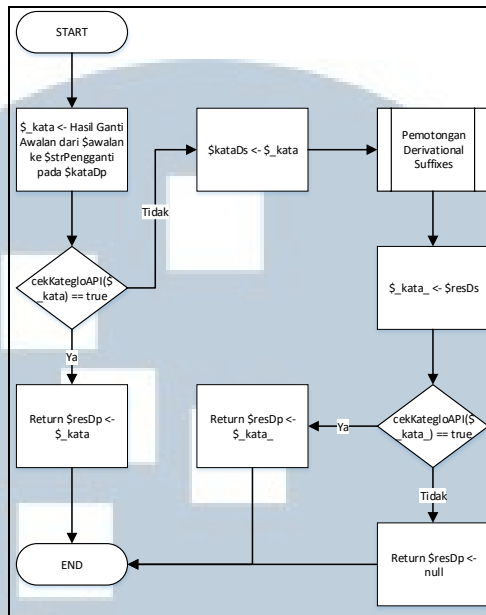
Gambar 3.17 menggambarkan *flowchart* proses pemotongan *derivational prefix* pada algoritma Nazief & Adriani. Kata dengan awalan ‘di-’, ‘ke-’, ‘se-’, ‘pe-’, ‘me-’, ‘memper-’, ‘meng-’, ‘peng-’, ‘mel-’, ‘mer-’, ‘pel-’, ‘per-’, ‘men-’, ‘pen-’, ‘mem-’, ‘pem-’, ‘diper-’, ‘keber-’, ‘keter-’, ‘te-’, ‘ter-’, ‘be-’, ‘ber-’, ‘bel-’, atau ‘tel-’ akan diproses pada proses pemotongan dan cek *derivational suffixes* yang digambarkan pada Gambar 3.17. Jika hasil *return* pada proses pemotongan dan cek

derivational suffixes tidak kosong, maka akan mengembalikan (return) hasilnya dari proses ini. Jika tidak ditemukan maka akan melakukan proses pencarian berikutnya. Proses pencarian berikutnya akan mengganti awalan ‘meng-‘ dan ‘peng-‘ dengan huruf ‘k’ dan melakukan proses pemotongan dan cek *derivational suffixes*. Jika hasil *return* pada proses pemotongan dan cek *derivational suffixes* tidak kosong, maka akan mengembalikan (return) hasilnya dari proses ini. Jika tidak ditemukan maka akan melakukan proses pencarian berikutnya. Proses pencarian berikutnya akan mengganti awalan ‘meny-‘ dan ‘peny-‘ dengan huruf ‘s’ dan melakukan proses pemotongan dan cek *derivational suffixes*. Jika hasil *return* pada proses pemotongan dan cek *derivational suffixes* tidak kosong, maka akan mengembalikan (return) hasilnya dari proses ini. Jika tidak ditemukan maka akan melakukan proses pencarian berikutnya. Proses pencarian berikutnya akan mengganti awalan ‘men-‘ dan ‘pen-‘ dengan huruf ‘t’ dan melakukan proses pemotongan dan cek *derivational suffixes*. Jika hasil *return* pada proses pemotongan dan cek *derivational suffixes* tidak kosong, maka akan mengembalikan (return) hasilnya dari proses ini. Jika tidak ditemukan maka akan melakukan proses pencarian berikutnya. Proses pencarian berikutnya akan mengganti awalan ‘mem-‘ dan ‘pem-‘ dengan huruf ‘p’ dan melakukan proses pemotongan dan cek *derivational suffixes*. Jika hasil *return* pada proses pemotongan dan cek *derivational suffixes* tidak kosong, maka akan mengembalikan (return) hasilnya dari proses ini. Jika tidak ditemukan maka akan mengembalikan \$dpKataAsal.

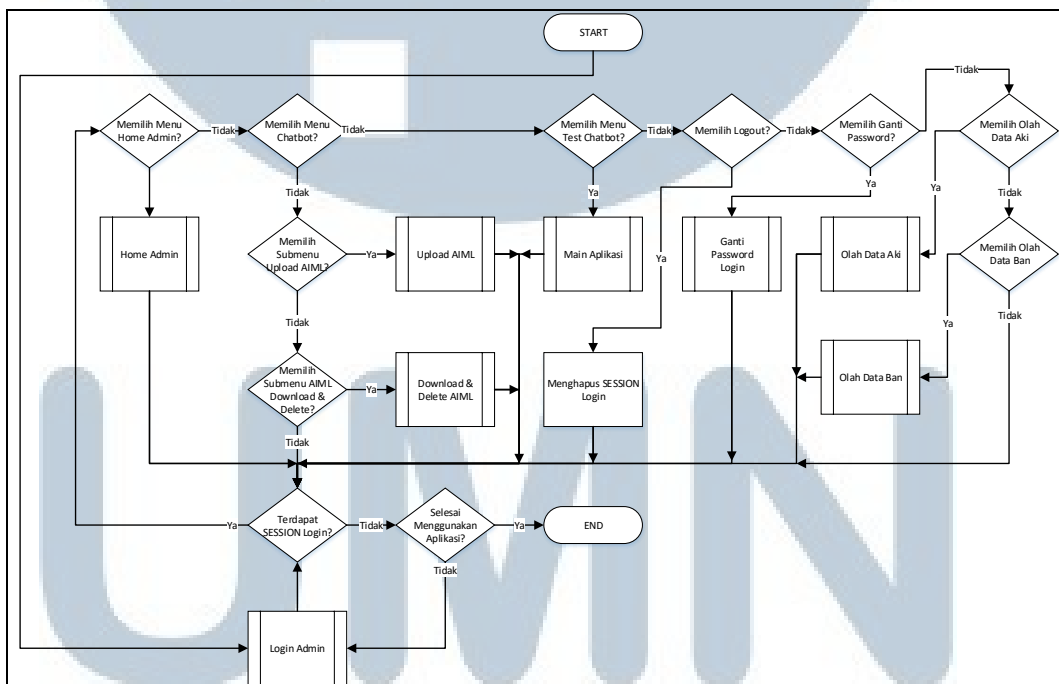


Gambar 3.17 Flowchart Proses Pemotongan Derivational Prefix

Gambar 3.18 menggambarkan *flowchart* proses pemotongan dan cek *derivational suffixes*. Pada proses isi \$kataDP akan yang memiliki awalan pada \$awalan akan digantikan dengan \$strPengganti. Kata tersebut akan dicek pada Kateglo API untuk mendapatkan kondisi sudah kata dasar atau belum bisa sudah maka kata akan dikembalikan (*return*). Jika tidak maka akan melakukan proses pemotongan *derivational suffixes* dan melakukan cek kata dasar ke Kateglo API. Jika ditemukan maka akan *return* hasilnya dan bila tidak akan *return null* atau kosong.



Gambar 3.18 Flowchart Proses Pemotongan dan Cek Derivational Suffixes

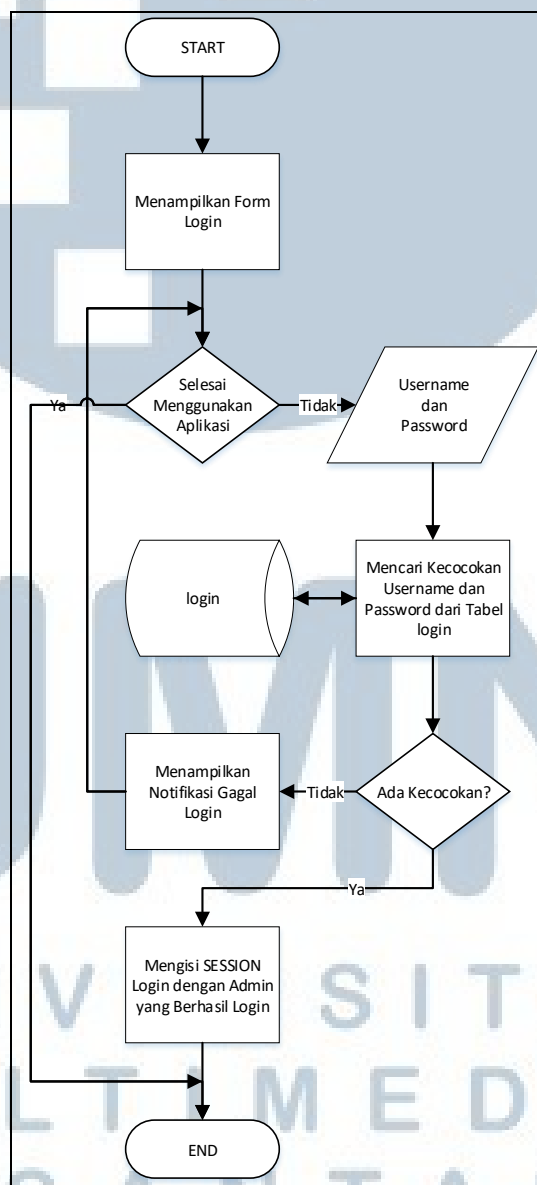


Gambar 3.19 Flowchart Main Web Admin

Gambar 3.19 menggambarkan *flowchart main web admin*. Pertama-tama admin harus melakukan proses *login* agar mendapatkan *session* agar dapat mendapat akses ke tiap halaman pada *web admin*. Admin dapat melakukan proses

