



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

Metode yang dilakukan dalam penelitian adalah sebagai berikut.

1. Studi Literatur

Studi literatur yang dilakukan dalam penelitian ini adalah mempelajari referensi-referensi yang akan dijadikan sebagai dasar dan acuan dalam proses perancangan dan pembuatan *chatbot* informasi objek wisata DKI Jakarta. Referensi tersebut antara lain teori tentang *Artificial Intelligence (AI)*, *Artificial Intelligence Markup Language*, *Chatbot*, *Natural Language Processing (NLP)*, *Algoritma Enhanced Confix Stripping Stemmer*, metode *Technology Acceptance Model (TAM)*, Skala Likert, dan Objek Wisata DKI Jakarta.

2. Perancangan dan Pembuatan Sistem

Tahap perancangan dan pembuatan sistem dimulai dengan merancang arsitektur dari *chatbot* informasi objek wisata. Arsitektur *Chatbot* objek wisata terdiri dari dua aplikasi, yaitu aplikasi *chatbot* yang diimplementasikan pada Line Messenger dan *Content Management System (CMS)*. Seluruh data pada aplikasi *Chatbot* akan tersimpan di dalam sebuah sistem basis data *server*. Aplikasi *chatbot* ditujukan sebagai media komunikasi antar pengguna dan sistem. Sedangkan CMS ditujukan kepada admin untuk mengelola informasi pada aplikasi *chatbot* dengan metode *CRUD (Create, Read, Update, Delete)*. Dalam pembuatan sistem mencakup perancangan desain antar muka aplikasi dan fitur-fitur tambahan yang dibutuhkan pada sistem informasi objek wisata menggunakan *chatbot*. Pembuatan

sistem dilakukan dengan mengimplementasikan PHP, *framework* CodeIgniter, dan aplikasi Line Messenger.

3. Pengujian Sistem

Pengujian sistem untuk memperoleh data terkait penggunaan *chatbot* dilakukan dengan menggunakan metode studi lapangan, dimana pengguna yang menjadi responden mencoba menggunakan aplikasi *chatbot* dan berkomunikasi dengan sistem untuk memperoleh informasi mengenai informasi objek wisata yang ada di DKI Jakarta. Sedangkan pengujian sistem untuk mengetahui hasil pemrosesan *input* oleh pengguna dengan *output* sistem yang dilakukan oleh algoritma ECS dilakukan secara manual (*manual testing*) yaitu dengan menguji aplikasi *chatbot* sebanyak sepuluh kali serta membandingkan hasil *stemming* dengan *knowledge base* sesuai dengan kalimat yang dimasukkan oleh pengguna.

4. Evaluasi

Evaluasi sistem dilakukan dengan menganalisis hasil kuisioner yang telah diisi oleh responden pada saat studi lapangan pengujian sistem. Hasil evaluasi akan disimpulkan menggunakan Skala Likert. Responden akan diberikan kuisioner mengenai pengalaman yang dirasakan ketika menggunakan aplikasi dengan menggunakan metode TAM sebagai dasar pembuatan kuisioner.

3.2 Variabel Penelitian

Variabel yang digunakan pada penelitian ini adalah variabel terikat dan variabel bebas. Variabel terikat pada penelitian ini adalah mengukur tingkat penerimaan pengguna terhadap aplikasi *chatbot* informasi objek wisata berbasis *Artificial Intelligence Markup Language* dengan menggunakan Algoritma

Enhanced Confix Stripping Stemmer Bahasa Indonesia. Sedangkan variabel bebas pada penelitian ini adalah *input* dari pengguna pada saat menanyakan informasi objek wisata DKI Jakarta melalui aplikasi *chatbot*.

3.3 Teknik Pengumpulan Data

Pada tahap pembuatan, data didapatkan dengan mengajukan surat permohonan pelaksanaan penelitian kepada Kelapa Dinas Pariwisata dan Kebudayaan Pemerintah Provinsi DKI Jakarta untuk menentukan permasalahan, topik dan data-data terkait lainnya untuk proses penelitian dan perancangan *chatbot* informasi objek wisata DKI Jakarta. Data juga didapat dengan melakukan pengumpulan data dari berbagai sumber di internet untuk menunjang kelengkapan data yang dibutuhkan. Pada tahap pengujian, data dikumpulkan dengan menggunakan metode studi lapangan dengan memberikan *link* atau *barcode chatbot* Line kepada responden untuk dicoba pada aplikasi Line mereka, kemudian menanyakan pengalaman dari penggunaan *chatbot* tersebut dalam bentuk kuisisioner. Kuisisioner yang terkumpul kemudian diukur dengan menggunakan Skala Likert lima tingkat dan dihitung dengan menggunakan metode TAM.

3.4 Teknik Pengambilan Sampel

Teknik yang digunakan untuk pengambilan sampel pada penelitian ini adalah dengan menggunakan teknik *Simple Random Sampling*. Pengambilan anggota sampel dari populasi dilakukan secara acak tanpa memperhatikan strata, jenis kelamin, usia, maupun asal tempat tinggal sanggota sampel yang ada dalam populasi tersebut. Sampel diambil secara acak untuk pengujian sistem dan

penyebaran kuisisioner. Jumlah sampel yang diperlukan adalah tiga puluh orang responden (Sugiyono, 2012).

3.5 Perancangan Sistem

Perancangan sistem yang dilakukan menghasilkan model berupa *Data Flow Diagram* (DFD), *Flowchart*, *Entity Relationship Diagram* (ERD), *Database Schema*, struktur tabel database, perancangan antarmuka pengguna, dan perancangan dialog komunikasi *chatbot*.

3.5.1 Data Flow Diagram (DFD)

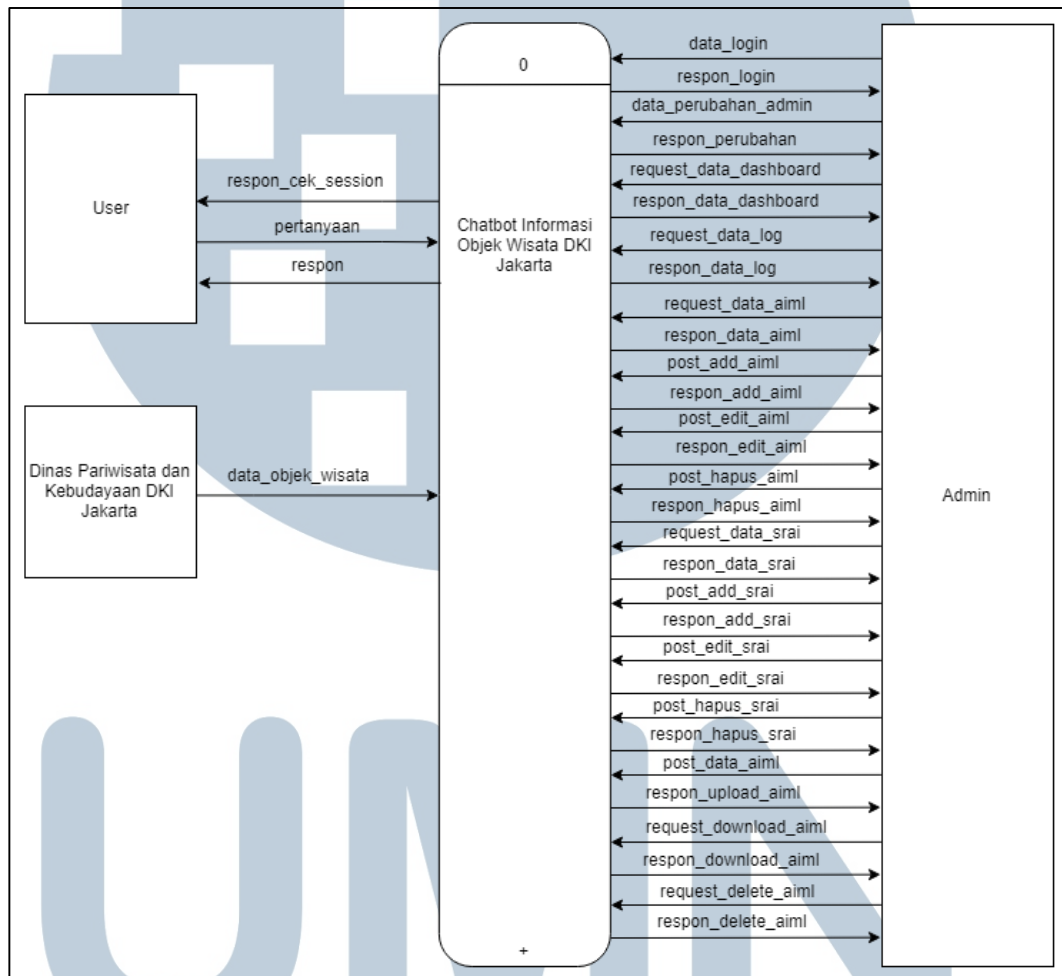
Data Flow Diagram menggambarkan alur data sistem yang saling terhubung antara satu dengan lainnya pada Chatbot Informasi Objek Wisata. Gambar 3.1 menggambarkan Diagram Konteks dari Chabot Informasi Objek Wisata. *Level* ini terdiri dari 3 entitas dan 1 proses. Entitas tersebut terdiri dari *User*, Admin dan Dinas Pariwisata dan Kebudayaan DKI Jakarta. Sedangkan prosesnya yaitu Chatbot Informasi Objek Wisata DKI Jakarta.

Entitas *user* memberikan masukan berupa kalimat pertanyaan pada *chatbot*. Kalimat pertanyaan yang telah dimasukkan kemudian akan diproses oleh sistem sehingga menghasilkan *output* berupa kalimat respon. *User* yang sudah terhubung dengan *bot* maka akan secara otomatis mendapatkan data *session* yang digunakan untuk memulai melakukan komunikasi dengan *chatbot*.

Entitas admin dapat melakukan *login* ke halaman admin dengan memasukkan *username* dan *password*. Setelah *login* admin berhasil, admin dapat melihat segala informasi terkait kegiatan komunikasi *user* dengan *chatbot*, melakukan

penambahan atau perubahan *knowledge base chatbot*, dan admin dapat mengunduh file-file AIML yang terdapat di *database*.

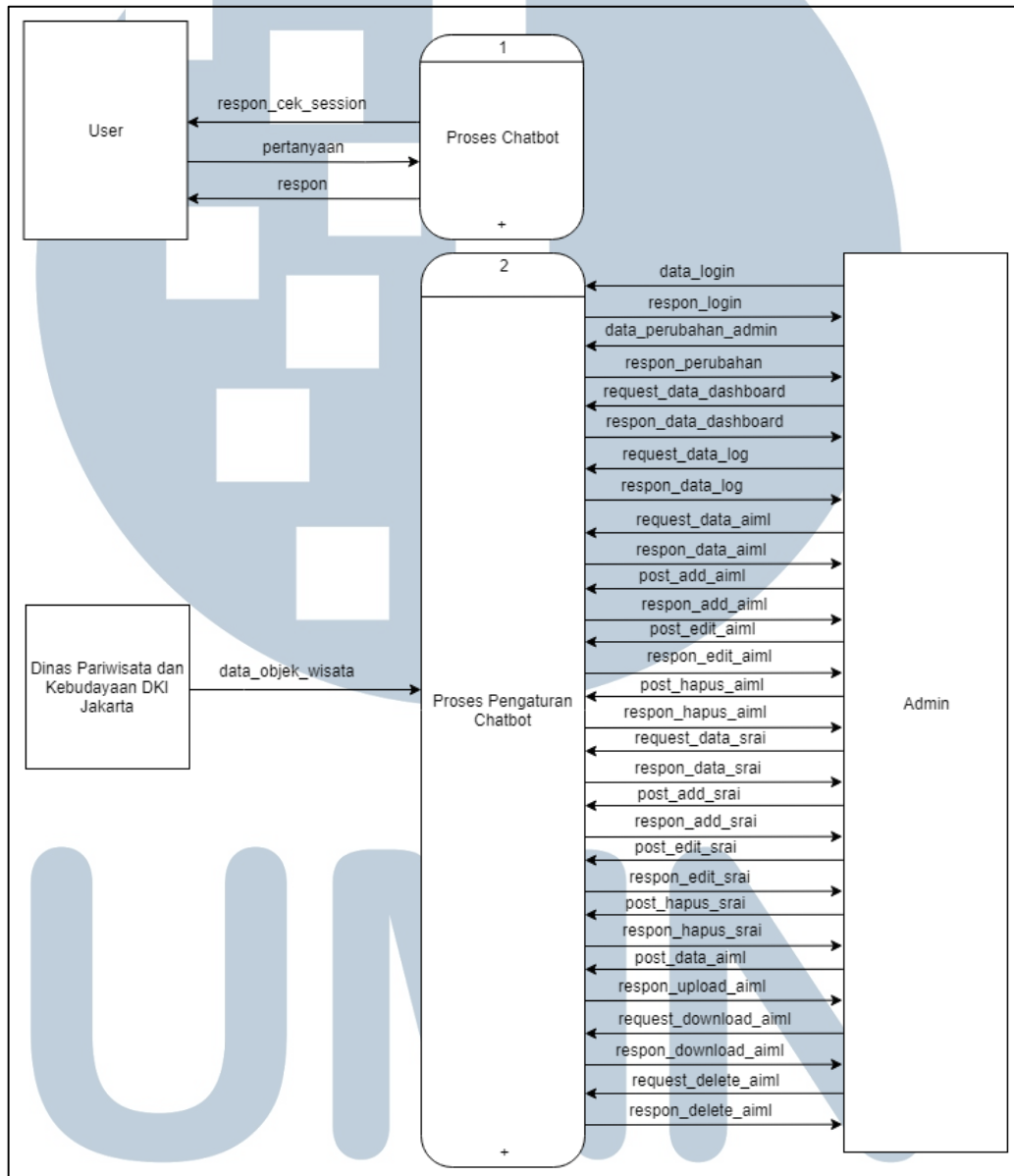
Entitas Dinas Pariwisata dan Kebudayaan DKI Jakarta memberikan data objek pariwisata kepada admin untuk kemudian dibuat menjadi *knowledge base* atau pola respon *chatbot* terhadap *user*.



