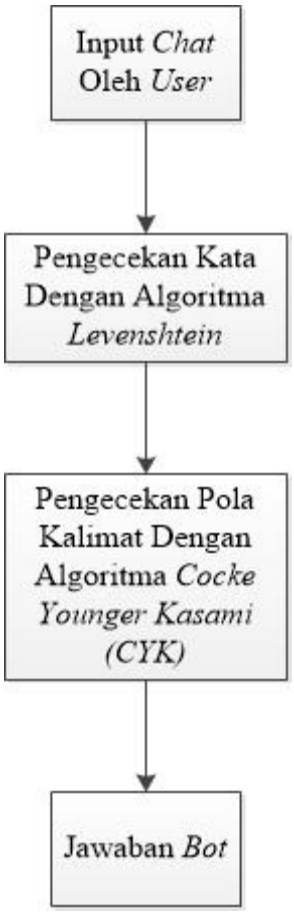
BAB 3
ANALISIS DAN PERANCANGAN APLIKASI

3.1 Analisis Aplikasi

Analisis dilakukan dengan tujuan untuk mengidentifikasi dan mengevaluasi seluruh komponen yang terkait dengan aplikasi yang akan dibangun.Menganalisis tahapan pembangunan aplikasi *chat bot* dengan menggunakan algoritma *Levenshtein* dan algoritma *Cocke Younger Kasami* (CYK). Analisis aplikasi ini akan menghasilkan beberapa data dan fakta yang akan dijadikan bahan uji dan analisis menuju pengembangan dan penerapan sebuah aplikasi yang diusulkan.



Gambar 3.1. Gambaran Umum Aplikasi

3.1.1 Analisis Masalah

Program *chat bot* pada umumnya menggunakan bahasa Inggris dalam percakapannya. *Chat bot* baik diterapkan sebagai aplikasi untuk meningkatkan kemampuan berbahasa Inggris karena tidak memerlukan lawan bicara langsung yang bisa berbahasa Inggris.

Masalah yang dihadapi dalam *chat bot* jika diterapkan di Brighten English Course yaitu dalam percakapan yang diinputkan oleh siswa tidak terdapat pembenaran kata jika pengguna salah menginput beberapa huruf dalam satu kata. Hal ini dapat menyebabkan siswa tidak mengetahui kata yang dimasukkan benar atau salah. Masalah lainnya yaitu tidak ada pengecekan *grammar* sehingga siswa tidak mengetahui kalimat yang telah dimasukkan benar atau salah.

Berdasarkan permasalahan tersebut maka dalam sebuah aplikasi *chat bot* dibutuhkan algoritma *Levenshtein* yang dapat melakukan pembenaran kata dan juga pengecekan grammar yang telah diinputkan oleh user dengan menggunakan algoritma *Cocke Younger Kasami* (*CYK*).

3.1.2 Analisis Data Masukan

Data masukan untuk chat bot ini dimasukkan oleh user (siswa). Untuk data atau informasi yang diberikan adalah seputar hobi khusunya olahraga, kuliner, dan wisata hiburan. Adapun detail informasi yang dapat diberikan oleh chat bot adalah sebagai berikut:

1. Informasi seputar olahraga :
    a. Tempat olahraga di Makassar seperti futsal, lari, basket , renang.
    b. Beberapa club sepak bola, dan basket yang terkenal.
    c. Pemain sepak bola, basket, MotoGP, dan Formula 1 yang terkenal.
2. Informasi seputar kuliner :
    a. Informasi mengenai beberapa makanan nasional dan internasional.
    b. Informasi mengenai tempat makan yang enak di Makassar.
    c. Informasi mengenai tempat berbelanja kue di Makassar.
3. Informasi seputar wisata hiburan:
    a. Informasi mengenai tempat wisata untuk liburan di Sulawesi Selatan.

b. Informasi mengenai tempat kumpul di akhir pekan.

3.1.3 Analisis Sintaksis Untuk Aturan Produksi

Berdasarkan pembahasan tentang sintaksis Bahasa Inggris dan pengelompokkan kata dengan Penn Treebank POS Tag[11], selanjutnya pada bagian ini akan dijelaskan bagaimana membuat aturan produksi dalam bentuk CNF.