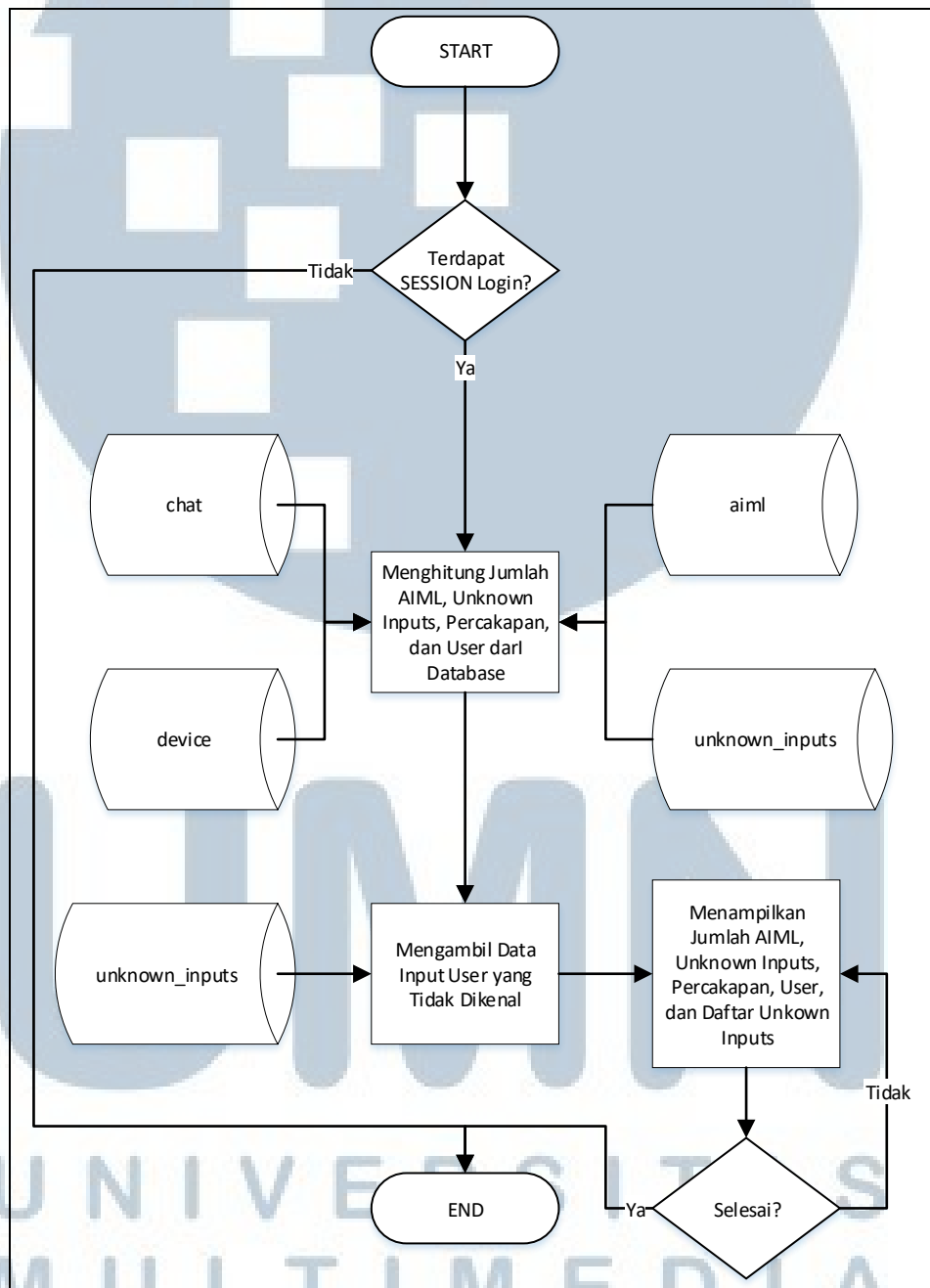
home admin, upload AIML, download & delete AIML, olah data ban, olah data aki, dan melakukan *log out*.

Gambar 3.20 menggambarkan *flowchart login* admin. Pada proses ini admin akan memberikan masukan berupa *username* dan *password*. Aplikasi akan melakukan pencocokan dengan tabel login dari input yang admin lakukan. Apa bila berhasil maka akan mendapatkan *session* dan bila gagal maka akan mendapatkan notifikasi gagal *login*.

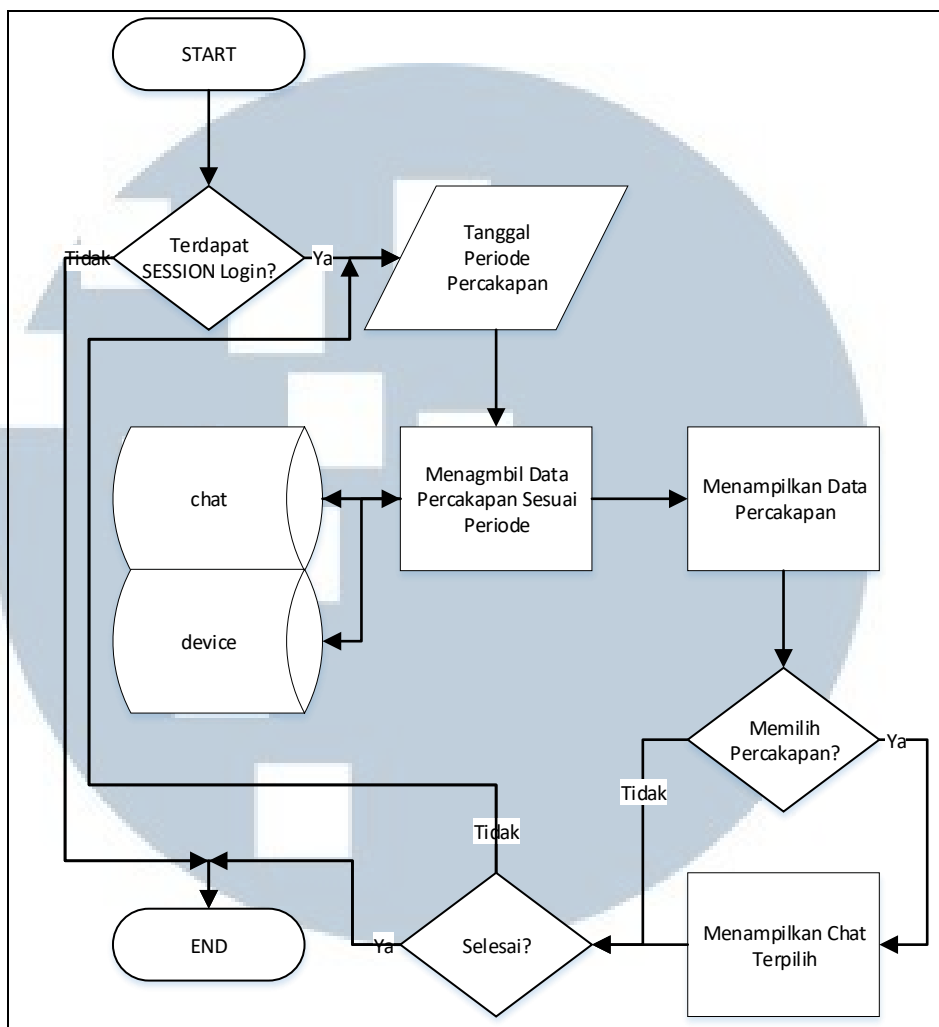


Gambar 3.20 Flowchart Login Admin

Gambar 3.21 menggambarkan *flowchart home admin*. Pada proses ini admin dapat melihat jumlah AIML, *unknown inputs*, dan percakapan, serta seluruh *unknown inputs*. Tabel yang digunakan pada proses ini adalah tabel *aiml*, tabel *unknown_inputs*, tabel *chat*, dan tabel *device*.