Gambar 3.1 Diagram Konteks Chatbot Informasi Objek Wisata

Gambar 3.2 menggambarkan DFD *level 0* dari Chatbot Informasi Objek Wisata. Pada *level* ini terdapat 2 proses yaitu proses *chatbot* dan proses pengaturan *chatbot*. Proses *chatbot* merupakan proses komunikasi antar *user* dan *chatbot* dengan menerima kalimat input dari *user* kemudian *chatbot* akan memberikan kalimat respon kepada *user*. Proses pengaturan *chatbot* merupakan proses yang dilakukan

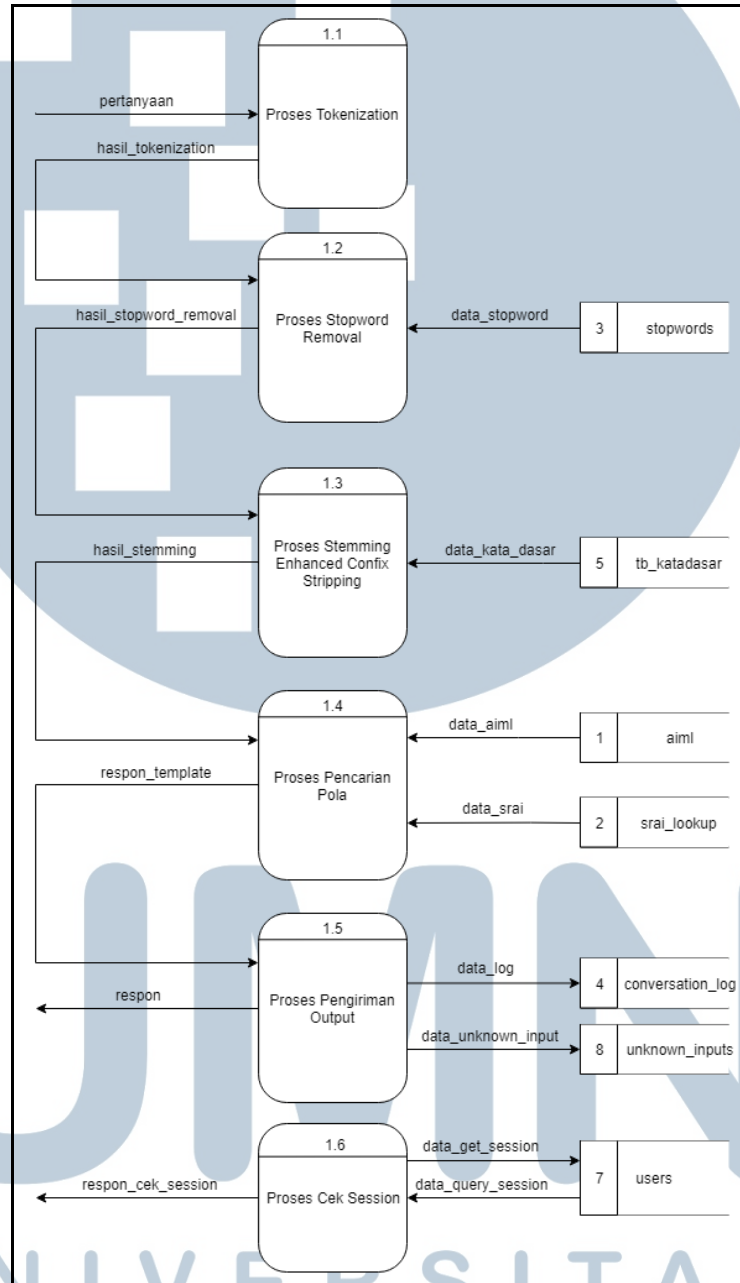
oleh entitas admin dan entitas Dinas Pariwisata dan Kebudayaan DKI Jakarta dalam melakukan penambahan atau perubahan pada *knowledge base chatbot*, dan juga melakukan pengawasan pada komunikasi antar *user* dengan *chatbot*.



Gambar 3.2 DFD Level 0 Chatbot Informasi Objek Wisata

Proses komunikasi *chatbot* memiliki 6 proses yang digambarkan pada DFD *level 1* pada Gambar 3.3. Proses awal yang dilakukan adalah dengan memeriksa *session user* pada tabel *users*. Jika pada proses cek *session* ditemukan *session* dari *user* tersebut maka komunikasi dapat dilanjutkan, tetapi jika pada proses ini tidak

terdapat *session* dari *user* tersebut, maka *user* tersebut wajib menambahkan sebagai teman *Line Official* dari Pariwisata Jakarta untuk mendapatkan *session* agar dapat berkomunikasi dengan *chatbot* tersebut.



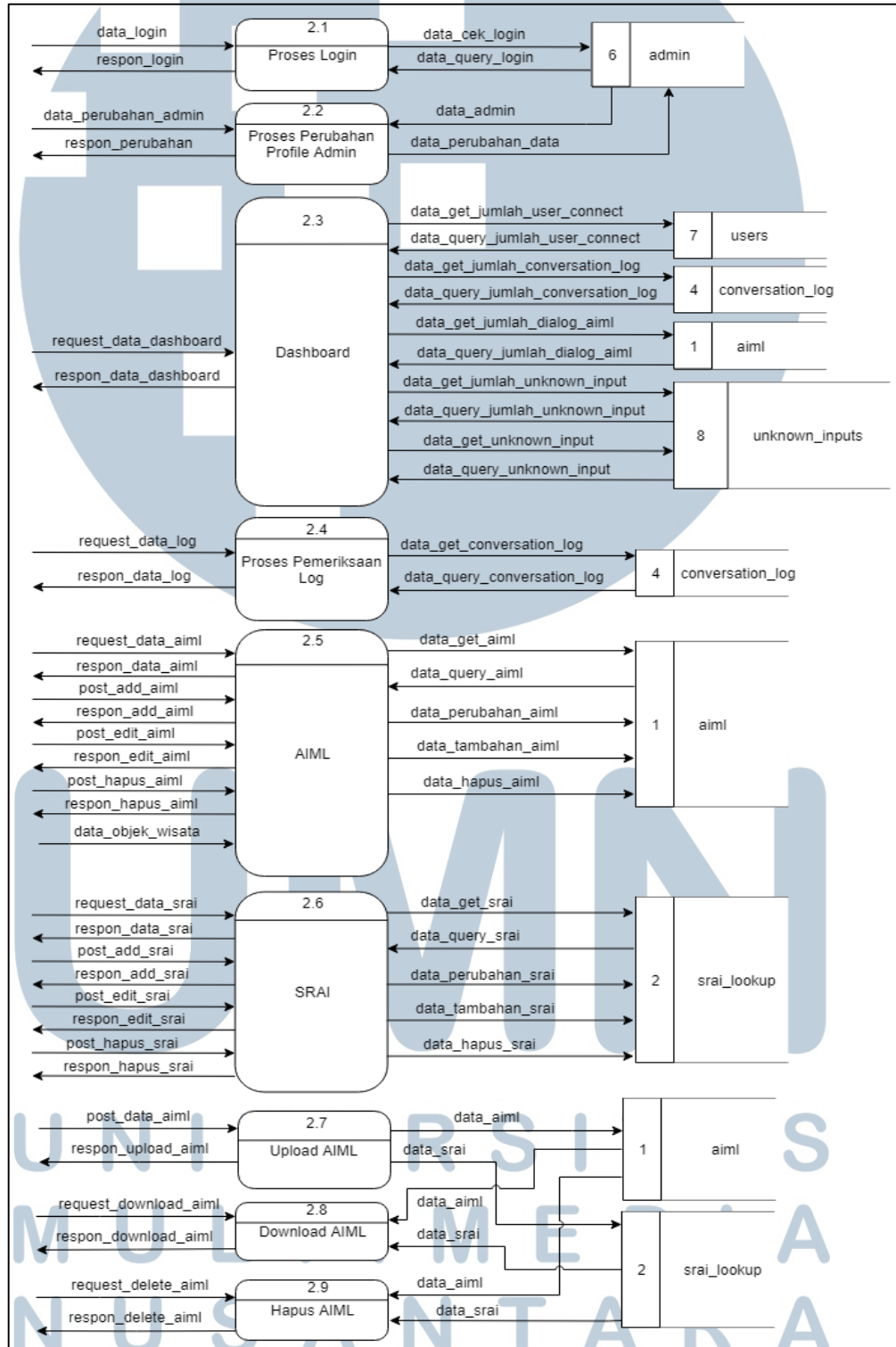
Gambar 3.3 DFD Level 1 Proses Chatbot

Selanjutnya, proses *tokenization* dilakukan dengan mengubah kalimat pertanyaan ke dalam bentuk *lowercase* kemudian membuang tanda baca yang terdapat pada kalimat tersebut dan mengubah bentuk kalimat tersebut menjadi

array of string. Setelah *tokenization*, kata-kata yang terdapat pada tabel stopwords akan dihilangkan, proses tersebut bernama *stopword removal*. Tabel stopwords yang digunakan merupakan data yang diambil dari penelitian milik Fadillah Z Tala (2003). Proses selanjutnya yaitu proses *stemming* dengan menggunakan algoritma *Enhanced Confix Stripping*. Hasil *stemming* yang sudah berupa kata dasar kemudian di cari polanya pada proses pencarian pola pada tabel aiml untuk menghasilkan respon dari pertanyaan *user*. Respon yang didapat kemudian dikirim kembali ke *user* melalui proses pengiriman *output* serta disimpan pada tabel *conversation_log*. Kalimat pertanyaan yang tidak dapat dijawab kemudian akan di catat pada tabel *unknown_inputs*.

Dalam proses pengaturan admin terdapat 9 proses yang digambarkan pada DFD *level 1* pada Gambar 3.4. Proses tersebut yaitu proses *login*, proses perubahan profile admin, *dashboard*, proses pemeriksaan log, aiml, *srai*, *upload aiml*, *download aiml*, dan *hapus aiml*. Sebelum dapat melakukan semua proses, admin terlebih dahulu wajib memasukkan *username* dan *password* yang kemudian akan di proses pada proses *login*. Jika *username* dan *password* sudah sesuai, maka admin akan mendapatkan *session* untuk dapat mengakses seluruh proses pada proses pengaturan admin. Proses perubahan admin berfungsi untuk mengubah segala informasi yang dimiliki admin. *Dashboard* merupakan halaman awal ketika admin sudah memiliki *session* dari proses *login*, dan dalam *Dashboard* admin dapat melihat berbagai informasi terkait *chatbot* Informasi Objek Wisata. Proses pemeriksaan *log* berfungsi untuk melihat semua komunikasi yang dilakukan *user* dengan *chatbot*. Pada AIML, selain dapat melihat *knowledge base chatbot*, admin dapat melakukan penambahan, perubahan dan juga menghapus aiml. SRAI

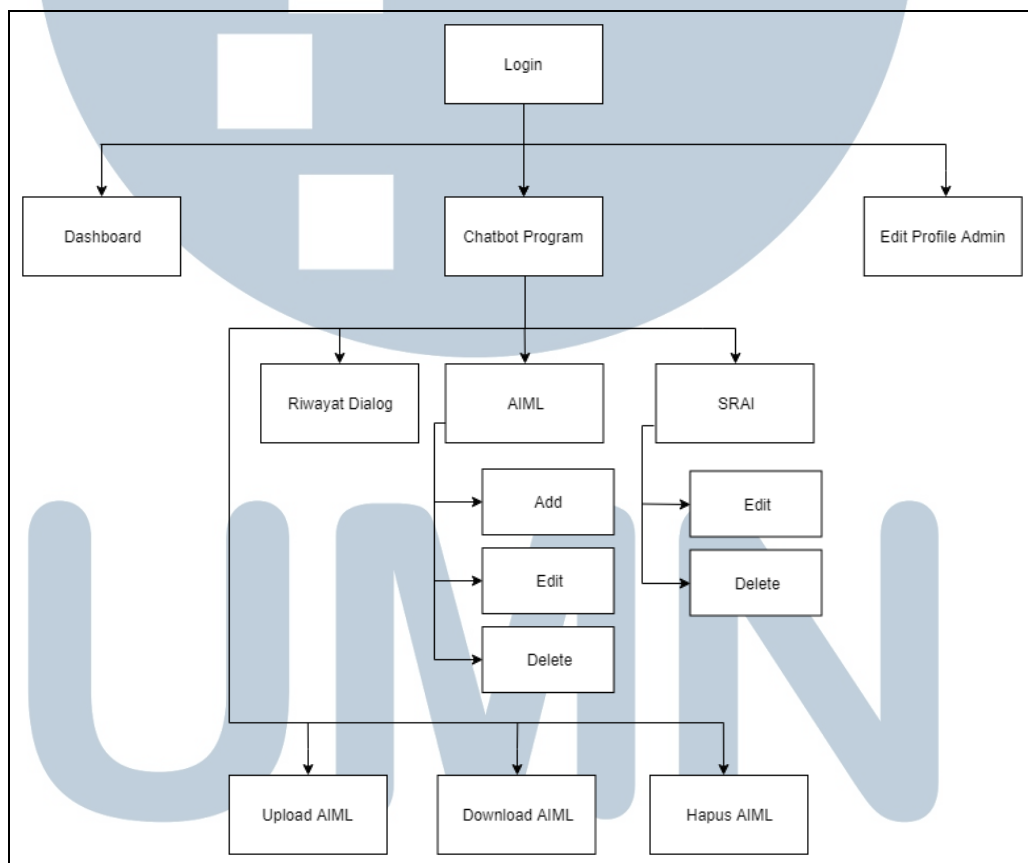
berfungsi untuk melihat hubungan antara respon yang satu dengan lainnya, selain itu admin dapat mengubah dan menghapus srail. Admin juga dapat melakukan *upload* file aiml, *download* file aiml dan menghapus file aiml yang ada.



Gambar 3.4 DFD Level 1 Proses Pengaturan Admin

3.5.2 Sitemap

Perancangan sistem pada suatu *website* dilakukan dengan membuat rancangan struktur halaman *web*. Struktur halaman *website* dapat dilihat dalam *Sitemap* pada Gambar 3.5. Halaman *back end* terdiri dari halaman *login*, halaman *dashboard*, halaman riwayat dialog, halaman *aiml*, halaman *srai*, halaman *upload aiml*, halaman *download aiml*, halaman *hapus aiml*, dan halaman *edit profile admin*. Pada halaman *aiml*, admin dapat menambahkan *aiml*, mengubah *aiml*, dan menghapus *aiml*. Pada halaman *srai*, admin dapat mengubah *srai* dan menghapus *srai*.

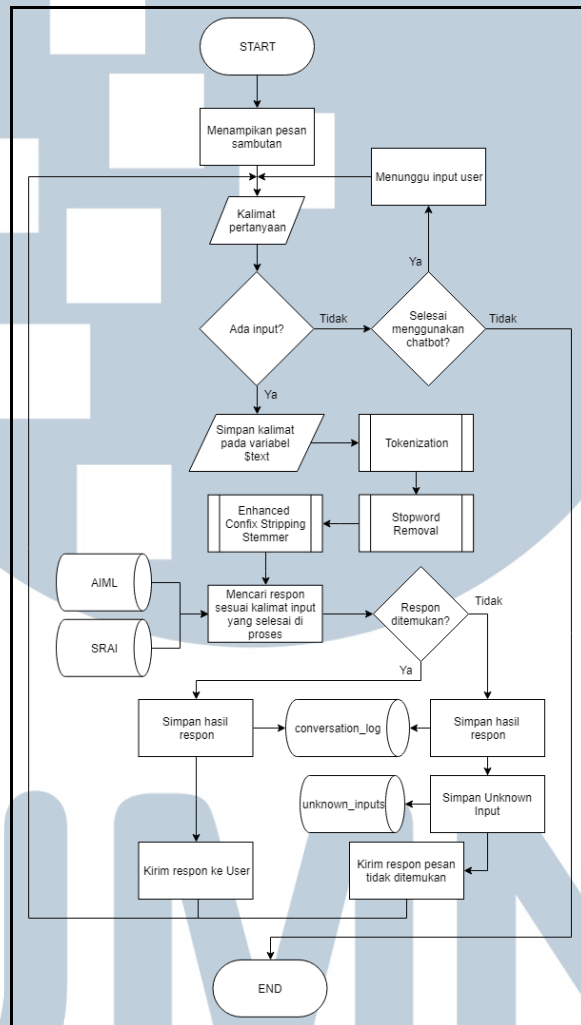


Gambar 3.5 Sitemap Back End

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.5.3 Flowchart

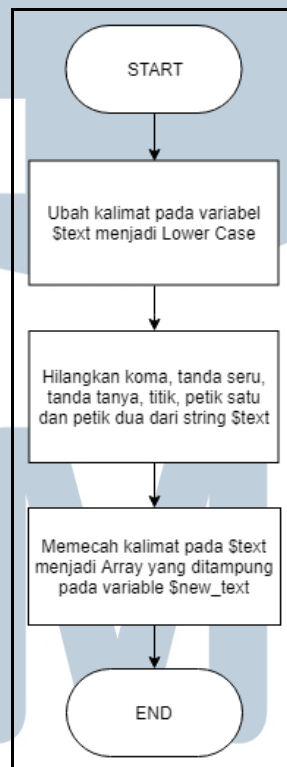
Flowchart merupakan diagram yang menggambarkan alur kerja program. Gambar 3.6 menunjukkan *flowchart* dari *chatbot* Informasi Objek Wisata yang dirancang dan dibangun pada penelitian ini.



