BAB III

ANALISIS DAN PERANCANGAN SISTEM

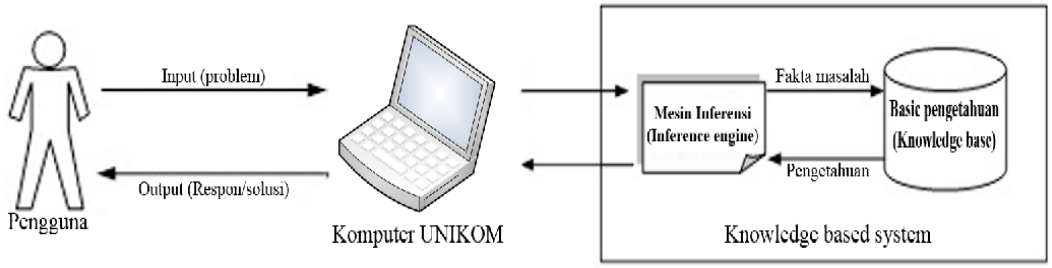
3.1 Analisis Sistem

Analisis dilakukan dengan tujuan untuk mengidentifikasi dan mengevaluasi seluruh komponen yang terkait dengan sistem yang akan dibangun. Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan pada tahap ini akan menyebabkan kesalahan pada tahap selanjutnya. Tahap analisis sistem dilakukan dengan cara menguraikan suatu sistem yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan sehingga ditemukan kelemahan dan keuntungan pada sistem tersebut, sehingga dalam membangun aplikasi menjadi lebih mudah dari hasil analisis sistem yang lama. Analisis sistem ini akan menghasilkan beberapa data dan fakta yang akan dijadikan bahan uji dan analisis menuju pengembangan dan penerapan sebuah aplikasi sistem yang diusulkan.

3.1.1 Analisis Masalah

Permasalahan utama pada tempat penelitian adalah ketidak efektifan aplikasi *chatting* yang digunakan, dimana pengguna pencari informasi menggantungkan informasi yang didapat dari operator yang *online*. Hal ini dapat menyebabkan penundaan penyampaian informasi dari operator ke pengguna pencari informasi atau bahkan pertanyaan yang diajukan tidak dijawab oleh operator, karena banyaknya pengguna yang bertanya.

Selain itu aplikasi *chat bot* yang pernah ada di tempat penelitian yaitu *chat bot* MILA, tidak berbasiskan web, sehingga penyampaian informasi hanya dibatasi di sekitar lingkungan UNIKOM saja.



Gambar 3.1 Skema *Knowledge Based System* Pada *Chat Bot* MILA

Dari gambar diatas dapat dijelas sebagai berikut :

1. Pengguna menginputkan string bahasa alami ke program chat bot yang berada dilingkungan Universitas Komputer Indonesia.
2. Program memproses *data input* dari pengguna menjadi bahasa yang dapat dimengerti oleh komputer.
3. Program komputer menyesuaikan *input* pengguna dengan *pattern* dari *database*.
4. Program *chat bot* memberikan respon ke pengguna.

Sedangkan dalam membangun sebuah perangkat lunak *Natural Language Processing* untuk pembuatan aplikasi *chat bot* berbasis web ini terdapat beberapa permasalahan yang harus ditangani adalah sebagai berikut :

1. Menjadikan bahasa alami atau bahasa yang dipakai oleh manusia sebagai bahasa perantara yang menghubungkan interaksi antara manusia dengan komputer (*chat bot*).

2. Pemrosesan bahasa alami yang dapat memahami kalimat yang mempunyai ambiguitas. Ambiguitas kalimat dapat terjadi saat dua kalimat atau lebih yang berbeda memiliki arti yang sama. Masalah utama yang menyebabkan terjadinya ambiguitas pada kalimat adalah saat ini penggunaan bahasa Indonesia tidak selalu terpaku pada pola yang baku. Maksud dari pernyataan ini adalah dalam kehidupan pergaulan sehari-hari pada saat ini para pengguna bahasa Indonesia tidak terlalu terpaku pada susunan subjek, predikat, ataupun objek (S,P,O), selain itu penggunaan sinonim atau kata-kata yang memiliki bentuk berbeda namun memiliki arti yang sama atau mirip sering kali digunakan. Dengan adanya kebebasan berbahasa seperti ini memungkinkan terbentuk banyak pola kalimat baru yang sebenarnya memiliki maksud atau arti yang sama. Contoh dari kasus ini adalah sebagai berikut:

Tabel 3.1 Penggunaan ambiguitas kalimat dan sinonim.

Input	**Respon**
Siapa nama kamu ?	Nama aku T-ara
Nama kamu siapa ?	Nama aku T-ara
Kamu siapa ?	Nama aku T-ara
Siapa kamu ?	Nama aku T-ara
Panggilan kamu apa ya ?	Nama aku T-ara
Aku panggil kamu apa ya ?	Nama aku T-ara

3. Dalam sebuah percakapan antara *chat bot* dengan pengguna mungkin saja alur pembicaraan menjadi sangat meluas. Dikarenakan terbatasnya pengetahuan pada *brain file* maka diperlukan kemampuan bagi *chat bot* untuk mengalihkan topik pembicaraan sehingga pengguna tidak membicarakan hal-hal yang tidak dikuasai *chat bot*.

4. Agar interaksi antara pengguna dengan *chat bot* layaknya berinteraksi antar sesama manusia, diperlukan kemampuan bagi *chat bot* untuk memberikan respon yang berbeda dan merespon pertanyaan atau pernyataan yang sama dari pengguna. Contoh dari kasus ini adalah sebagai berikut :

   <**Pengguna**> Apa kabar ?

   <**Bot** > *Kabarku baik-baik saja.*

   <**Pengguna**> Apa kabar ?

   <**Bot** > ....*?*

   <**Pengguna**> Apa kabar ?

   <**Bot** > *Mengapa kamu mengulangi perkataanmu ?*

   <**Pengguna**> Ok

   <**Bot** > *Ok.*

   <**Pengguna**> Apa kabar ?

   <**Bot** > *Baik-baik saja.*

   Dari percakapan di atas dapat dilihat bahwa *chat bot* memberikan respon terhadap pertanyaan ”Apa kabar ?” yang diulang beberapa kali dengan mengomentari atau bertanya kembali inputan yang sama dari

pengguna. Selain itu *chat bot* akan memberikan respon yang berbeda dari pertanyaan tersebut. Dengan adanya kemampuan chat bot untuk merespon pertanyaan atau pernyataan yang sama dari pengguna seperti contoh di atas, maka *chat bot* tidak akan terkesan statis dan pengguna tidak akan merasa bosan.

5. Kepintaran *chat bot* ditentukan melalui pengetahuan (*Knowledge*) yang ada dalam *brain file*. Semakin banyak pengetahuan yang ada dalam *brain file* maka akan semakin pintar *chat bot* tersebut. Maka diperlukan sebuah proses *learning* agar *brain file* pada *chat bot* agar pengetahuan *chat bot* dapat dimanipulasi.

3.1.2 Analisis Masukan

Pada pembuatan chat bot berbasis web berbahasa indonesia ini, ada beberapa hal yang tidak dapat diimplementasikan kedalam sistem, seperti penggunaan *stemming*.

3.1.2.1 *Stemming*

Stemming merupakan suatu proses untuk menemukan kata dasar dari sebuah kata dengan menghilangkan imbuhan dari kata tersebut. Penggunaan imbuhan pada suatu kata dalam aturan tata bahasa indonesia dirasa penting karena suatu kata akan memiliki makna yang berbeda dengan kata dasarnya misalnya kata perwalian dengan wali memiliki makna yang berbeda, perwalian merupakan kata kerja, sedangkan wali merupakan pelaku. Oleh karena itu proses

menghilangkan imbuhan pada suatu kata atau disebut dengan *Stemming* tidak cocok diterapkan pada aplikasi *chat bot*.

3.1.3 Analisis Data

Sebagaimana telah dijelaskan pada batasan masalah, *chat bot* mampu untuk memberikan data atau informasi seputar lingkungan jurusan Teknik Informatika. Adapun detail informasi yang dapat diberikan oleh *chat bot* adalah sebagai berikut:

1. Informasi umum mengenai UNIKOM seperti :
    a. Kepanjangan UNIKOM, visi, misi, tujuan dan motto UNIKOM
    b. Ruangan-ruangan yang ada di gedung UNIKOM.
    c. Fakultas dan program studi yang ada.
    d. Nama rektor dan nama dekan setiap fakultas.
2. Informasi seputar perkuliahan di UNIKOM
    a. Penilaian prestasi akademik mahasiswa.
    b. Informasi mengenai alih jurusan dan alih program studi.
    c. Informasi mengenai batas waktu studi.
    d. Informasi mengenai pindahan dari perguruan/ institusi lain.
    e. Informasi mengenai perwalian.
    f. Persyaratan jumlah sks yang dapat diambil.
    g. Standarisasi kelulusan dan yudisium kelulusan.
    h. Sanksi-sanksi akademik.
    i. Informasi mengenai tidak aktif kuliah dan aktif kembali.
    j. Ketentuan dan prosedur pindah jurusan.

k. Ketentuan dan prosedur pengunduran diri.

l. Ketentuan dan prosedur cuti akademik.

m. Informasi dan prosedur registrasi.

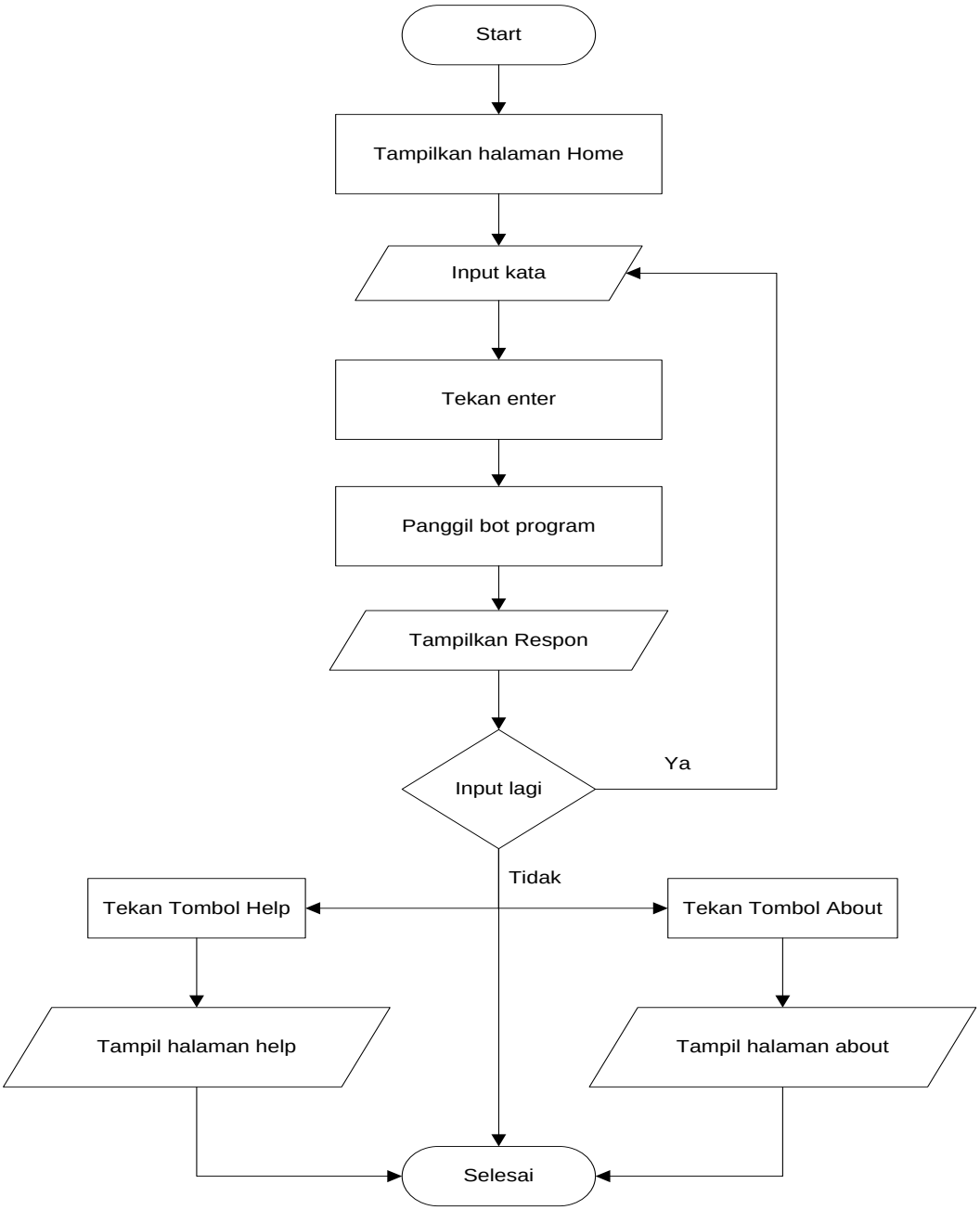
n. Kode etik mahasiswa UNIKOM

3.1.4 Model Kerja Sistem Program

Perancangan bot program merupakan salah satu bagian penting dalam pembuatan *chat bot* berbasis web ini, pada tahap ini digambarkan bagaimana sebuah aplikasi berjalan.

3.1.4.1 Prosedur Tiara

Prosedur TIARA (*Technology of Automatic Responses*) merupakan prosedur utama di mana semua dialog dengan lawan bicaranya dilakukan. Berikut adalah **Gambar 3.2** dari prosedur dari program *chat bot* Tiara :



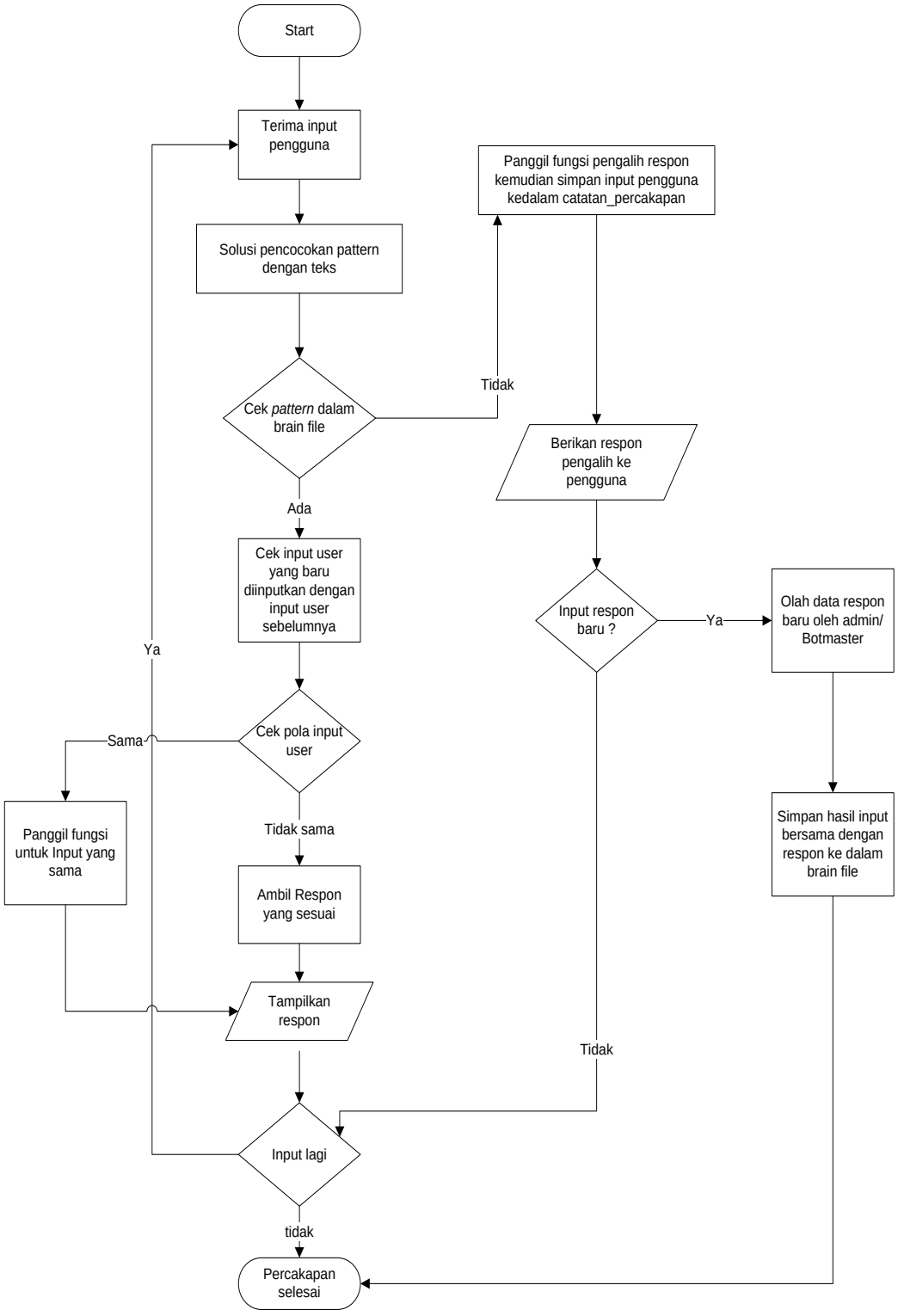
Gambar 3.2 Model Prosedur Tiara

3.1.4.2 Model Kerja Global Bot Program

Pada bagian ini merupakan gambaran bagaimana *bot program* menangani percakapan antara pengguna dengan *chat bot*, sehingga aplikasi dapat berjalan dengan baik.

Inputan yang diberikan oleh pengguna berupa bahasa alami (*Natural Language*) akan melalui proses *parsing* yaitu memecah kalimat menjadi *token-token array*, kemudian *bot program* akan mencari pola kalimat atau *pattern (keyword)* yang sesuai dengan database (*brain file*) untuk menentukan respon yang tepat untuk disampaikan kepada pengguna. Bot program terlebih dahulu akan memerisa apakah inputan pengguna sama dengan yang sebelumnya atau tidak jika pengguna menginputkan kalimat yang sama maka bot program akan meresponnya, kemudian jika tidak ditemukan pola kalimat atau *pattern* yang dimaksud, maka bot program akan memberikan respon atau menanyakan kembali apa yang ditanyakan pengguna, untuk mengalihkan topik pembicaraan.

Pada perancangan *bot program* Tiara ini juga diberikan kemampuan bagi *chat bot* untuk menambah respon baru, jika pola kalimat atau *pattern* tidak ada dalam *brain file.* Berikut adalah *flowchart* rancangan *global chat bot* Tiara :



Gambar 3.3 Rancangan *Global Chat Bot* T-ARA

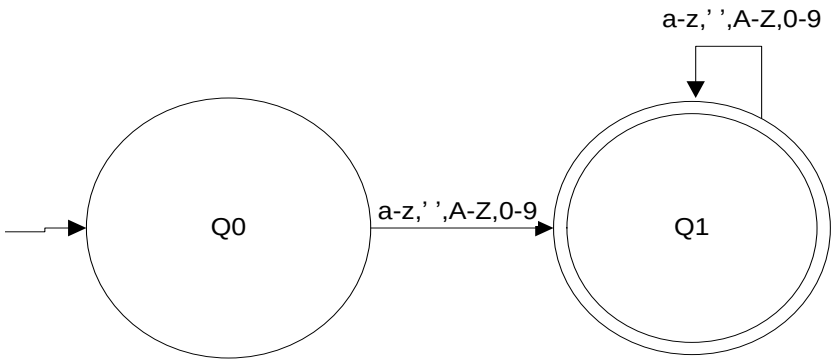
3.1.4.3 Analisis *Parsing*

Proses *parsing* (penguraian kalimat) merupakan proses yang dilakukan untuk menterjemahkan masukan dari pengguna agar dapat dimengerti oleh sistem. Secara *default*, seluruh kalimat masukan pengguna yang dimasukan akan dianggap sebagai kata-kata yang harus ada pada data yang akan dicari. Misal pengguna memasukan kalimat sebagai berikut :

Penggunaan bahasa alami

Maka sistem akan melakukan pencarian data pada dokumen yang memiliki kata *bahasa* **atau** *alami.* Artinya jika terdapat dokumen yang mengandung kata *bahasa inggris* maka dokumen tersebut dianggap sesuai dengan yang dicari pengguna, karena mengandung kata *bahasa.* Hal ini membuat pengguna yang ingin mencari informasi mengenai *bahasa alami* akan mendapatkan hasil yang tidak sesuai dengan yang diinginkan.

Jika pengguna ingin melakukan proses pencarian yang lebih spesifik, agar informasi yang didapat lebih tepat sasaran, maka pengguna haru mengikuti aturan proses penguraian kalimat yang dimiliki sistem, yaitu :



Gambar 3.4 *Finite State Automata* Proses *Parsing*

Konfigurasi FSA diatas secara formal dinyatakan sebagai berikut :

Q = {Q0, Q1}

∑ = {a-z,' ',A-Z,0-9}

S = Q0

F = {Q1}

Fungsi transisi yang ada sebagai berikut :

δ(Q0, a-z|' '|A-Z|0-9) = Q1

δ(Q1, a-z|' '|A-Z|0-9) = Q1

Fungsi tersebut bisa disajika dalam *table* transisi berikut :

Table 3.2 Tabel Transisi

δ	a-z|' '|A-Z|0-9
Q0	Q1
Q1	Q1

Dari diagram state gambar 3.4, didapat aturan produksi sebagai berikut:

S -> aA..zA| AA..ZA | 0A-9A|<space>A|a..Z|A..Z|0..9|<space>

A -> aA..zA| AA..ZA |0A-9A| <space>A|a..Z|A..Z|0..9|<space>

Dimana secara formal dapat ditulis :

V = {S, A}

T = {a-z,' ' A-Z,0-9}

P = {S->aA..zA|AA..ZA|0A..9A|<space>A|a..Z|A..Z|0..9|<space>,
A->aA..zA| AA..ZA |0A..9A|<space>A|a..Z|A..Z|0..9|<space>}

S = S

Aturan produksi diatas menggunakan *left to right* sehingga jika pengguna memasukan pertanyaan “hai juga”, maka proses akan dilakukan dari depan (*left*) ke belakang (*right*), seperti berikut: :

S -> hA

S -> haA

S -> haiA

S -> hai A

S -> hai jA

S -> hai juA

S -> hai jugA

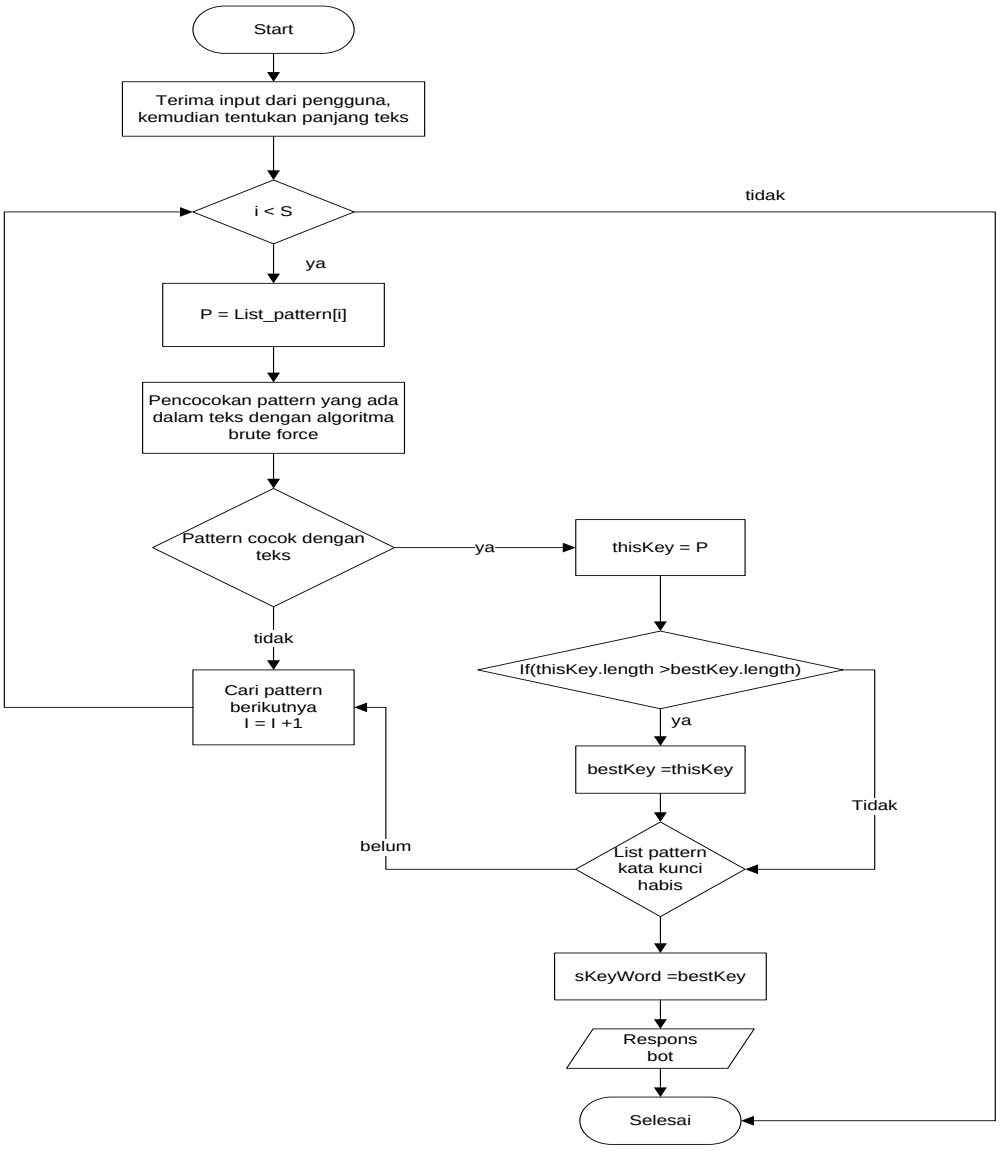
S -> hai jugaA

S -> hai juga

Pada proses di atas setiap kalimat atau kata akan dipecah menjadi perbagian karakter dalam bentuk array.

3.1.4.4 *Flowchart* Penulusuran *Pattern*

Pencarian *pattern* dilakukan saat *chat bot* menerima inputan berupa teks dari pengguna. Dimana *list_pattern* didefinisikan sebagai daftar *pattern* yang ada dalam database, P merupakan *pattern*, S merupakan banyaknya pattern yang ada, dan bestKey diinisialisasikan sebagai variable kosong. Berikut ini *flowchart* bagaimana sistem melakukan penulusuran *pattern* pada database.