Gambar 3.21 Flowchart Home Admin

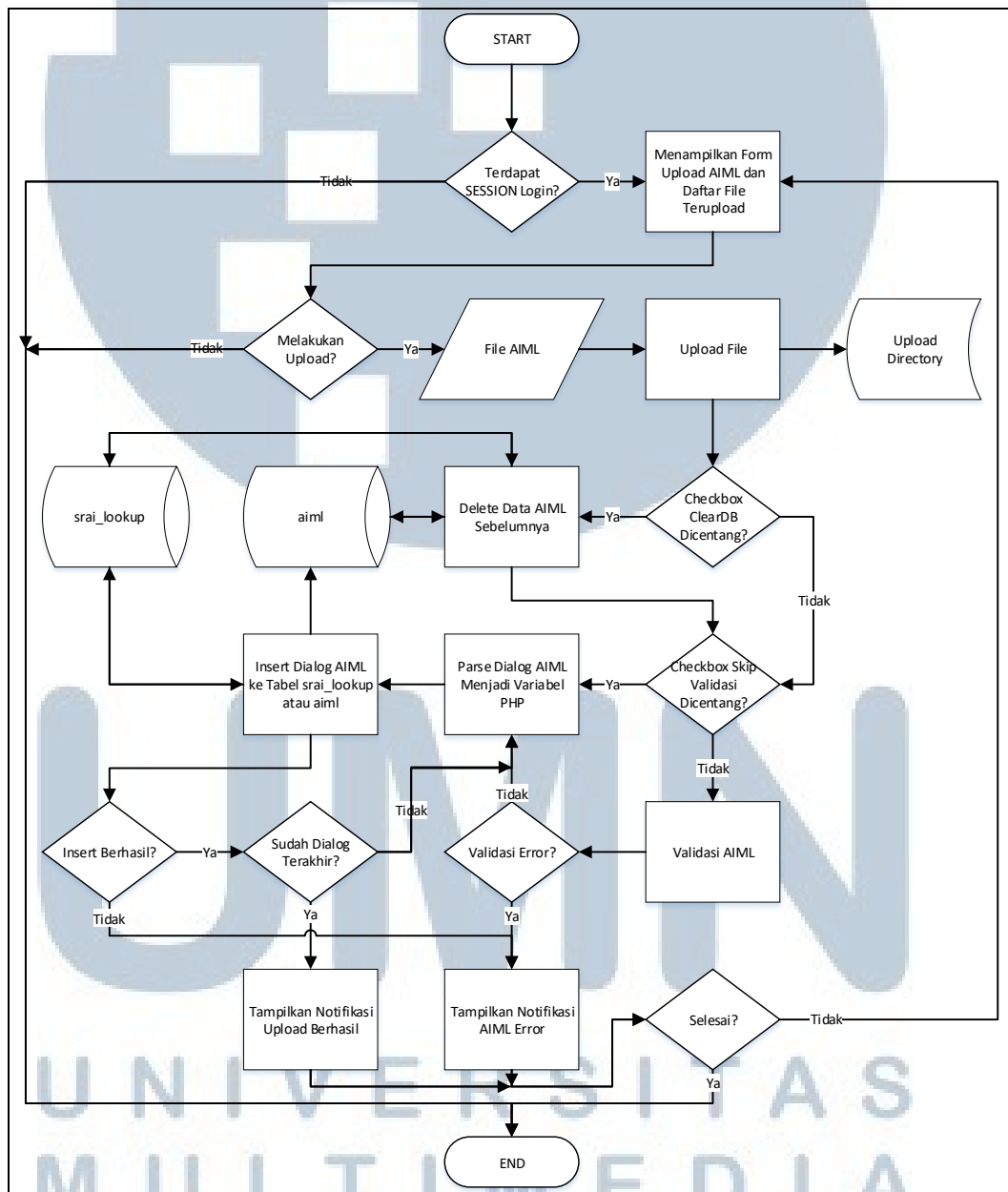


Gambar 3.22 Flowchart Chatlog

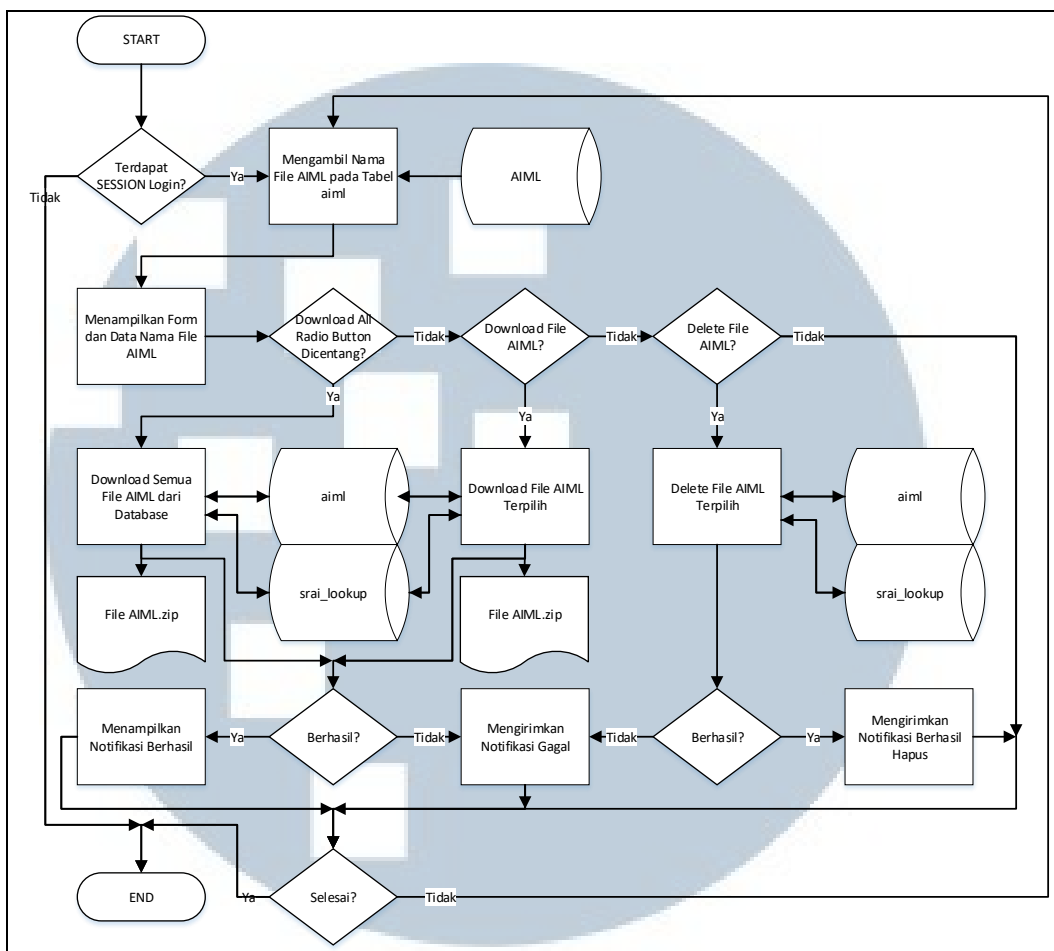
Gambar 3.22 menggambarkan *flowchart chatlog*. Tujuan proses ini adalah dapat melihat percakapan yang terjadi yang dilakukan oleh aplikasi *chatbot*. Pada proses ini admin harus memasukkan tanggal periode yang ingin dilihat. Maka akan muncul seluruh *device* yang melakukan percakapan selama periode yang dimasukan. Admin akan memilih *device* mana yang ingin dilihat. Jika terpilih *device*, maka akan muncul seluruh percakapan yang terjadi.

Gambar 3.23 menggambarkan *flowchart upload AIML*. Pada proses ini menampilkan *file* AIML apa saja yang sudah tercatat pada aplikasi. Admin harus melakukan *upload file* AIML ke aplikasi. Admin dapat memilih apakah mau

menghapus *file* lama atau tidak dan dapat memilih apakah melakukan validasi AIML atau tidak. Proses memasukan ke dalam *database* akan selalu dicek keberhasilannya. Apabila gagal maka akan diberi notifikasi. Jika seluruh proses berhasil, aplikasi akan memberi notifikasi keberhasilan kepada admin atau bila gagal maka akan ada notifikasi gagal.



Gambar 3.23 Flowchart Upload AIML

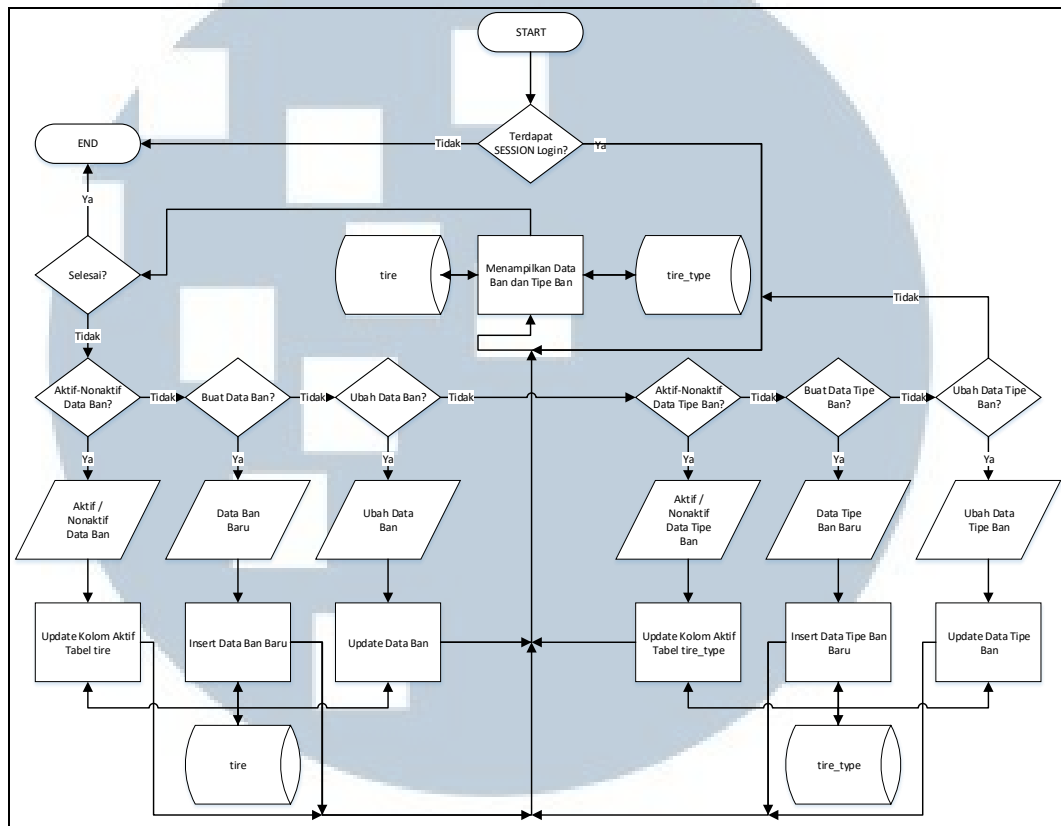


Gambar 3.24 Flowchart Download & Delete AIML

Gambar 3.24 menggambarkan *flowchart download & delete AIML*. Pada proses ini menampilkan *file AIML* apa saja yang sudah tercatat pada aplikasi. Admin dapat memilih ingin *download* seluruh *file* atau memilih *download file* terpilih saja, atau menghapus *file AIML* terpilih. *File AIML* yang dapat diunduh diperoleh dari data pada tabel *aiml* dan *srai_lookup* yang diubah menjadi *file AIML*. Keberhasilan akan dikonfirmasi kepada admin.