3.1.3.1 Aturan Produksi Untuk Kalimat Simple Present Tense

Susunan kelompok kata yang dapat dibentuk dari kalimat simple present tense adalah sebagai berikut:

1. Kalimat Positif
   a. S + V1 (s/es) (+ O + Adv)
   b. N + V + N + Adv
   c. N + V + Pron + Adv
   d. Pron + V + N + Adv
   e. Pron + V + Pron + Adv
2. Kalimat Negatif
   a. S + do, does + not + V1 (+ O + Adv)
   b. N + V + Prep + V + N + Adv
   c. N + V + Prep + V+ Pron + Adv
   d. Pron + V + Prep + V + N + Adv
   e. Pron + V + Prep + V + Pron + Adv
3. Kalimat Interogatif
   a. Do, Does + S + V1 (+ O + Adv) ? atau
   b. Dont, Doesn’t + S + V1 (+ O + Adv) ?
   c. V + N + V + N + Adv + Z
   d. V + N + V + Pron + Adv + Z
   e. V + Pron + V + N + Adv + Z
   f. V + Pron + V + N + Pron + Adv + Z

Aturan produksi (P) yang terbentuk untuk kalimat Simple Present Tense adalah sebagai berikut.

S → S PRE1 | S CONJUNCTION| N CONJUCTION|PP1 VB| PP2 VBZ | PP1 PRE1 | PP2 PRE2 | PP1 JJ |PP2 JJ| N JJ| N PRE2 | N PRE1 |PRE1 PRE1 | PRE2 PRE1 | DT S

PRE1 → VB N |VB PP1| VB TO | VB IN|N JJ |VB COMB_OBJ | VBB COMB_OBJ |PRE1 JJ | VB NOT | VB PP1 | VBB PP1| VBB PP2| COMB_OBJ

PRE2 → VBZ N | VBZ PP1|VBZ TO |VBZ IN|VBZ COMB_OBJ |PRE2 JJ | VBZ NOT | VBZ PP2 | VBZ N

N → NN | NNS | NNP| NNPS

IN → in | on | at

COMB_OBJ → ARTICLE N

CONJUNCTION → CC S | CC PRE1 | IN N

DT → the

CC → and | or | but

ARTICLE → a | an

PP1 → i | you | they | we

PP2 → he | she | it

VB → eat

VBB → am | are |is

VBZ → eats

NN → apple |him | them | monkey

NNS → apples | monkeys

NNP → lina | budi | amanda

NNPS → americans

JJ → everyday |everynight | always | usually

Dimana secara formal dapat dinyatakan dengan V = himpunan variabel/non terminal, T = himpunan terminal, P = kumpulan aturan produksi, S = simbol awal. Variabel dengan uraian beserta keterangannya dapat dilihat pada tabel 3.1. Pada tabel 3.2 dapat dilihat pengelompokan variable di dalam V.

Tabel 3.1 Pengelompokan Variabel untuk Simple Present Tense

Variabel	Uraian Variabel	Keterangan
V	S, PRE1, CONJUNCTION, PP1, VB, PP2, VBZ, PRE2, JJ, N, DT, IN, TO, COMB_OBJ, VBB, NOT, NN, NNS, NNP, NNPS, ARTICLE	Himpunan non terminal
T	in, on, it, the, and, or, but, i, you, they, we, he, she, it, eat, am, are, is	Himpunan terminal

Tabel 3.2 Pengelompokan Variabel Di Dalam V untuk Simple Present Tense

Variabel Di Dalam V	Uraian Variabel	Keterangan
PRE1	VB N, VB PP1, VB TO, VB IN, N JJ, VB COMB_OBJ, VBB COMB_OBJ, PRE1 ADV, VB NOT, VB PP1, VBB PP1, VBB PP2, COMB_OBJ	Himpunan simbol yang terdapat pada simbol non terminal PRE1 sebagai pembentuk kata kerja (verb 1)
PRE2	VBZ N, VBZ PP1, VBZ TO, VBZ IN, VBZ COMB_OBJ, PRE2 JJ,	Himpunan simbol yang terdapat pada simbol non terminal PRE2 sebagai pembentuk kata kerja

	VBZ NOT, VBZ PP2	(verb 2)
CONJUNTION	CC S, CC PRE1, IN N	Himpunan simbol yang terdapat pada simbol non terminal CONJUTION sebagai pembentuk kata penghubung
COMB_OBJ	ARTICLE N	Himpunan simbol yang terdapat pada simbol non terminal COMB_OBJ untuk menyatakan sebuah benda
N	NN, NNS, NNP, NNPS	Himpunan simbol yang terdapat pada simbol non terminal N sebagai pembentuk kata benda

3.1.3.2 Aturan Produksi Untuk Kalimat Simple Past Tense

Susunan kelompok kata yang dapat dibentuk dari kalimat simple past tense adalah sebagai berikut:

1. Kalimat Positif
   a. S + V2 + (+ O + Adv)
   b. N + V + N + Adv
   c. N + V + Pron + Adv
   d. Pron + V + N + Adv
   e. Pron + V + Pron + Adv
2. Kalimat Negatif
   a. S + did + not + V1 (+ O + Adv)
   b. N + V + Prep + V + N + Adv
   c. N + V + Prep + V+ Pron + Adv
   d. Pron + V + Prep + V + N + Adv
   e. Pron + V + Prep + V + Pron + Adv
3. Kalimat Interogatif
   a. Did + S + V1 (+ O + Adv) ?

b. V + N + V + N + Adv + Z

c. V + N + V + Pron + Adv + Z

d. V + Pron + V + N + Adv + Z

e. V + Pron + V + Pron + Adv + Z

Aturan produksi yang terbentuk untuk kalimat Simple Past Tense adalah sebagai berikut.

S → S PRE1 | S CONJUNCTION| N CONJUCTION|PP1 PRE1 | PP2 PRE2 | PP1 ADV | N ADV| N PRE1 |PRE1 PRE1| DT S

PRE1 → VBD N |VBD PP1| VBD TO | VBD IN|N ADV |VBD COMB_OBJ |PRE1 ADV | VDB COMB_OBJ | VDB PP1 | VDB PP2 |VBD NOT | VBD PP1 | VBD PP2

N → NN | NNS | NNP

IN → in | on | at

COMB_OBJ → ARTICLE N

CONJUNCTION → CC S | CC PRE1 | IN N

DT → the

CC → and | or | but

ARTICLE → a | an

PP1 → i | you | they | we

PP2 → he | she | it

VB → ate

VDB → was | were

NN → apple |him | them | monkey

NNS → apples | monkeys

NNP → lina | budi | amanda

NNPS → americans

ADV → yesterday

Dimana secara formal dapat dinyatakan dengan V = himpunan variabel/non terminal, T = himpunan terminal, P = kumpulan aturan produksi, S = simbol awal. Variabel dengan uraian beserta keterangannya dapat dilihat pada tabel 3.3. Pada tabel 3.4 dapat dilihat pengelompokan variable di dalam V.

Tabel 3.3 Pengelompokan Variabel untuk Simple Past Tense

Variabel	Uraian Variabel	Keterangan
V	S, PRE1, CONJUNCTION, PP1, VB, PP2, VBZ, PRE2, ADV, N, DT, IN, TO, COMB_OBJ, VBB, NOT, NN, NNS, NNP, NNPS, ARTICLE	Himpunan non terminal
T	in, on, it, the, and, or, but, i, you, they, we, he, she, it, eat, am, are, is	Himpunan terminal

Tabel 3.4 Pengelompokan Variabel Di Dalam V untuk Simple Past Tense

Variabel Di Dalam V	Uraian Variabel	Keterangan
PRE1	VB N, VB PP1, VB TO, VB IN, N ADV, VB COMB_OBJ, VBB COMB_OBJ, PRE1 ADV, VB NOT, VB PP1, VBB PP1, VBB PP2, COMB_OBJ	Himpunan simbol yang terdapat pada simbol non terminal PRE1 sebagai pembentuk kata kerja (verb 1)
PRE2	VBZ N, VBZ PP1, VBZ TO, VBZ IN, VBZ COMB_OBJ, PRE2 ADV,	Himpunan simbol yang terdapat pada simbol non terminal PRE2 sebagai pembentuk kata kerja

	VBZ NOT, VBZ PP2	(verb 2)
CONJUNTION	CC S, CC PRE1, IN N	Himpunan simbol yang terdapat pada simbol non terminal CONJUTION sebagai pembentuk kata penghubung
COMB_OBJ	ARTICLE N	Himpunan simbol yang terdapat pada simbol non terminal COMB_OBJ untuk menyatakan sebuah benda
N	NN, NNS, NNP, NNPS	Himpunan simbol yang terdapat pada simbol non terminal N sebagai pembentuk kata benda

3.1.3.3 Aturan Produksi Untuk Kalimat Simple Future Tense

Susunan kelompok kata yang dapat dibentuk dari kalimat simple future tense adalah sebagai berikut:

1. Kalimat Positif
   a. S + will/shall + V1 + (+ O + Adv)
   b. N + V +V + N + Adv
   c. N + V + V + Pron + Adv
   d. Pron + V + V + N + Adv
   e. Pron + V + V + Pron + Adv
2. Kalimat Negatif
   a. S + will/shall + not + V1 (+ O + Adv)
   b. N + V + Prep + V + N + Adv
   c. N + V + Prep + V + Pron + Adv
   b. Pron + V + Prep + V +N + Adv
   c. Pron + V + Prep + V + Pron + Adv
3. Kalimat Interogatif
   a. Will/shall + S + V1 (+ O + Adv) ?

b. V + N + V + N + Adv + Z
c. V + N + V + Pron + Adv + Z
d. V + Pron + V + N + Adv + Z
e. V + Pron + V + Pron + Adv + Z

Aturan produksi yang terbentuk untuk kalimat Simple Future Tense adalah sebagai berikut.