Gambar 3.6 Flowchart Chatbot

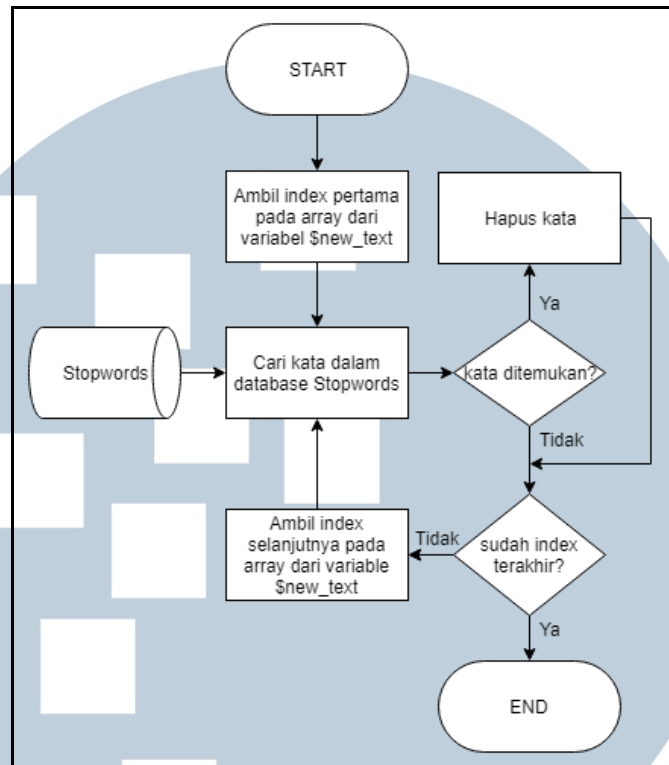
User yang sudah terhubung dengan *Line Official* dari Pariwisata Jakarta akan mendapatkan pesan sambutan pertama kali. Pesan sambutan tersebut terdapat beberapa bantuan yang dapat dilakukan oleh *user* ketika *user* akan melakukan komunikasi dengan *chatbot*. Proses ini akan memproses masukan yang diberikan *user* dan kemudian diolah menjadi respon yang akan diterima kembali oleh *user*.

Dalam proses ini melibatkan proses *tokenization*, *stopword removal*, *stemming* dengan algoritma *Enhanced Confix Stripping* yang digunakan untuk memperoleh kata dasar. Kalimat dengan kata dasar kemudian dicocokkan dengan data AIML dan SRAI yang didapat dari tabel *aiml* dan tabel *srai_lookup*. Respon yang diperoleh pada proses pencocokan pola akan dikirim kembali ke *user* dan respon akan disimpan pada tabel *conversation_log* sedangkan pada respon yang tidak diperoleh pada proses pencocokan pola akan dikirim ke tabel *unknown_input* dan tetap disimpan pada tabel *conversation_log* dengan mengirim kalimat respon *default* yang dimiliki *chatbot*.



Gambar 3.7 Flowchart Tokenization

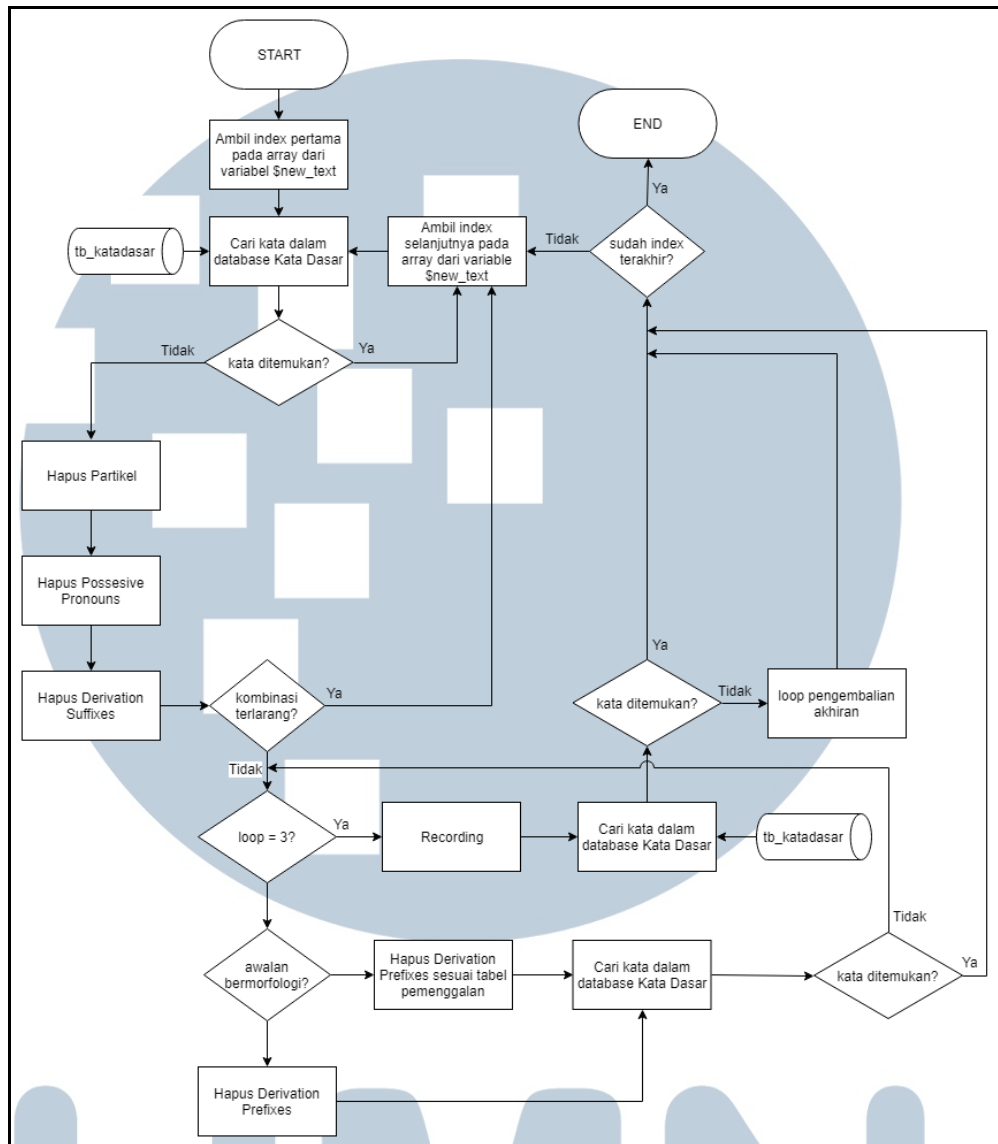
Gambar 3.7 menggambarkan alur proses *tokenization* yaitu proses mengubah kalimat menjadi huruf kecil, menghilangkan tanda baca yang terdapat pada kalimat dan memecah kalimat tersebut menjadi *array of string* yang akan digunakan pada proses selanjutnya yaitu proses *stopword removal*.



Gambar 3.8 Flowchart Stopword Removal

Gambar 3.8 menunjukkan alur dalam proses *Stopword Removal* yaitu menghapus kata-kata yang tidak penting pada setiap *array* dengan mencocokkan *list array* dengan tabel stopwords. Jika pada *array* terdapat kata yang terdapat pada tabel stopwords maka kata yang terdapat di *array* tersebut akan dihapus dan akan diperiksa hingga *array* terakhir. Jika sudah di *array* terakhir dan sudah tidak ada kata yang terdapat pada tabel stopwords maka proses stopwords akan dihentikan dan *array* yang baru sudah merupakan *array* yang bersih dari *stopword*.

Setelah didapat *array* yang sudah bersih dari *stopword*, selanjutnya *array* akan di *stemming* dengan menggunakan algoritma *Enhanced Confix Stripping*. Proses ini mengubah bentuk kata menjadi kata dasar. Gambar 3.9 menggambarkan alur kerja dari proses algoritma *enhanced confix stripping* pada *chatbot*. Alur kerja algoritma *enhanced confix stripping* telah dijabarkan pada Bab 2.

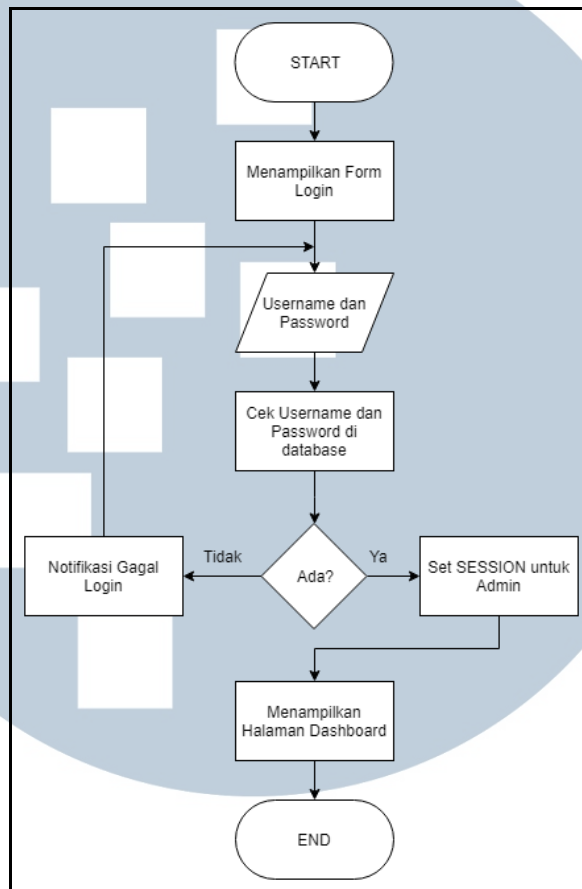


Gambar 3.9 Flowchart Enhanced Confix Stripping Stemmer

Untuk dapat memasuki halaman *back end* dari *chatbot* informasi objek wisata, admin diwajibkan untuk melalui proses *login* terlebih dahulu untuk memperoleh *session* agar dapat mengakses seluruh halaman di *website back end* aplikasi *chatbot*.

Alur proses *login* digambarkan pada Gambar 3.10. Saat admin memasuki halaman *login*, admin akan disuguhkan *form login* yang berisi *username* dan *password* yang harus diisi oleh admin dengan data yang sesuai. Jika data yang dimasukkan oleh admin sesuai dengan data yang terdapat di tabel admin, maka admin akan diberikan *session* dan diarahkan ke halaman *dashboard*. Namun jika data yang diisikan oleh

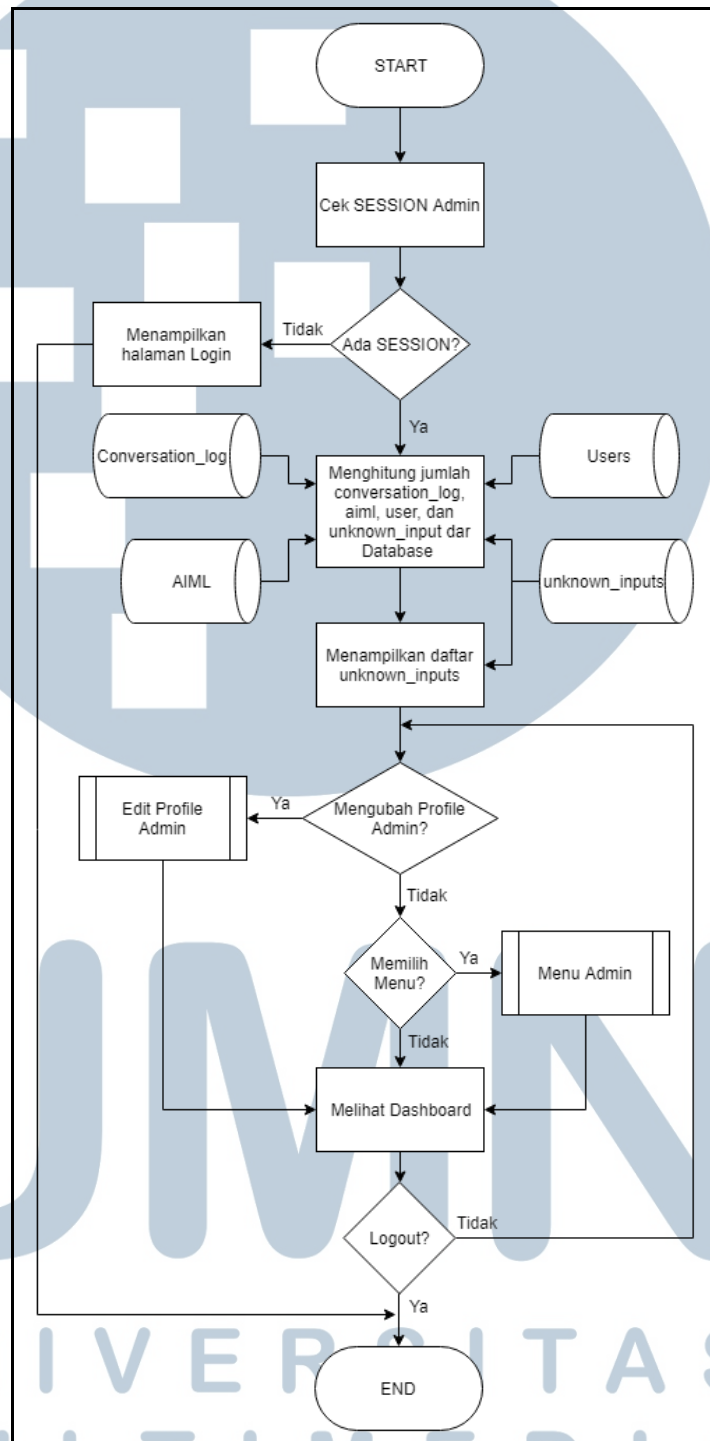
admin tidak sesuai, maka admin akan dikembalikan ke halaman *login* dengan muncul notifikasi bahwa login gagal.