Gambar 3.25 menggambarkan *flowchart* olah data ban. Pada proses ini akan diawali dengan menampilkan data ban dan data tipe ban. Data ban adalah kumpulan ban dengan berbagai macam ukurannya. Data tipe ban adalah kumpulan merek dan

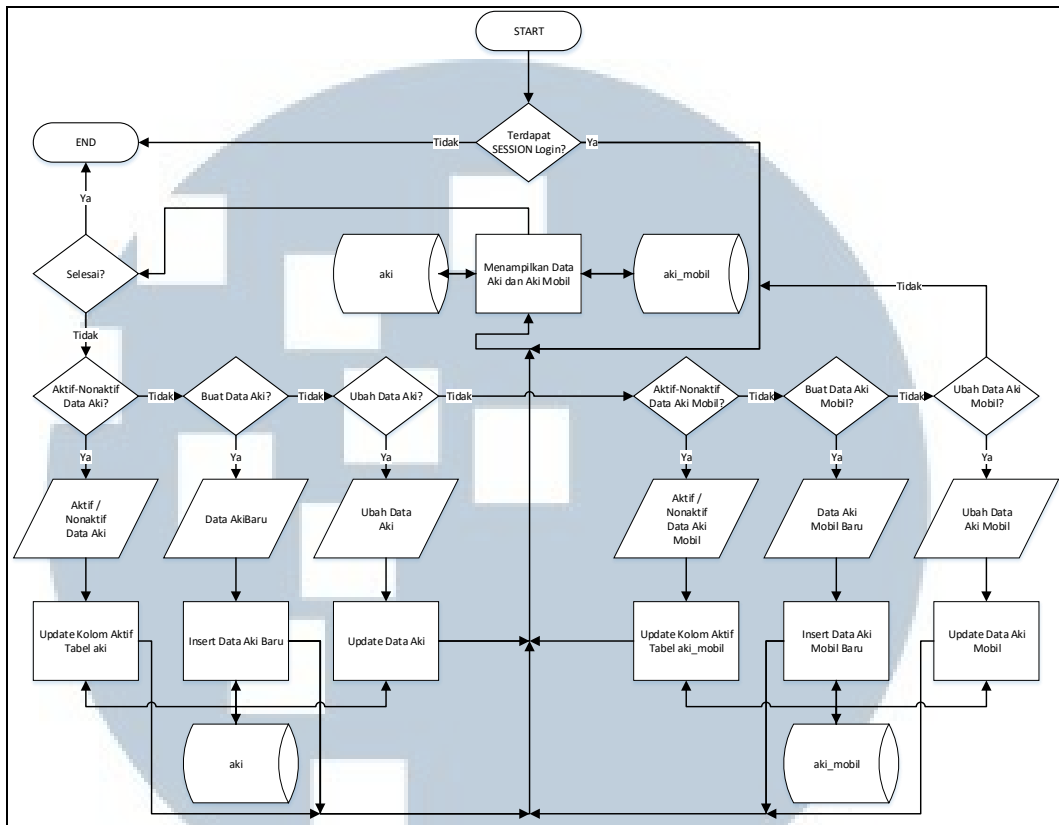
informasi produk dari tiap tipe ban. Kedua data tersebut dapat dibuat data baru, diperbarui datanya, dan melakukan aktif/non-aktif dari tiap datanya.



Gambar 3.25 Flowchart Olah Data Ban

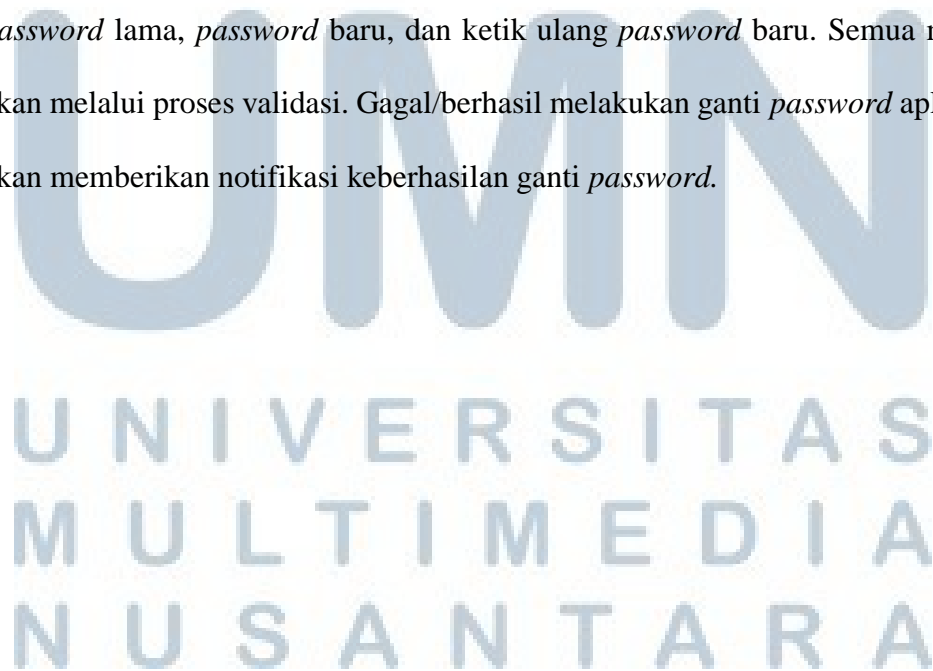
Gambar 3.26 menggambarkan *flowchart* olah data aki. Pada proses ini akan diawali dengan menampilkan data aki dan data aki mobil. Data aki adalah kumpulan aki dengan berbagai merek, ukuran, dan spesifikasinya. Data aki mobil adalah kumpulan kecocokan mobil dengan tipe aki yang digunakan. Kedua data tersebut dapat dibuat data baru, diperbarui datanya, dan melakukan aktif/non-aktif dari tiap datanya.

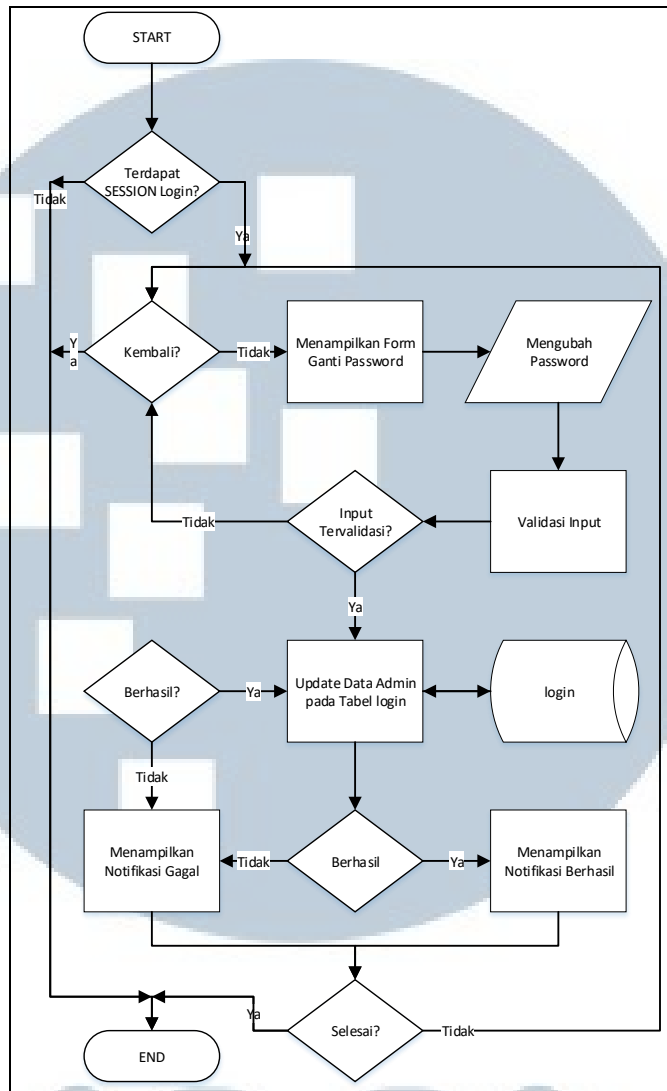
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.26 Flowchart Olah Data Aki

Gambar 3.27 menggambarkan *flowchart* ganti *password login*. Pada halaman ini admin dapat mengganti *password* dengan cara mengisi *text input* untuk *password* lama, *password* baru, dan ketik ulang *password* baru. Semua masukan akan melalui proses validasi. Gagal/berhasil melakukan ganti *password* aplikasi ini akan memberikan notifikasi keberhasilan ganti *password*.