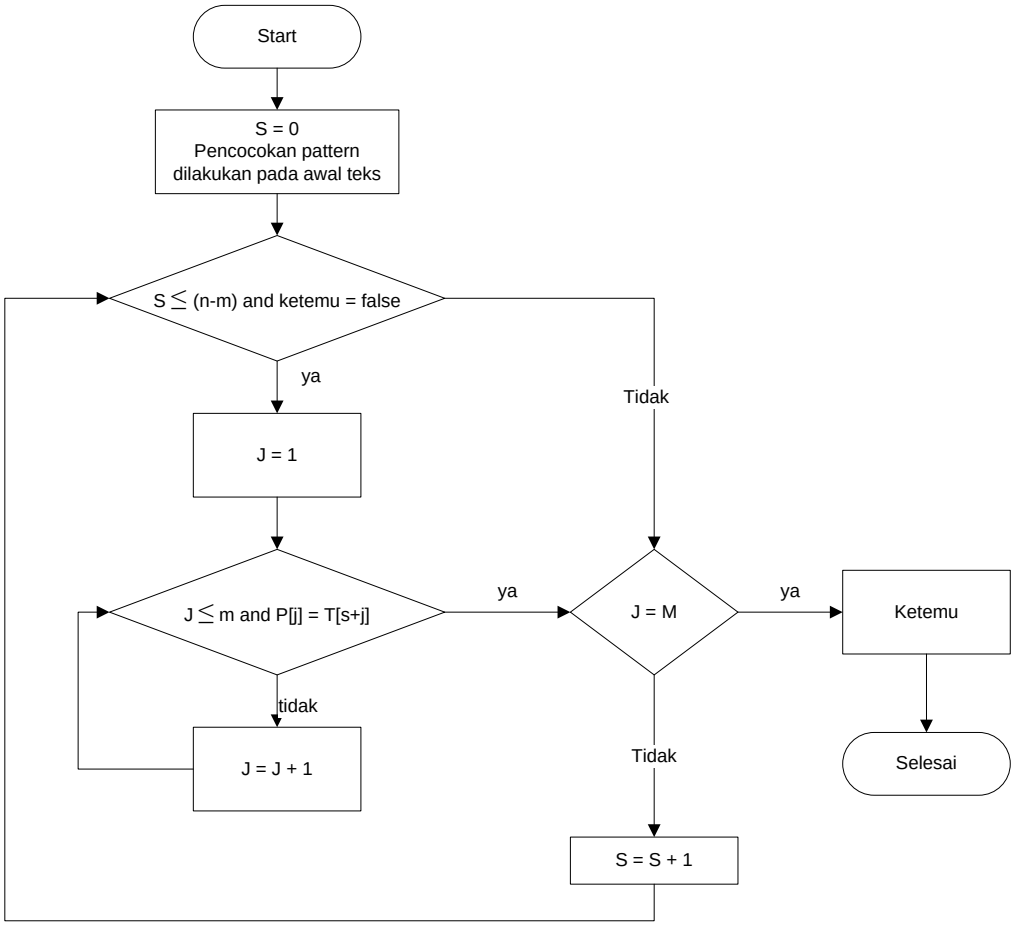
Gambar 3.5 Flowchart Penulusuran *Pattern*

1. Sistem akan membandingkan semua *pattern* yang ada dalam *database* satu persatu dengan *teks* pengguna.
2. Pencocokan pattern pada *teks* dilakukan dengan menggunakan algoritma brute force
3. Jika ditemukan kecocokan *pattern* pada teks, maka langkah berikutnya adalah 5a.
4. Bandingkan *pattern* berikutnya.
5. Jika ditemukan kecocokan pattern pada teks maka lakukan :
   a. Isi *variable thisKey* dengan pattern yang baru ditemukan.
   b. Bandingkan panjang karakter isi *variable thisKey* dengan panjang karakter isi *variable bestKey*.
   c. Jika panjang karakter isi *variable thisKey* lebih besar maka pindahkan isi *variable thisKey* ke *variable bestKey*

      (jika perbandingan panjang karakter *variable thisKey* dengan *variable bestKey* sama besar maka karakter *pattern* yang dipilih berdasarkan urutan paling akhir dari database (*descending*)).
6. Jika tidak ditemukan ketidakcocokan dengan pattern dengan teks maka lakukan pencocokan *pattern* berikutnya.
7. Jika list *pattern* berikutnya masih ada ulangi langkah 4, jika semua pattern sudah diperiksa maka *variable bestKey* yang ditemukan adalah *keyword*.

3.1 4.5 Flowchart Algoritma Brute Force

Salah satu algoritma dalam penyelesaian aplikasi *chat bot* ini adalah algoritma Brute Force. Dimana N didefinisikan sebagai panjang teks dan M didefinisikan sebagai panjang *pattern*.



Gambar 3.6 Flowchart Algoritma Brute Force

Cara Kerja Algoritma Brute Force :

1. Mula-mula *pattern* dicocokan pada awal teks.
2. Dengan bergerak dari kiri ke kanan, bandingkan setiap karakter di dalam *pattern* dengan karakter yang bersesuaian di dalam teks sampai :
    a. Semua karakter yang dibandingkan cocok atau sama (pencarian berhasil), atau
    b. Dijumpai sebuah ketidakcocokan karakter (pencarian belum berhasil)
3. Bila *pattern* belum ditemukan kecocokannya dan teks belum habis, geser *pattern* satu karakter ke kanan dan ulangi langkah 2.

3.1.4.6 Contoh Implementasi Algoritma Brute Force

Berikut ini merupakan contoh bagaimana sistem berjalan pada aplikasi *chat bot* ini. Dimana isi *brain file* memiliki 3 pattern dan teks dari pengguna adalah ”SIAPA REKTOR UNIKOM YAH”.

Diketahui :

Teks yang diinputkan pengguna : ”SIAPA REKTOR UNIKOM YAH”.

Daftar *Pattern* dapat dilihat pada tabel 3.3 :

Tabel 3.3 Tabel Daftar *Pattern* Yang Ada Dalam Database

NO.	PATTERN	RESPON
1.	REKTOR	Rektor adalah orang yang memimpin suatu perguruan tinggi atau Universitas
2.	REKTOR UNIKOM	Dr. Ir. Eddy Suryanto Soegoto, M.Sc.
3.	SINGKATAN UNIKOM	Universitas Komputer Indonesia

1. Sistem akan membandingkan semua *pattern* yang ada dalam *database* satu-persatu dengan teks pengguna, pencocokan *pattern* dengan teks dilakukan dengan algoritma brute force.

2. Pada implementasi algoritma brute force, perulangan satu *pattern* dilakukan dengan perhitungan (panjang karakter teks – panjang karakter *pattern*) + 1.

   Contoh :

   Teks : ”SIAPA REKTOR UNIKOM YAH”, panjang karakter dari teks adalah 23

a. Pencarian Untuk *Pattern* Pertama Yaitu ”REKTOR”

Diketahui :

Panjang karakter ”REKTOR” adalah 6.

Banyak perulangan adalah (panjang karakter teks - panjang karakter *pattern*) + 1

sehingga (23 - 6) + 1 = 18

Tabel 3.4 Tabel Contoh Implementasi Algoritma Brute Force Pencarian *Pattern* 1

	TEKS																					
	S	I	A	P	A		R	E	K	T	O	R		U	N	I	K	O	M		Y	A	H
1	R	E	K	T	O	R																	
2		R	E	K	T	O	R																
3			R	E	K	T	O	R															
4				R	E	K	T	O	R														
5					R	E	K	T	O	R													
6						R	E	K	T	O	R												
7							R	E	K	T	O	R											
8								R	E	K	T	O	R										
9									R	E	K	T	O	R									
10										R	E	K	T	O	R								
11											R	E	K	T	O	R							
12												R	E	K	T	O	R						
13													R	E	K	T	O	R					
14														R	E	K	T	O	R				
15															R	E	K	T	O	R			
16																R	E	K	T	O	R		
17																	R	E	K	T	O	R	
18																		R	E	K	T	O	R

1. Pada pencarian ditemukan *pattern* ”REKTOR” pada teks sehingga simpan *pattern* pada variabel thisKey.

   thisKey ← “REKTOR”.

2. Bandingkan panjang karakter isi variabel bestKey dengan isi variabel thisKey (nilai pertama bestKey adalah ” ”).

3. Karena panjang karakter thisKey lebih besar dibandingkan panjang karakter bestKey, maka isi variabel bestKey dengan thisKey.

4. Sehingga bestKey ← “REKTOR”.

b. Pencarian Untuk Pattern Kedua Yaitu ”REKTOR UNIKOM”

Diketahui :

Panjang karakter ”REKTOR UNIKOM” adalah 13.

Banyak perulangan adalah (panjang karakter teks - panjang karakter *pattern*) + 1

sehingga (23 - 13) + 1 = 11

Tabel 3.5 Tabel Contoh Implementasi Algoritma Brute Force Pencarian *Pattern* 2

	TEKS																					
	S	I	A	P	A		R	E	K	T	O	R		U	N	I	K	O	M		Y	A	H
1	R	E	K	T	O	R		U	N	I	K	O	M									
2		R	E	K	T	O	R		U	N	I	K	O	M								
3			R	E	K	T	O	R		U	N	I	K	O	M							
4				R	E	K	T	O	R		U	N	I	K	O	M						
5					R	E	K	T	O	R		U	N	I	K	O	M					
6						R	E	K	T	O	R		U	N	I	K	O	M				
7							R	E	K	T	O	R		U	N	I	K	O	M			
8								R	E	K	T	O	R		U	N	I	K	O	M		
9									R	E	K	T	O	R		U	N	I	K	O	M	
10										R	E	K	T	O	R		U	N	I	K	O	M
11											R	E	K	T	O	R		U	N	I	K	O	M

1. Pada pencarian ditemukan *pattern*”REKTOR UNIKOM” pada teks sehingga simpan *pattern* pada variabel thisKey.

   thisKey ← “REKTOR UNIKOM”.

2. Kemudian bandingkan panjang karakter yang dimiliki variabel bestKey dengan variabel thisKey (nilai bestKey yang ada adalah ”REKTOR”).

3. Karena panjang karakter thisKey lebih besar dibandingkan panjang karakter bestKey, maka isi variabel bestKey dengan thisKey.

4. Sehingga bestKey ← “REKTOR UNIKOM”.

c. Pencarian Untuk Pattern Ketiga Yaitu ”SINGKATAN UNIKOM”

Diketahui :

Panjang karakter ”SINGKATAN UNIKOM” adalah 13.

Banyak perulangan adalah (panjang karakter teks - panjang karakter *pattern*) + 1 sehingga (23 - 16) + 1 = 8

Tabel 3.6 Tabel Contoh Implementasi Algoritma Brute Force Pencarian *Pattern* 3

	TEKS																					
	S	I	A	P	A		R	E	K	T	O	R		U	N	I	K	O	M		Y	A	H
1	S	I	N	G	K	A	T	A	N		U	N	I	K	O	M							
2		S	I	N	G	K	A	T	A	N		U	N	I	K	O	M						
3			S	I	N	G	K	A	T	A	N		U	N	I	K	O	M					
4				S	I	N	G	K	A	T	A	N		U	N	I	K	O	M				
5					S	I	N	G	K	A	T	A	N		U	N	I	K	O	M			
6						S	I	N	G	K	A	T	A	N		U	N	I	K	O	M		
7							S	I	N	G	K	A	T	A	N		U	N	I	K	O	M	
8								S	I	N	G	K	A	T	A	N		U	N	I	K	O	M

Pada pencarian tidak ditemukan *pattern* ”SINGKATAN UNIKOM” pada teks. Maka pattern ”SINGKATAN UNIKOM” tidak diinputkan pada variabel thisKey. Sehingga nilai bestKey tetap ”REKTOR UNIKOM”.

bestKey ← ”REKTOR UNIKOM”.

Karena semua *pattern* sudah dibandingkan dengan teks maka nilai bestKey terakhir yang didapatkan dari proses diatas adalah *keyword* dari teks pengguna.

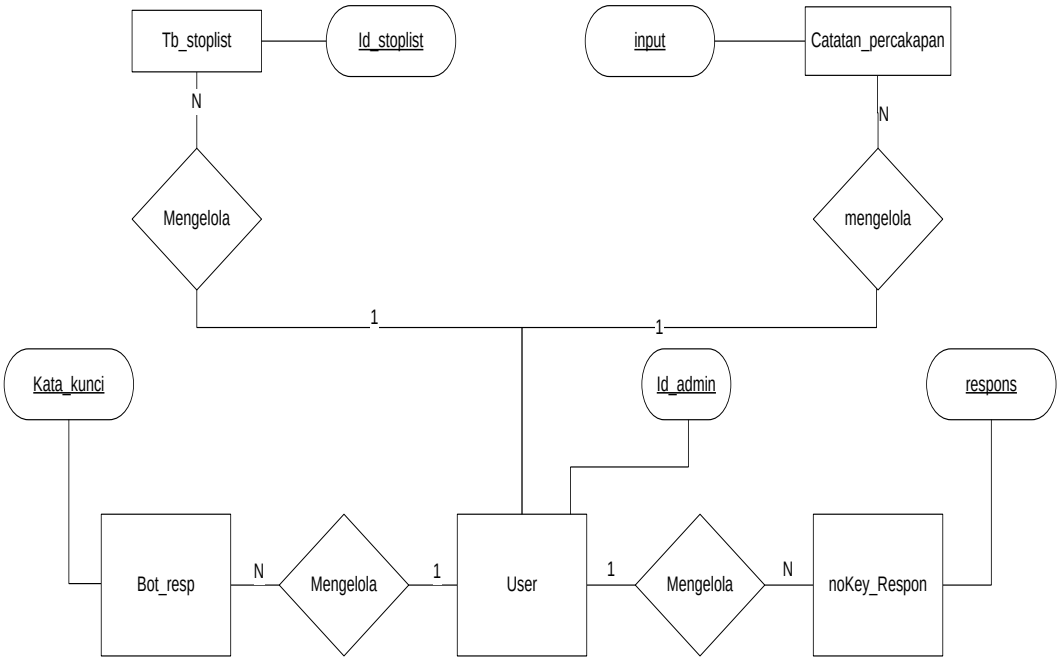
sKeyword ← bestKey

sKeyword ← ”REKTOR UNIKOM”

3.1.5 Analisis Basis Data

Analisis basis data adalah kegiatan menganalisis data yang akan diolah dan disimpan dalam database. Dalam analisis ini direpresentasikan dari mana data berasal dan atribut dari data tersebut.

Basis data merupakan kumpulan data yang saling berkaitan satu dengan lainnya yang direalisasikan dengan *relation key* yang digambarkan dalam *entity relationship diagram* (ERD). Adapun ERD yang terbentuk adalah sebagai berikut



Gambar 3.7 ERD Aplikasi *Natural Language Processing Chat Bot* Berbasis Web

3.1.6 Analisis Kebutuhan Nonfungsional

Kebutuhan non fungsional adalah usulan yang direkomendasikan kepada pengguna agar perangkat lunak yang akan dibangun menjadi *user friendly* dan perangkat keras yang mendukung secara maksimal terhadap kinerja perangkat lunak.

3.1.6.1 Analisis Perangkat Keras

Perangkat keras adalah seluruh komponen atau unsur peralatan yang digunakan untuk menunjang pembangunan aplikasi. Adapun perangkat keras yang digunakan untuk pemakaian aplikasi ini secara optimal memerlukan spesifikasi minimum komputer sebagai berikut :

1. Processor dengan kecepatan minimal 1 GHz+ Motherboard
2. Kapasitas Hardisk minimal 40 GB
3. VGACard minimal 64 MB
4. RAM minimal 256 MB
5. Monitor
6. Lan Card 10/100 Mbps
7. Koneksi internet minimal 64 Kbps
8. Mouse
9. Keybord

3.1.6.2 Analisis Perangkat Lunak

Spesifikasi perangkat lunak yang dibutuhkan agar aplikasi dapat berjalan adalah sebagai berikut :

1. WAMP Server sebagai *Web Server*
2. Adobe Dreamweaver sebagai editor PHP
3. MySQL sebagai DBMS

4. Tool antarmuka design *Microsoft Visio 2007*
5. Google Chrome sebagai *browser*

3.1.6.3 Analisis Pengguna

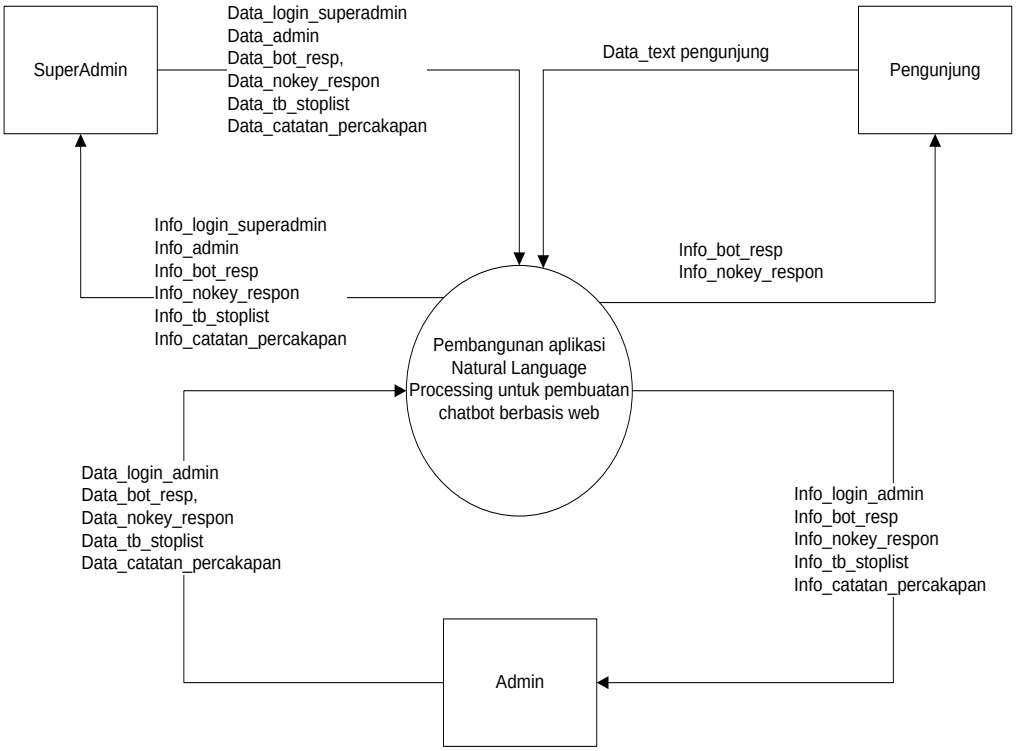
Analisis pengguna aplikasi ini memiliki karakteristik antara lain :

1. Admin
   a. Memiliki hak akses terhadap keseluruhan aplikasi
   b. Terbiasa menggunakan komputer
   c. Menguasai bahasa pemograman PHP dan Javascript
2. Masyarakat umum
   a. User hanya menginputkan teks untuk berinteraksi dengan *chat bot*

3.1.7 Analisis Kebutuhan Fungsional

3.1.7.1 Diagram Konteks

Diagram konteks adalah diagram yang menggambarkan masukan, proses dan keluaran secara umum yang terjadi pada sistem. Diagram konteks untuk aplikasi *Natural Language Processing* pada *chat bot* berbasis web dapat digambarkan sebagai berikut :

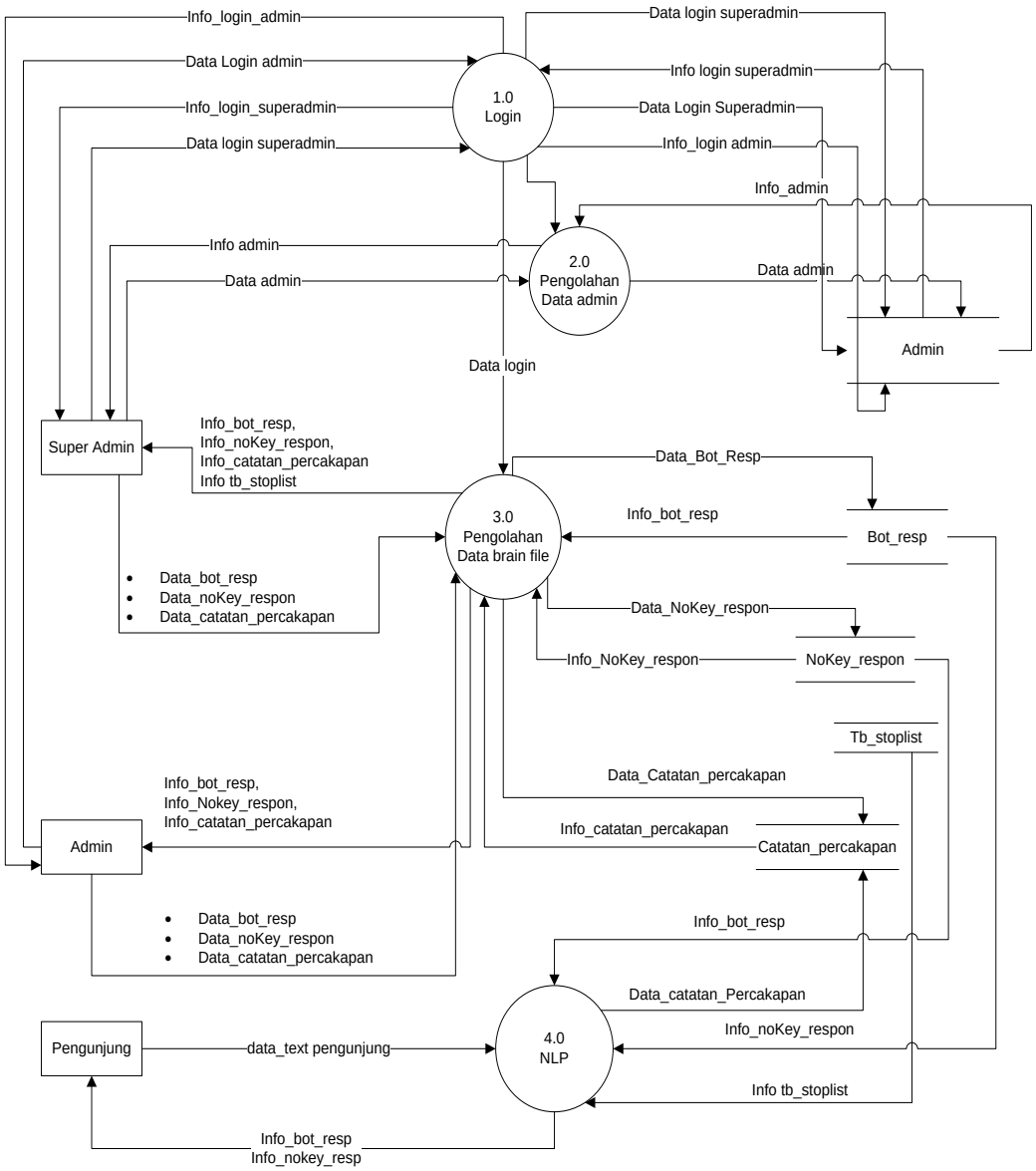


Gambar 3.8 Diagram Konteks

3.1.7.2 DFD (*Data Flow Diagram*)

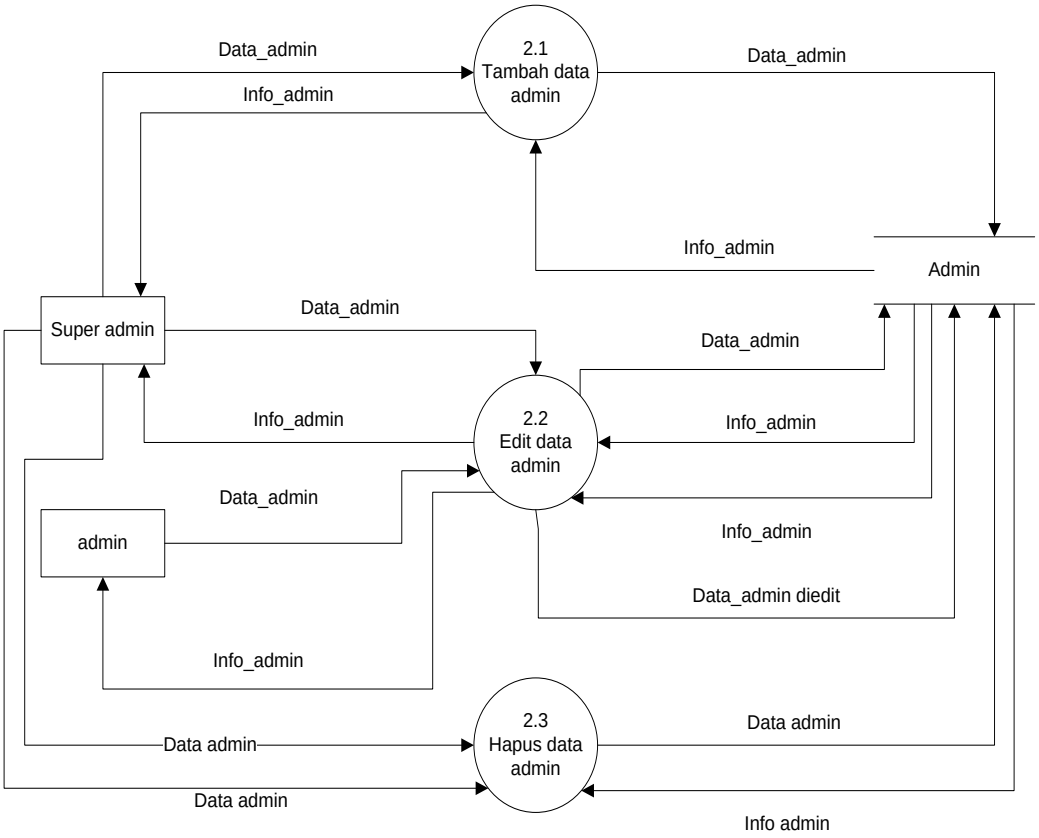
Data Flow Diagram merupakan suatu media yang digunakana untuk menggambarkan aliran data yang mengalir pada suatu sistem. DFD aplikasi *natural language processing* untuk pembuatan *chat bot* yang diusulkan terdiri dari beberapa bagian, berikut *data flow diagram* selengkapnya :