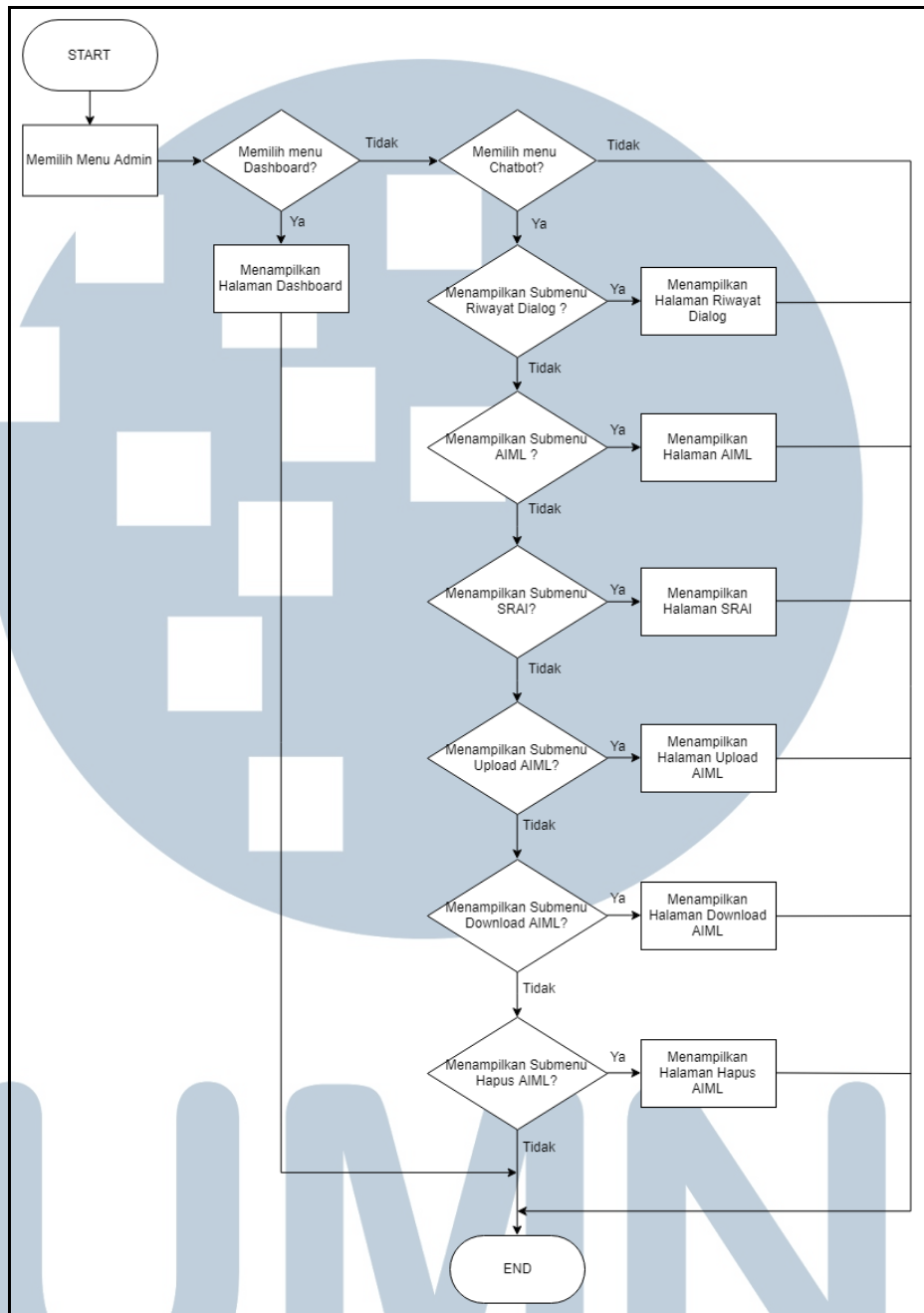
Gambar 3.10 Flowchart Login Admin

Setelah *login* berhasil, maka admin akan diarahkan kedalam halaman *dashboard*. *Flowchart* halaman *dashboard* admin digambarkan pada Gambar 3.11. Admin akan disuguhkan data terkait jumlah dialog komunikasi antara *user* dengan *chatbot*, jumlah dialog *aiml* atau *knowledge base chatbot*, jumlah *user* yang terkoneksi dengan aplikasi *chatbot*, dan jumlah kalimat yang tidak dikenali oleh *chatbot*. Data-data tersebut secara berturut-turut didapatkan dari tabel *conversation_log*, *aiml*, *users*, dan *unknown_inputs*. Selain itu pada halaman *dashboard*, admin juga dapat melihat daftar kalimat yang tidak dikenali oleh *chatbot* yang didapat dari tabel *unknown_inputs*. Dari halaman *dashboard* ini,

admin dapat mengubah data-data admin yang tersimpan pada *database*, dan admin juga dapat memilih menu lain yang dapat diakses.

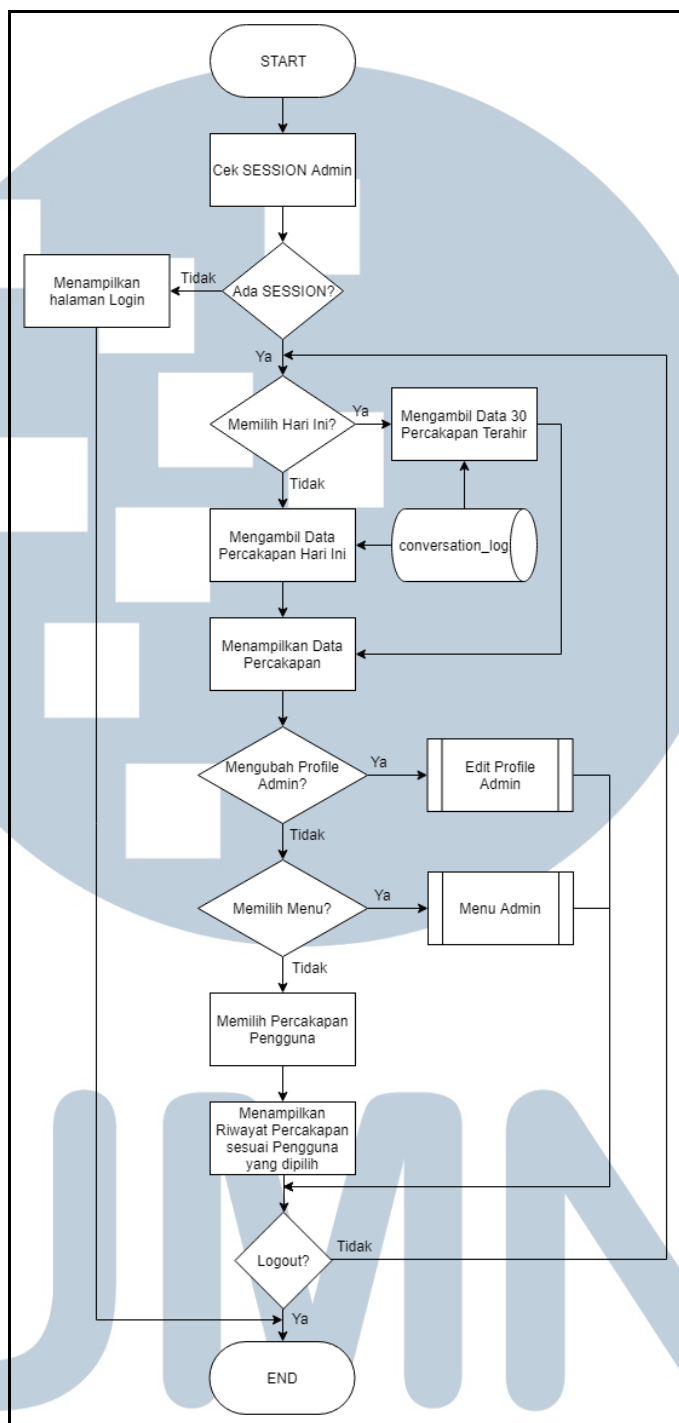


Gambar 3.11 Flowchart Halaman Dashboard Admin



Gambar 3.12 Flowchart Memilih Menu Admin

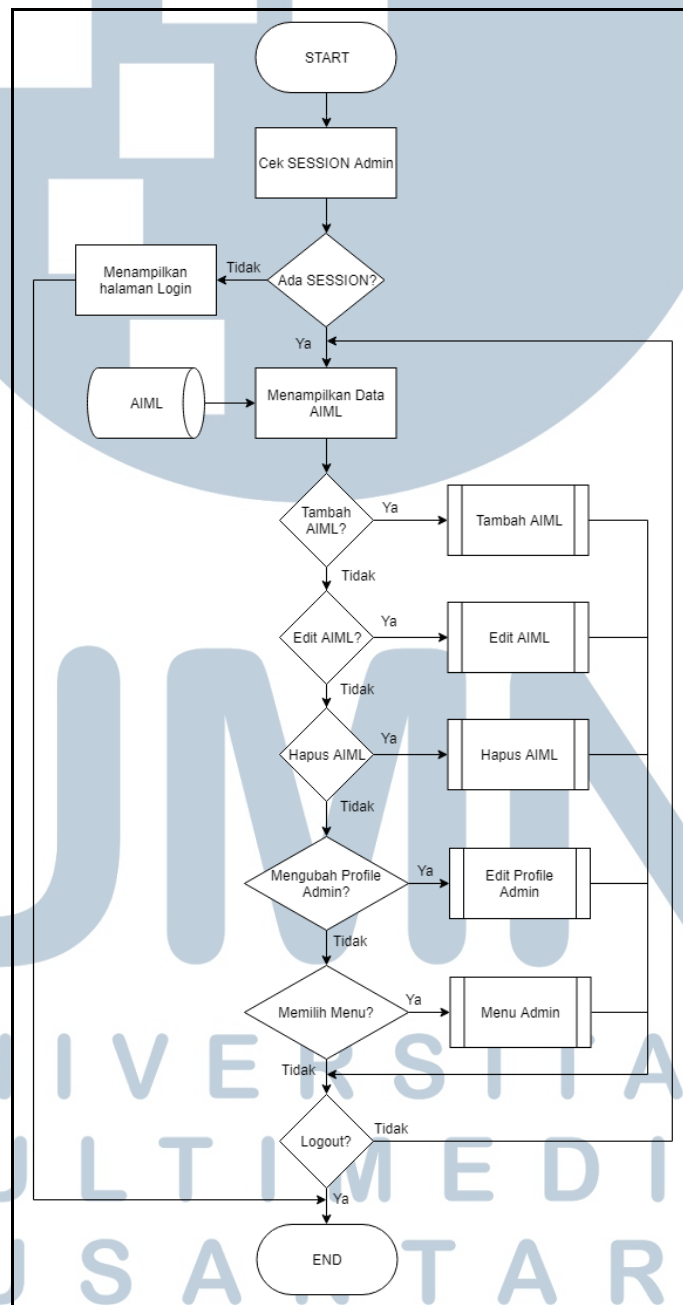
Gambar 3.12 menunjukkan *flowchat* dalam memilih menu yang ada di *website back end chatbot* Infomasi Objek Wisata. Admin dapat memilih menu *dashboard*, *riwayat dialog*, *aiml*, *srai*, *upload aiml*, *download aiml*, dan *hapus aiml*. Jika admin memilih salah satu dari menu tersebut, admin akan diarahkan ke halaman yang ingin dituju.



Gambar 3.13 Flowchart Halaman Riwayat Dialog

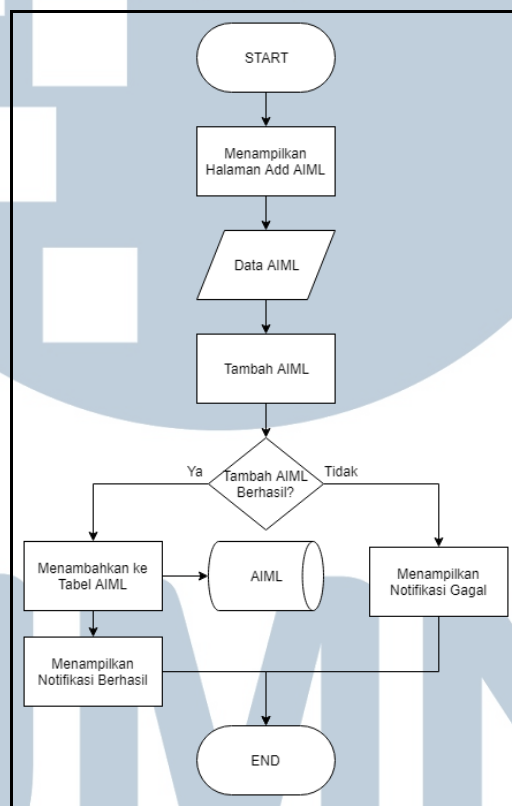
Untuk melihat riwayat komunikasi antara *user* dengan *chatbot*, admin dapat mengakses halaman riwayat dialog pada *website back end chatbot* informasi objek wisata. Gambar 3.13 menggambarkan keseluruhan alur proses yang terjadi pada halaman riwayat dialog. Secara *default*, sistem akan menampilkan 30 percakapan

terakhir antara *user* dengan *chatbot*. Jika admin ingin melihat seluruh komunikasi yang terjadi di hari tersebut, admin dapat mengganti kategori yang ada menjadi hari ini dan melakukan *submit*. Maka akan secara otomatis sistem akan menampilkan seluruh percakapan yang terjadi di hari tersebut sesuai dengan *user* masing-masing. Selain itu, admin dapat mengubah data-data admin yang tersimpan pada *database*, dan admin juga dapat memilih menu lain yang dapat diakses.



Gambar 3.14 Flowchart Halaman AIML

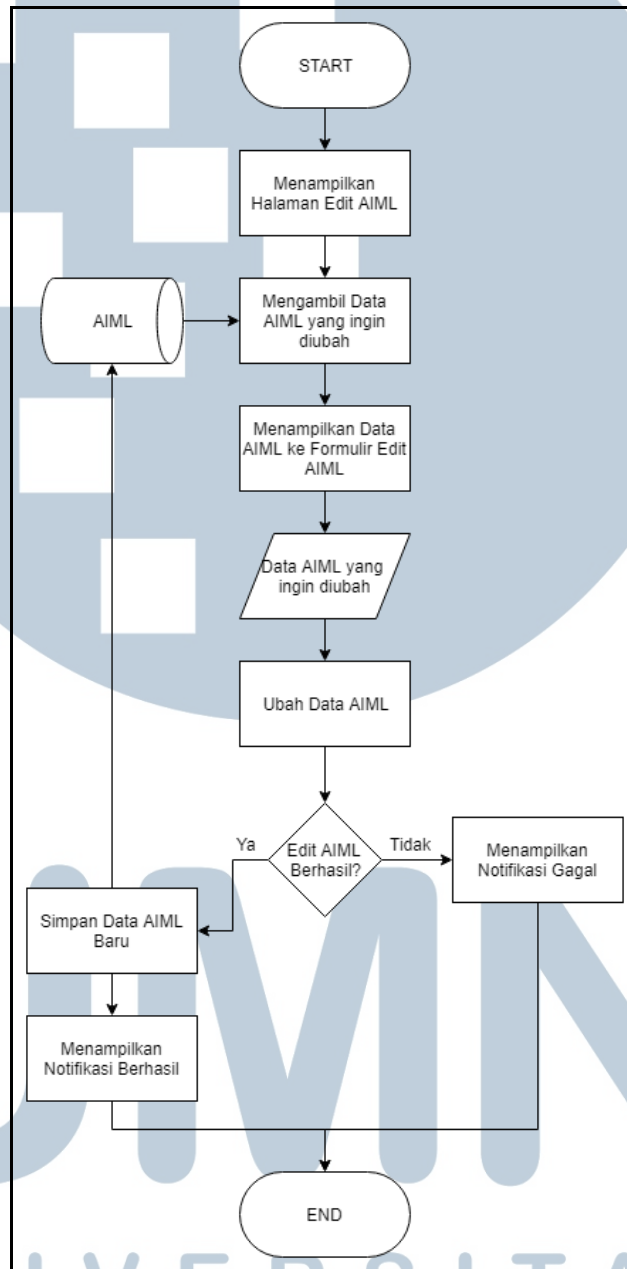
Pada menu selanjutnya admin dapat memilih menu aiml yang akan diarahkan ke halaman aiml. Secara sederhana halaman aiml merupakan halaman untuk mengakses *knowledge base* aplikasi *chatbot*. Alur proses aiml ditunjukkan pada Gambar 3.14. Saat admin mengakses halaman aiml, admin akan disuguhkan data-data terkait data *knowledge base chatbot* yang didapat dari tabel aiml. Selain itu, admin juga dapat menambahkan data aiml baru, mengubah data aiml yang sudah ada dan menghapus data aiml.