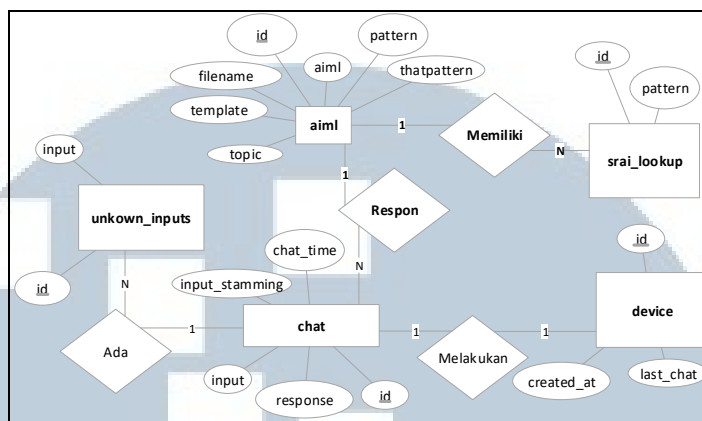




Gambar 3.27 Flowchart Ganti Password Login

3.2.5 Entity Relationship Diagram (ERD)

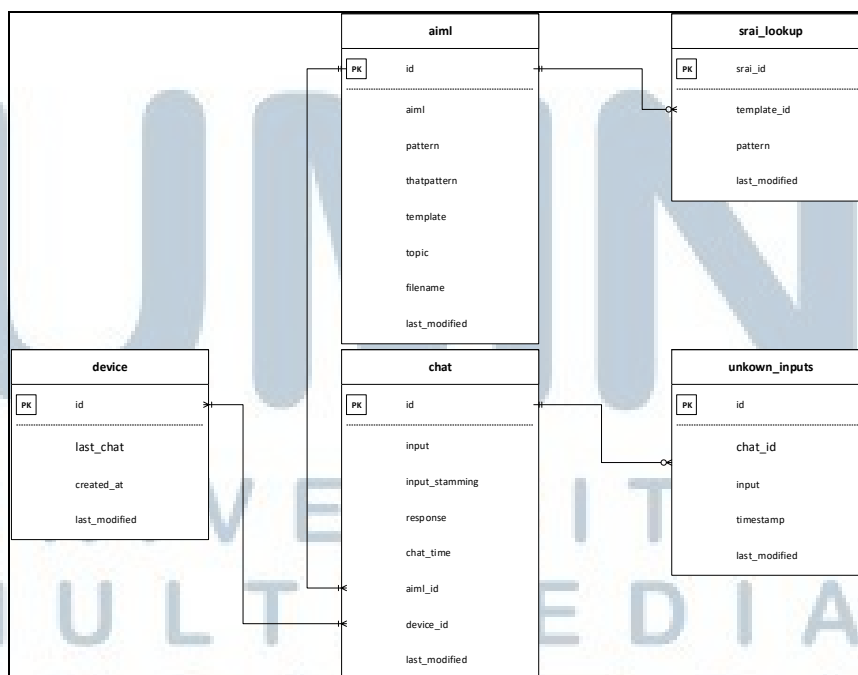
Entity Relationship Diagram (ERD) digunakan untuk menggambarkan hubungan antara suatu entitas dengan entitas lainnya. Terdapat 5 entitas pada Aplikasi Chatbot Penyedia Informasi Bengkel yang ditunjukkan pada Gambar 3.28. Entitas tersebut adalah *unknown_inputs*, *chat*, *aiml*, *srai_lookup*, dan *device*. Relasi *one to many* terdapat pada relasi *chat* dengan *unknown_inputs*, relasi *chat* dengan *device*, dan relasi *aiml* dengan *srai_lookup*. Relasi *many to one* terdapat pada relasi *chat* dengan *aiml*. Relasi *one to one* terdapat pada relasi *chat* dengan *device*.



Gambar 3.28 ERD Aplikasi Chatbot Penyedia Informasi Bengkulu

3.2.6 Database Schema

Berdasarkan ERD pada Gambar 3.28, maka dirumuskan *database schema* yang menunjukkan bentuk hubungan tabel dengan tabel lain. *Database schema* Aplikasi Chatbot Penyedia Informasi Bengkulu digambarkan pada Gambar 3.29. Aplikasi ini memiliki 4 tabel yang saling berelasi satu dengan yang lainnya, yang terdiri dari tabel device, chat, aiml, srai_lookup, dan unkown_inputs



Gambar 3.29 Database Schema Aplikasi Chatbot Penyedia Informasi Bengkulu

3.2.7 Struktur Tabel

Struktur tabel pada Aplikasi Chatbot Penyedia Informasi Bengkel adalah sebagai berikut.

a. Tabel aiml

Tabel 3.1 Struktur Tabel aiml

Name	Type	Default	Other
id	int	None	Primary Auto Increment
pattern	varchar(255)	None	
thatpattern	varchar(255)	None	
template	text	None	Unique
topic	varchar(255)	None	
filename	varchar(255)	None	
last_modified	timestamp	CURRENT_TIMESTAMP	

Tabel 3.1 menunjukkan struktur tabel aiml. Tabel ini memiliki 7 kolom. Tabel ini memiliki 1 *primary key*, yaitu kolom “id”. Kolom “pattern” menyimpan pola masukan *user*. Kolom “thatpattern” digunakan untuk menyimpan data pada tag <that> pada *file* AIML. Kolom “template” digunakan untuk menyimpan respon yang akan diberikan ke *user*. Kolom “topic” digunakan untuk menyimpan topik pembicaraan. Kolom “filename” untuk menyimpan nama *raw file* dari *file* AIML saat melakukan *upload*. Pada Tabel 3.1 hanya kolom “thatpattern” yang boleh *null*.

b. Tabel chat

Tabel 3.2 menunjukkan struktur tabel chat. Tabel ini memiliki 8 kolom. *primary key* dari tabel ini hanya kolom “id”. Kolom “input” digunakan untuk menyimpan masukan *user*. Kolom “response” digunakan untuk menyimpan jawaban dari AIML. Kolom “input_stemming” digunakan untuk menyimpan masukan *user* setelah proses *stemming*. Kolom “aiml_id” merupakan *foreign key* dari tabel aiml. Kolom “device_id” merupakan *foreign key* dari tabel device. Kolom

“chat_time” digunakan untuk menyimpan waktu percakapan terjadi. Semua kolom pada tabel ini tidak boleh *null*.

Tabel 3.2 Struktur Tabel chat

Name	Type	Default	Other
id	int	None	Primary Auto Increment
input	text	None	Unique
response	longtext	None	
input_stemming	text	None	Unique
aiml_id	int	None	Foreign Key
device_id	int	None	Foreign Key
chat_time	datetime	None	
last_modified	timestamp	CURRENT_TIMESTAMP	

c. Tabel stopwords

Tabel 3.3 menunjukkan struktur tabel stopwords. Tabel ini memiliki 2 kolom. Kolom “id” merupakan *primary key*. Kolom “stopword” digunakan untuk menyimpan *stopwords*. Semua kolom pada tabel ini tidak boleh *null*.

Tabel 3.3 Struktur Tabel stopwords

Name	Type	Default	Other
id	int	None	Primary Auto Increment
stopword	varchar(50)	None	

d. Tabel srail_lookup

Tabel 3.4 Struktur Tabel srail_lookup

Name	Type	Default	Other
id	int	None	Primary Auto Increment
pattern	Text	None	Unique
aiml_id	int	None	Foreign Key

Tabel 3.4 menunjukkan struktur tabel srail_lookup. Tabel ini terdapat 3 kolom. Kolom “id” merupakan *primary key*. Kolom “pattern” digunakan untuk menyimpan pola masukan *user*. Kolom “aiml_id” merupakan *foreign key* dari tabel aiml. Semua kolom pada tabel ini tidak boleh *null*.

e. Tabel unknown_inputs

Tabel 3.5 Struktur Tabel unknown_inputs

Name	Type	Default	Other
id	int	None	Primary Auto Increment
input	Text	None	
chat_id	int	None	Foreign Key
last_modified	timestamp	CURRENT_TIMESTAMP	

Tabel 3.5 menunjukkan struktur tabel unknown_inputs. Tabel ini terdiri dari 4 kolom. Kolom “id” merupakan *primary key*. Kolom “input” digunakan untuk menyimpan masukan yang tidak dapat dijawab oleh aplikasi. Kolom “chat_id” merupakan *foreign key* dari tabel chat. Semua kolom pada tabel ini tidak boleh *null*.

f. Tabel device

Tabel 3.6 menunjukkan struktur tabel device. Tabel ini memiliki 4 kolom. Kolom “id” merupakan *primary key*. Kolom “last_chat” digunakan untuk menyimpan kapan *device* melakukan *chat* terakhir. Kolom “created_at” digunakan untuk menyimpan kapan *device* mengakses Aplikasi Chatbot Penyedia Informasi Bengkel pertama kali. Semua kolom pada tabel ini tidak boleh *null*.

Tabel 3.6 Struktur Tabel device

Name	Type	Default	Other
id	varchar(255)	None	Primary
last_chat	datetime	None	
created_at	datetime	None	
last_modified	timestamp	CURRENT_TIMESTAMP	

g. Tabel login

Tabel 3.7 menunjukkan struktur tabel login. Tabel ini memiliki 5 kolom. Kolom “user_name” merupakan *primary key*. Kolom “password” digunakan untuk menyimpan kata sandi. Kolom “last_ip” digunakan untuk menyimpan ip terakhir

admin melakukan login ke *web* pengaturan *chatbot*. Kolom “last_login” digunakan untuk menyimpan kapan admin melakukan login ke *web* pengaturan *chatbot*.

Tabel 3.7 Struktur Tabel login

Name	Type	Default	Other
user_name	varchar(255)	None	Primary
password	varchar(255)	None	
last_ip	varchar(25)		
last_login	timestamp	None	
last_modified	timestamp	CURRENT_TIMESTAMP	

h. Tabel tire

Tabel 3.8 Struktur Tabel tire

Name	Type	Default	Other
cai	varchar(10)		Primary
rim	int		
section_width	int		
aspect_ratio	int		
li	int		
si	varchar(5)		
marking	varchar(10)		
car_fitment	text		
currency	varchar(10)		
price	int		
type_id	int		
active	int		
last_modified	timestamp	CURRENT_TIMESTAMP	

Tabel 3.8 menunjukkan struktur tabel tire. Tabel ini terdiri dari 13 kolom. Kolom “cai” adalah kode unik untuk setiap ukuran ban dan setiap merek ban. Kolom “rim”, “section_width”, “aspect_ratio”, “li”, “si”, dan “marking” adalah kolom-kolom yang menjelaskan spesifikasi setiap ban. Kolom “car_fitment” adalah kolom yang menandakan ban tersebut cocok/dapat digunakan pada mobil apa. Kolom “currency” dan “price” adalah kolom yang menandakan harga ban tersebut. Kolom “type_id” adalah kode tipe dan ukuran ban tersebut. Kolom “active” adalah kolom yang menandakan data itu dapat di akses oleh pencarian data pada *chatbot*.

i. Tabel tire_type

Tabel 3.9 Struktur Tabel tire_type

Name	Type	Default	Other
id	int		Primary
type	varchar(500)		
excellence	text		
link	varchar(500)		
img	varchar(500)		
active	int		
last_modified	timestamp	CURRENT_TIMESTAMP	