1. DFD Level 0



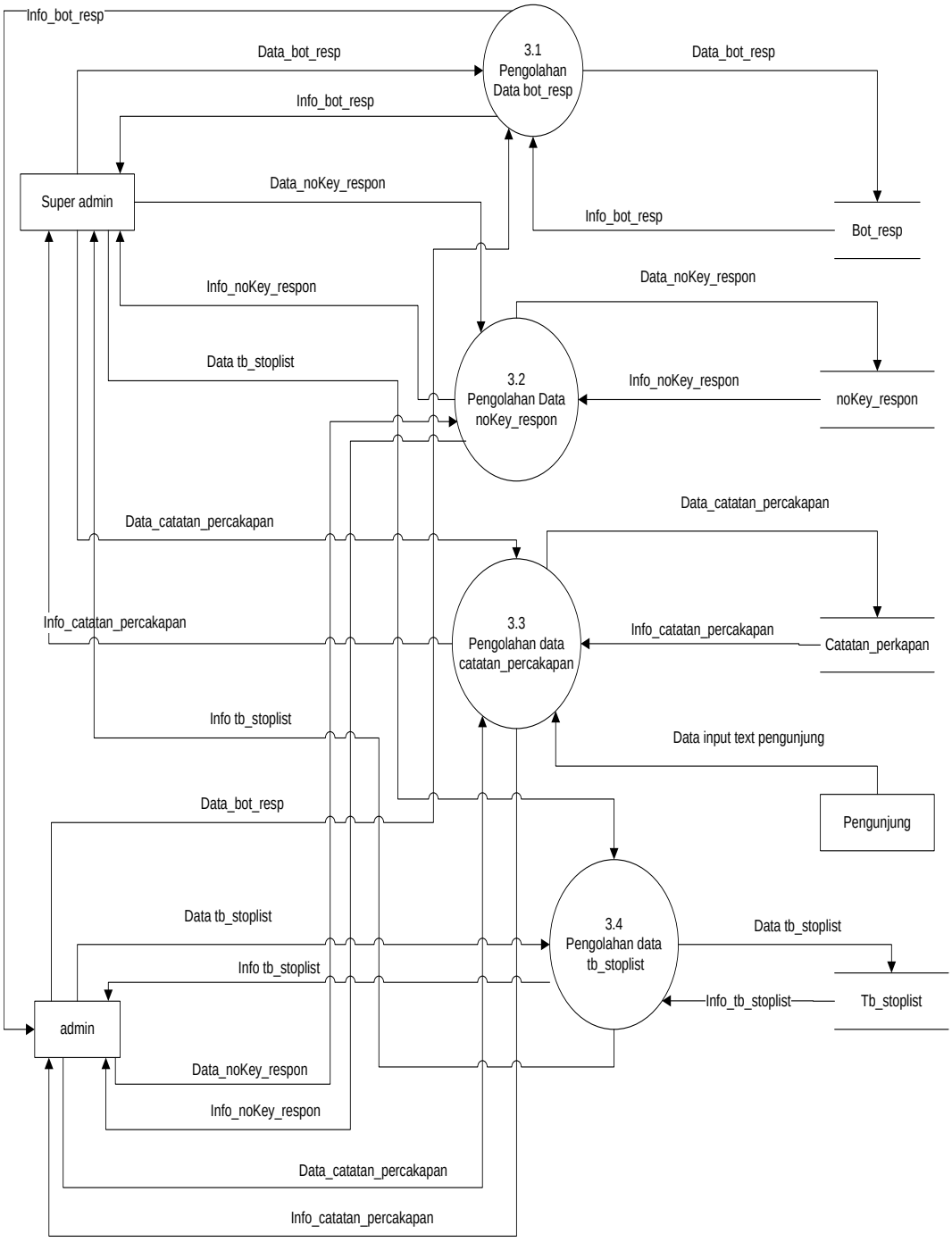
Gambar 3.9 DFD Level 0

2. DFD Level 1 Proses 1 (Pengolahan Data Admin)



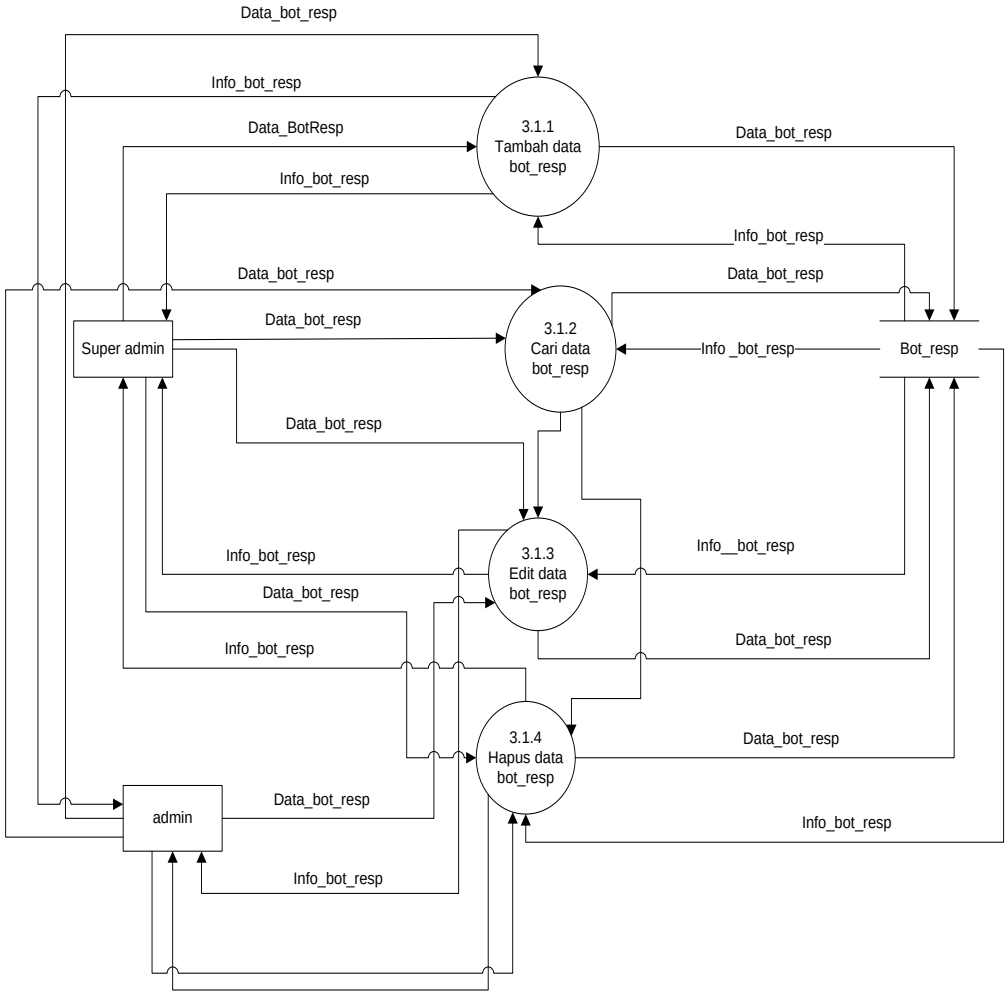
Gambar 3.10 DFD Level 1 Pengolahan Data Admin

3. DFD Level 1 Proses 3 (Pengolahan Data *Brainfile*)



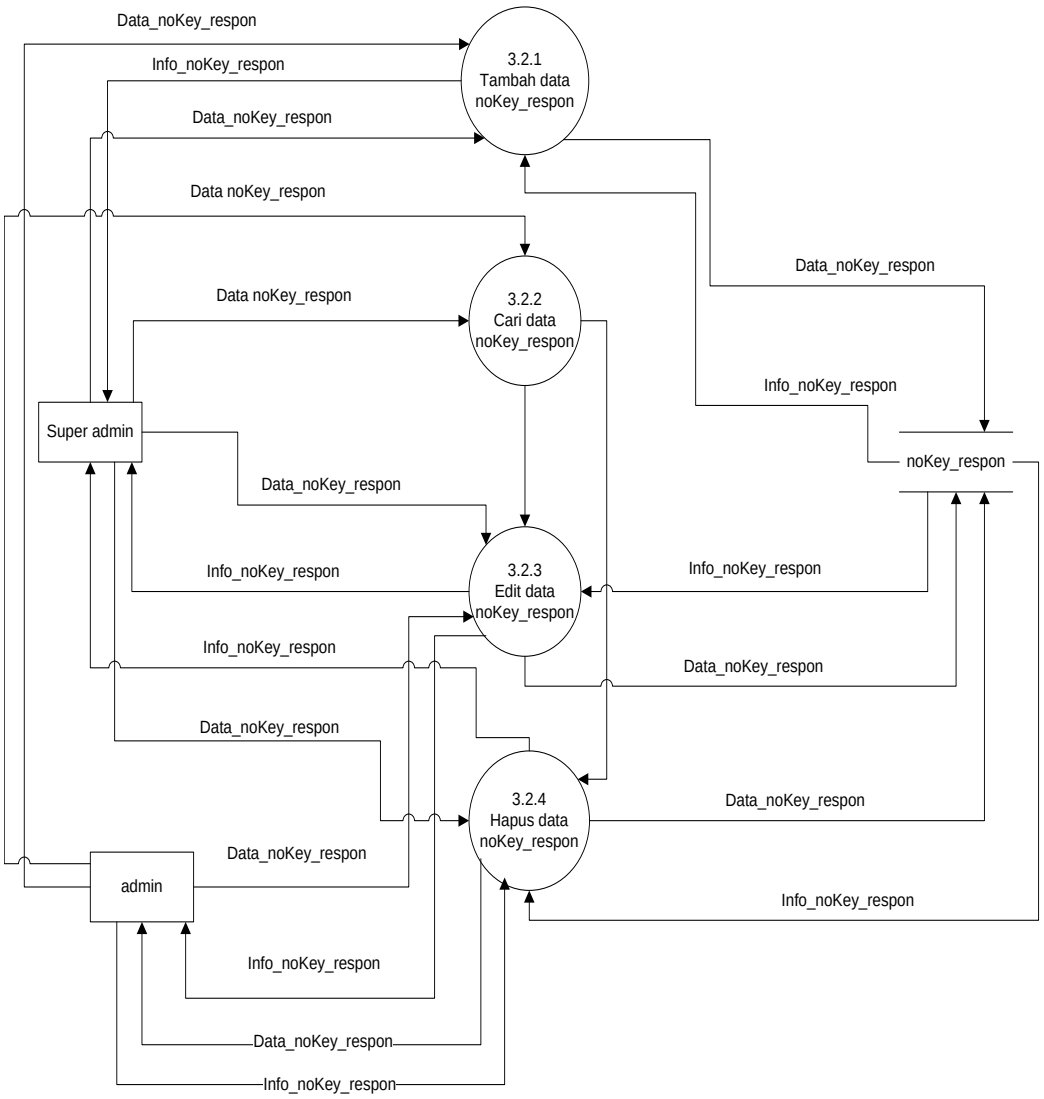
Gambar 3.11 DFD Level 1 Proses 3 (Pengolahan Data *Brain File*)

4. DFD Level 2 Proses 3.1 (Proses Pengolahan Data Bot_resp)



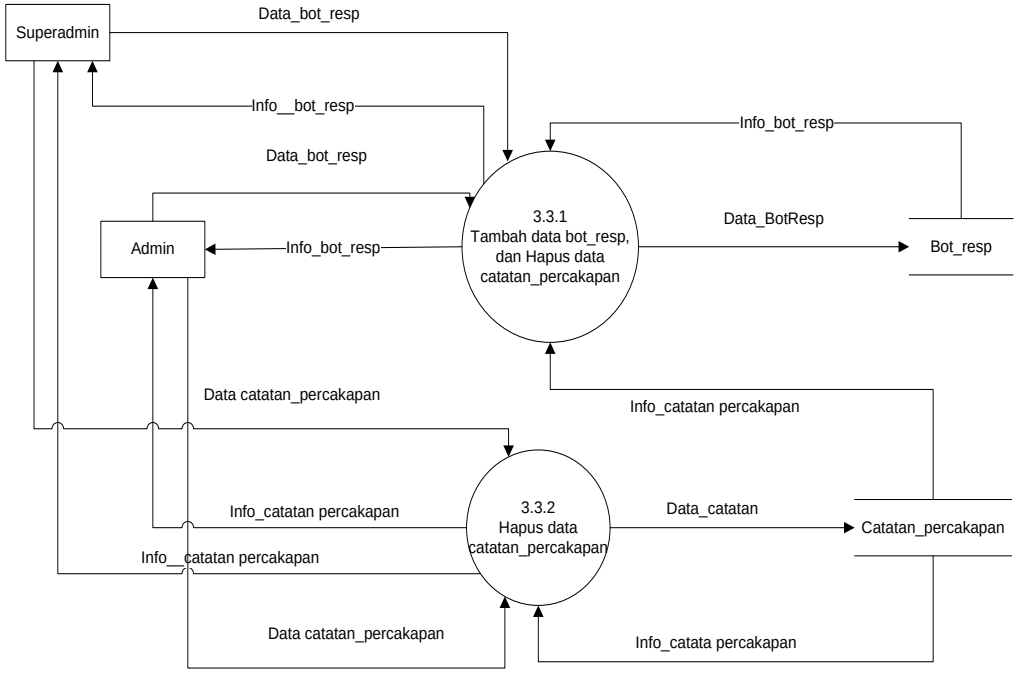
Gambar 3.12 DFD Level 2 Proses 3.1 (Proses Pengolahan Data Bot_resp)

5. DFD Level 2 Proses 3.2 (Proses Pengolahan Data NoKey_Respon)



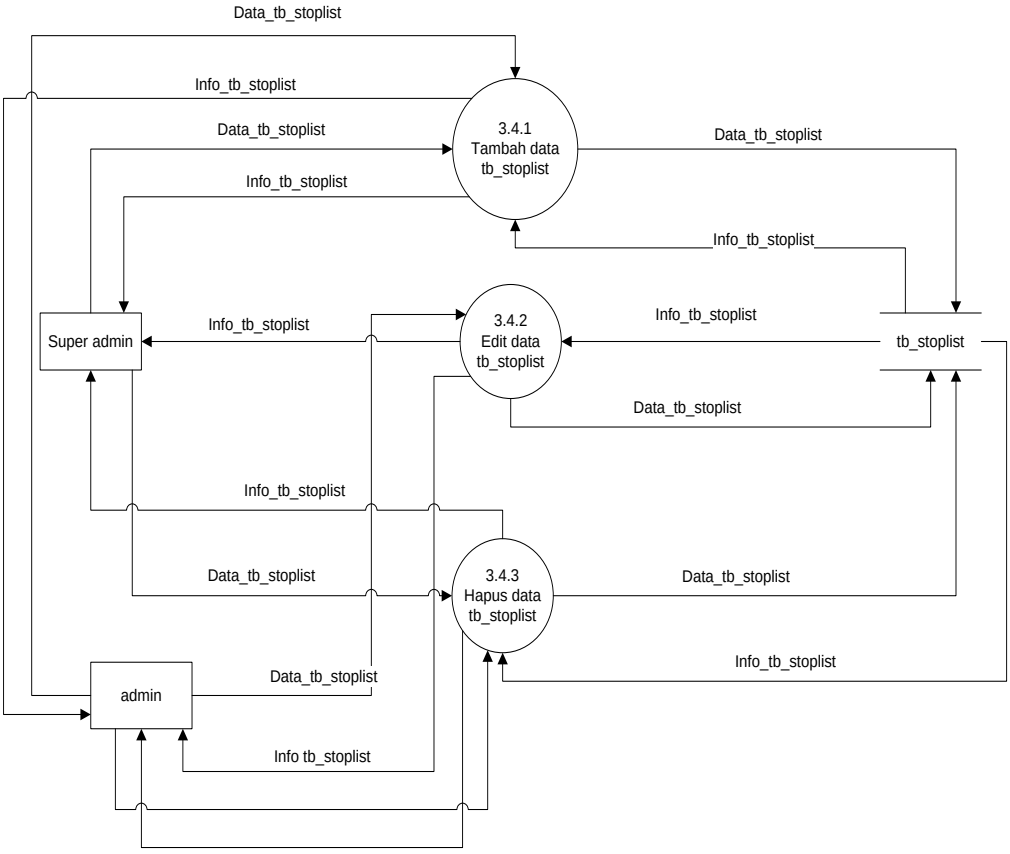
Gambar 3.13 DFD Level 2 Proses Pengolahan Data NoKey_Respon

6. DFD Level 2 Proses 3.3 (Pengolahan Data Catatan Percakapan)



Gambar 3.14 DFD Level 2 Proses 3.4 (Proses *Learning*)

7. DFD Level 2 Proses 3.4 (Proses Pengolahan Data Tb_stoplist)



Gambar 3.15 DFD Level 2 (Proses Pengolahan Data Tb_stoplist)

3.1.8 Spesifikasi Proses

3.7 Tabel Spesifikasi Proses

No.	Proses	Keterangan
1.	No.Proses	1.0
	Nama Proses	Login
	Source (sumber)	Superadmin, Admin
	Input	Data login superadmin, data login admin
	Output	Info login superadmin, info admin

	Destination (tujuan)	Superadmin, Admin
	Logika Proses	1. sistem menampilkan form login 2. superadmin atau admin mengisi data login pada form login 3. apabila data login valid sistem akan menampilkan halaman akun superadmin untuk superadmin sedangkan untuk admin sistem akan menampilkan halaman akun admin 4. apabila data login tidak valid, sistem akan menampilkan pesan kesalahan dan meminta superadmin atau admin untuk memasukan kembali data login pada form login
2.	No. Proses	2.0
	Nama Proses	Pengolahan Data Admin
	Source	Superadmin
	Input	Data admin
	output	Info Admin
	Destination	Superadmin
	Logika Proses	1. Superadmin memilih menu pengolahan data admin yang terdiri dari : a. Tambah b. Edit c. Hapus 2. superadmin memilih menu 3. jika superadmin memilih tambah maka ke proses 2.1 4. jika superadmin memilih edit maka ke proses 2.2 5. jika superadmin memilih hapus maka ke proses 2.3

3.	No. Proses	3.0
	Nama Proses	Pengolahan Data Brainfile
	Source	Superadmin, Admin
	Input	- Data Bot_resp - Data NoKey_respon - Data catatan_percakapan - Data tb_stoplist
	Ouput	- Info Bot_resp - Info NoKey_Respon - Info catatan_percakapan - Info tb_stoplist
	Destination	Superadmin, admin
	Logika Proses	1. User memilih menu data brainfile yang terdiri dari a. Pengolahan data bot_resp b. Pengolahan data nokey_resp c. Pengolahan data catatan_percakapan d. Pengolahan data tb_stoplist
4.	No. Proses	4.0
	Nama Proses	NLP
	Source	Pengujung web
	Input	Data text pengunjung
	Output	info Bot_resp, info nokey_resp
	Destination	catatan_percakapan
	Logika proses	1. Sistem menampilkan halaman depan pengunjung 2. Pengunjung menginputkan data text di

		bagian form untuk chatting 3. Jika text yang diinputkan memilki *pattern* maka tampilkan info bot_resp 4. Jika text yang diinputkan tidak memiliki *pattern* tampilkan info nokey_resp
5.	No. Proses	2.1
	Nama Proses	Tambah Data Admin
	Source	Superadmin
	Input	Data Admin
	Output	Informasi admin yang telah ditambah
	Destination	Superadmin
	Logika Proses	1. Sistem menampilkan form tambah admin 2. Superadmin mengisi data admin baru 3. Superadmin mengklik tombol simpan 4. Validasi data admin 5. Jika data admin yang dimasukan valid maka tampilkan pesan data sudah tersimpan 6. Jika data admin yang dimasukan tidak valid maka tampilkan pesan galat
6.	No. Proses	2.2
	Nama Proses	Edit Data Admin
	Source	Superadmin
	Input	Data admin
	Output	Informasi admin yang telah diedit
	Destination	Superadmin

	Logika Proses	1. Anggota sudah login sebagai admin atau superadmin 2. Jika login sebagai superadmin maka data profile semua dapat diubah 3. Jika login sebagai admin maka data yang diubah hanya data profile admin yang login saja. 4. Anggota memilih menu edit profile 5. Apabila data tidak kosong maka data baru akan disimpan didalam database 6. Apabila data ada yang kosong maka akan ada informasi data tidak boleh kosong dan data yang kosong harus diisi 7. Sistem menampilkan info profile anggota yang telah diubah
7.	No. Proses	3.1
	Nama Proses	Pengolahan Data Bot_resp
	Source	Superadmin, Admin
	Input	Data Bot_resp
	Output	Info Bot_resp
	Destination	Superadmin, admin
	Logika Proses	1. Superadmin atau admin memilih menu pengolahan data bot_resp yang terdiri dari : a. Tambah b. Cari c. Edit d. Hapus 2. Jika superadmin atau admin memilih tambah maka ke proses 3.1.1 3. Jika superadmin atau admin memilih cari maka ke proses 3.1.2 4. Jika superadmin atau admin memilih ubah maka ke proses 3.1.3

		5. Jika superadmin atau admin memilih hapus maka ke proses 3.1.4
8.	No. Proses	3.2
	Nama Proses	Pengolahan Data Nokey_respon
	Source	Superadmin, admin
	Input	Data Nokey_respon yang akan ditambah, diedit, dan dihapus
	Output	Informasi Nokey_respon yang telah ditambah, diedit, dan dihapus
	Destination	Superadmin, Admin
	Logika Proses	1. superadmin atau admin memilih menu pengolahan nokey_respon yang terdiri dari : a. Tambah b. Cari c. Edit d. Hapus 2. Jika superadmin atau admin menekan tambah maka ke proses 3.2.1 3. Jika superadmin atau admin menekan cari maka ke proses 3.2.2 4. Jika superadmin atau admin menekan edit maka ke proses 3.2.3 5. Jika superadmin atau admin menekan hapus maka ke proses 3.2.4
	No. Proses	3.3
	Nama Proses	Pengolahan data Catatan_percakapan

9.	Source	Pengunjung, Superadmin, Admin
	Input	1. Data *text* pengunjung web 2. Data catatan_percakapan yang akan diedit dan dihapus di proses *learning*
	Output	1. Informasi catatan_percakapan yang sudah ditambah oleh pengunjung 2. Informasi yang sudah diedit dan dihapus oleh admin
	Destination	Superadmin, Admin
	Logika Proses	1. sistem menampilkan halaman utama pengunjung 2. Pengunjung menginputkan text di dalam form 3. jika input text yang dimasukan pengunjung tidak mempunyai *pattern* kemudian tambah input text pengunjung ke catatan_percakapan 4. Data catatan_percakapan yang ada, diedit untuk dimasukan kedalam Bot_resp, kemudian data yang diedit akan terhapus setelah masuk Bot_resp.
10.	No. Proses	3.1.1
	Nama Proses	Tambah Data Bot_resp
	Source	Superadmin, Admin
	Input	Data Bot_resp
	Output	Info Bot_resp
	Destination	Superadmin, Admin
	Logika Proses	1. sistem menampilkan form isi data bot_resp baru. 2. Superadmin atau admin mengisi data

		bot_resp . 3. Superadmin atau admin mengklik tombol simpan. 4. Validasi data yang diisi user. 5. Jika data yang dimasukan valid simpan data 6. Jika data yang dimasukan tidak valid tampilkan alert data yang dimasukan tidak valid.
12.	No. Proses	3.1.2
	Nama Proses	Cari Data Bot_resp
	Source	Superadmin, Admin
	Input	Data Bot_resp
	Output	Info Bot_resp
	Destination	Superadmin, Admin
	Logika Proses	1. sistem menampilkan halaman bot_resp. 2. Superadmin atau admin mengisi form untuk kata yang dicari. 3. Superadmin atau admin mencari berdasarkan *pattern* atau respons. 4. Superadmin atau admin mencari berdasarkan topic yang dipilih. 5. Tekan tombol cari. 6. Sistem akan memberikan info bot_resp yang dicari.
	No. Proses	3.1.3
	Nama Proses	Edit data Bot_resp
	Source	Superadmin, Admin

13.	Input	Data Bot_resp
	Output	Informasi Bot_resp yang sudah diedit
	Destination	Superadmin, Admin
	Logika Proses	1. sistem berada dihalaman bot_resp 2. Superadmin atau admin memilih data yang akan diubah 3. Klik tombol edit 4. Sistem akan menampilkan *form* edit bot_resp 5. Superadmin memasukan data bot_resp yang akan diubah 6. Sistem menyimpan data bot_resp yang sudah diubah ke *database* 7. Sistem menampilkan data bot_resp yang sudah diubah
14.	No. Proses	3.1.4
	Nama Proses	Hapus data Bot_resp
	Source	Superadmin, Admin
	Input	Data Bot_resp
	Output	Info_Bot_resp
	Destination	Superadmin, Admin
	Logika Proses	1. Sistem berada pada halaman bot_resp 2. Superadmin atau admin memilih data yang akan dihapus 3. Superadmin atau admin mengklik tombol hapus 4. Sistem menampilkan alert untuk menanyakan kepada superadmin atau admin untuk cek data yang akan dihapus 5. Sistem menampilkan info data bot_resp

		yang sudah dihapus
15.	No. Proses	3.2.1
	Nama Proses	Tambah Data Nokey_respon
	Source	Superadmin, Admin
	Input	Data Nokey_respon
	Output	Info_Nokey_resp
	Destination	Superadmin, Admin
	Logika Proses	1. sistem menampilkan form isi data nokey_resp baru. 2. Superadmin atau admin mengisi data nokey_resp. 3. Superadmin atau admin mengklik tombol simpan. 4. Validasi data yang diisi user. 5. Jika data yang dimasukan valid simpan data 6. Jika data yang dimasukan tidak valid tampilkan alert data yang dimasukan tidak valid.
16.	No. Proses	3.2.2
	Nama Proses	Cari Data Nokey_respon
	Source	Superadmin, Admin
	Input	Data_Nokey_resp
	Output	Info_Nokey_respon
	Destination	Superadmin, Admin
	Logika Proses	1. sistem menampilkan halaman nokey_respons.