Gambar 3.15 Flowchart Halaman Tambah AIML

Gambar 3.15 menunjukkan alur yang terjadi pada saat admin ingin menambahkan data aiml baru ke *database*. Pada saat mengakses halaman tambah aiml, admin akan disuguhkan *form* yang harus diisi oleh admin. Setelah admin mengisi *form* yang telah disediakan, maka data akan diverifikasi untuk mengetahui apakah data aiml berhasil ditambahkan atau tidak. Jika berhasil maka data aiml

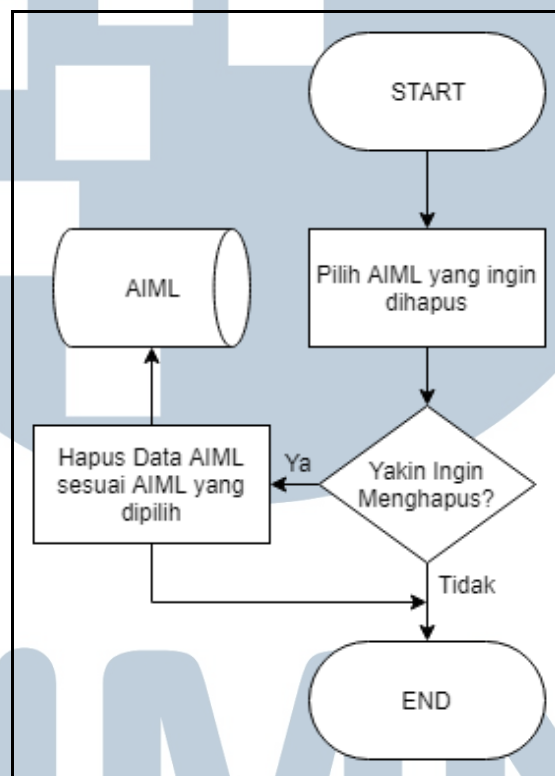
tersebut akan ditambahkan ke dalam tabel aiml dan menampilkan notifikasi berhasil, tetapi jika gagal maka sistem akan menampilkan notifikasi gagal dan akan diarahkan kembali ke halaman aiml.



Gambar 3.16 Flowchart Halaman Edit AIML

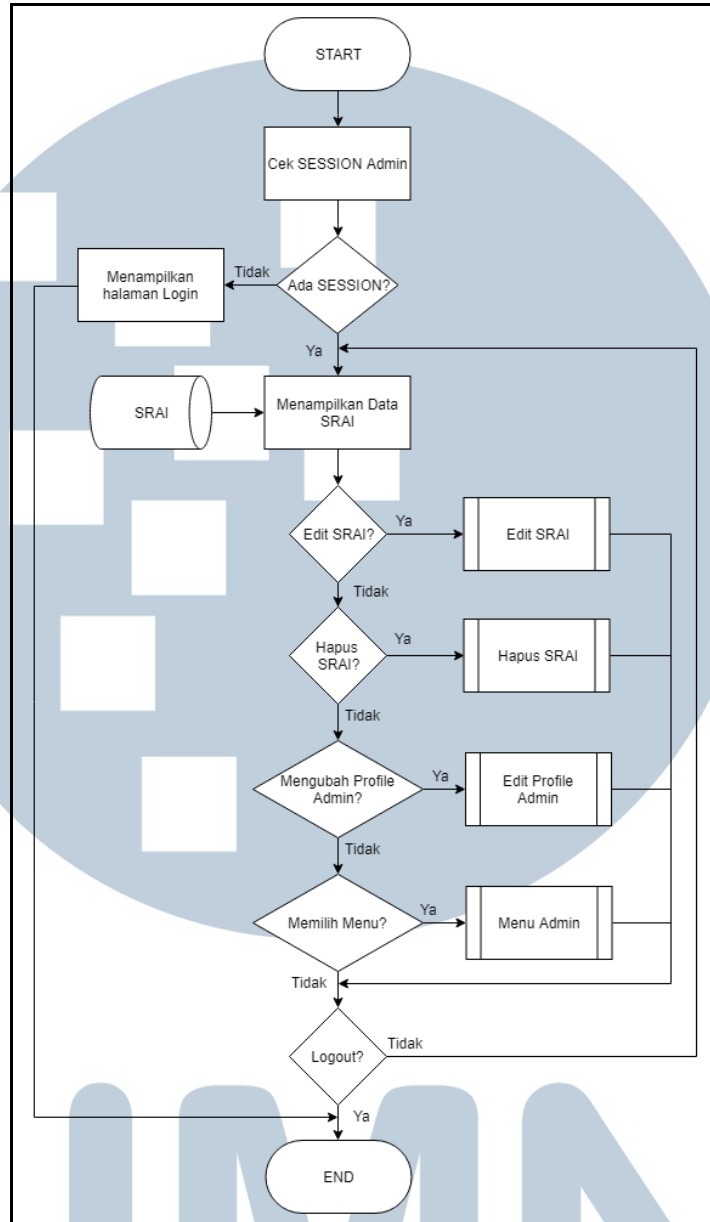
Gambar 3.16 menunjukkan alur yang terjadi pada saat admin ingin mengubah data aiml yang sudah ada ke *database*. Pada saat mengakses halaman *edit aiml*, admin akan disuguhkan *form* yang sudah terisi dengan data aiml sesuai data aiml

yang ingin diubah yang didapatkan dari tabel aiml. Setelah admin mengubah isi dari *form* yang telah disediakan, maka data akan diverifikasi untuk mengetahui apakah data aiml berhasil diubah atau tidak. Jika berhasil maka data aiml tersebut akan diubah pada data yang baru dalam tabel aiml dan menampilkan notifikasi berhasil, tetapi jika gagal maka sistem akan menampilkan notifikasi gagal dan akan diarahkan kembali ke halaman aiml.



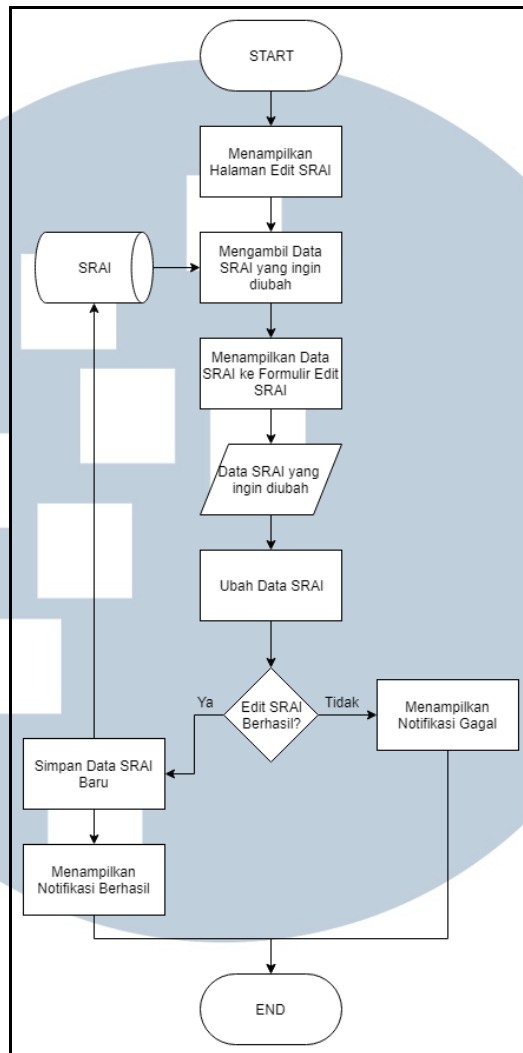
Gambar 3.17 Flowchart Hapus AIML

Gambar 3.17 menggambarkan alur proses untuk menghapus data aiml sesuai data yang ingin dihapus oleh admin. Pada halaman aiml, jika admin memilih salah satu data aiml yang ingin dihapus maka sistem akan memberikan notifikasi untuk melanjutkan proses hapus data aiml atau tidak. Jika admin menyetujui untuk menghapus file tersebut maka secara otomatis data aiml tersebut akan terhapus dari tabel aiml, tetapi jika admin tidak menyetujui untuk menghapus file tersebut maka data tersebut tidak akan terhapus dari *database*.



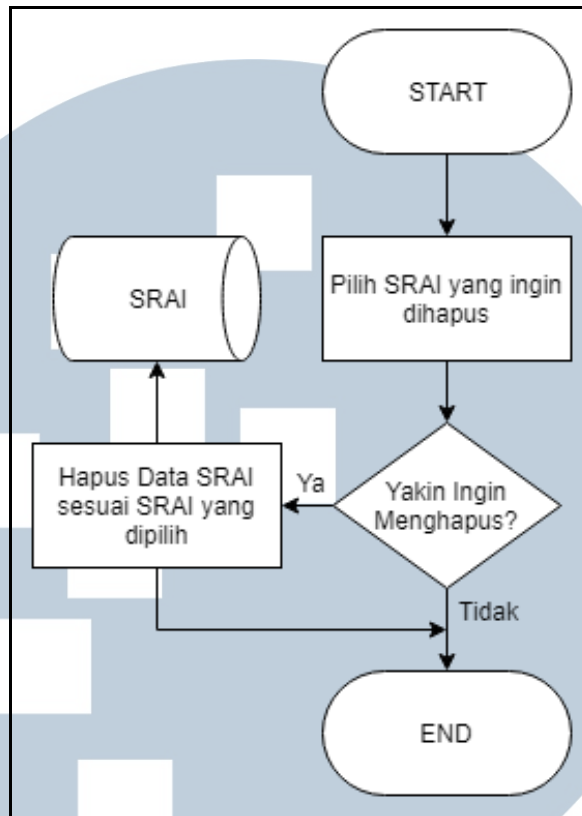
Gambar 3.18 Flowchart Halaman SRAI

Pada menu selanjutnya admin dapat memilih menu srai yang akan diarahkan ke halaman srai. Secara sederhana halaman srai merupakan halaman untuk mengetahui hubungan antara respon yang satu dengan lainnya di tabel aiml. Alur proses srai ditunjukkan pada Gambar 3.18. Saat admin mengakses halaman srai, admin akan disuguhkan data-data terkait data srai *chatbot* yang didapat dari tabel srai_lookup. Selain itu, admin juga dapat mengubah data srai yang sudah ada dan menghapus data srai yang ingin dihapus.



Gambar 3.19 Flowchart Halaman Edit SRAI

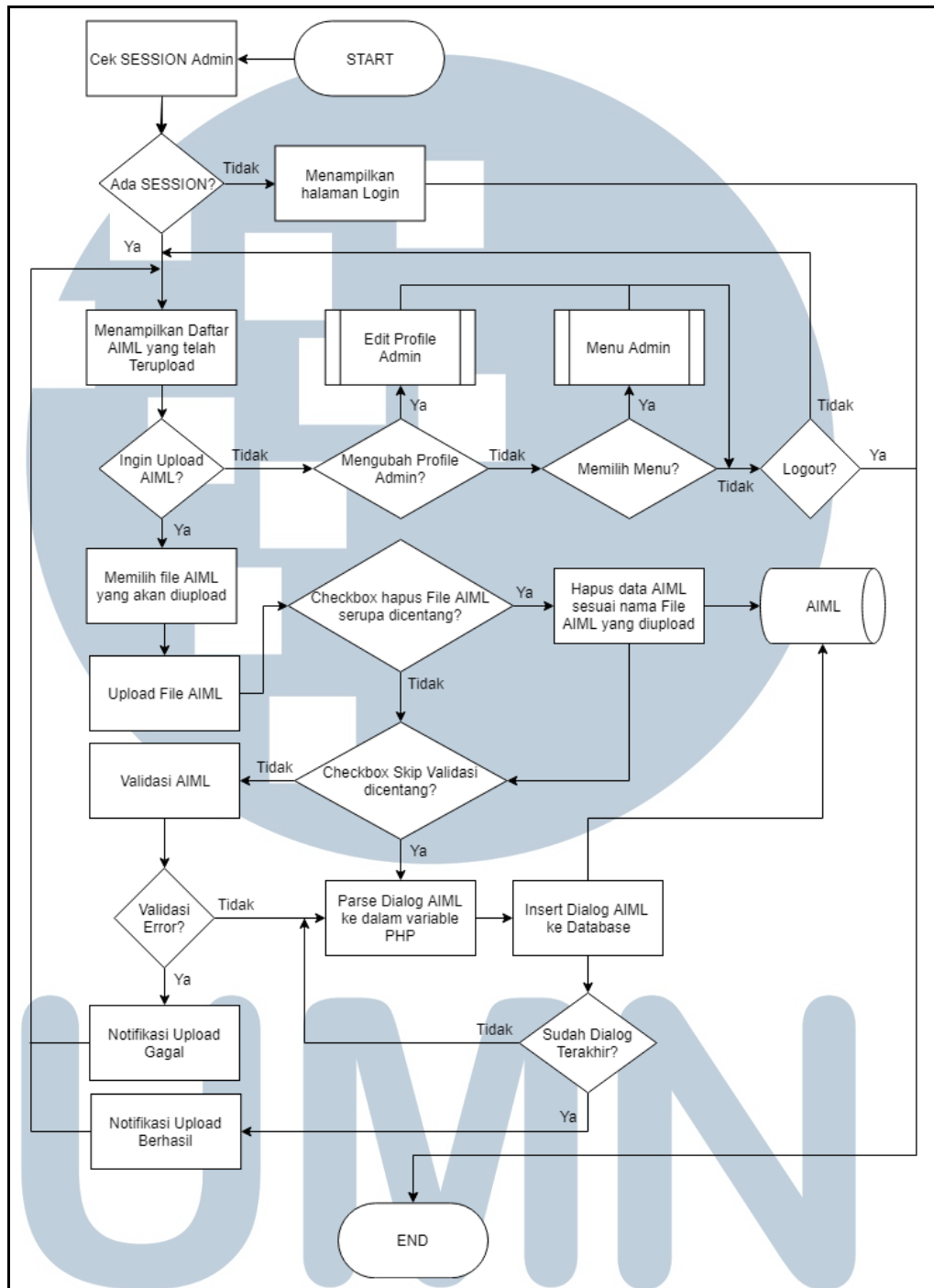
Gambar 3.19 menunjukkan alur yang terjadi pada saat admin ingin mengubah data srai yang sudah ada ke *database*. Pada saat mengakses halaman *edit* srai, admin akan disuguhkan *form* yang sudah terisi dengan data srai sesuai data srai yang ingin diubah yang didapatkan dari tabel *srai_lookup*. Setelah admin mengubah isi dari *form* yang telah disediakan, maka data akan diverifikasi untuk mengetahui apakah data srai berhasil diubah atau tidak. Jika berhasil maka data srai tersebut akan diubah pada data yang baru dalam tabel *srai_lookup* dan menampilkan notifikasi berhasil, tetapi jika gagal maka sistem akan menampilkan notifikasi gagal dan akan diarahkan kembali ke halaman srai.



Gambar 3.20 Flowchart Delete SRAI

Gambar 3.20 menggambarkan alur proses untuk menghapus data srai sesuai data yang ingin dihapus oleh admin. Pada halaman srai, jika admin memilih salah satu data srai yang ingin dihapus maka sistem akan memberikan notifikasi untuk melanjutkan proses hapus data srai atau tidak. Jika admin menyetujui untuk menghapus file tersebut maka secara otomatis data srai tersebut akan terhapus dari tabel *srai_lookup*, tetapi jika admin tidak menyetujui untuk menghapus file tersebut maka data tersebut tidak akan terhapus dari *database*.

Pada menu selanjutnya admin dapat memilih menu *upload* aiml yang akan diarahkan ke halaman *upload* aiml. Alur proses *upload* aiml ditunjukkan pada Gambar 3.21. Pada halaman ini admin dapat menambahkan data aiml yang dimiliki dalam bentuk file berekstensi “.aiml”. Untuk melakukan *upload* file aiml, terdapat beberapa pilihan yang dapat dipilih oleh admin sebelum melakukan proses *upload*.