Tabel 3.9 menunjukkan struktur tabel tire_type. Tabel ini terdiri dari 7 kolom. Kolom “id” merupakan *primary key* dari tabel ini. Kolom “type” adalah kolom menunjukkan nama merek dan tipe dari ban. Kolom “excellence” adalah kolom yang menjelaskan keunggulan sebuah tipe ban. Kolom “link” adalah kolom yang berisikan alamat *url* yang berisi informasi suatu tipe ban. Kolom “img” adalah kolom yang berisikan alamat *url* yang berisi gambar dari suatu tipe ban. Kolom “active” adalah kolom yang menandakan data itu dapat di akses oleh pencarian data pada *chatbot*.

j. Tabel aki

Tabel 3.10 Struktur Tabel aki

Name	Type	Default	Other
id	int		Primary
merk	int		
tipe_aki	int		
volt	int		
kap	int		
panjang	varchar(5)		
lebar	varchar(10)		
tinggi	text		
price	varchar(10)		
active	int		
last_modified	timestamp	CURRENT_TIMESTAMP	

Tabel 3.10 menunjukkan struktur tabel aki. Tabel ini terdiri dari 11 kolom. Kolom “id” merupakan *primary key* dari tabel ini. Kolom “merk”, “tipe_aki”,

“volt”, “kap”, “panjang”, “lebar”, dan “tinggi” merupakan kolom-kolom yang menjelaskan spesifikasi suatu aki. Kolom “price” adalah kolom yang menunjukkan harga dari suatu aki. Kolom “active” adalah kolom yang menandakan data itu dapat di akses oleh pencarian data pada *chatbot*.

k. Tabel aki_mobil

Tabel 3.11 Struktur Tabel aki_mobil

Name	Type	Default	Other
id	int		Primary
mobil	varchar(500)		
tipe_aki	varchar(500)		
active	int		
last_modified	timestamp	CURRENT_TIMESTAMP	

Tabel 3.11 menunjukkan struktur tabel aki_mobil. Kolom “id” merupakan *primary key* dari tabel ini. Kolom “mobil” dan “tipe_aki” adalah kolom yang berisi kecocokan tipe mobil dengan tipe aki yang digunakan. Kolom “active” adalah kolom yang menandakan data itu dapat di akses oleh pencarian data pada *chatbot*.

3.2.8 Perancangan Antarmuka Pengguna

Perancangan antar muka Aplikasi Chatbot Penyedia Informasi Bengkel dibagi menjadi untuk aplikasi *chatbot* dalam aplikasi *mobile* (Android) dan aplikasi pengaturan *chatbot* dalam *website*.

a. Antarmuka Aplikasi Chatbot

Gambar 3.30 menggambarkan *wireframe* Halaman Splash Screen. Halaman ini akan memberikan animasi saat *user* membuka aplikasi. Halaman ini akan tampil dengan waktu 3 detik dan langsung menuju halaman berikutnya.

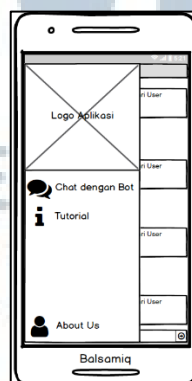


Gambar 3.30 Wireframe Halaman Splash Screen



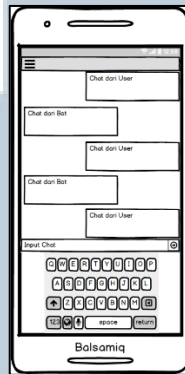
Gambar 3.31 Wireframe Halaman Tutorial

Gambar 3.31 menggambarkan *wireframe* Halaman Tutorial. Halaman ini akan menampilkan cara menggunakan Aplikasi Chatbot Penyedia Informasi Bengkel serta bantuan apabila *user* mengalami kesulitan saat berkomunikasi dengan *bot*. Halaman ini akan muncul ketika *user* baru pertama melakukan pemasangan/*install* aplikasi ini atau *user* melakukan navigasi dari *sidemenu* pada Halaman Home.



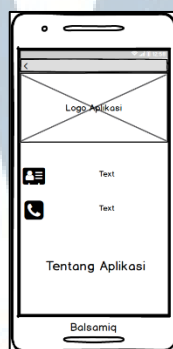
Gambar 3.32 Wireframe Tampilan Sidemenu

Gambar 3.32 menggambarkan *wireframe* tampilan *sidemenu* aplikasi *chatbot*. Menu-menu yang disediakan adalah *chat* dengan *bot*, *tutorial*, dan *about us*. Dari side menu ini, *user* dapat bernavigasi ke halaman yang tersedia. *Sidemenu* akan muncul ketika *user* menekan *menubar* di kiri atas layar pada Halaman Home atau mengusap tepi kiri layar ke arah kanan layar.



Gambar 3.33 Wireframe Halaman Home

Gambar 3.33 menggambarkan *wireframe* Halaman Home. Pada halaman ini *user* dapat melakukan komunikasi dengan *bot*. *User* dapat mengajukan pertanyaan pada *text field* yang tersedia. Sisi kanan akan menampilkan pertanyaan *user* dan sisi kiri menampilkan respon dari *chatbot*.



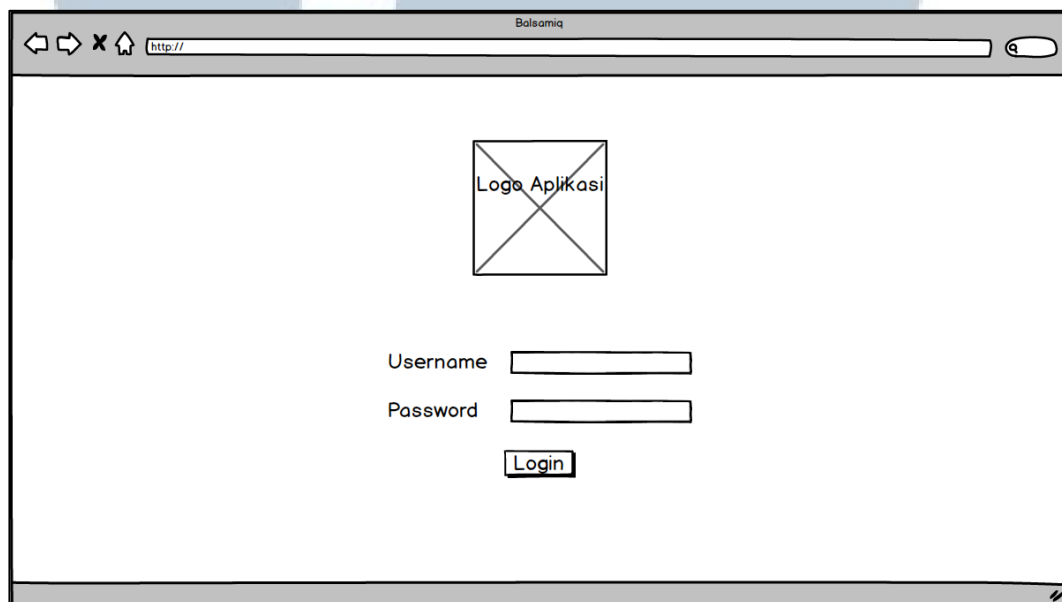
Gambar 3.34 Wireframe Halaman About Us

Gambar 3.34 menggambarkan *wireframe* Halaman About Us. Halaman ini akan menyediakan gambar logo aplikasi, alamat bengkel SUM Hexaputra, nomor

telepon bengkel SUM Hexaputra, informasi singkat seputar Aplikasi Chatbot Penyedia Informasi Bengkel.

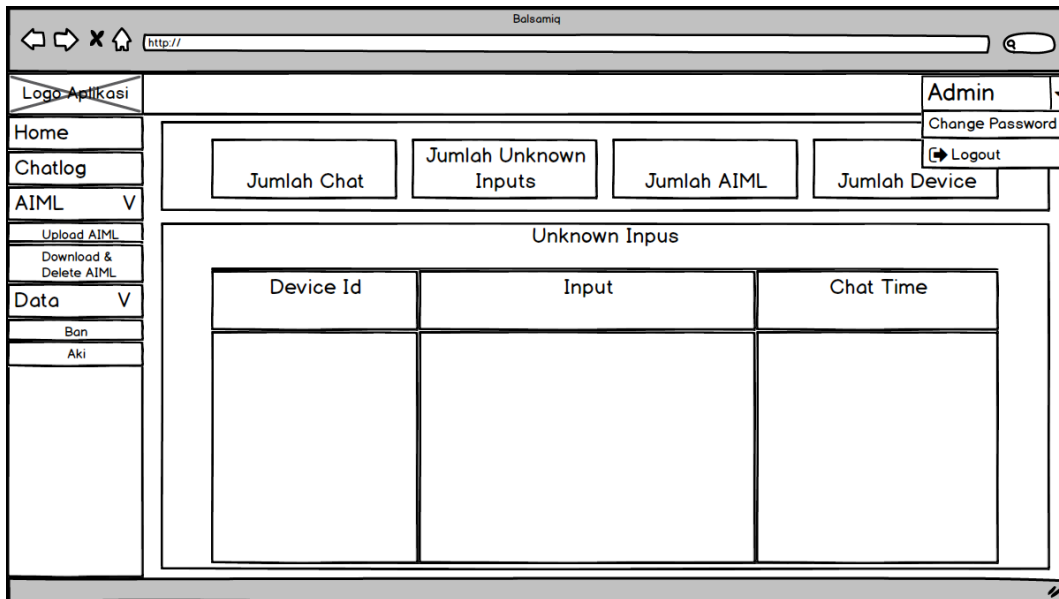
b. Antarmuka Website Admin

Gambar 3.35 menggambarkan *wireframe* Halaman Login Admin. Pada halaman ini tersedia logo aplikasi pada bagian atas, dan terdapat 2 input untuk admin melakukan login. Ketika admin sudah mengisi input *username* dan *password*, admin dapat menekan tombol Login untuk memasuki web *admin* Aplikasi Chatbot Penyedia Informasi Bengkel.