		2. Superadmin atau admin mengisi form untuk kata yang dicari. 3. Superadmin atau admin mencari berdasarkan respons. 4. Superadmin atau admin mencari berdasarkan topic yang dipilih. 5. Tekan tombol cari. 6. Sistem akan memberikan info nokey_respons yang dicari.
17.	No. Proses	3.2.3
	Nama Proses	Edit data Nokey_respon
	Source	Superadmin, Admin
	Input	Data_Nokey_resp
	Output	Info_Nokey_resp
	Destination	Superadmin, Admin
	Logika Proses	1. sistem berada dihalaman nokey_respons 2. Superadmin atau admin memilih data yang akan diubah 3. Klik tombol edit 4. Sistem akan menampilkan *form* edit nokey_respons 5. Superadmin memasukan data nokey_respons yang akan diubah 6. Sistem menyimpan data nokey_respons yang sudah diubah ke *database* Sistem menampilkan data bot_resp yang sudah diubah
	No. Proses	3.2.4

18.	Nama Proses	Hapus data Nokey_respon
	Source	Superadmin, Admin
	Input	Data_Nokey_resp
	Output	Info_Nokey_resp
	Destination	Superadmin, Admin
	Logika Proses	1. Sistem berada pada halaman nokey_respons 2. Superadmin atau admin memilih data yang akan dihapus 3. Superadmin atau admin mengklik tombol hapus 4. Sistem menampilkan alert untuk menanyakan kepada superadmin atau admin untuk cek data yang akan dihapus 5. Sistem menampilkan info data nokey_respons yang sudah dihapus
19.	No. Proses	3.3.1
	Nama Proses	Tambah data bot_resp dan hapus data catatan_percakapan
	Source	Superadmin, Admin
	Input	Data_Bot_resp
	Output	Informasi Bot_resp yang telah ditambah dari data catatan_percakapan
	Destination	Superadmin, Admin
	Logika Proses	1.Sistem menampilkan halaman catatan_percakapan 2.Superadmin atau admin memilih data percakapan kemudian klik edit 3.Sistem menampilkan form tambah bot_resp

		dengan kata_kunci dari input catatan_percakapan 4.User mengklik tombol simpan 5.Validasi data yang sudah ditambahkan ke bot_resp 6.Jika data bot_resp valid simpan data dan hapus data catatan_percakapan yang telah ditambah ke bot_resp 7.Jika tidak valid tampilkan pesan data yang dimasukan tidak valid
20.	No. Proses	3.3.2
	Nama Proses	Hapus data catatan _percakapan
	Source	Superadmin, Admin
	Input	Data_catatan_percakapan
	Output	Informasi data catatan_percakapan yang telah dihapus
	Destination	Superadmin, Admin
	Logika Proses	1.sistem menampilkan halaman catatan_percakapan 2.Superadmin atau admin memilih data catatan_percakapan yang akan dihapus 3.Sistem menampilkan pesan apakah data yang dipilih yakin akan dihapus 4.Jika yakin dihapus tampilkan info catatan_percakapan telah dihapus 5.Jika tidak maka sistem menampilkan halaman catatan_percakapan kembali
	No. Proses	3.4.1

21.	Nama Proses	Tambah Data Tb_stoplist
	Source	Superadmin, Admin
	Input	Data_tb_stoplist
	Output	Info_tb_stoplit
	Destination	Superadmin, Admin
	Logika Proses	1. sistem menampilkan form isi data Tb_stoplist baru. 2. Superadmin atau admin mengisi data Tb_stoplist. 3. Superadmin atau admin mengklik tombol simpan. 4. Validasi data yang diisi user. 5. Jika data yang dimasukan valid simpan data 6. Jika data yang dimasukan tidak valid tampilkan alert data yang dimasukan tidak valid.
22	No. Proses	3.4.2
	Nama Proses	Edit data tb_stoplist
	Source	Superadmin, Admin
	Input	Data_tb_stoplist
	Output	Info_tb_stoplist
	Destination	Superadmin, Admin
	Logika Proses	1. sistem berada dihalaman tb_stoplist 2. Superadmin atau admin memilih data yang akan diubah 3. Klik tombol edit 4. Sistem akan menampilkan *form* edit tb_stoplist 5. Superadmin memasukan data tb_stoplist

		yang akan diubah 6. Sistem menyimpan data tb_stoplist yang sudah diubah ke *database* 7. Sistem menampilkan data tb_stoplist yang sudah diubah
23	No. Proses	3.4.3
	Nama Proses	Hapus data tb_stoplist
	Source	Superadmin, Admin
	Input	Data_tb_stoplist
	Output	Info_tb_stoplist
	Destination	Superadmin, Admin
	Logika Proses	1. Sistem berada pada halaman tb_stoplist 2. Superadmin atau admin memilih data yang akan dihapus 3. Superadmin atau admin mengklik tombol hapus 4. Sistem menampilkan alert untuk menanyakan kepada superadmin atau admin untuk cek data yang akan dihapus 5. Sistem menampilkan info data tb_stoplist yang sudah dihapus

3.1.9 Kamus Data

Kamus data merupakan deskripsi formal mengenai seluruh elemen yang tercakup dalam DFD. Kamus data untuk diagram arus data pada sistem aplikasi *Natural Language Processing* untuk pembuatan *chat bot* berbasis web adalah :

3.8 Tabel Kamus Data

Nama aliran Data	**Data Admin**
Where used/how used	Proses 1.0 Login Proses 2.0 Pengolahan data admin Proses 2.1 Tambah Admin Proses 2.2 Edit Admin
Keterangan	Berisi seluruh data admin yang digunakan sebagai untuk fungsi-fungsi yang ada dalam sistem
Struktur data	Id_user+nama_lengkap+password+level
Deskripsi	Id_user=[A-Z | a-z | 0-9] Nama_lengkap=[A-Z | a-z] Password= [A-Z | a-z | 0-9 | simbol] Level=[A-Z | a-z] Simbol= [!|@|#|$|%|^|&|*|(|)|_|+|}|{|?|>|<|.|,|;|:]
Nama aliran data	**Data Bot_resp**
Where used / how used	Proses 3.0 Pengolahan data *brainfile* Proses 3.1 Pengolahan data Bot_resp Proses 3.1.1 Tambah data Bot_resp Proses 3.1.2 Cari data Bot_resp

	Proses 3.1.3 Ubah data Bot_resp Proses 3.1.4 Hapus data Bot_resp Proses 3.3.1 Tambah Data Bot_resp
Keterangan	Data ini berisi seluruh data Bot_resp yang digunakan sebagai informasi yang ingin ditampilkan
Struktur data	kata_kunci+respons+id_topic+id_admin+nomor
Deskripsi	Kata_kunci=[A-Z | a-z | 0-9] Respons=[A-Z | a-z | 0-9] Id_topic=[0-9] Id_Admin =[A-Z | a-z | 0-9] Nomor =[0-9]
Nama aliran data	**Data Nokey_respons**
Where used / how used	Proses 3.0 Pengolahan data *brainfile* Proses 3.2 Pengolahan data Nokey_respons Proses 3.2.1 Tambah data Nokey_respons Proses 3.2.2 Cari data Nokey_respons Proses 3.2.3 Ubah data Nokey_respons Proses 3.2.4 Hapus data Nokey_respons
Keterangan	Data ini berisi seluruh data Nokey_respons yang digunakan sebagai informasi yang ingin ditampilkan
Struktur data	respons+id_topic+id_user
Deskripsi	Respons=[A-Z | a-z | 0-9] Id_topic=[0-9] id_user =[A-Z | a-z | 0-9]

Nama aliran data	**Data Catatan_percakapan**
Where used / how used	Proses 3.0 Pengolahan data *brainfile* Proses 3.3 Pengolahan data catatan_percakapan Proses 4.0 NLP Proses 3.3.1 Tambah data Bot_resp dari data catatan_percakapan Proses 3.3.2 Hapus data catatan_percakapan yang sudah diproses
Keterangan	Berisi data input dari pengunjung yang tidak diketahui *patternnya* oleh *bot program*
Struktur data	Input + tanggal + id_admin
Deskripsi	Input=[A-Z | a-z | 0-9 | simbol] Tanggal=[0-9] Id_admin = [A-Z | a-z | 0-9] Simbol=[!|@|#|$|%|^|&|*|(|)|_|+|}|{|?|>|<|.|,|;|:]
Nama aliran Data	**Data tb_stoplist**
Where used/how used	Proses 3.0 Pengolahan Data Brain file Proses 4.0 NLP Proses 3.4 Pengolahan data tb_stoplist Proses 3.4.1 Tambah data Tb_stoplist Proses 3.4.2 Edit data Tb_stoplist Proses 3.4.3 Hapus data Tb_stoplist
Keterangan	Berisi seluruh data tb_stoplist yang digunakan sebagai untuk fungsi-fungsi yang ada dalam sistem

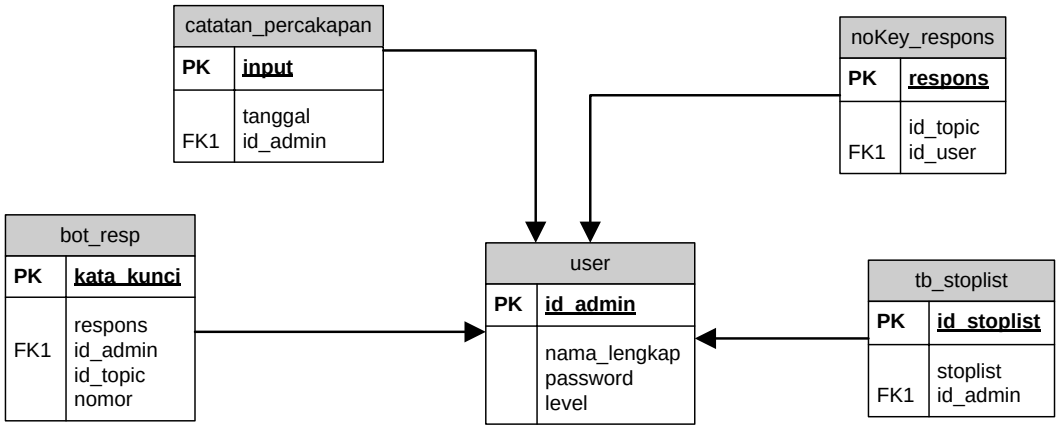
Struktur data	Id_stoplist+stoplist+id_admin
Deskripsi	Id_stoplist=[0-9] stoplist=[A-Z | a-z | 0-9] id_admin =[A-Z | a-z | 0-9]

3.1.10 Perancangan Basis Data

Perancangan basis data yaitu perancangan yang bisa menciptakan atau merancang kumpulan data yang terhubung dan tersimpan secara bersama-sama.

3.1.10.1 Skema Relasi

Skema relasi menggambarkan hubungan antar data, arti data dan batasannya dijelaskan dengan baris dan kolom.



Gambar 3.16 Skema Relasi

3.1.10.2 Perancangan Struktur Tabel

Tabel adalah sekumpulan data informasi spesifik tentang subjek tertentu yang disusun dalam bentuk kolom dan baris. Tabel adalah komponen utama dan

pertama dari sebuah database. Struktur tabel untuk setiap tabel yang terdapat dalam database aplikasi Natural Language Processing untuk pembuatan chat bot berbasis web adalah :

1. Tabel User

   Tabel admin digunakan untuk mencatat data penting yang merupakan identitas admin. Secara umum, struktur tabel yang akan digunakan untuk mencatat data user dapat dilihat pada tabel 3.5 Berikut :

Tabel 3.9 Tabel Admin

Nama Field	Tipe	Size	Keterangan
Id_admin	Varchar	50	Primary Key
Nama_lengkap	Varchar	50	
Password	Varchar	50	
Level	Varchar	30	

2. Tabel Bot_resp

   Tabel Bot_resp digunakan untuk mencatat data penting yang menyimpan pengetahuan dari chat bot. Secara umum, struktur tabel yang akandigunakan untuk mencatat data Bot_resp dapat dilihat pada tabel 3.6 :

Tabel 3.10 Tabel Bot_resp

Nama Field	Tipe	Size	Keterangan
Kata_kunci	varchar	300	Primary Key
Respons	Text		
Id_topic	Integer	30	
Id_admin	Varchar	100	
Nomor	integer	50	

3. Tabel Nokey_respons

Tabel nokey_respons digunakan untuk mencatat data penting yang menyimpan respon-respon yang tidak memakai *pattern* untuk merespon inputan pengguna dalam *chatting*. Secara umum, struktur tabel yang akan digunakan untuk mencatat data Bot_resp dapat dilihat pada tabel 3.7 :

Tabel 3.11 Nokey_respons

Nama Field	Tipe	Size	Keterangan
Respon	varchar	450	Primary Key
Id_topic	integer	10	
Id_user	Varchar	100	

4. Tabel catatan_percakapan

Tabel catatan_percakapan digunakan untuk mencatat semua percakapan dengan pengguna yang tidak ditemukan *pattern* pada proses mencocokan data. Secara umum, struktur tabel yang akan digunakan untuk mencatat data catatan_percakapan dapat dilihat pada tabel 3.8 :

Tabel 3.12 Tabel Catatan_percakapan

Nama Field	Tipe	Size	Keterangan
Input	varchar	450	Primary Key
Tanggal	Timestamp		
Id_admin	Varchar	50	

5. Tabel Tb_stoplist

Tabel tb_stoplist digunakan untuk tempat penyimpanan kata–kata yang dianggap tidak perlu, sehingga inputan pengguna yang mengandung kata yang ada dalam tabel tb_stoplist akan difilter.

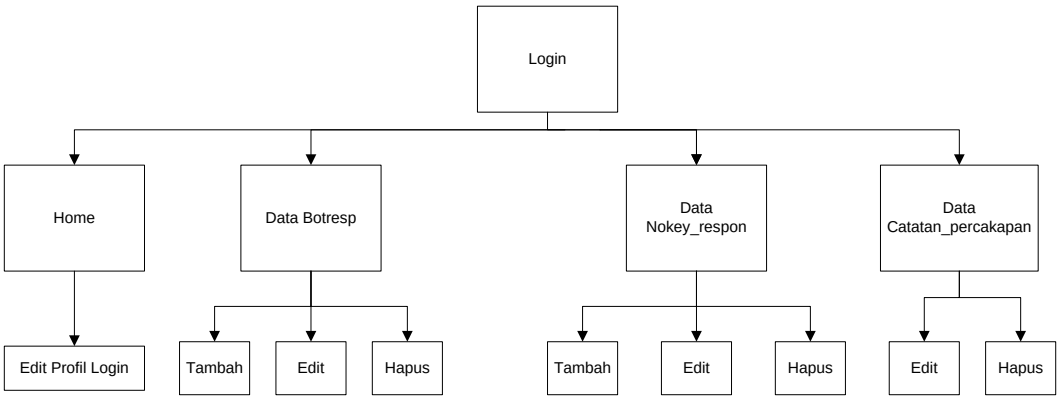
Tabel 3.13 Tabel Tb_stoplist

Nama Field	Tipe	Size	Keterangan
Id_stoplist	integer	10	Primary Key
Stoplist	Varchar	50	
Id_admin	Varchar	50	

3.1.11 Perancangan Struktur Menu

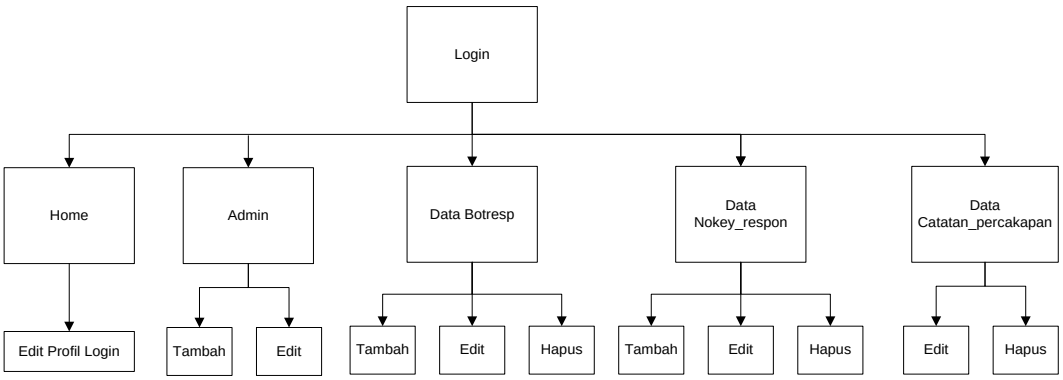
Dalam pembuatan suatu sistem informasi diperlukan suatu struktur menu yang berisikan mengenai menu dan submenu. Berikut ini struktur menu dari sistem aplikasi natural language processing untuk pembuatan chat bot berbasis web

a. Perancangan Struktur Menu Superadmin



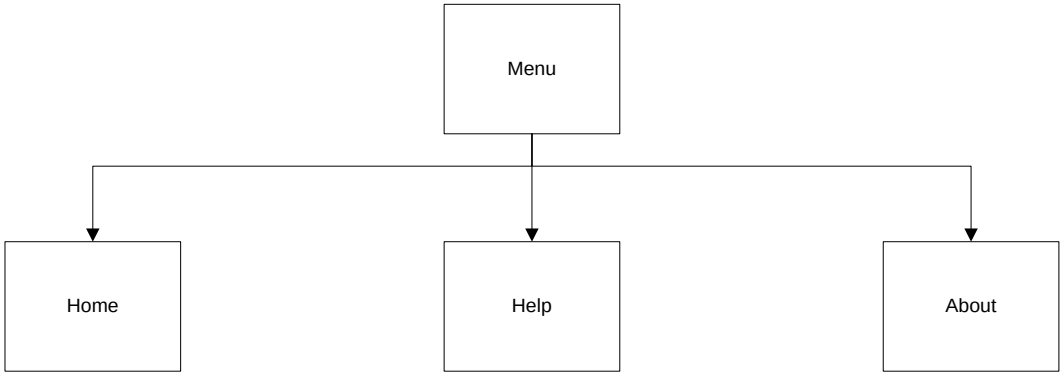
3.17 Struktur Menu Superadmin

b. perancangan Sturktur Menu Admin



Gambar 3.18 Perancangan Struktur Menu Admin

c. Perancangan Struktur Menu Pengunjung

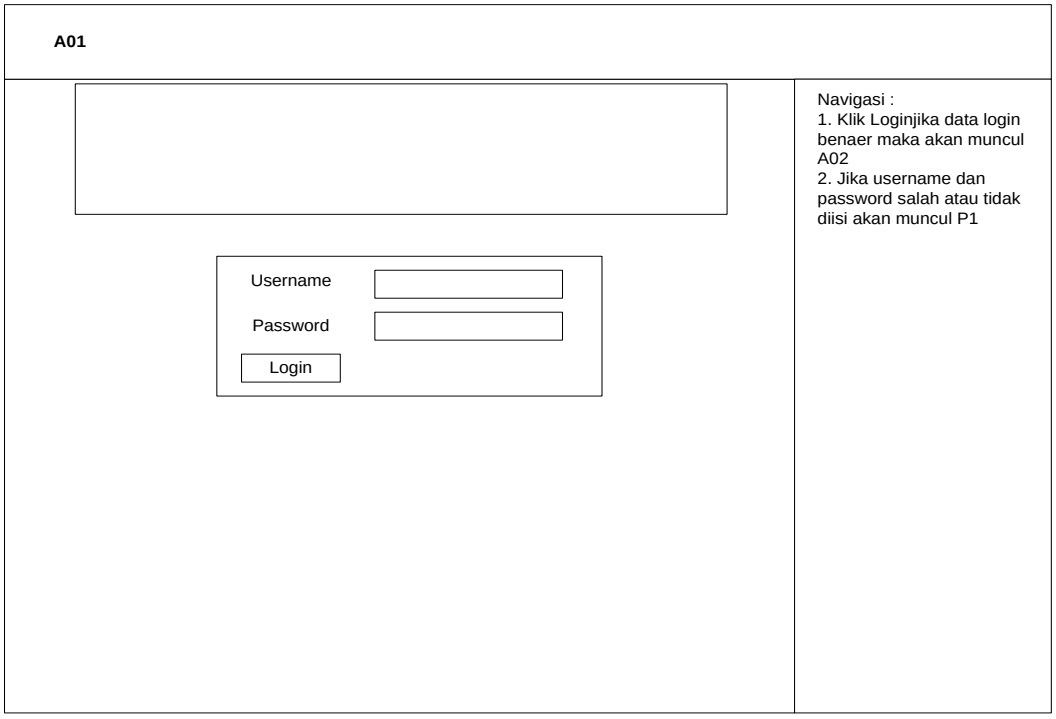


Gambar 3.19 Perancangan Struktur Menu Pengunjung

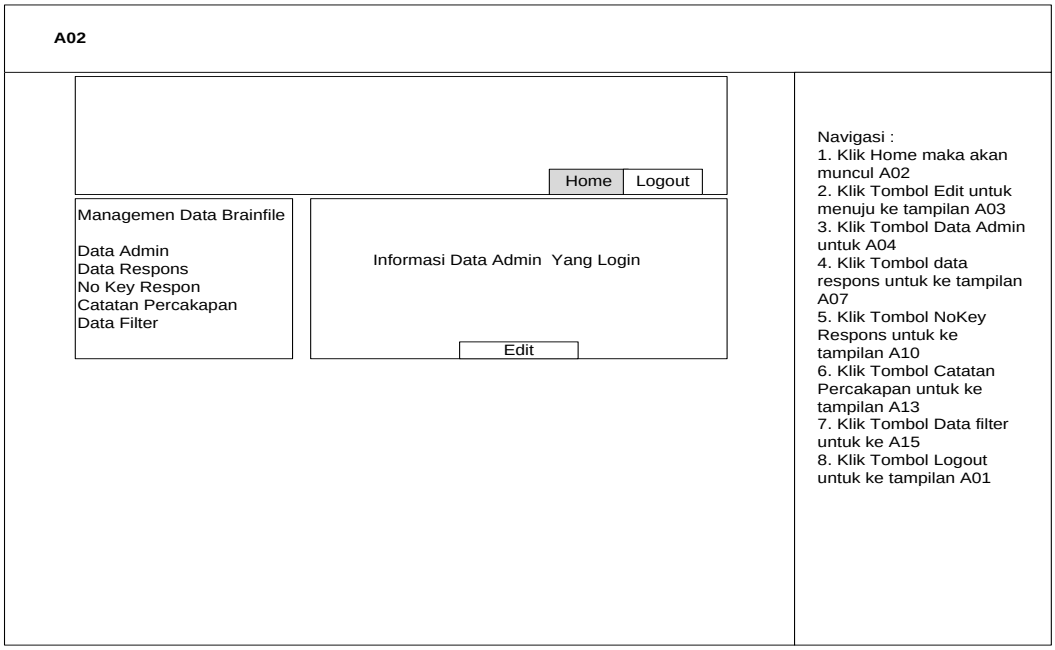
3.1.12 Perancangan Antarmuka

Perancangan antarmuka (*interface*) menggambarkan bagaimana perangkat lunak berkomunikasi dengan dirinya sendiri dengan sistem yang berinteroperasi dengannya, dan dengan manusia yang menggunakannya.

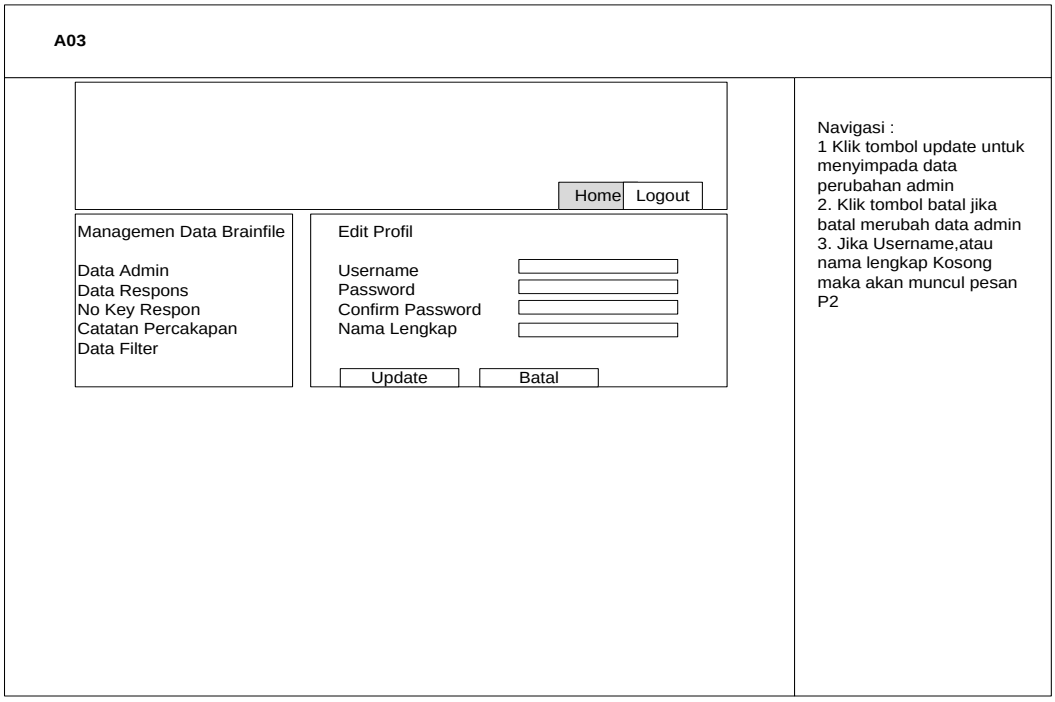
3.1.12.1 Perancangan Antarmuka Superadmin



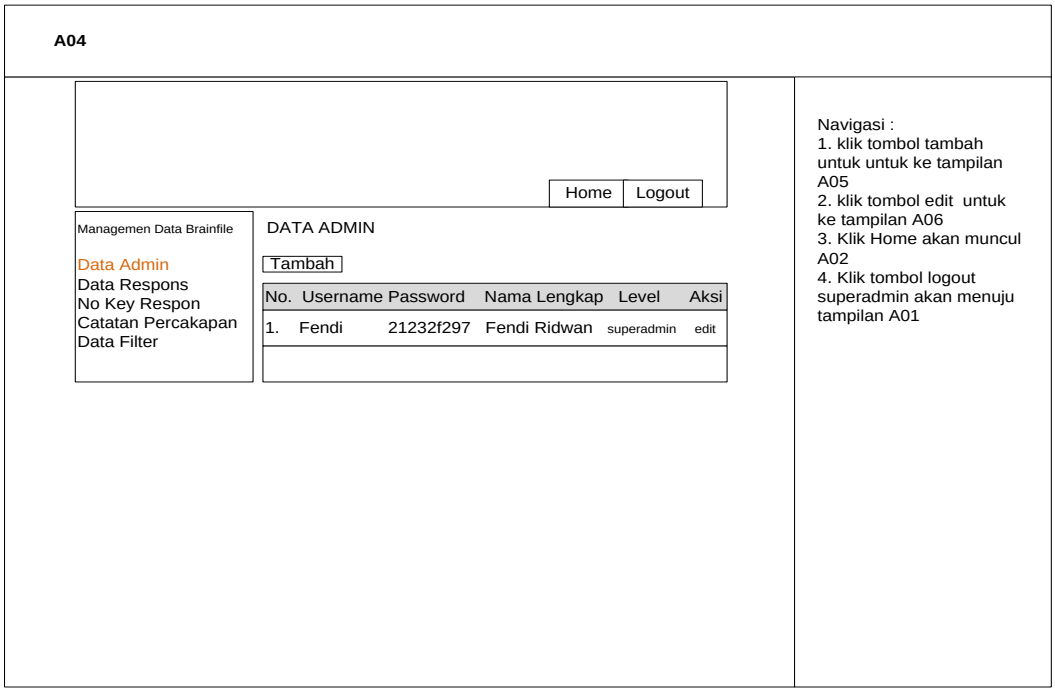
Gambar 3.20 Antarmuka Superadmin A01 Login



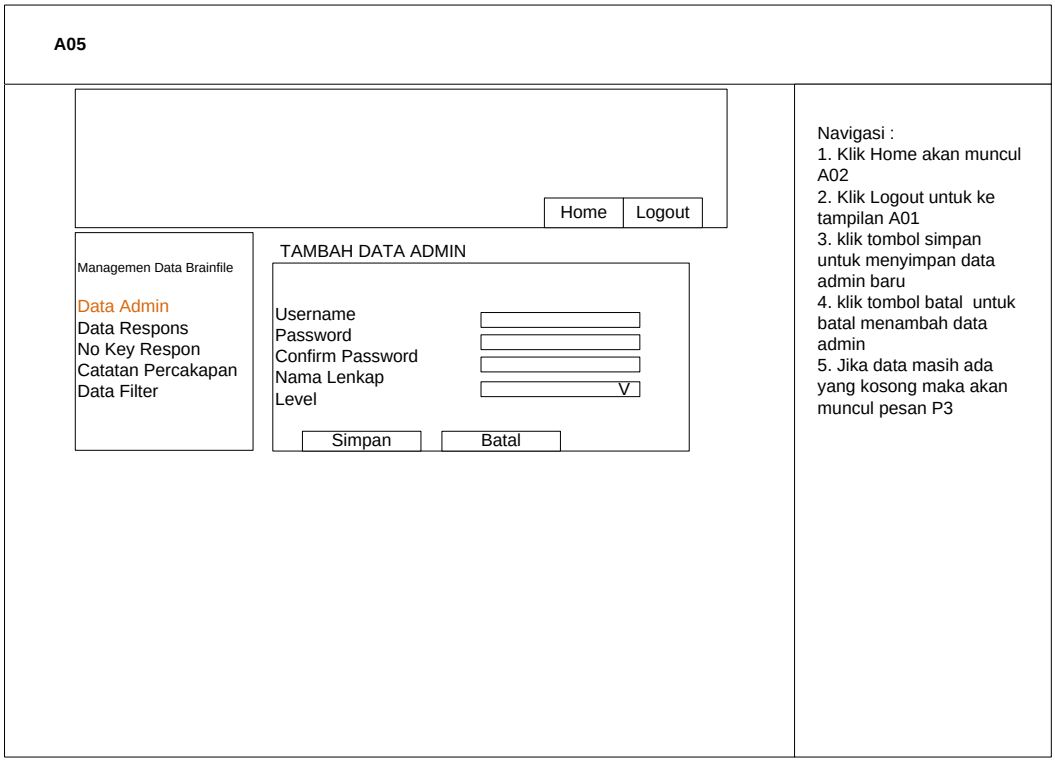
Gambar 3.21 Antarmuka Superadmin A02 Menu Utama Superadmin