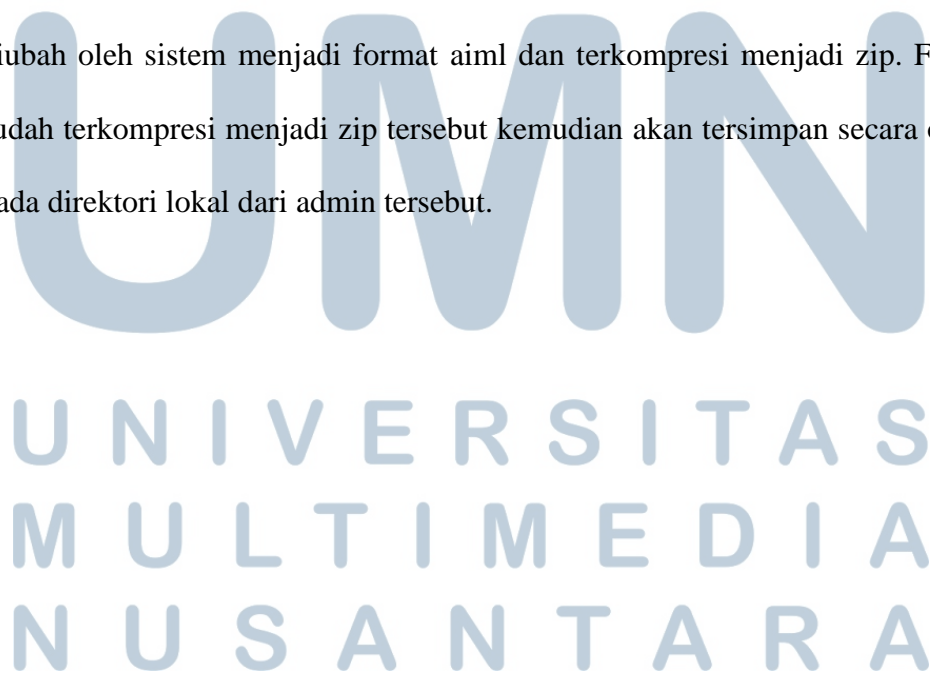


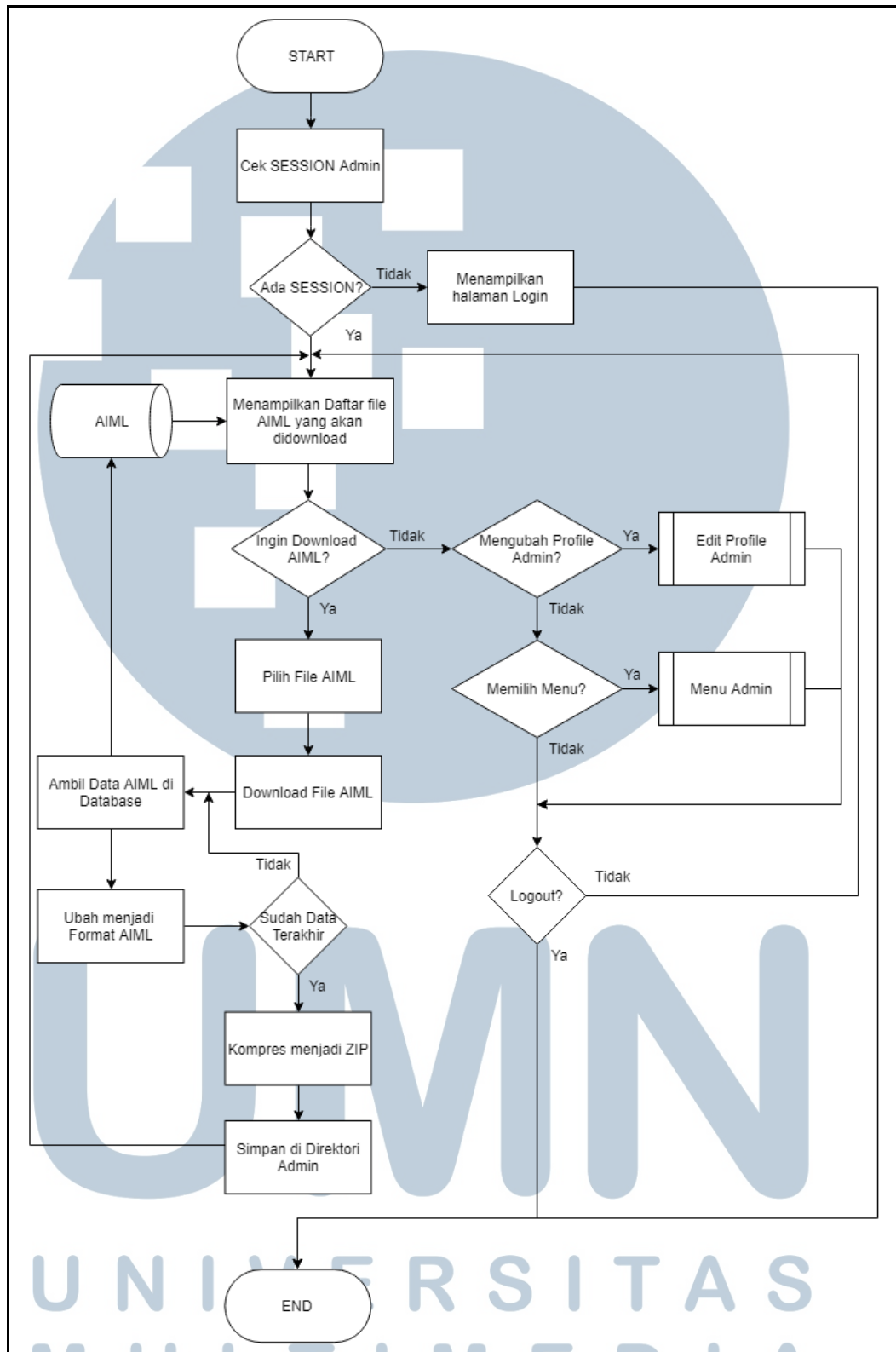
Gambar 3.21 Flowchart Halaman Upload AIML

Pilihan yang dapat dipilih oleh admin yaitu hapus file aiml yang sama dengan file yang akan diupload dan melewati proses validasi dari file yang akan di-upload. Jika admin memilih pilihan pertama, maka pada saat *upload* aiml berhasil, data dengan nama file yang sama di *database* akan dihapus terlebih dahulu kemudian

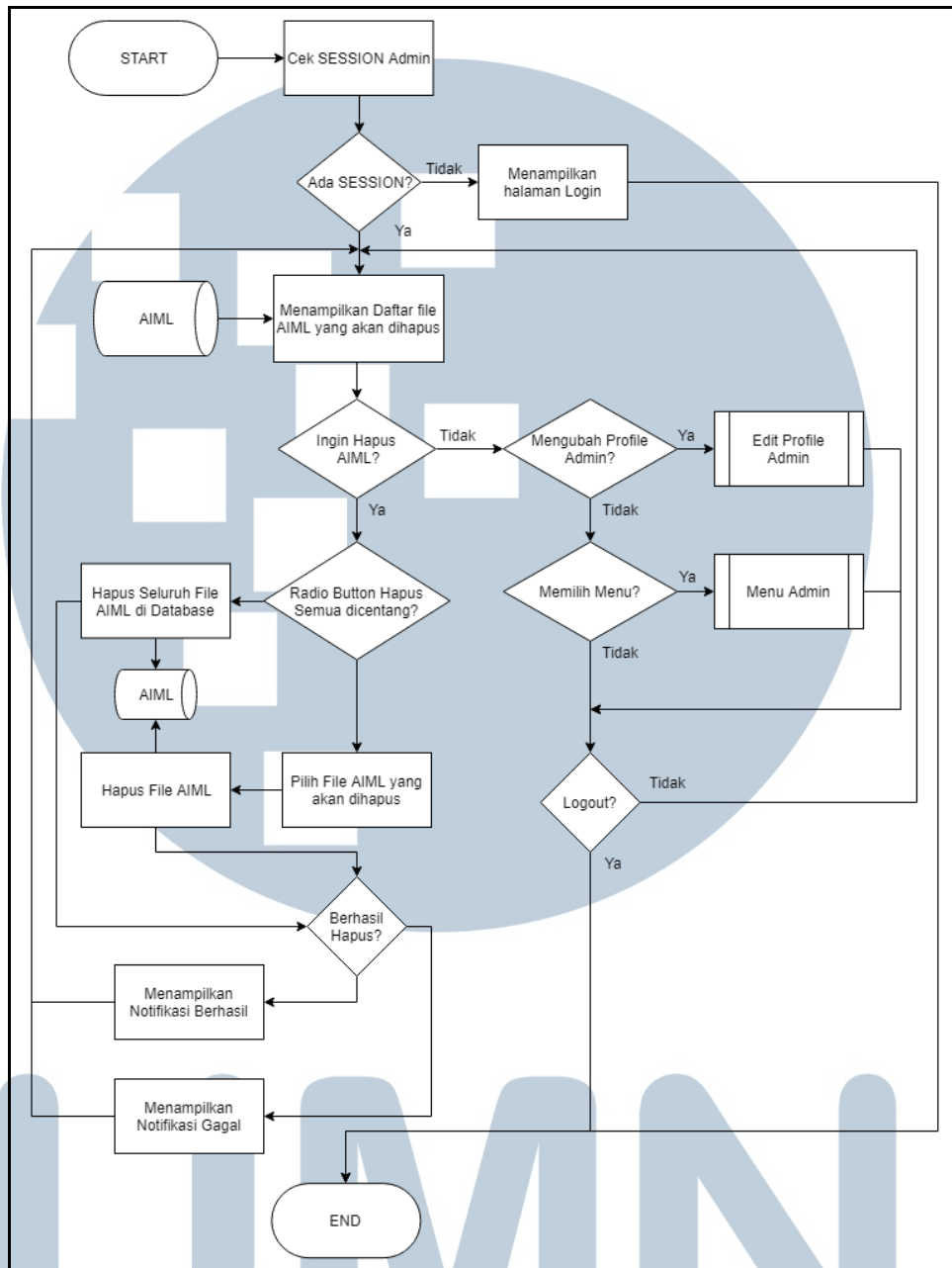
diubah dengan data yang baru. Pada saat admin memilih pilihan kedua, data dari file aiml yang baru akan diproses ke database tanpa melalui proses validasi terlebih dahulu. Jika data yang masuk tidak sesuai standar aiml, maka pada saat komunikasi dengan *chatbot*, respon yang dikirimkan ke *user* dikhawatirkan akan tidak sesuai dengan respon yang diharapkan. Admin juga dapat memilih kedua kategori tersebut secara bersamaan sehingga pada saat proses upload berhasil, data yang di-*upload* tidak akan melewati proses validasi dan data sebelumnya dengan nama file yang sama akan dihapus terlebih dahulu dari *database* kemudian diganti dengan data yang baru.

Menu selanjutnya yaitu *Download AIML* yang akan diarahkan ke halaman *download aiml*. Alur proses *download aiml* digambarkan pada Gambar 3.22. Saat admin memasuki halaman ini, sistem akan menampilkan daftar file aiml yang terdapat di *database*. File yang ditampilkan dapat dipilih untuk di *download* oleh admin. Jika admin memilih salah satu file tersebut, maka seluruh data yang berhubungan dengan file tersebut akan ter-*download* dari *database* kemudian diubah oleh sistem menjadi format aiml dan terkompresi menjadi zip. File yang sudah terkompresi menjadi zip tersebut kemudian akan tersimpan secara otomatis pada direktori lokal dari admin tersebut.





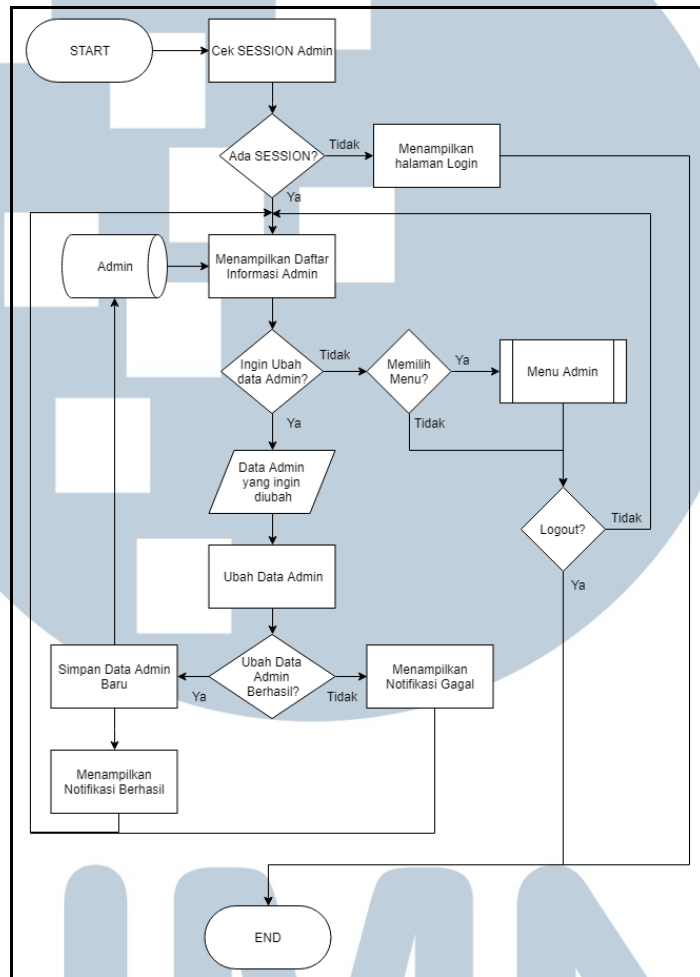
Gambar 3.22 Flowchart Halaman Download AIML



Gambar 3.23 Flowchart Halaman Hapus AIML

Gambar 3.23 menggambarkan *flowchart* hapus aiml. Pada halaman ini, admin dapat memilih file aiml yang dapat dihapus atau admin dapat menghapus seluruh file sekaligus yang terdapat di *database*. Jika admin memilih file tertentu yang akan dihapus maka hanya file itulah yang akan terhapus dari *database*, tetapi jika admin memilih untuk menghapus seluruh file, maka seluruh file aiml yang terdapat di *database* akan dihapus. Dalam proses ini, saat admin melakukan penghapusan data

aiml, data yang telah terhapus tidak dapat dikembalikan lagi seperti semula. Admin hanya dapat memasukkan data satu per satu atau meng-*upload* data aiml kembali pada menu *upload* aiml jika admin menginginkan data tersebut kembali.