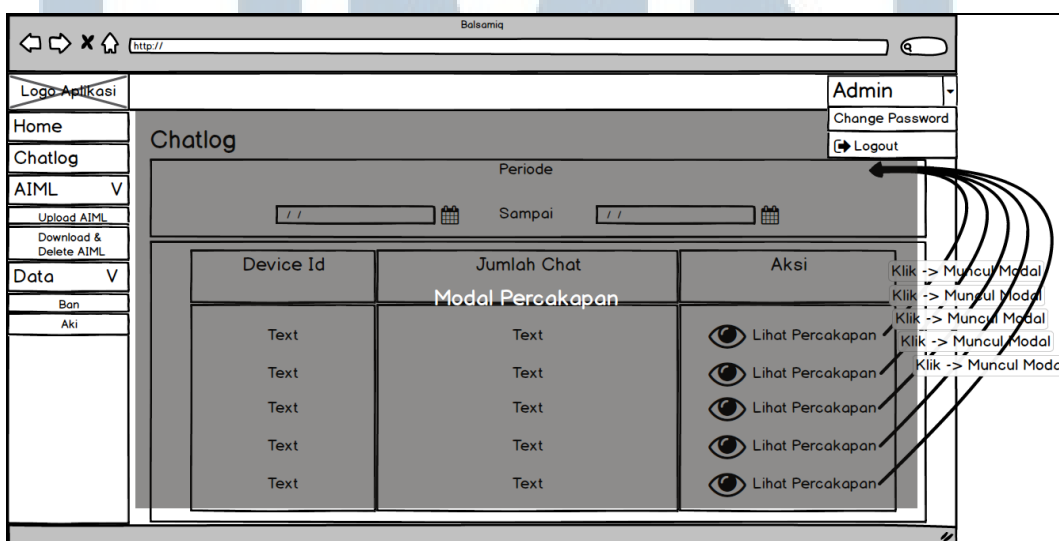
Gambar 3.35 Wireframe Halaman Login Admin

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.36 Wireframe Halaman Home Admin

Gambar 3.36 menggambarkan *wireframe* Halaman Home Admin. Pada halaman ini admin akan disediakan informasi jumlah *chat*, *unknown inputs*, AIML, dan *device* yang terdapat pada Aplikasi Chatbot Penyedia Informasi Bengkel. Pada bagian bawah terdapat tabel yang menyediakan informasi *unknown inputs* yang terdapat pada komunikasi dengan *chatbot*. Data yang masuk ke dalam tabel tersebut adalah input *user* yang tidak dapat *bot* jawab.



Gambar 3.37 Wireframe Halaman Chatlog

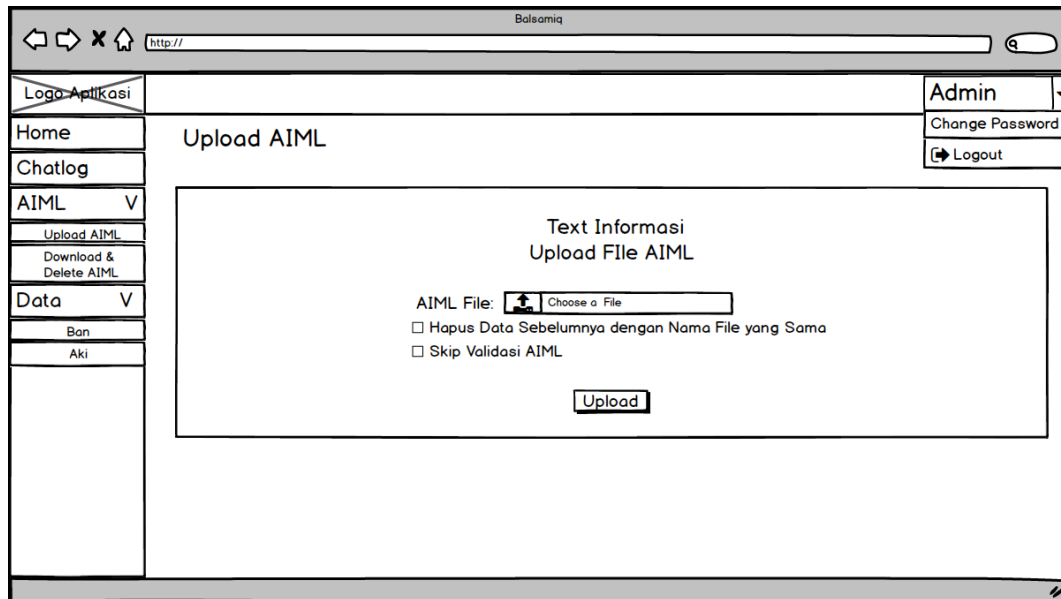
Gambar 3.37 menggambarkan *wireframe* Halaman Chatlog. Pada halaman ini admin dapat melihat sejarah percakapan yang terjadi antara *chatbot* dengan *user*. Admin harus menentukan periode percakapan yang ingin dilihat. Setelah menentukan periode percakapan, data percakapan yang dikelompokkan berdasarkan *device* akan muncul di dalam tabel. Admin dapat memilih percakapan dengan menekan tombol pada salah satu data di dalam kolom paling kanan dan akan memunculkan modal percakapan.

Gambar 3.38 menggambarkan *wireframe* Modal Chatlog. *Modal* ini merupakan sebuah *pop-up* yang muncul dan tidak menutupi keseluruhan layar pada Halaman Chatlog. *Modal* ini bersikan tabel yang memuat informasi percakapan dari input sampai respon *bot*.

Chatlog				
Text Info Keterangan				Tutup
Device Id	Input Setelah Stamping	Input User	Respon	Chat Time

Gambar 3.38 Wireframe Modal Chatlog

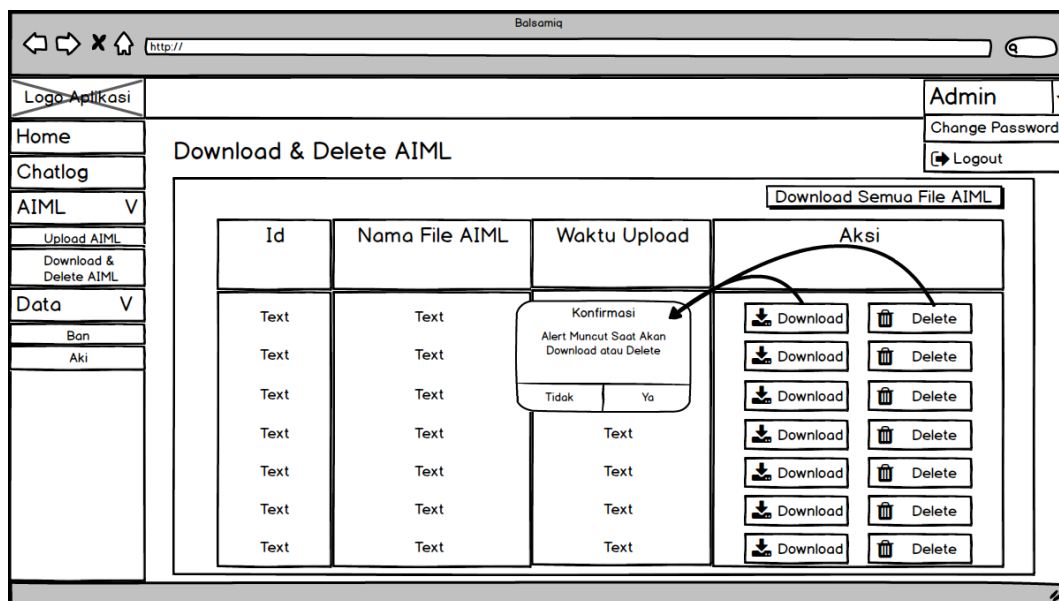
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.39 Wireframe Halaman Upload AIML

Gambar 3.39 menggambarkan *wireframe* Halaman Upload AIML. Halaman ini berisi informasi yang diperlukan untuk melakukan *upload* AIML. Untuk melakukan *upload* AIML, admin harus memilih *file* AIML dan memasukan pada *input file*. Pada halaman ini tersedia 2 fitur untuk melakukan penghapusan pada data sebelumnya dan tidak melakukan validasi saat memasukan *file* AIML ke Aplikasi Chatbot Penyedia Informasi Bengkel. Keberhasilan proses *upload* akan memunculkan notifikasi keberhasilan.