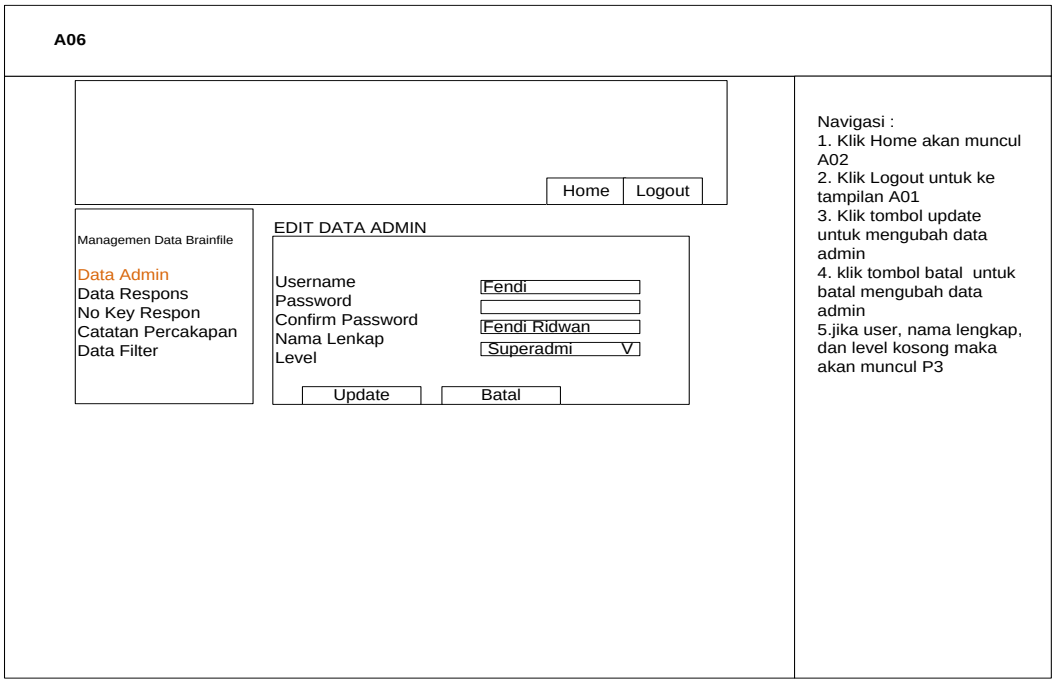
Gambar 3.22 Antarmuka Superadmin A03 Edit Profil Login



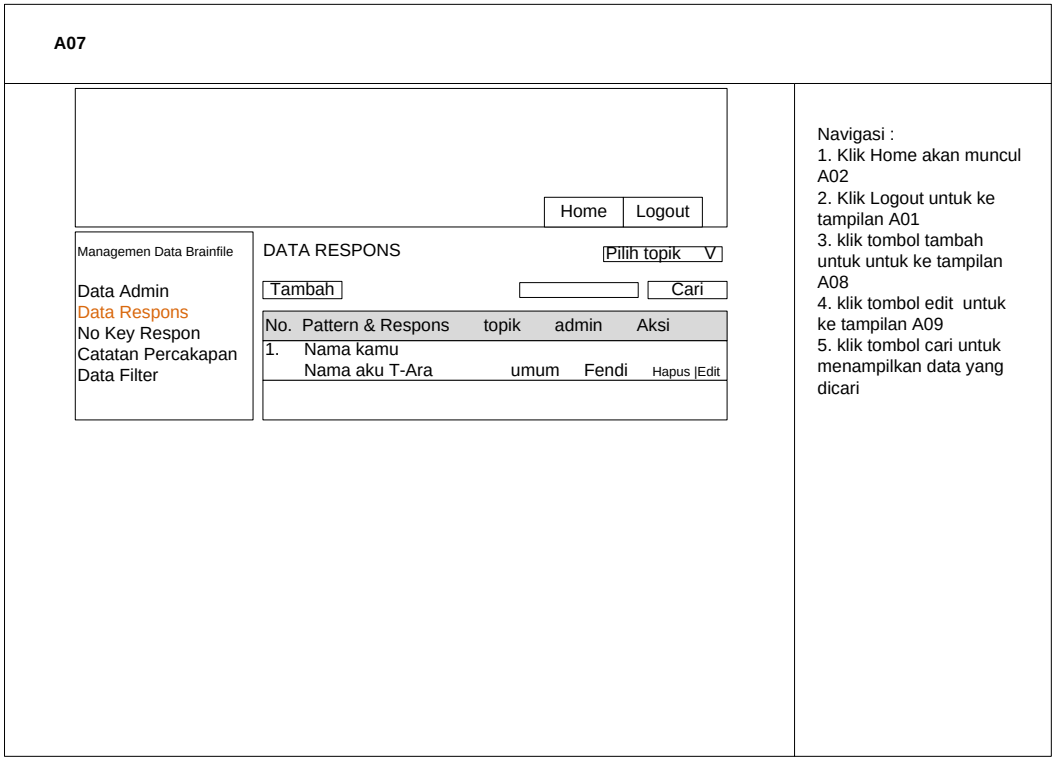
Gambar 3.23 Antarmuka Superadmin A04 Data Admin



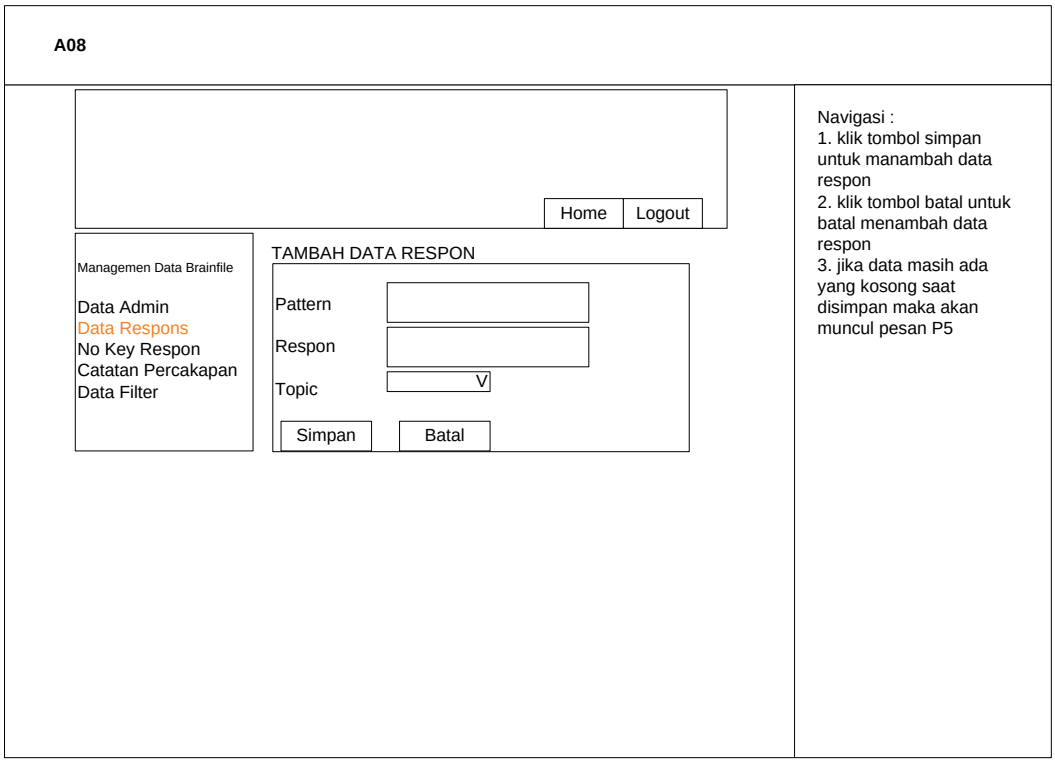
Gambar 3.24 Antarmuka Superadmin A05 Tambah Data Admin



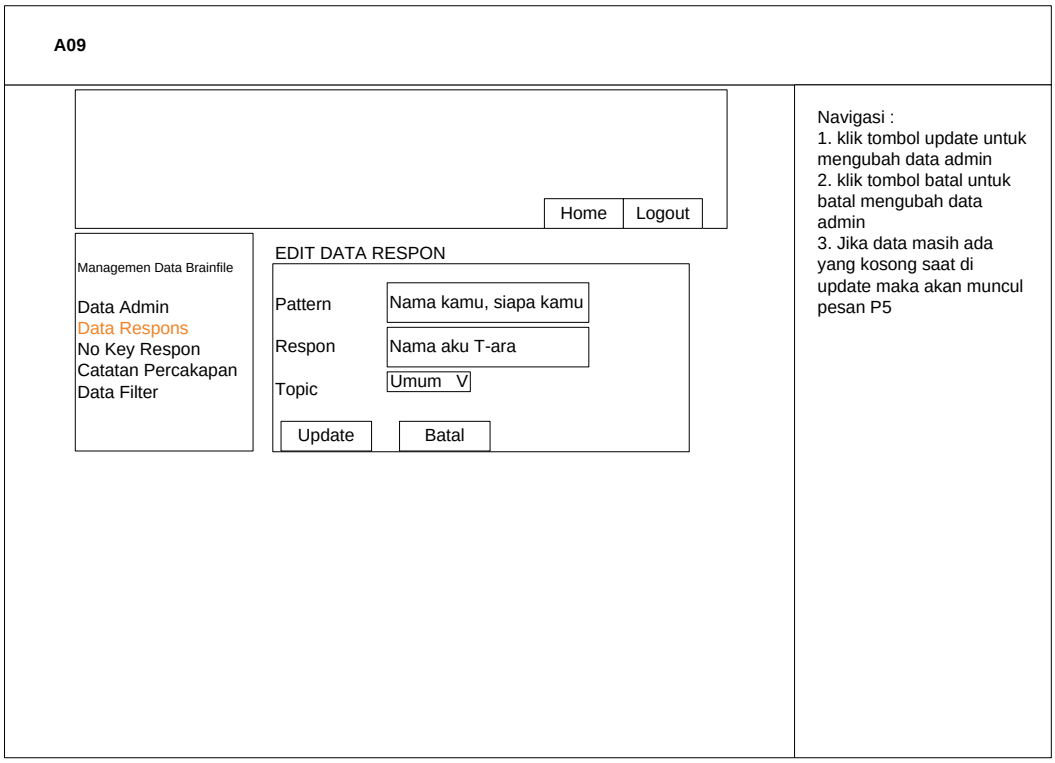
Gambar 3.25 Antarmuka Superadmin A06 Edit Data Admin



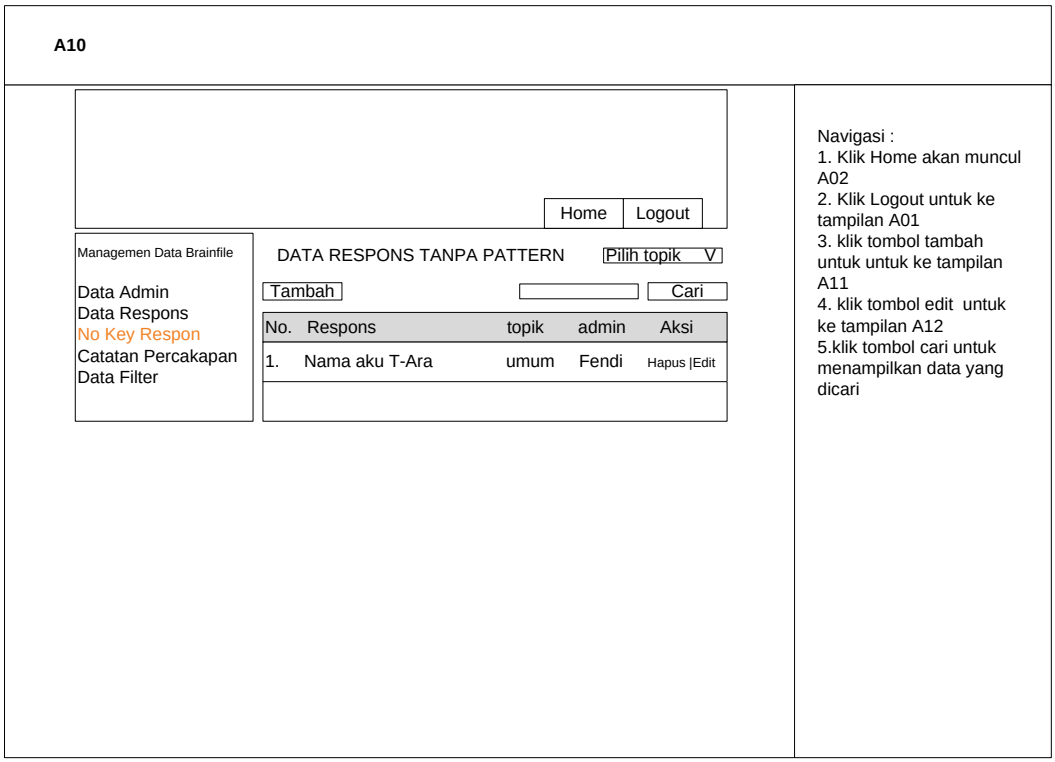
Gambar 3.26 Antarmuka Superadmin A07 Data Respons



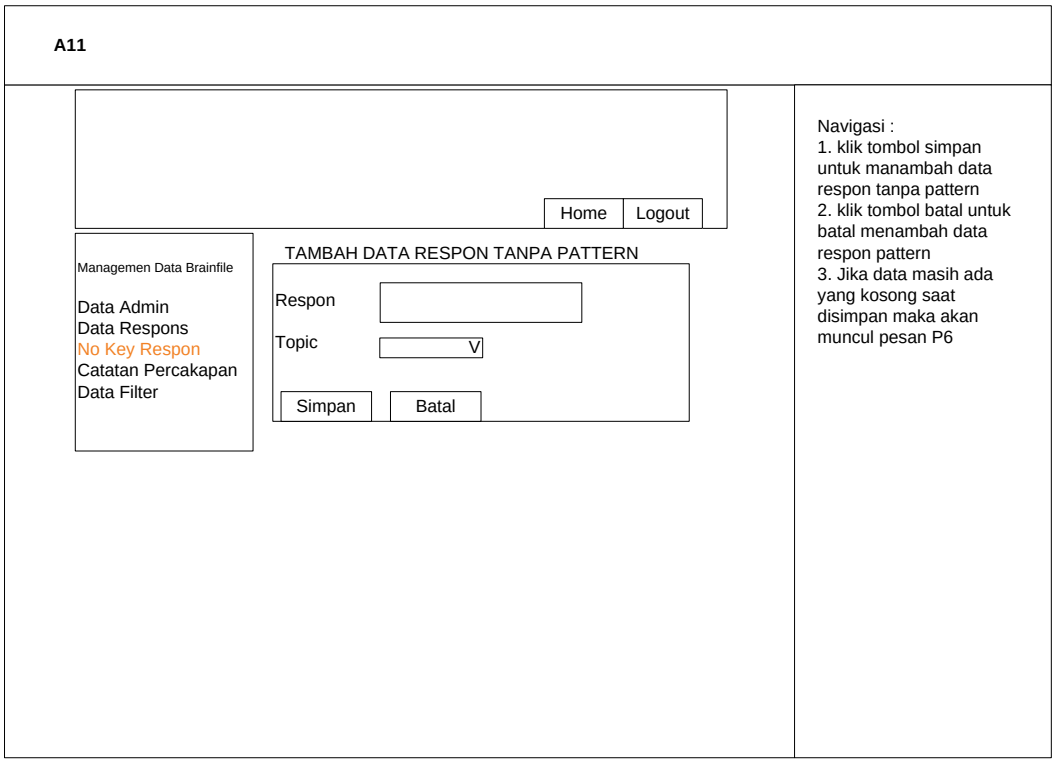
Gambar 3.27 Antarmuka Superadmin A08 Tambah Data Respons



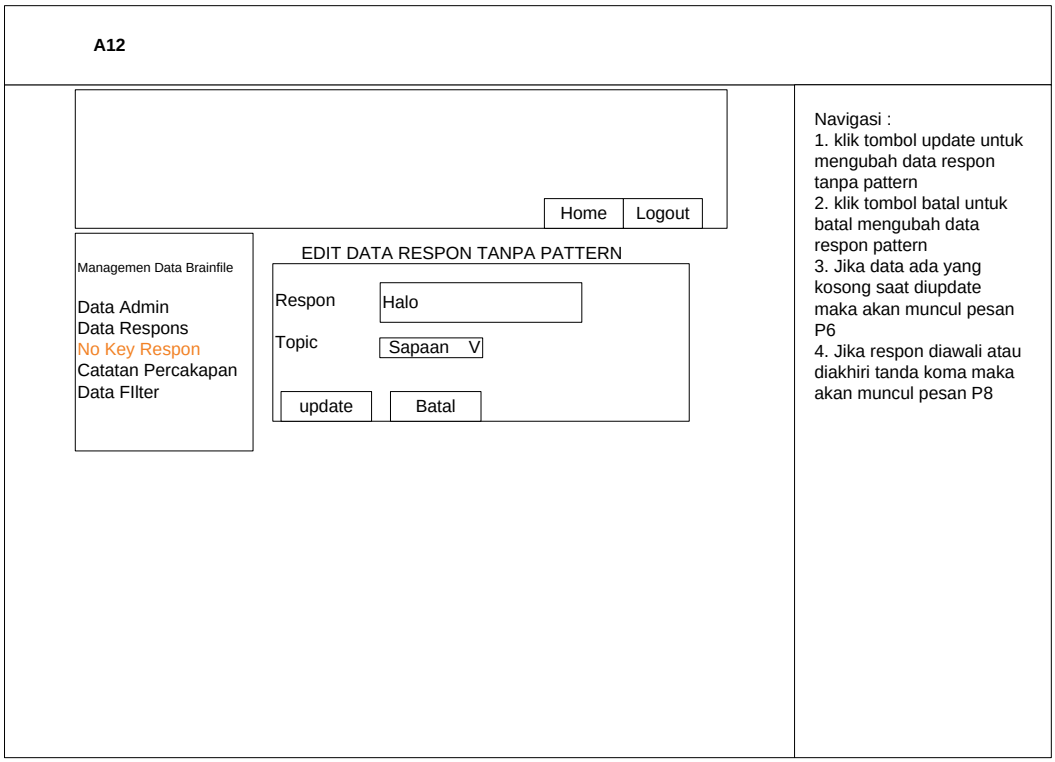
Gambar 3.28 Antarmuka Superadmin A09 Edit Data Respons



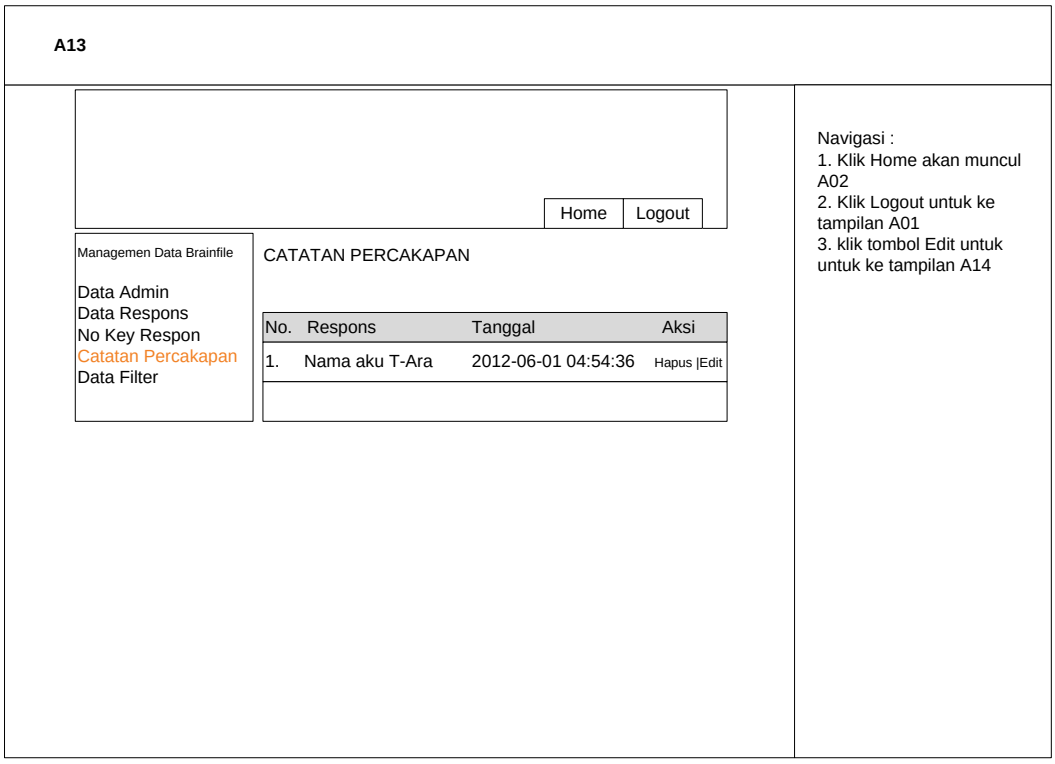
Gambar 3.29 Antarmuka Superadmin A10 Data Respon Tanpa *Pattern*



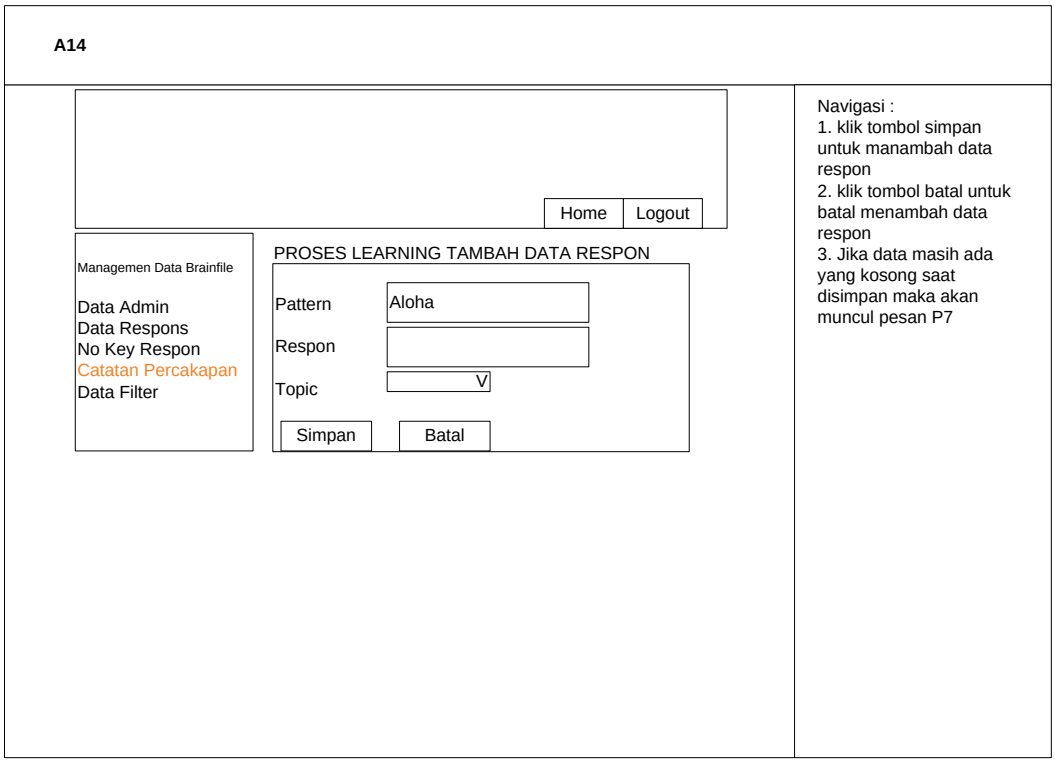
Gambar 3.30 Antarmuka Superadmin A11 Tambah Data Respon Tanpa *Pattern*



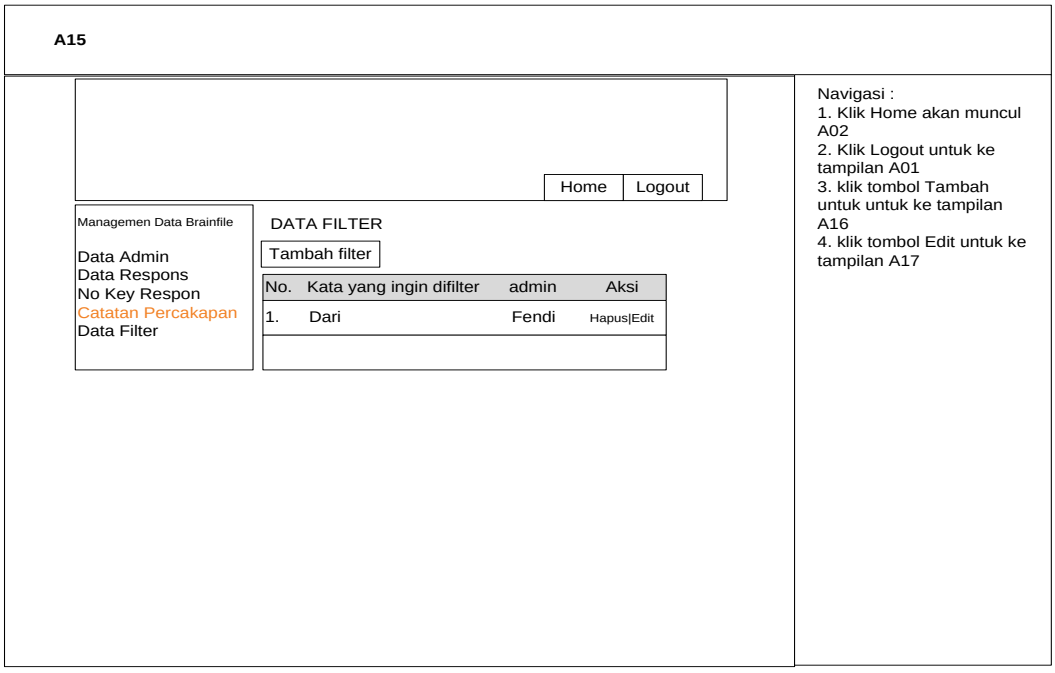
Gambar 3.31 Antarmuka Superadmin A12 Edit Data Respon Tanpa *Pattern*