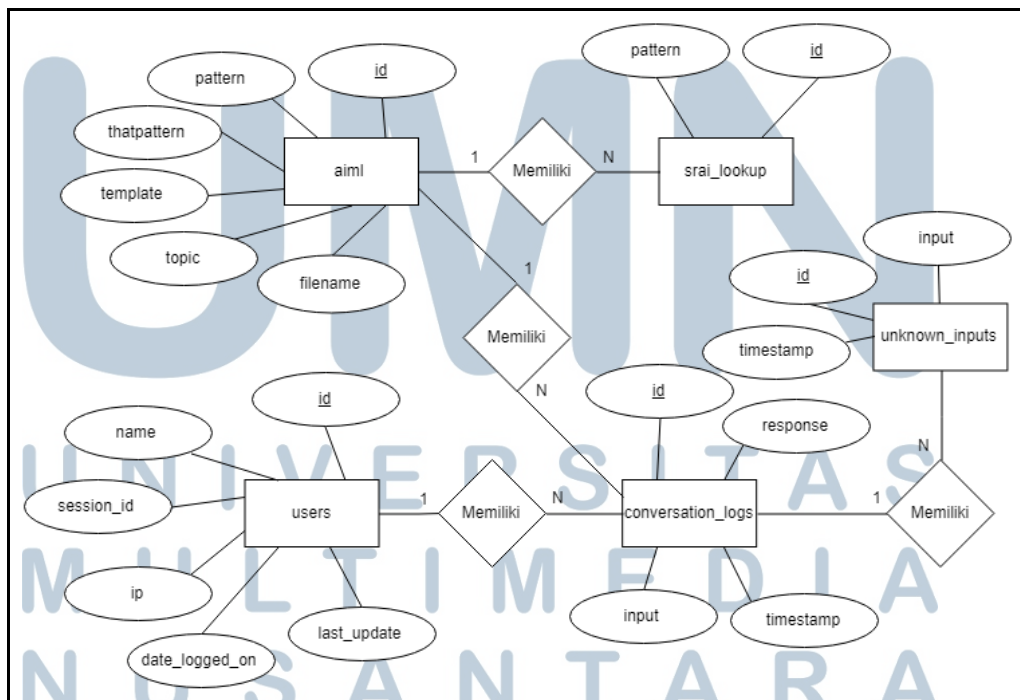
Gambar 3.24 Flowchart Halaman Edit Profil Admin

Gambar 3.24 menunjukkan *flowchart* untuk *edit* profile admin. Untuk masuk pada halaman ini, admin dapat mengaksesnya dengan cara men-*click* nama admin di bagian *header* halaman *web chatbot* dan pilih profile, maka secara otomatis admin akan diarahkan ke halaman ubah profil admin. Pada halaman ini, semua informasi admin yang tersimpan di *database* akan dimunculkan pada *form* untuk merubah data admin. Jika admin ingin merubah data tersebut, admin dapat merubahnya pada *form* yang telah disediakan dan melakukan *submit* data. Data

yang ter-*submit* akan diverifikasi apakah proses perubahan data tersebut berhasil atau tidak. Jika berhasil, maka data admin yang baru akan disimpan pada tabel admin dan akan muncul notifikasi bahwa perubahan data admin berhasil dilakukan. Tetapi jika gagal, maka admin akan diarahkan ke halaman perubahan admin dan akan muncul notifikasi bahwa perubahan data admin gagal. Namun jika admin tidak ingin merubah data, admin dapat keluar pada halaman tersebut dengan menekan tombol keluar yang akan diarahkan ke halaman *dashboard* atau admin dapat memilih menu lainnya untuk masuk ke halaman yang diinginkan.

3.5.4 Entity Relationship Diagram (ERD)

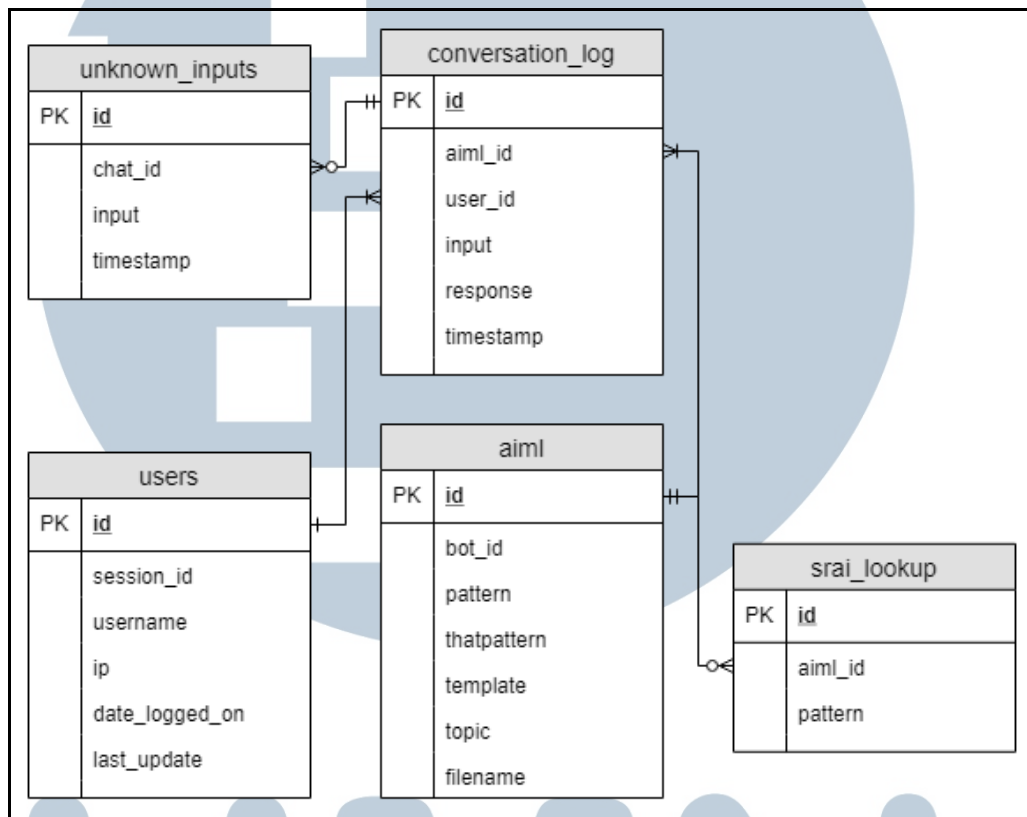
Entity Relationship Diagram (ERD) digunakan untuk menggambarkan hubungan antara suatu entitas dengan entitas yang lainnya. Terdapat 5 entitas pada Aplikasi Chatbot Informasi Objek Wisata yang ditunjukkan pada Gambar 3.25. Entitas tersebut yaitu *aiml*, *conversation_log*, *srai_lookup*, *unknown_inputs*, *users*. Relasi yang terdapat pada setiap entitas yang terhubung yaitu relasi *one to many*.



Gambar 3.25 ERD Chatbot Informasi Objek Wisata

3.5.5 Database Schema

Berdasarkan ERD pada Gambar 3.25, dirumuskan *Database Schema* Chatbot Informasi Objek Wisata yang merujuk pada hubungan antara tabel dengan tabel lainnya. *Database Schema* ditunjukkan pada Gambar 3.26.



Gambar 3.26 Database Schema Chatbot Informasi Objek Wisata

3.5.6 Struktur Tabel

Struktur tabel pada Chatbot Informasi Objek Wisata adalah sebagai berikut.

a. Tabel aiml

Tabel ini berisikan *knowledge base* berbentuk aiml yang disimpan dalam format SQL. Tabel aiml terdiri dari 6 kolom, yaitu kolom “id” yang merupakan *Primary Key* dari tabel aiml, kolom “pattern” yang menyimpan pola masukan *user*, kolom “thatpattern” yang menyimpan data dengan *tag* <that> pada file aiml, kolom “template” yang berisikan respon terhadap pertanyaan dari *user*, kolom

“topic” yang berisikan topik pembicaraan *user*, dan kolom “filename” yang berisi nama *raw* file dari dialog *aiml* tersebut. Struktur tabel ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Struktur Tabel AIML

Name	Type	Default	Other
id	int(11)	None	Primary Auto Increment
pattern	varchar(255)	None	
thatpattern	varchar(255)	None	
template	text	None	
topic	varchar(255)	None	
filename	varchar(255)	None	

b. Tabel *srai_lookup*

Tabel ini berisi daftar *template* AIML yang menggunakan *tag* <*srai*>. Pada tabel ini terdapat 3 kolom, yaitu kolom “id” yang merupakan *Primary Key* dari tabel *srai_lookup*, kolom “*aiml_id*” yang merupakan *Foreign Key* dari tabel *aiml*, dan kolom “*pattern*” yang menyimpan pola masukkan *user*. Struktur tabel ini dapat dilihat pada Tabel 3.2.

Tabel 3.2 Struktur Tabel *Srai_lookup*

Name	Type	Default	Other
id	int(11)	None	Primary Auto Increment
<i>aiml_id</i>	int(11)	None	Foreign Key
<i>pattern</i>	varchar(255)	None	Unique

c. Tabel *stopwords*

Tabel ini berisi daftar kata yang akan dibuang dari kalimat pertanyaan *user*. Pada tabel ini terdapat 2 kolom, yaitu kolom “id” yang merupakan *Primary Key* dari tabel *stopwords*, dan kolom “*stopword*” yang berisi kata yang akan dibuang pada saat proses *Stopword Removal*. Struktur tabel ini dapat dilihat pada Tabel 3.3.

Tabel 3.3 Struktur Tabel Stopwords

Name	Type	Default	Other
id	int(11)	None	Primary Auto Increment
stopword	varchar(255)	None	Unique

d. Tabel conversation_log

Tabel ini berisi riwayat percakapan yang terjadi antara *user* dengan *chatbot*. Pada tabel ini terdapat 6 kolom, yaitu kolom “id” yang merupakan *Primary Key* dari tabel stopwords, dan kolom “user_id” yang merupakan *Foreign Key* dari tabel users, kolom “aiml_id” yang merupakan *Foreign Key* dari tabel aiml, kolom “input” merupakan kolom yang menampung pertanyaan *user*, kolom “response” merupakan kolom yang menampung respon dari *chatbot*, dan kolom “timestamp” merupakan kolom yang mencatat waktu yang diterima oleh sistem pada saat *chatbot* berhasil memberikan respon kepada *user*. Struktur tabel ini dapat dilihat pada Tabel 3.4.

Tabel 3.4 Struktur Tabel Conversation_log

Name	Type	Default	Other
id	int(11)	None	Primary Auto Increment
user_id	int(11)	None	Foreign Key
aiml_id	int(11)	None	Foreign Key
input	text	None	
response	longtext	None	
timestamp	timestamp	CURRENT_TIMESTAMP	

e. Tabel tb_katadasar

Tabel ini berisi daftar kata dasar yang akan digunakan pada proses pencocokan pola. Pada tabel ini terdapat 2 kolom, yaitu kolom “id” yang merupakan *Primary Key* dari tabel stopwords, dan kolom “katadasar” yang berisi kata yang akan dicocokkan pada saat proses *Stemming*. Struktur tabel ini dapat dilihat pada Tabel 3.5.

Tabel 3.5 Struktur Tabel Tb_katadasar

Name	Type	Default	Other
id	int(11)	None	Primary Auto Increment
katadasar	varchar(255)	None	Unique

f. Tabel admin

Tabel ini berisikan segala informasi yang berhubungan dengan admin. Struktur tabel admin dijabarkan pada Tabel 3.6.

Tabel 3.6 Struktur Tabel Admin

Name	Type	Default	Other
id	int(11)	None	Primary Auto Increment
username	varchar(50)	None	Unique
password	varchar(100)	None	
name	varchar(50)	None	
last_ip	varchar(20)	None	
last_login	timestamp	on update CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP

g. Tabel users

Tabel ini berisikan segala informasi *user* yang akan berkomunikasi dengan *chatbot* Informasi Objek Wisata. Pada tabel ini, data *user* diperoleh pada saat *user* menambahkan *Official Line* Pariwisata Jakarta pada aplikasi Line Messenger. Jika *user* melakukan *block* pada *Official Line* Pariwisata Jakarta, maka secara otomatis data yang ada di tabel ini akan terhapus dan *user* tidak dapat berkomunikasi dengan *chatbot*. Struktur tabel *user* dijabarkan pada Tabel 3.7.

Tabel 3.7 Struktur Tabel Users

Name	Type	Default	Other
id	int(11)	None	Primary Auto Increment
session_id	varchar(255)	None	Unique
name	varchar(50)	None	
ip	varchar(20)	None	
date_logged_on	timestamp	CURRENT_TIMESTAMP	
last_update	timestamp	CURRENT_TIMESTAMP	

h. Tabel *unknown_inputs*

Tabel ini berisikan data pertanyaan *user* yang tidak dapat dijawab oleh *chatbot*. Tabel ini berisi 4 kolom, yaitu kolom “id” yang merupakan *Primary Key* dari tabel *unknown_inputs*, kolom “chat_id” yang merupakan *Foreign Key* dari tabel *conversation_log*, kolom “input” yang berisikan daftar pertanyaan yang tidak dapat dijawab oleh *chatbot*, dan kolom “timestamp” yang merupakan kolom yang mencatat waktu yang diterima oleh sistem pada saat *chatbot* berhasil memberikan respon kepada *user*. Struktur tabel *unknown_inputs* dijabarkan pada Tabel 3.8.

Tabel 3.8 Struktur Tabel *Unknown_inputs*

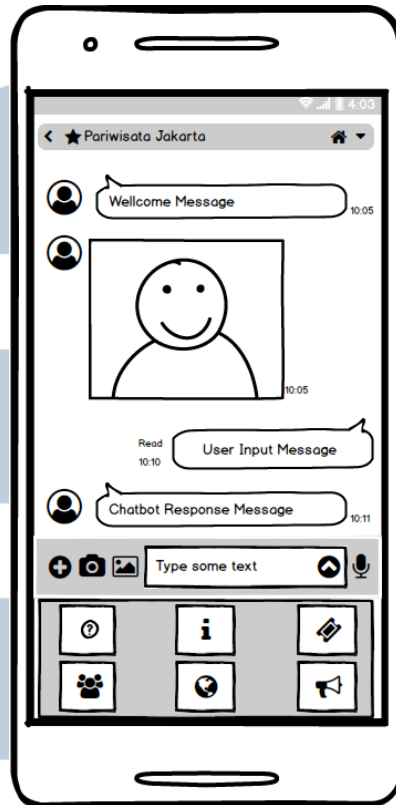
Name	Type	Default	Other
id	int(11)	None	Primary Auto Increment
chat_id	int(11)	None	Foreign Key
input	text	None	
timestamp	timestamp	CURRENT_TIMESTAMP	

3.5.7 Perancangan Antarmuka Pengguna

Perancangan antarmuka dibuat untuk menggambarkan tampilan *chatbot* yang menggunakan aplikasi Line Messenger dan *Content Management System* (CMS) pada perancangan penelitian ini.

a. Antarmuka Aplikasi Chatbot

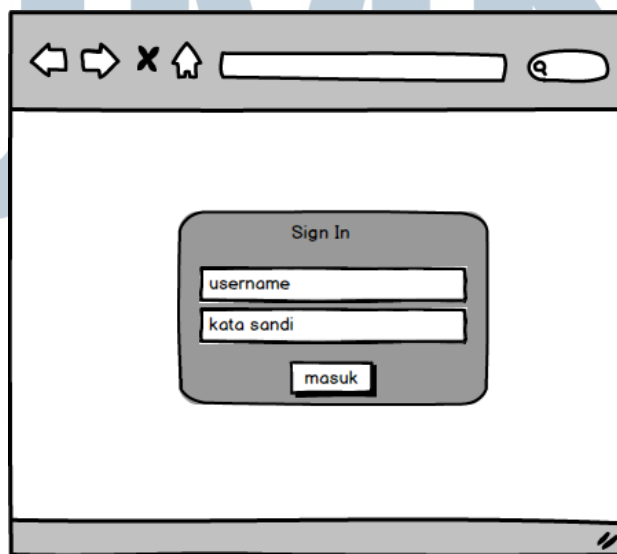
Gambar 3.27 menggambarkan *wireframe* Halaman Chatbot Informasi Objek Wisata pada aplikasi Line Messenger. Halaman ini akan muncul ketika *user* menambahkan sebagai teman *Official Account* dari Pariwisata Jakarta. Jika berhasil menambahkan teman, maka *chatbot* akan mengirimkan pesan awal kepada *user*. Setelah itu *user* dapat berkomunikasi langsung dengan *chatbot* dengan menanyakan segala hal tentang informasi objek wisata yang ada di DKI Jakarta.