U M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

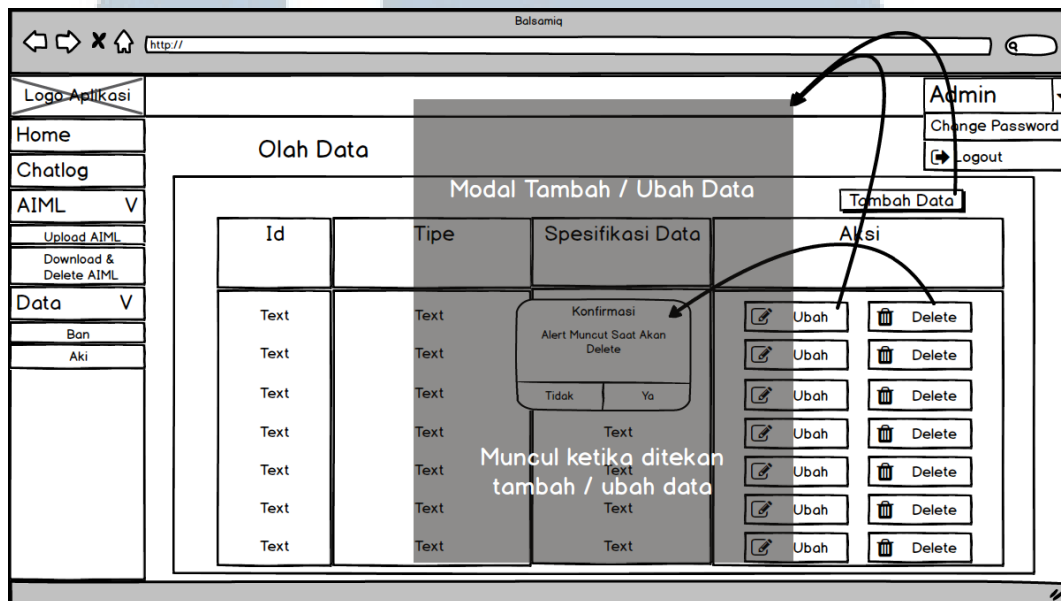


Gambar 3.40 Wireframe Halaman Download & Delete AIML

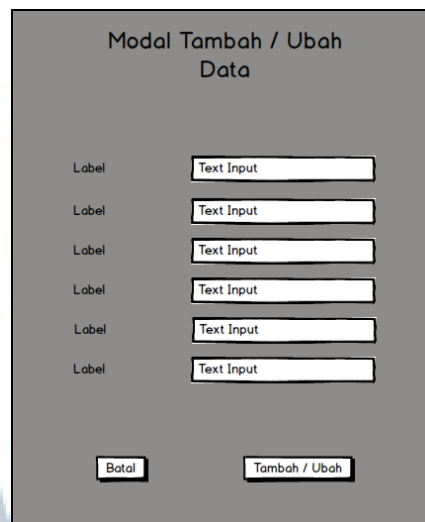
Gambar 3.40 menggambarkan *wireframe* Halaman Download & Delete AIML. Pada halaman ini akan tersedia tabel yang menampilkan seluruh data AIML yang terdapat pada tabel *aiml*. Informasi yang diberikan adalah *id*, nama *file* AIML, waktu *upload*, dan tombol *Download* atau *Delete*. Pada bagian atas tabel juga terdapat 1 tombol yang tujuannya untuk melakukan *download* untuk semua *file* AIML yang terdapat pada tabel *aiml*. Setiap akan melakukan *download* atau *delete*, akan memunculkan sebuah *alert* yang akan menkonfirmasi tindakan yang dilakukan. Jika menjawab *ya*, maka perintah akan langsung dieksekusi. Keberhasilan dari proses *delete* atau *download* akan diberikan melalui notifikasi keberhasilan.

Gambar 3.41 menggambarkan *wireframe* Halaman Olah Data. Pada halaman ini terdapat satu atau lebih tabel sesuai dengan kriteria data. Halaman ini digunakan untuk olah data aki dan olah data ban. Pada olah data ban terdapat dua tabel yang berisikan data ban dan data tipe ban. Pada olah data aki terdapat dua tabel yang berisikan data aki dan data kecocokan aki dengan mobilnya. Pada

halaman ini, admin dapat melakukan tambah data, ubah data, dan aktivasi/non-aktivasi data. Konfirmasi akan muncul ketika admin ingin melakukan aktivasi/non-aktivasi data. Ketika admin ingin menambah atau mengubah data, maka akan muncul sebuah *modal* yang dapat diisi.



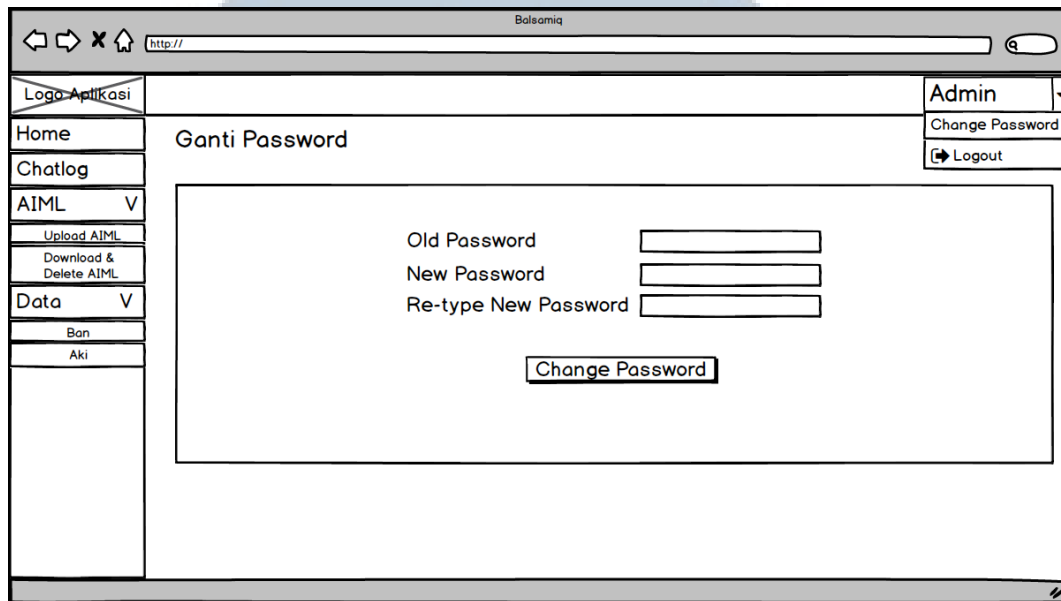
Gambar 3.41 Wireframe Halaman Olah Data



Gambar 3.42 Wireframe Modal Tambah atau Data

Gambar 3.42 menggambarkan *wireframe* Modal Tambah atau Data. Pada halaman ini admin harus mengisi semua masukkan sesuai dengan labelnya. Admin

dapat menutup *modal* dengan cara menekan tombol batal. Admin harus menekan tombol tambah/ubah ketika selesai melakukan tambah data atau mengubah data.



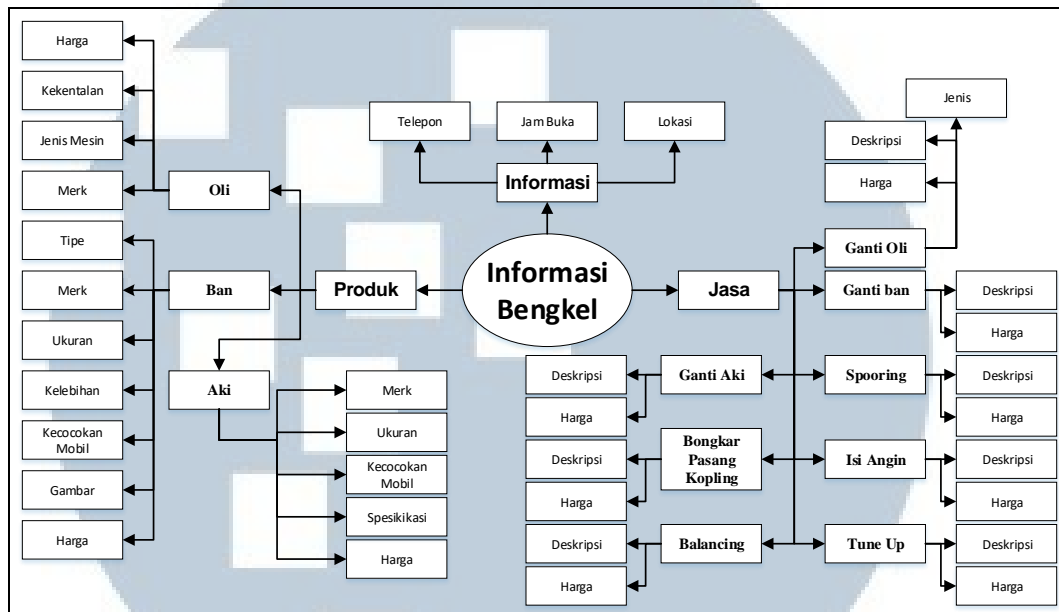
Gambar 3.43 Wireframe Halaman Ganti Password

Gambar 3.43 menggambarkan *wireframe* Halaman Ganti Password. Pada halaman ini terdapat 3 *input password*. Yang pertama merupakan input yang harus diisi dengan *password login* yang lama. Dua input di bawahnya merupakan input yang harus diisi dengan isi yang sama di mana merupakan *password* baru yang ingin diganti. Halaman ini akan melakukan validasi saat melakukan *change password*. Keberhasilan dari mengganti *password* akan ditampilkan melalui notifikasi keberhasilan.

3.2.9 Perancangan Dialog Komunikasi Chatbot

Gambar 3.44 menunjukkan dialog komunikasi *chatbot* pada Aplikasi Chatbot Penyedia Informasi Bengkel. Topik besar yang dapat dibahas pada aplikasi ini adalah jasa, informasi dan produk. Untuk topik produk dan jasa hanya dapat membahas produk yang dijual dan disediakan pada bengkel SUM Hexaputra. Topik

informasi memberikan informasi umum seputar bengkel SUM Hexaputra, seperti lokasi, jam buka, dan nomor telepon bengkel.



Gambar 3.44 Dialog Komunikasi Chatbot

Topik produk menampilkan ciri-ciri tiap produk dengan kekurangan dan kelebihan, tipe, harga dan merk pada produk ban, aki, dan oli. Khusus topik ban juga dapat menyajikan gambar dari setiap produk yang dijual serta dapat menanyakan berapa ukuran produknya yang sesuai dengan mobil *user*. Pada produk oli, *user* juga dapat menanyakan pertanyaan tentang kekentalan oli. Pada produk aki, *user* juga dapat menanyakan pertanyaan tentang spesifikasi aki beserta kecocokan dengan mobil yang digunakannya.

Topik jasa memberikan informasi berupa deskripsi dan harga dari layanan yang diberikan. Jasa yang disediakan adalah ganti oli, ganti ban, *spooiring*, isi angin, *tune up*, pengecekan, bongkar pasang kopling, dan *balancing*. Pada bagian deskripsi juga menjelaskan bagaimana jasa tersebut dikerjakan dan manfaatnya. Untuk jasa ganti oli, *user* juga dapat membahas jenis ganti oli.