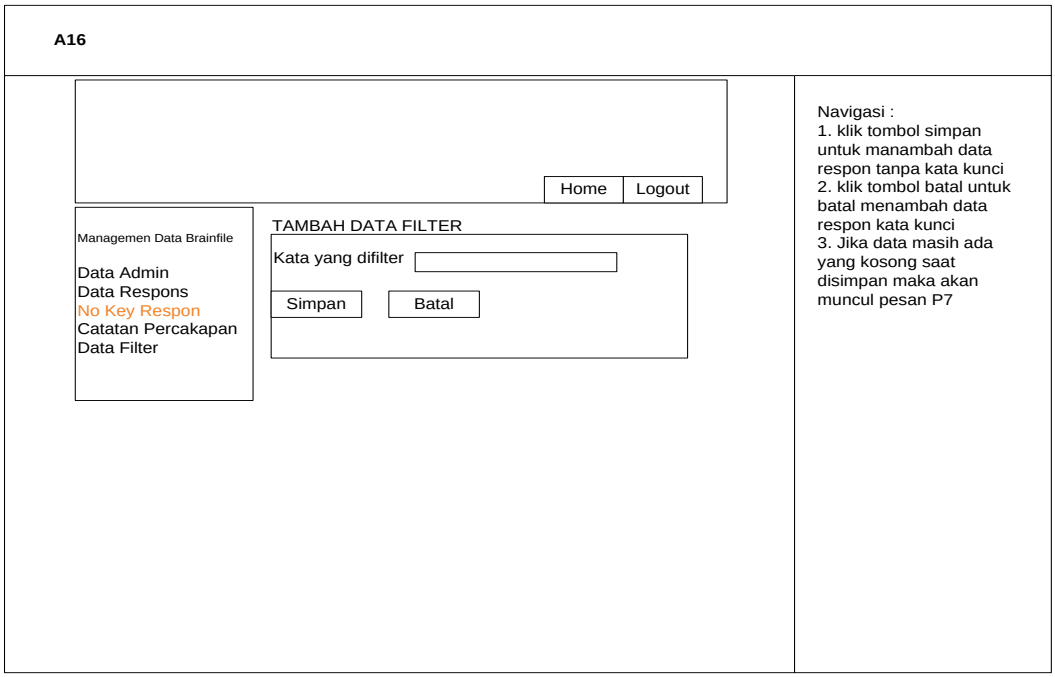
Gambar 3.32 Antarmuka Superadmin A13 Data Catatan Percakapan



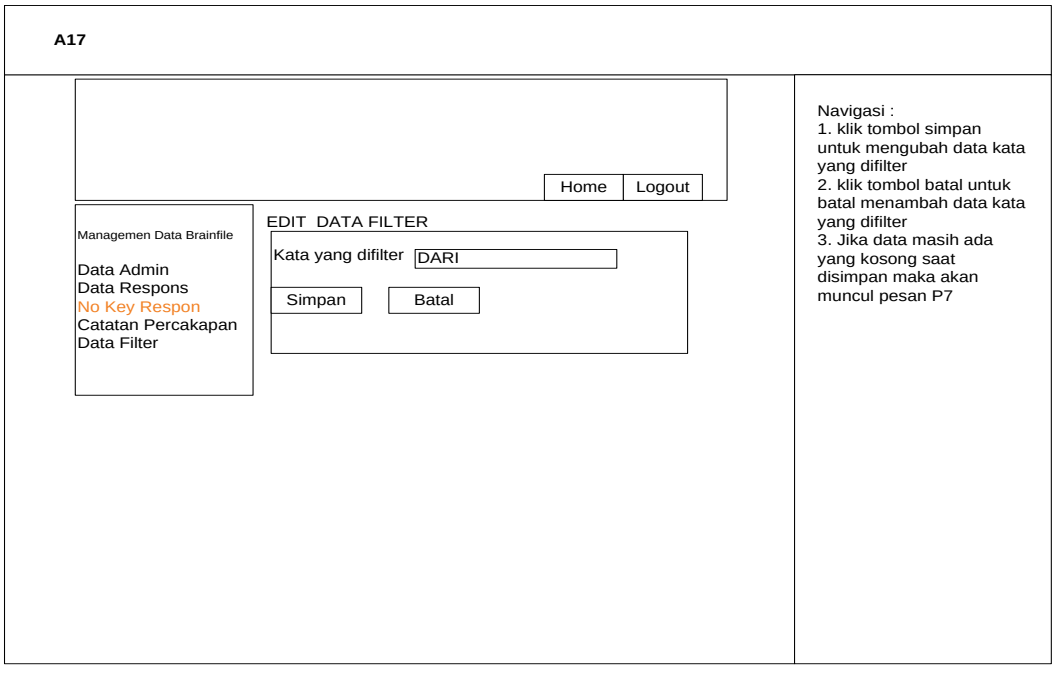
Gambar 3.33 Antarmuka Superadmin A14 Proses Learning Tambah Data Respon



Gambar 3.34 Antarmuka Superadmin A15 Data Filter

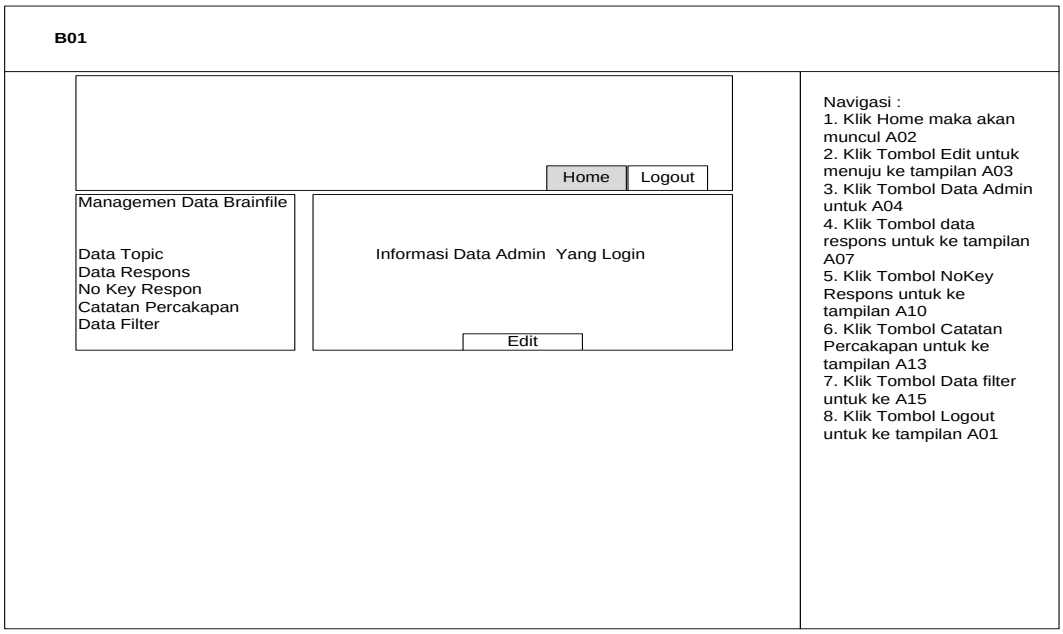


Gambar 3.35 Antarmuka Superadmin A16 Tambah Data Filter



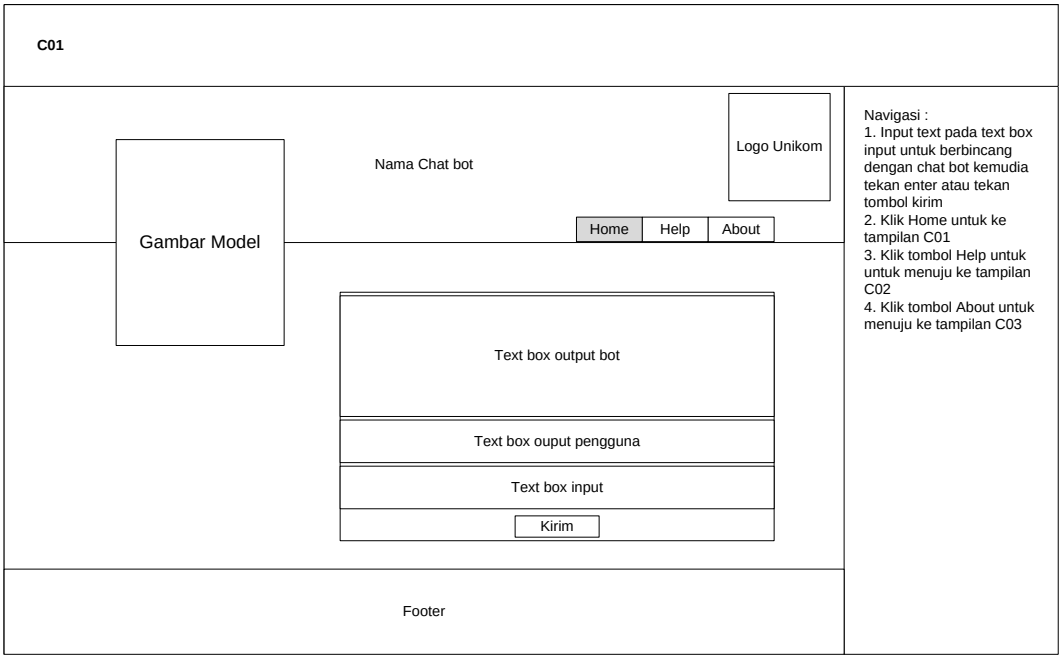
Gambar 3.36 Antarmuka Superadmin A17 Edit Data Filter

3.1.12.2Perancangan Antarmuka Admin

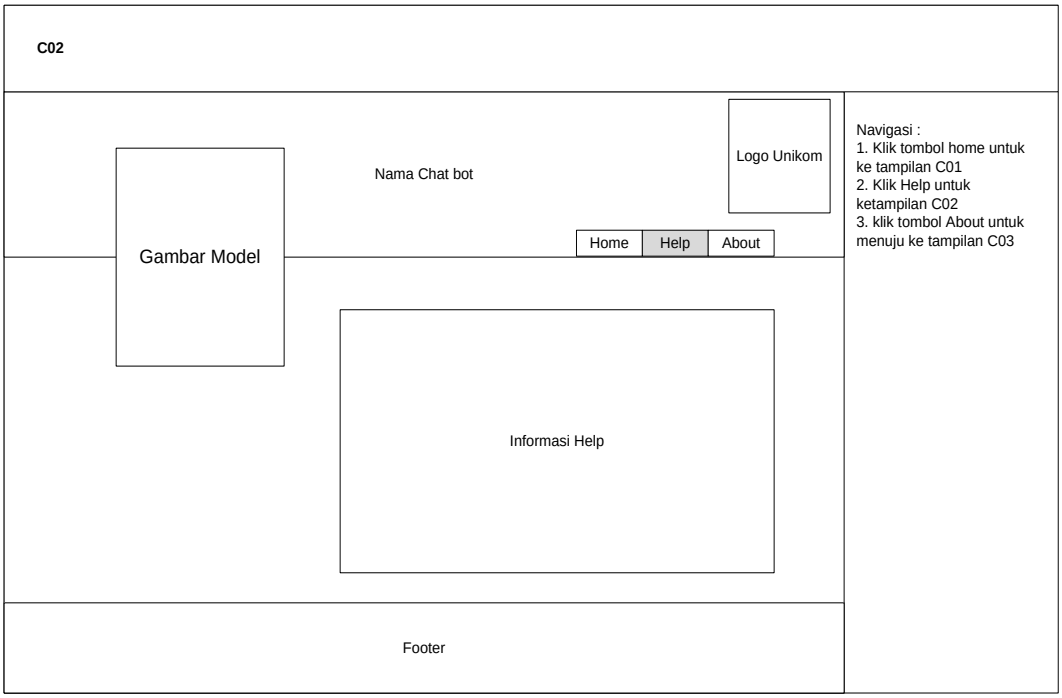


Gambar 3.37 Antarmuka Admin B01 Menu Utama Admin

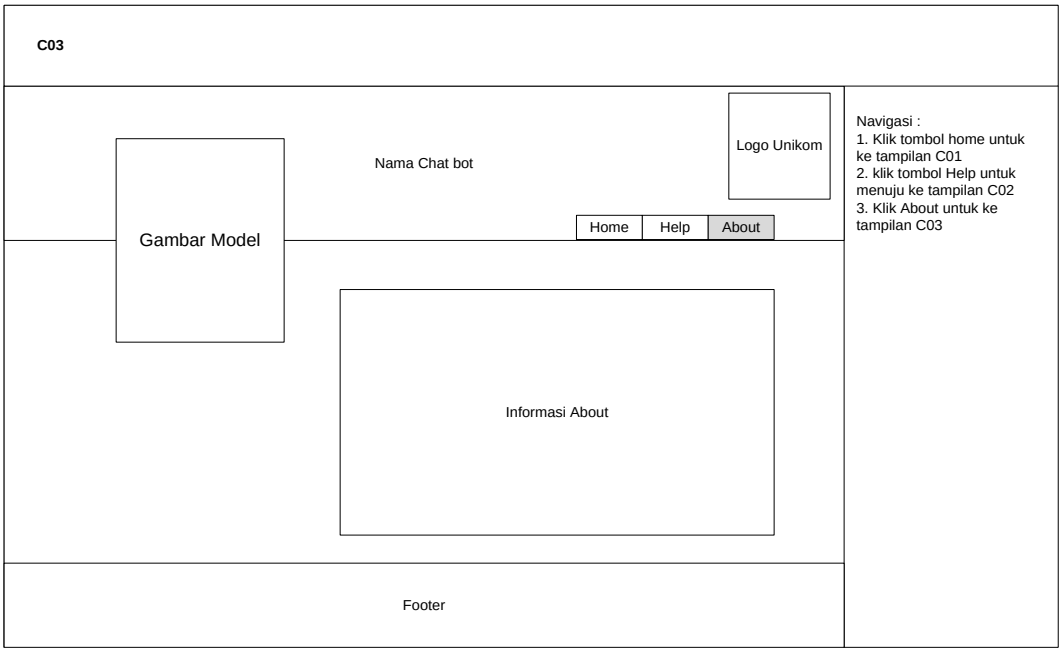
3.1.12.3 Perancangan Antamuka Pengunjung



Gambar 3.38 Antarmuka Pengunjung C01 Halaman Depan Pengunjung



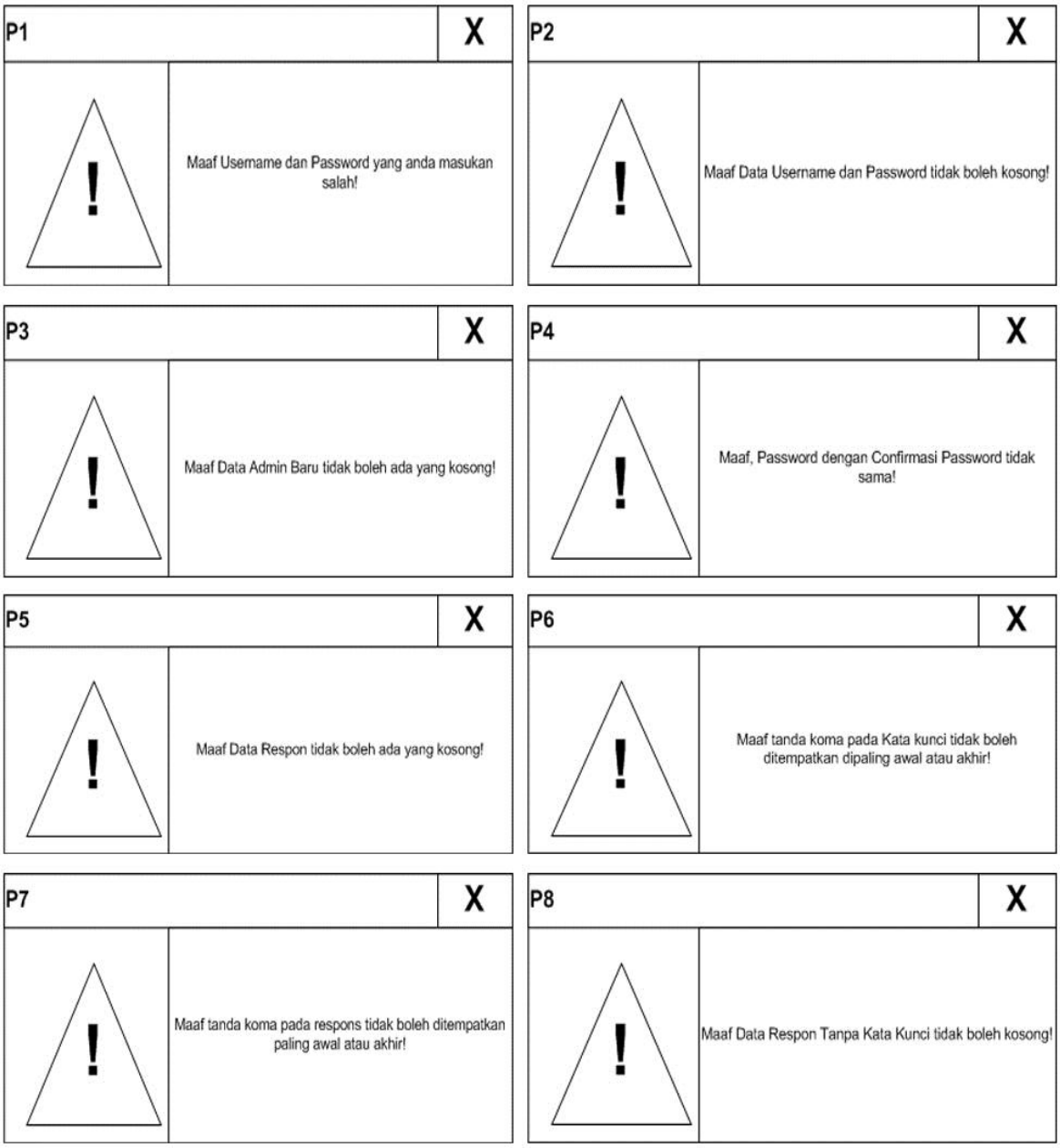
Gambar 3.39 Antarmuka Pengunjung C02 Halaman Help



Gambar 3.40 Antarmuka Pengunjung C03 Halaman About

3.1.12.4 Perancangan Pesan

Perancangan pesan yang akan dibuat pada Aplikasi Natural Language Processing untuk pembuatan chat bot dapat dilihat pada gambar :

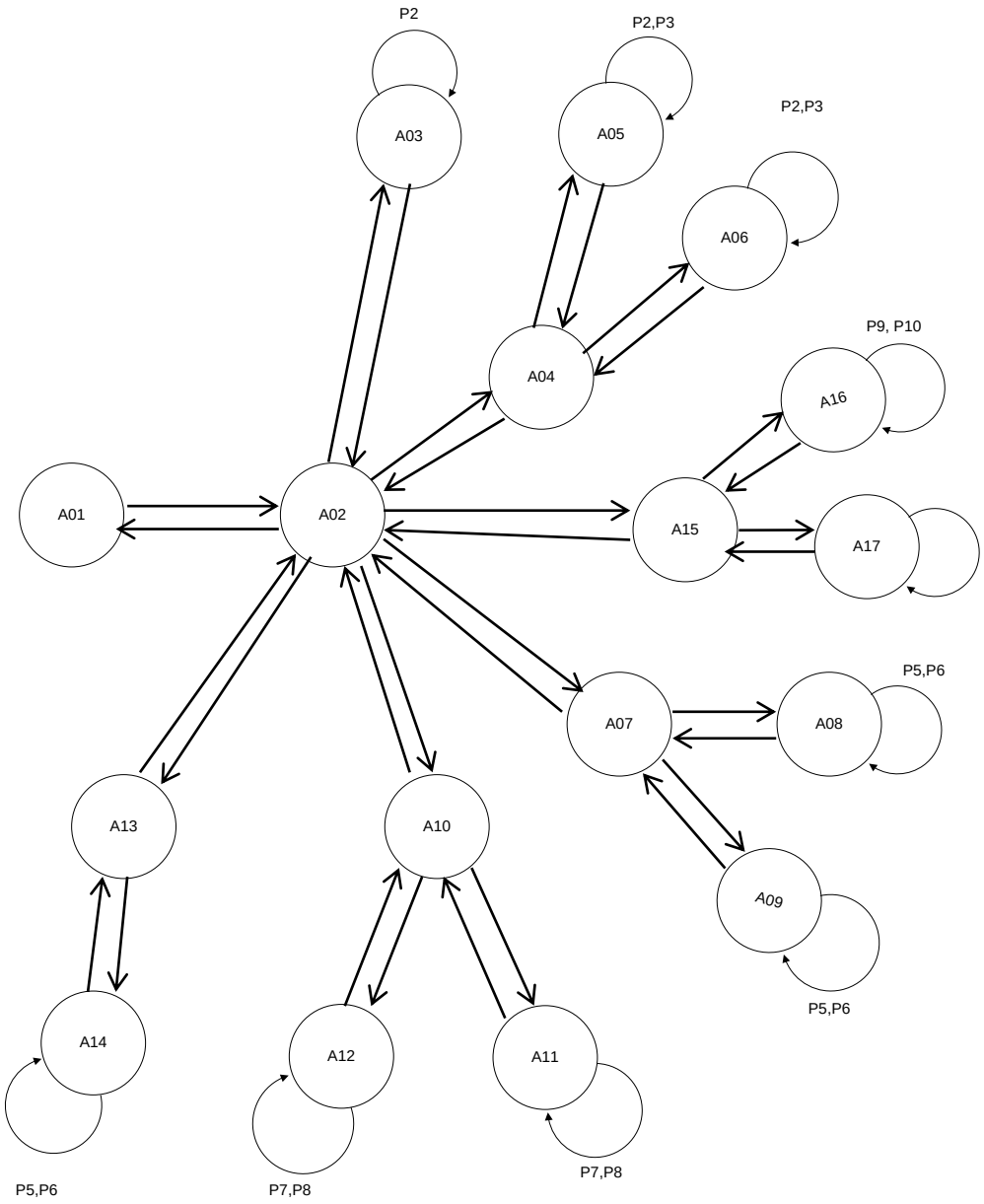


Gambar 3.41 Perancangan Pesan

3.1.13 Jaringan Semantik

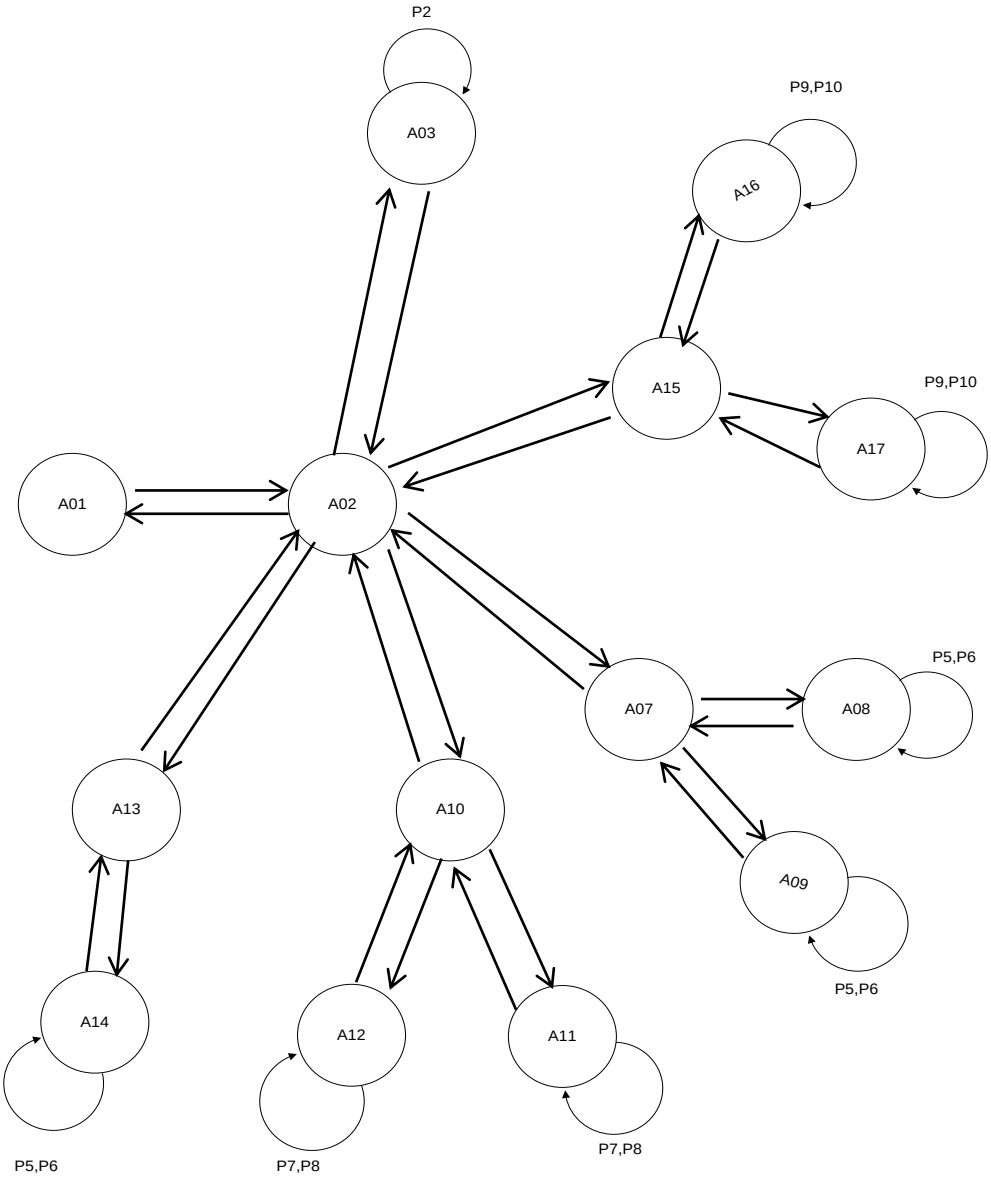
Jaringan semantic yang terbentuk pada Aplikasi Natural Language Processing untuk pembuatan chat bot berbasis web ini adalah :

a. Jaringan Semantik Superadmin



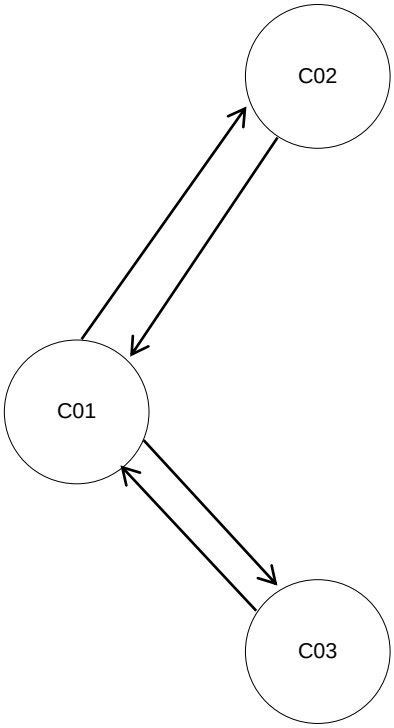
Gambar 3.42 Jaringan Semantik Superadmin

b. Jaringan Semantik admin



Gambar 3.43 Jaringan Semantik Admin

c. Jaringan Semantik Pengunjung

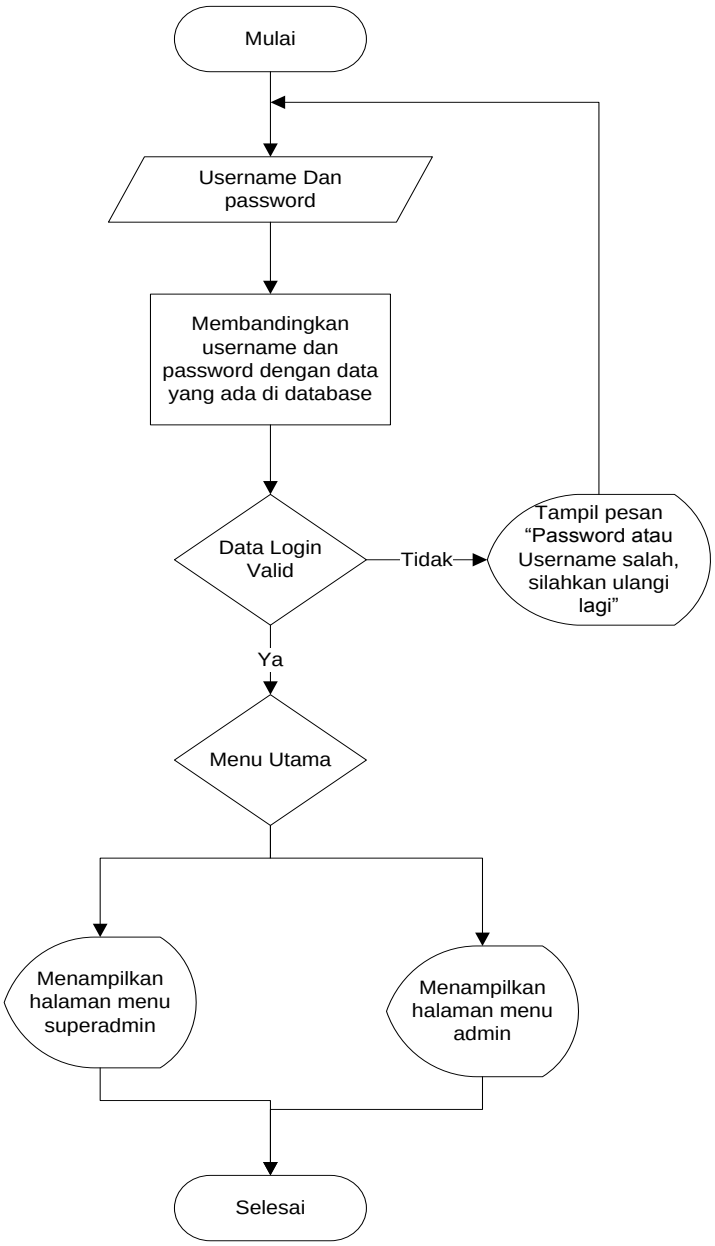


Gambar 3.44 Jaringan Semantik Pengunjung

3.1.14 Perancangan Prosedural

Perancangan procedural merupakan perancangan yang dilakukan untuk menetapkan detail algoritma yang akan dinyatakan kedalam suatu program. Adapun perancangan procedural untuk Aplikasi *Natural language Processing* untuk pembuatan *chat bot* berbasis web yang akan dibangun sebagai berikut :