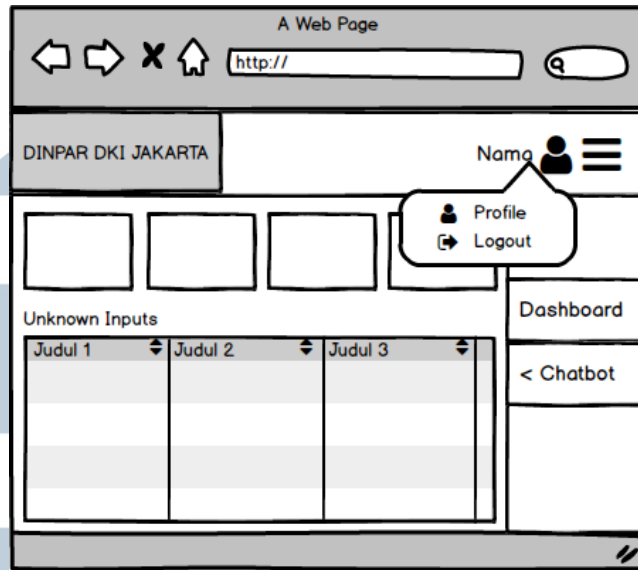
Gambar 3.27 Wireframe Halaman Chatbot Aplikasi Line Messenger

b. Antarmuka Website Admin

Gambar 3.28 menggambarkan *wireframe* Halaman Login Admin. Untuk dapat masuk ke halaman selanjutnya admin diharuskan memasukkan *username* dan *password* yang tepat.

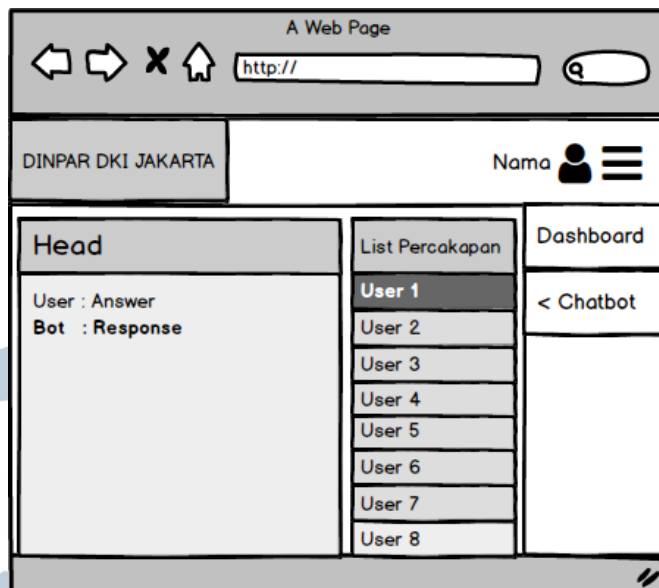


Gambar 3.28 Wireframe Halaman Login Admin



Gambar 3.29 Wireframe Halaman Dashboard Admin

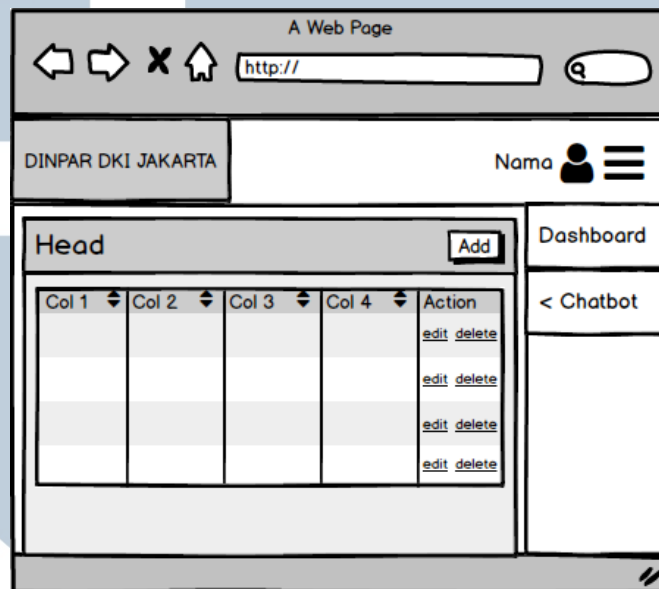
Gambar 3.29 menggambarkan *wirefram* Halaman Dashboard Admin. Terdapat data yang akan ditampilkan oleh sistem kepada admin pada saat admin masuk ke halaman *dashboard*, yaitu data jumlah *user*, jumlah *aiml*, jumlah riwayat percakapan dan jumlah masukan yang tidak dikenali. Selain itu admin juga akan disuguhkan data terkait pertanyaan *user* yang tidak dapat dijawab oleh *chatbot*.



Gambar 3.30 Wireframe Halaman Riwayat Dialog

Gambar 3.30 menggambarkan Halaman Riwayat Dialog. Pada halaman ini, diskusi antara *user* dengan *chatbot* akan ditampilkan sesuai dengan keinginan

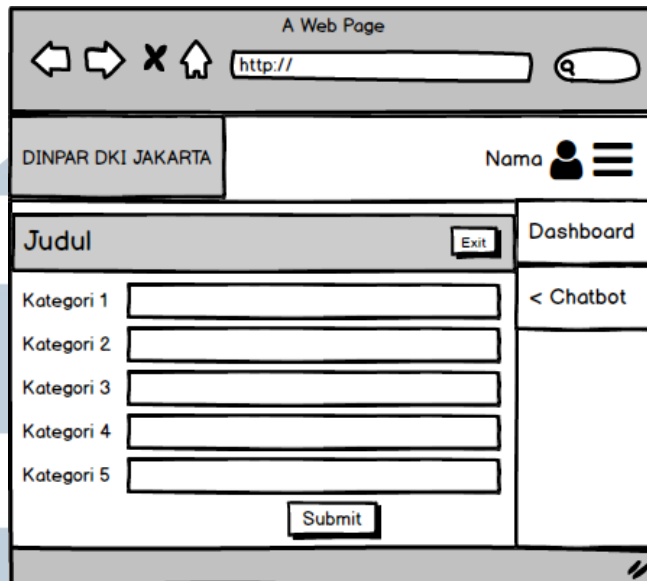
admin. Admin dapat memilih percakapan dari salah satu *user* kemudian percakapan tersebut akan ditampilkan oleh sistem. Selain itu, admin juga dapat memilih kategori dalam melihat percakapan antara *user* dengan *chatbot*. Kategori yang disediakan yaitu 30 percakapan terakhir dan percakapan hari ini.



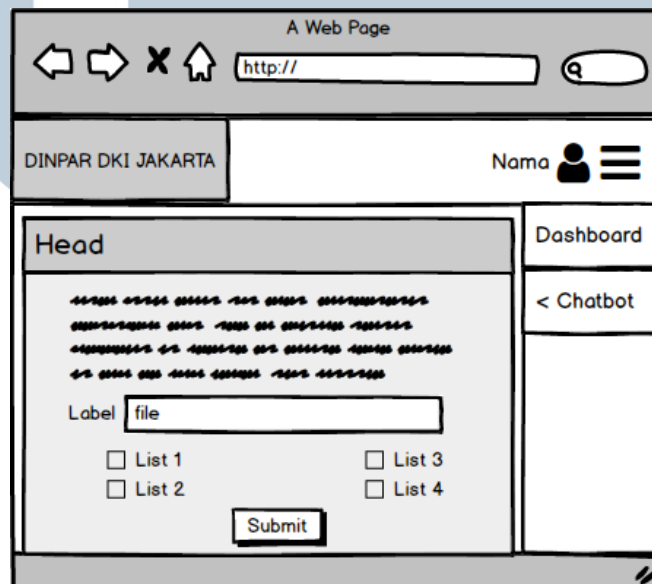
Gambar 3.31 Wireframe Halaman AIML dan SRAI

Gambar 3.31 menggambarkan *wireframe* Halaman AIML dan SRAI. Tampilan untuk halaman AIML dan SRAI seperti Gambar 3.31, hanya saja pada setiap kategori, data yang ditampilkan akan berbeda sesuai dengan menu pilihan admin. Jika menu AIML yang dipilih, maka data yang ditampilkan adalah data AIML. Sedangkan jika menu SRAI yang dipilih oleh admin, maka data yang akan ditampilkan adalah data SRAI.

Halaman Tambah AIML, Edit AIML dan Edit SRAI dirancang sesuai Gambar 3.32. Pada *wireframe* ini, baik untuk menambahkan maupun mengubah data aiml atau srai, tampilan *form* yang digunakan sama satu dengan yang lain.



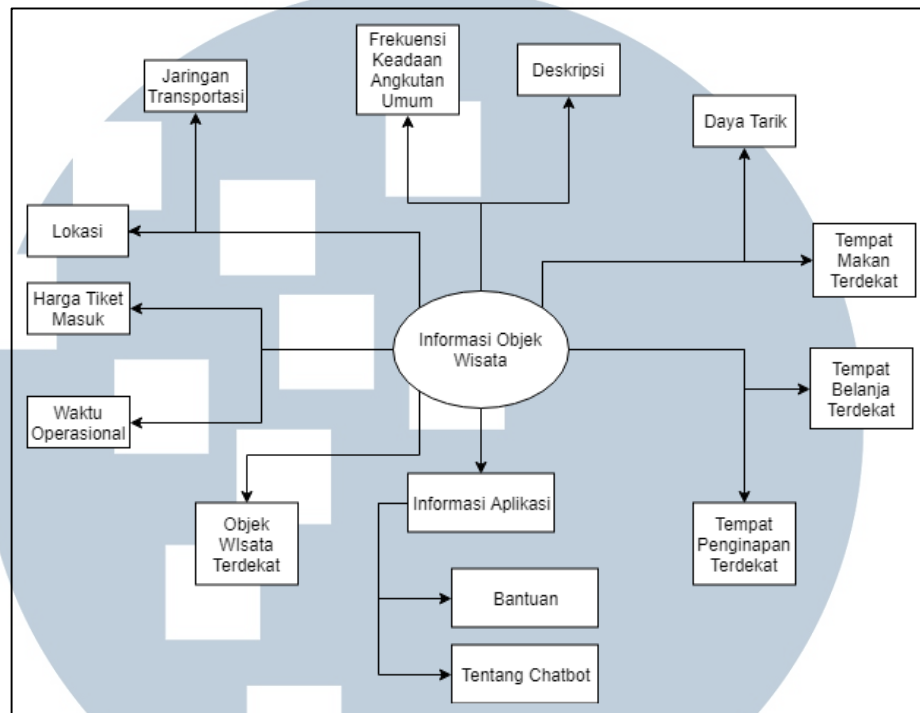
Gambar 3.32 Wireframe Halaman Add dan Edit AIML SRAI



Gambar 3.33 Wireframe Halaman Upload, Download dan Hapus AIML

Halaman Upload AIML, Download AIML, dan Hapus AIML berisi *form* pada bagian tengah halaman. Pada halaman *upload* aiml, terdapat sebuah *input* file yang berfungsi untuk mencari file aiml yang akan di-*upload*. Pada halaman *download* aiml, terdapat *list* data aiml yang dapat diunduh oleh admin. Sedangkan pada halaman hapus aiml, admin dapat menghapus file dari *database* sesuai dengan file yang dipilih sebelumnya.

3.5.8 Perancangan Dialog Komunikasi Chatbot

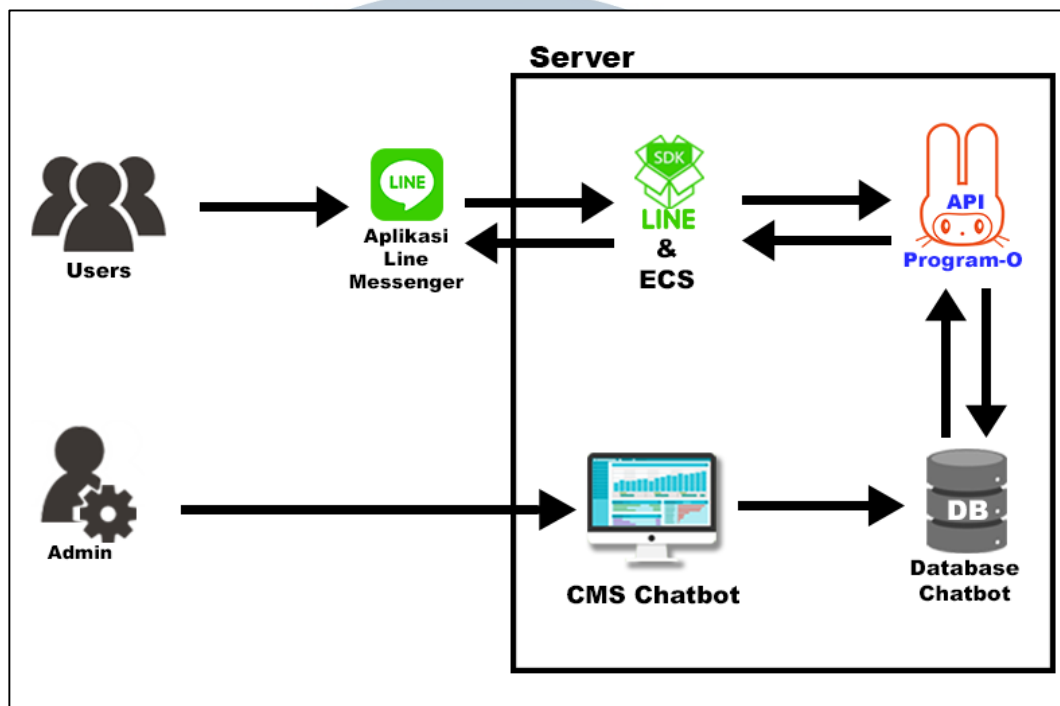


Gambar 3.34 Dialog Komunikasi Chatbot

Gambar 3.34 menunjukkan topik dialog komunikasi *chatbot* pada Aplikasi *Chatbot* Informasi Objek Wisata. Topik utama yang digunakan sebagai acuan awal atau *knowledge base chatbot* yaitu deskripsi, lokasi, harga tiket masuk, waktu operasional, daya tarik, tempat makan terdekat, tempat belanja terdekat, tempat penginapan terdekat, objek wisata terdekat, jaringan transportasi, frekuensi keadaan angkutan umum dan informasi terkait aplikasi *chatbot*.

Data-data yang digunakan sebagai dialog komunikasi diperoleh melalui hasil survei yang diperkuat dengan persetujuan dari Dinas Pariwisata DKI Jakarta. Dialog komunikasi dirancang dengan melakukan uji coba terhadap setiap kemungkinan pertanyaan yang dapat ditanyakan oleh *user*. Perancangan dialog komunikasi dilakukan dengan bantuan *website* admin dengan menambahkan file *aiml* kedalam *database* sebagai *knowledge base chatbot*.

3.5.9 Topologi Chatbot



Gambar 3.35 Topologi Chatbot

Gambar 3.35 menggambarkan topologi *chatbot* informasi objek wisata. *User* dapat berkomunikasi dengan *chatbot* dengan menggunakan aplikasi Line Messenger. Pertanyaan yang ditanyakan oleh *user* melalui aplikasi Line Messenger akan diolah oleh SDK Line dan algoritma ECS. Pada tahap ini kalimat pertanyaan akan diolah hingga menghasilkan kata dasar oleh algoritma ECS, dan kalimat dengan kata dasar akan dicocokkan polanya pada *database* dengan menggunakan API Program-O yang berkomunikasi langsung dengan *database chatbot* di *server*. Jika respon ditemukan maka akan dikirimkan kembali *respon* yang sesuai ke API Program-O dan akan diteruskan ke SDK Line. Lalu SDK Line akan melakukan *reply message* ke aplikasi Line Messenger. Namun jika respon tidak ditemukan pada *database*, SDK Line akan memberikan *default respon* ke aplikasi Line.

Sedangkan CMS Chatbot, hanya dapat diakses oleh admin. Untuk dapat mengakses CMS Chatbot, admin dapat menggunakan *web browser* dengan

mengarahkan *browser* tersebut ke *link* CMS *chatbot*. CMS *chatbot* dapat digunakan oleh admin untuk mengatur keperluan yang berhubungan dengan *chatbot* informasi objek wisata. CMS Chatbot, SDK Line dan algoritma ECS, API Program-O dan *database chatbot* terletak di *server* yang sama sehingga memudahkan komunikasi antar komponen pembentuk *chatbot* informasi objek wisata.