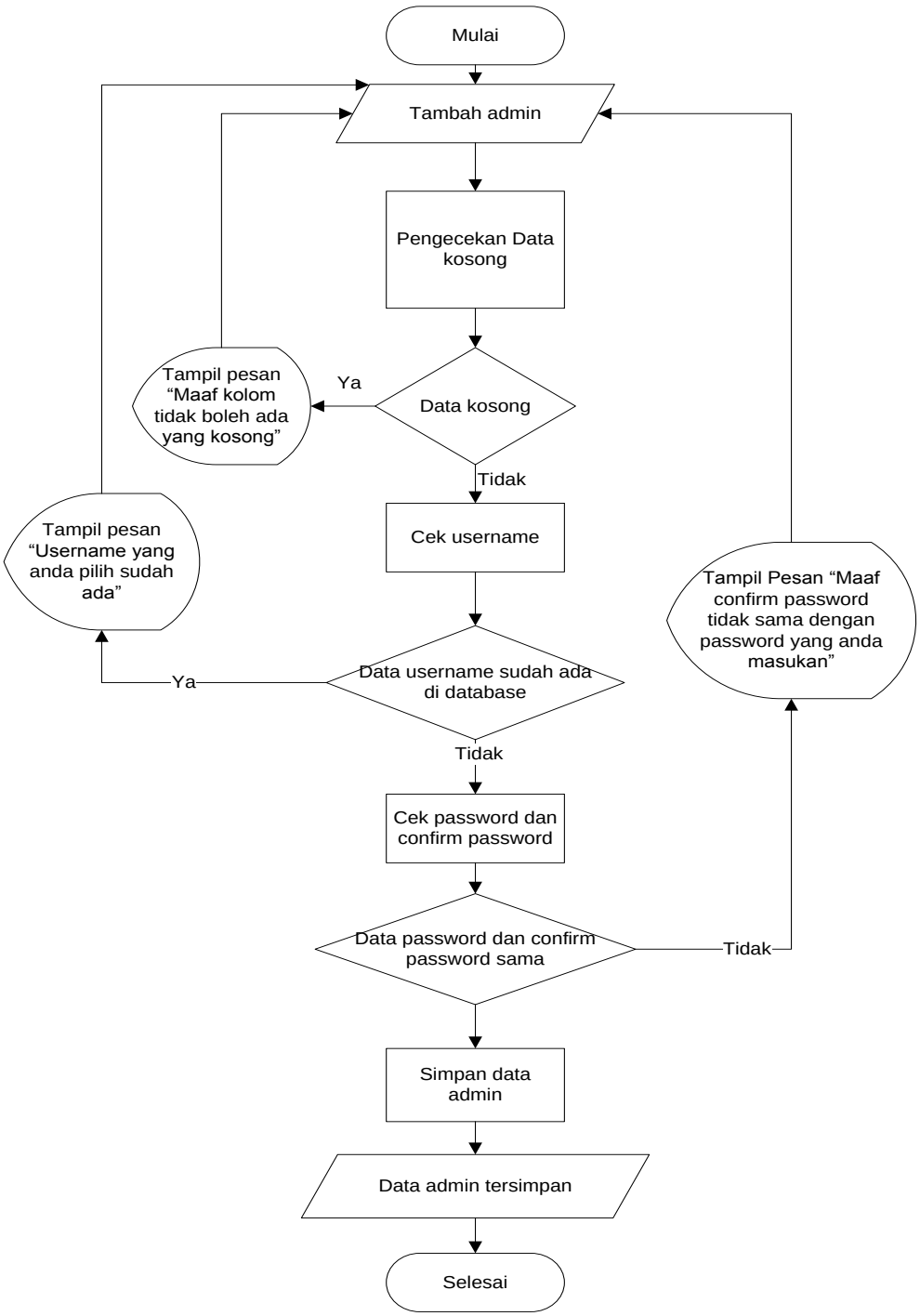
1. Prosedural *login* merupakan prosedur yang terjadi ketika Superadmin, Admin mulai mengakses Aplikasi.



Gambar 3.45 Tampilan *Flowchart Login*

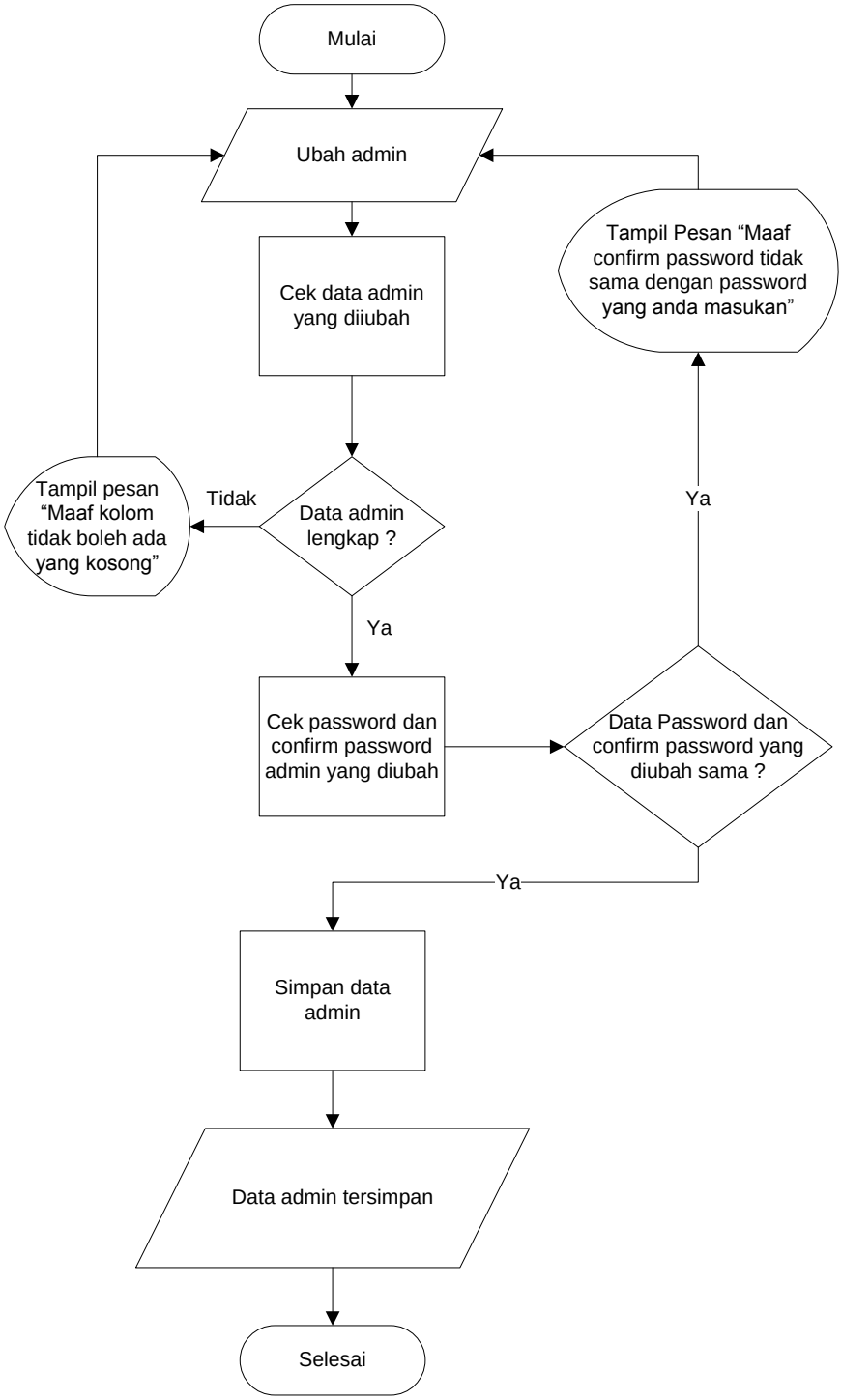
2. Pengolahan Data Admin

a. Prosedur Tambah Data Admin



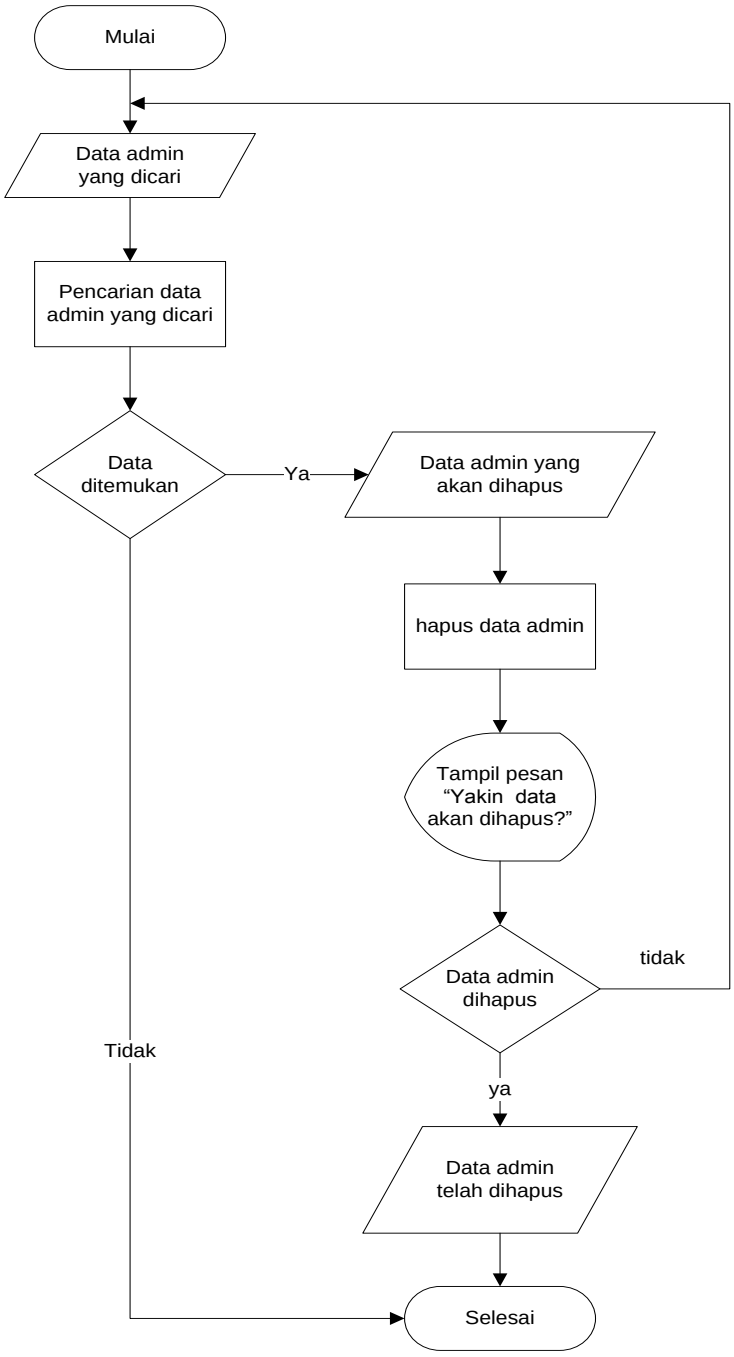
Gambar 3.46 *Flowchart* Tambah Data Admin

b. Prosedur Ubah Data Admin



Gambar 3.47 Tampilan *Flowchart* Ubah Data Admin

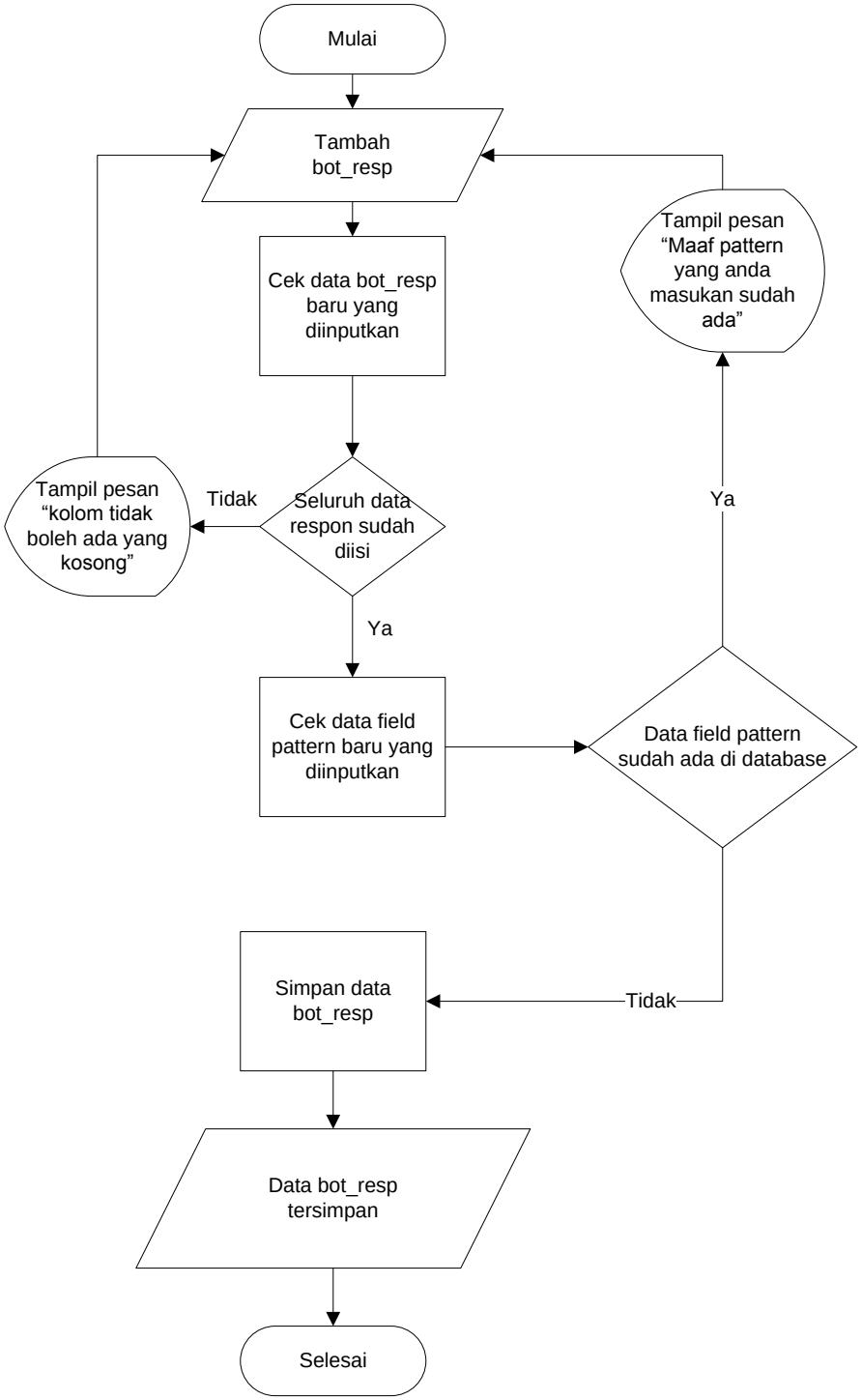
c. Prosedural Hapus Data Admin



Gambar 3.48 Tampilan *Flowchart* Hapus Data Admin

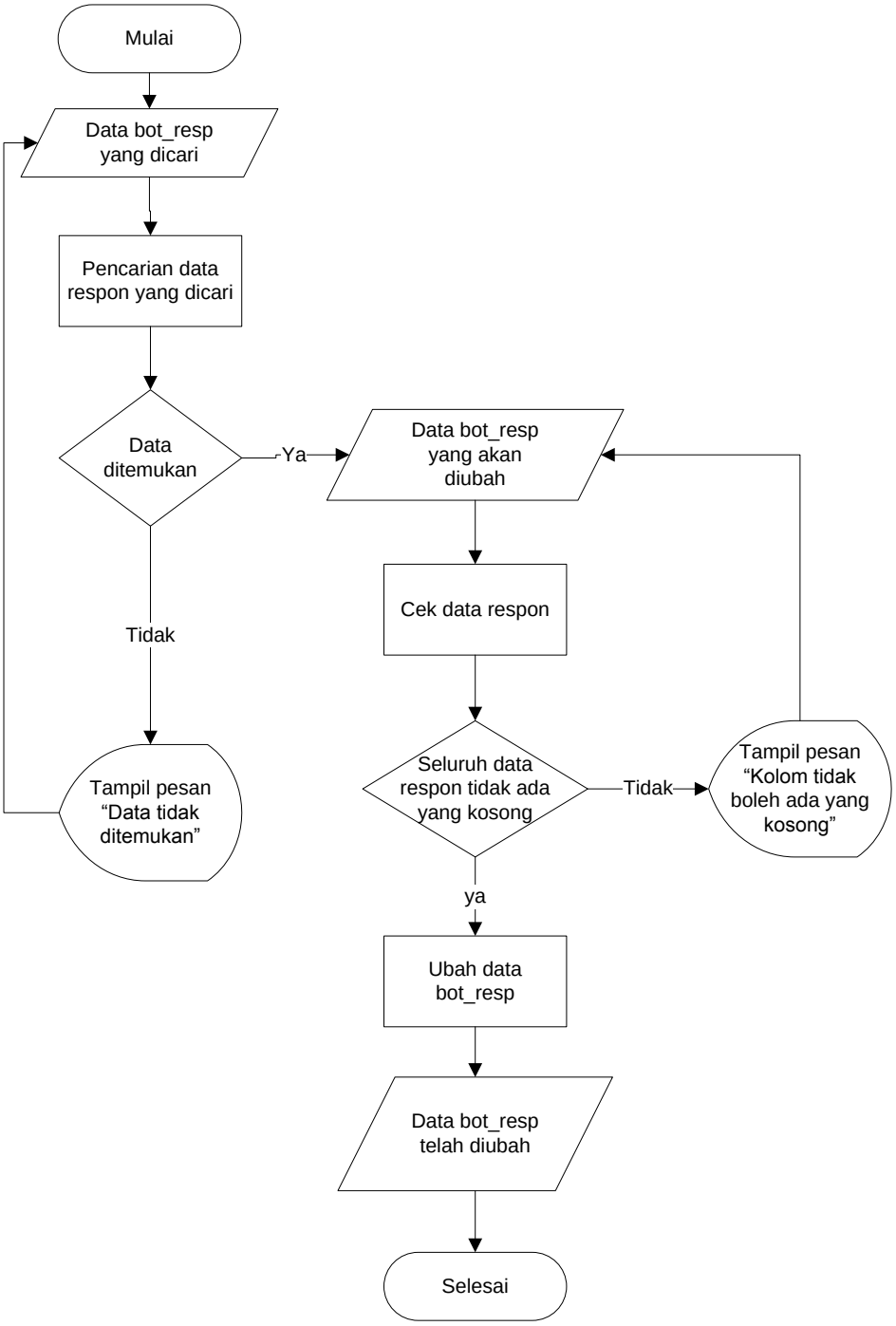
3. Pengolahan Data Bot_resp

   a. Prosedur Tambah Data Bot_resp



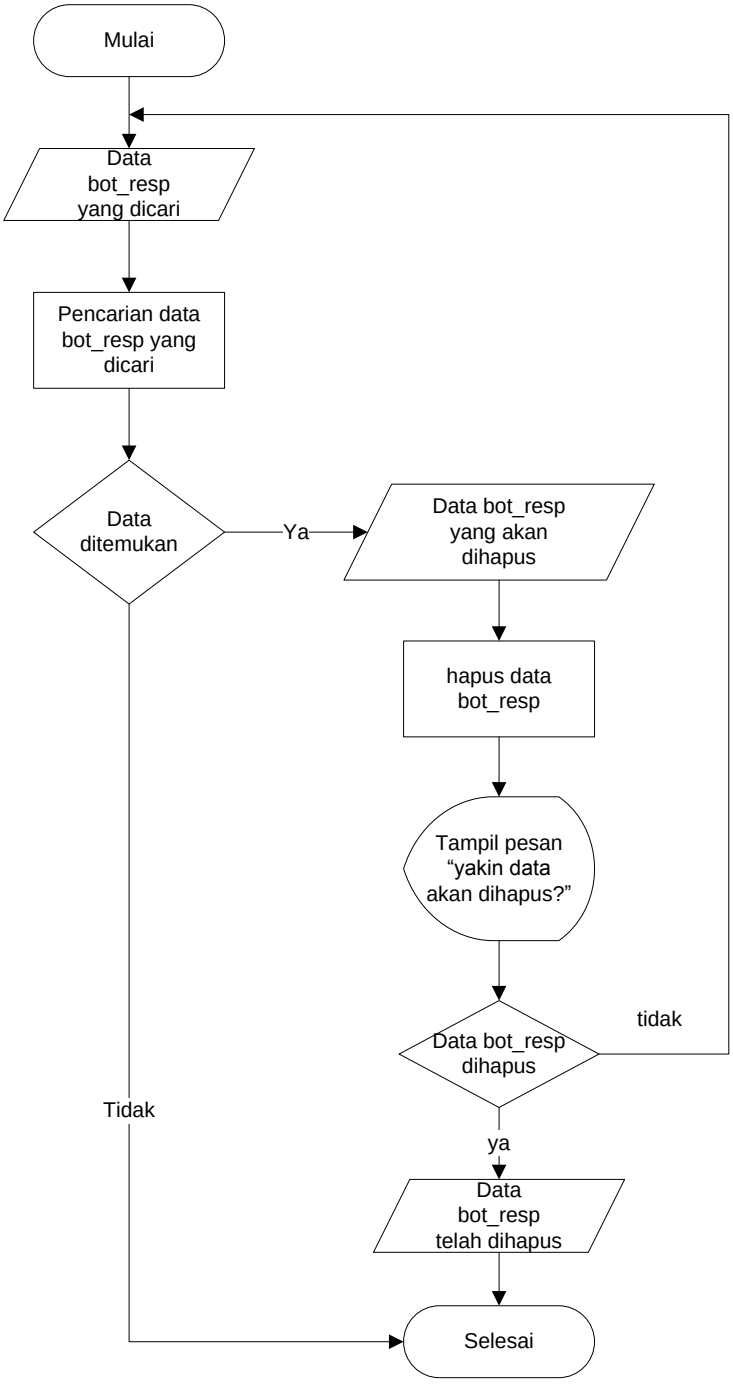
Gambar 3.49 Tampilan *Flowchart* Tambah Data Bot_resp

b. Prosedur Ubah Data Bot_resp



Gambar 3.50 Tampilan *Flowchart* Edit Data Bot_resp

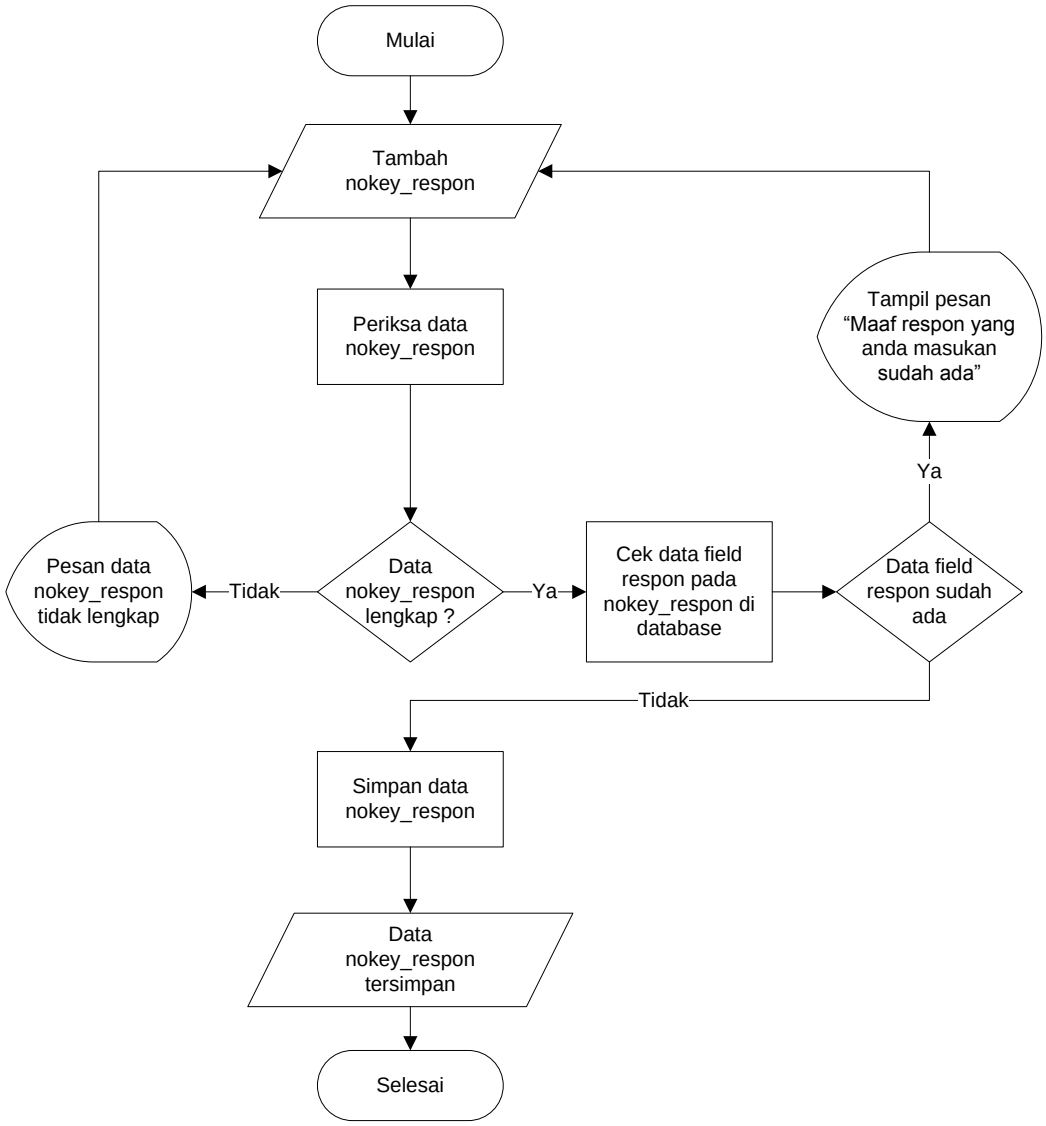
c. Prosedur Hapus Data Bot_resp



Gambar 3.51 *Flowchart* Hapus Data Respon

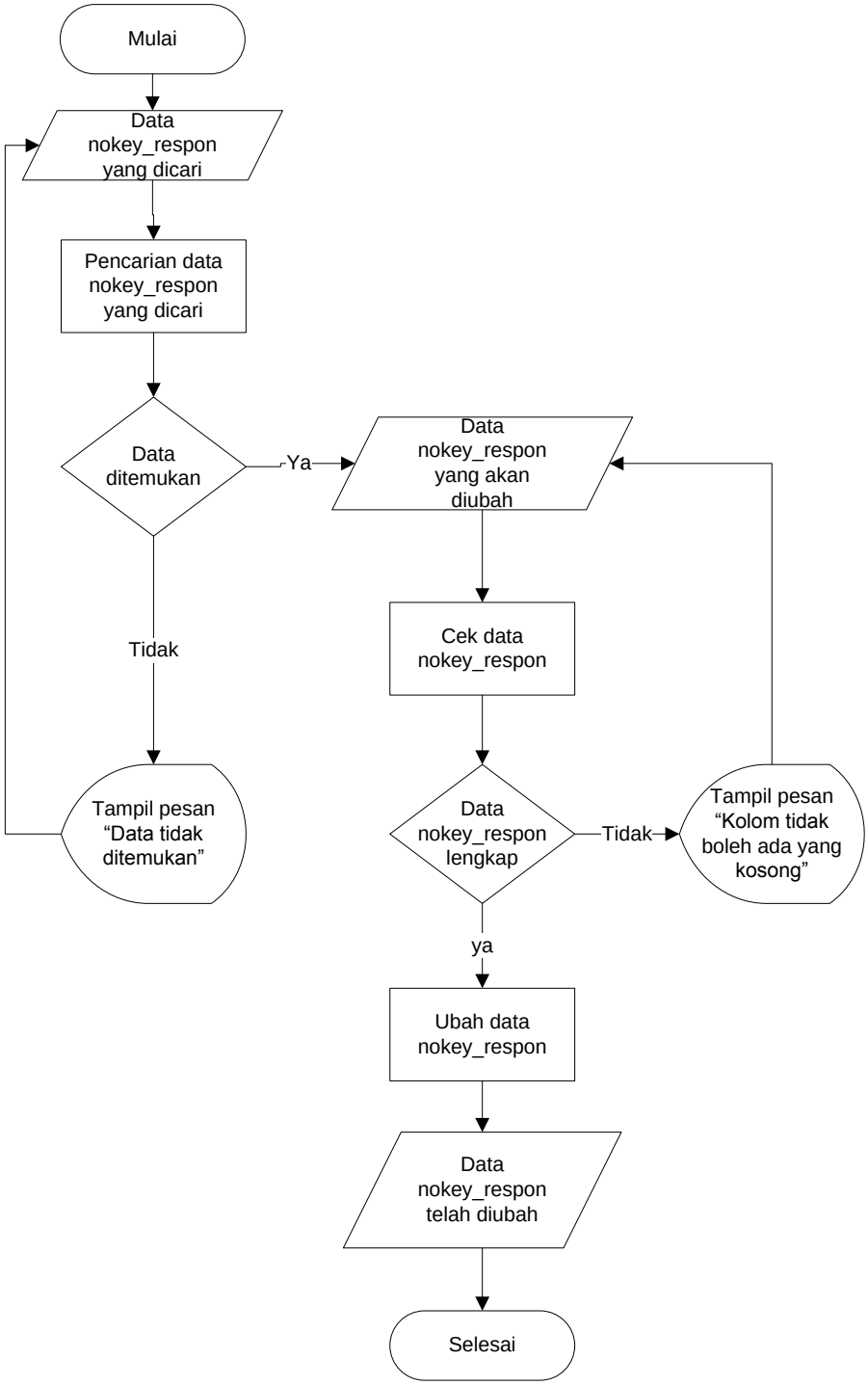
4. Pengolahan Data *Nokey_respons*

   a. Prosedur Tambah Data *Nokey_respons*



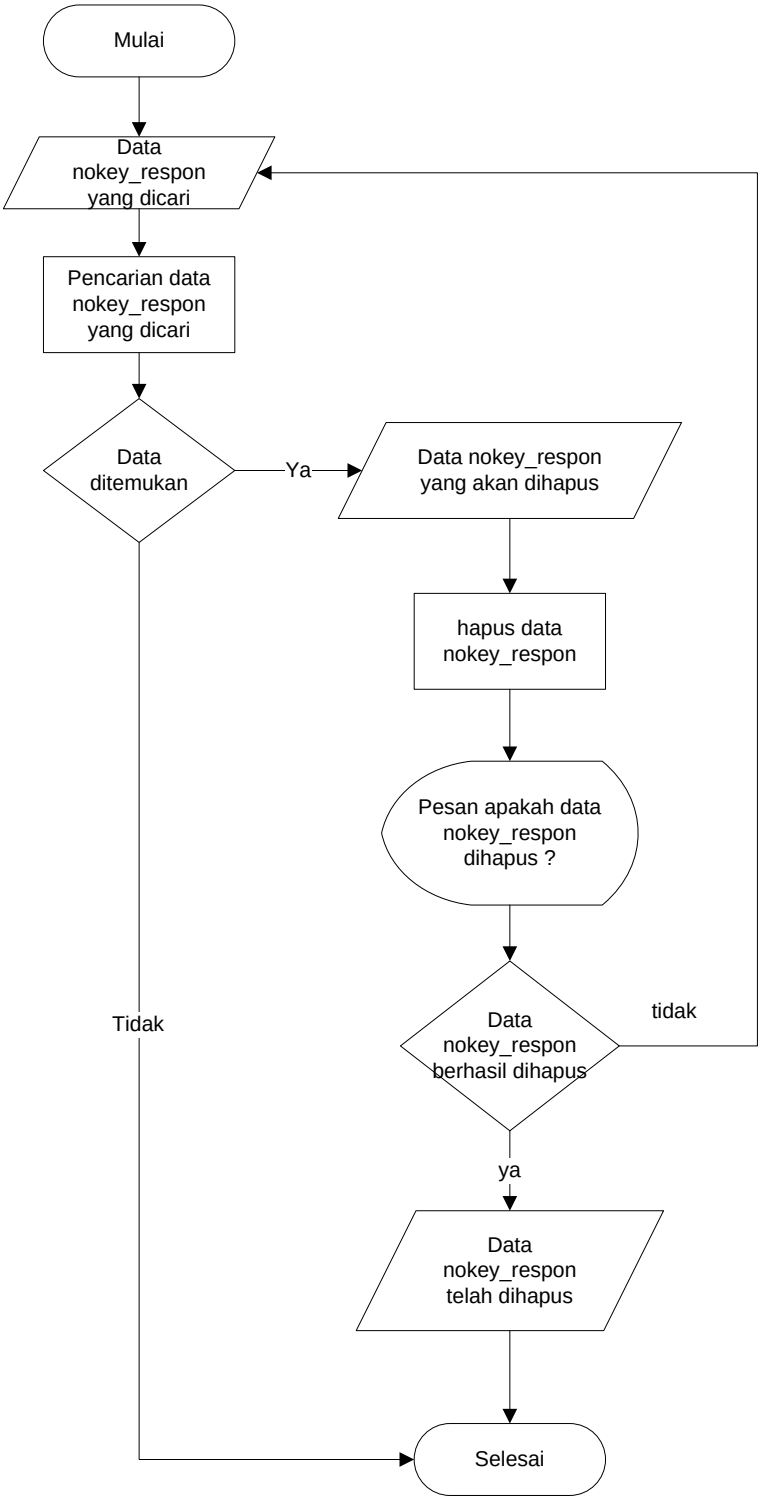
Gambar 3.52 Tampilan *Flowchart* Tambah Data *Nokey_respons*

b. Prosedur Edit Data Nokey_respon



Gambar 3.53 Tampilan *Flowchart* Edit Data *Nokey_respons*

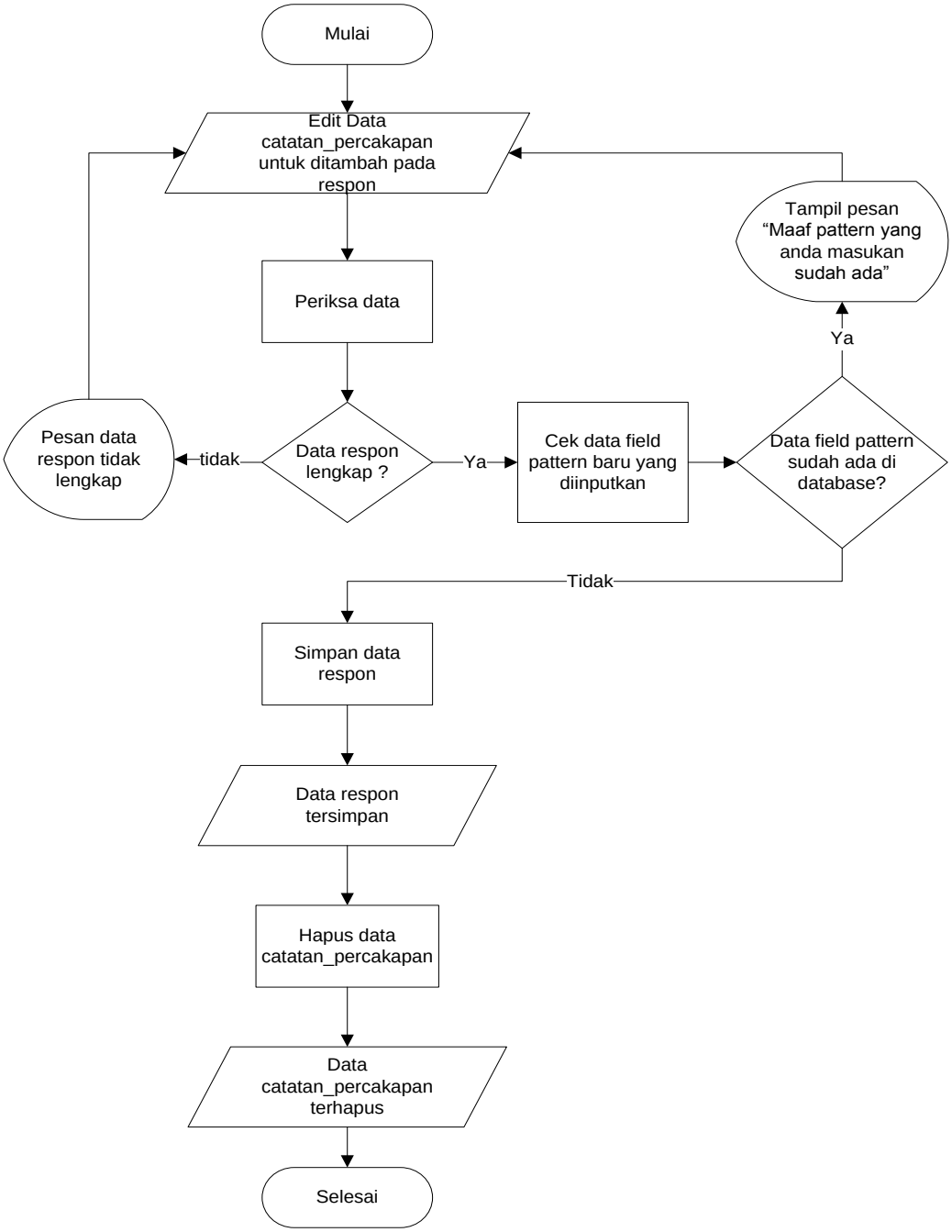
c. Prosedur Hapus Data *Nokey*_respons



Gambar 3.54 Tampilan *Flowchart* Hapus Data *Nokey_respons*

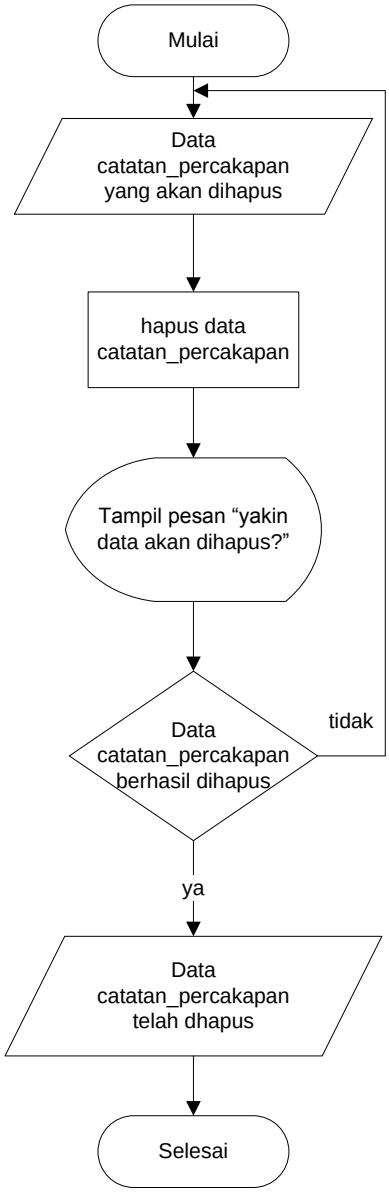
5. Pengolahan Data Catatan Percakapan

a. Prosedur Edit Catatan Percakapan



Gambar 3.55 Tampilan Flowchart Edit Data Catatan Percakapan

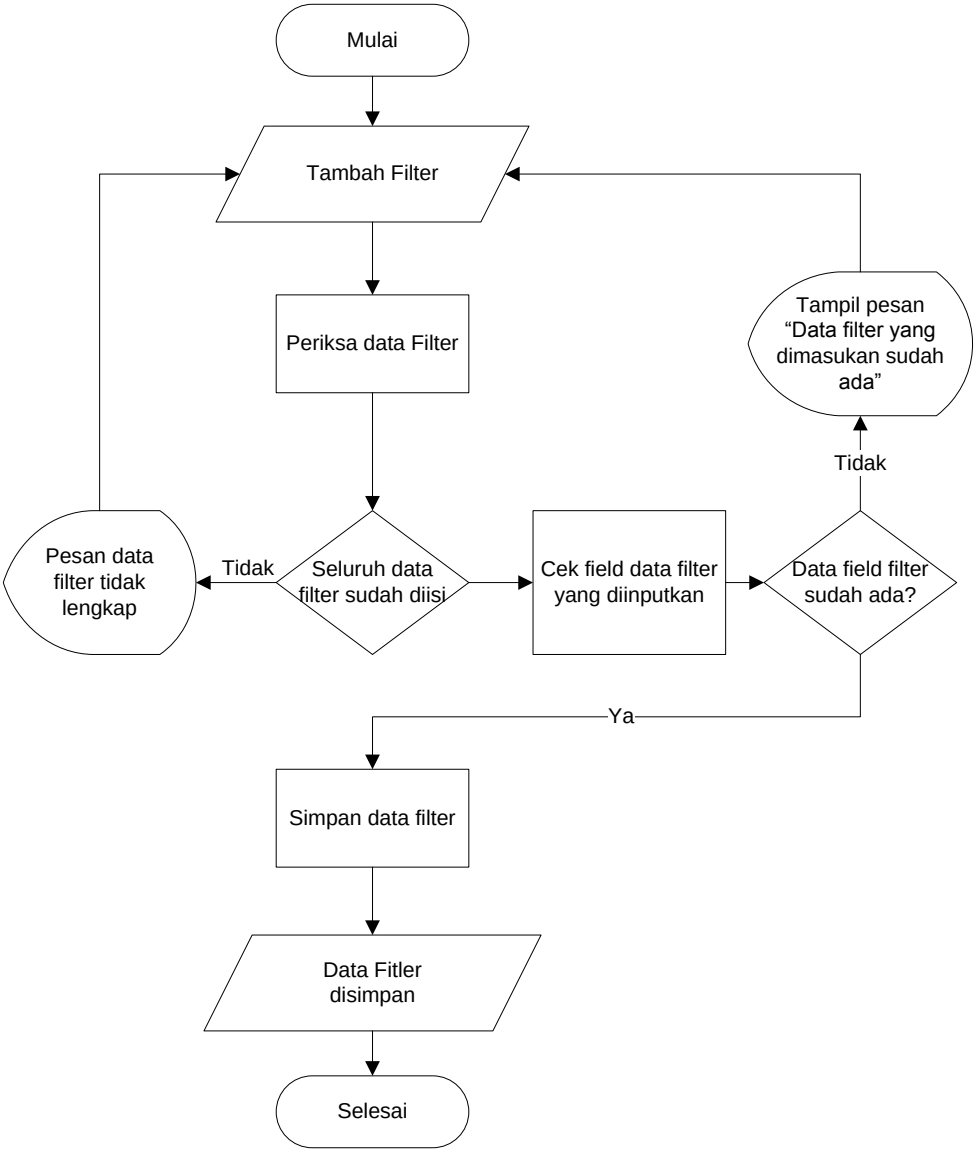
b. Prosedur Hapus Data Catatan_percakapan



Gambar 3.56 Tampilan *Flowchart* Hapus Data Catatan Percakapan

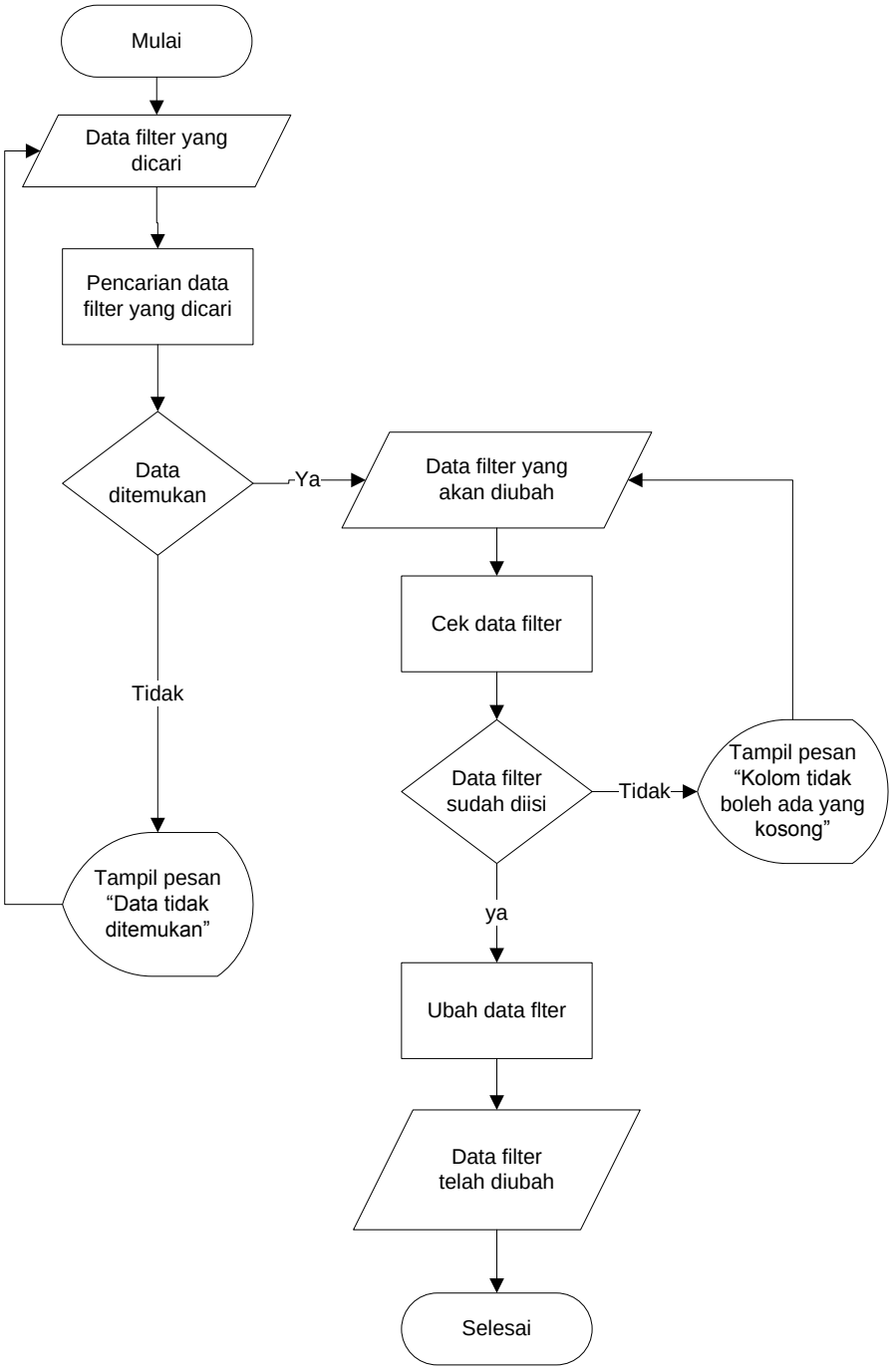
6. Pengolahan Data Filter

   a. Prosedur Tambah Data Filter



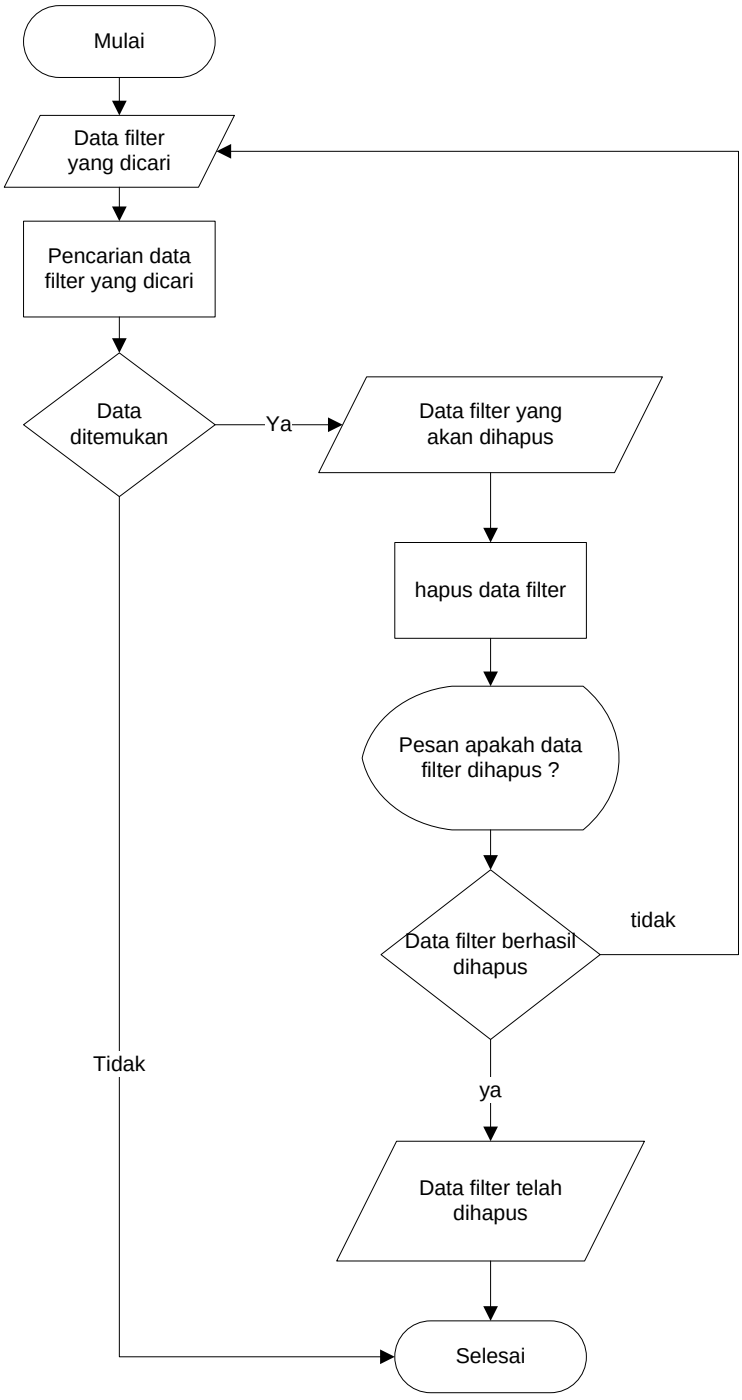
Gambar 3.57 Tampilan *Flowchart* Tambah Data Filter

b. Prosedur Edit Data Filter



Gambar 3.58 Tampilan *Flowchart* Edit Data Filter

c. Prosedure Hapus Data Filter



Gambar 3.59 Tampilan *Flowchart* Hapus Data Filter