S → S PRE1 | S CONJUNCTION| N CONJUCTION|PP1 PRE1 | ADV N | N PRE1 |PRE1 PRE1 | DT S
PRE1 → VB2 PRE1| VB N |VBN PP1 |VBN TO | VBN IN|N ADV |VBN COMB_OBJ |PRE1 ADV | VB1 NOT | VB2 PP1 | VB2 PP2 | VB2 N
N → NN | NNS | NNP
IN → in | on | at
COMB_OBJ → ARTICLE N
CONJUNCTION → CC S | CC PRE1 | IN N
VB2 → will | shall
DT → the
CC → and | or | but
ARTICLE → a | an
PP1 → i | you | they | we
PP2 → he | she | it
VB→ eat
NN → apple |him | them | monkey
NNS → apples | monkeys
NNP → lina | budi | amanda
NNPS → americans
ADV → tomorrow

Dimana secara formal dapat dinyatakan dengan V = himpunan variabel/non terminal, T = himpunan terminal, P = kumpulan aturan produksi, S = simbol awal.

Variabel dengan uraian beserta keterangannya dapat dilihat pada tabel 3.5. Pada tabel 3.6 dapat dilihat pengelompokan variable di dalam V.

Tabel 3.5 Pengelompokan Variabel untuk Simple Future Tense

Variabel	Uraian Variabel	Keterangan
V	S, PRE1, CONJUNCTION, PP1, VB, PP2, VBZ, PRE2, ADV, N, DT, IN, TO, COMB_OBJ, VBB, NOT, NN, NNS, NNP, NNPS, ARTICLE	Himpunan non terminal
T	in, on, it, the, and, or, but, i, you, they, we, he, she, it, eat, am, are, is	Himpunan terminal

Tabel 3.6 Pengelompokan Variabel Di Dalam V untuk Simple Future Tense

Variabel Di Dalam V	Uraian Variabel	Keterangan
PRE1	VB N, VB PP1, VB TO, VB IN, N ADV, VB COMB_OBJ, VBB COMB_OBJ, PRE1 ADV, VB NOT, VB PP1, VBB PP1, VBB PP2, COMB_OBJ	Himpunan simbol yang terdapat pada simbol non terminal PRE1 sebagai pembentuk kata kerja (verb 1)
PRE2	VBZ N, VBZ PP1, VBZ TO, VBZ IN, VBZ COMB_OBJ, PRE2 ADV, VBZ NOT, VBZ PP2	Himpunan simbol yang terdapat pada simbol non terminal PRE2 sebagai pembentuk kata kerja (verb 2)
CONJUNTION	CC S, CC PRE1, IN N	Himpunan simbol yang terdapat pada simbol non terminal

		CONJUTION sebagai pembentuk kata penghubung
COMB_OBJ	ARTICLE N	Himpunan simbol yang terdapat pada simbol non terminal COMB_OBJ untuk menyatakan sebuah benda
N	NN, NNS, NNP, NNPS	Himpunan simbol yang terdapat pada simbol non terminal N sebagai pembentuk kata benda

3.1.4 Analisis Algoritma *CYK* dan *Levenshtein* pada Perbaikan Kalimat

Pada tahap ini dilakukan analisis sebuah kalimat untuk dilakukan perbaikan pola kalimat berdasarkan analisis aturan produksi yang telah dibuat sebelumnya. Alur proses algoritma *CYK* dan *Levenshtein* dapat dilihat pada Gambar 3.2.



Gambar 3.2 Alur Proses Perbaikan Pola Kalimat

3.1.4.1 Proses Pemecahan Kalimat Menjadi Kata

Pada tahap ini kalimat-kalimat yang sudah terbentuk akan dipisahkan berdasarkan spasi. Setiap kata dikelompokkan sesuai dengan kamus yang diambil berdasarkan aturan dari POS Tag. Pengelompokkan kata yang terbentuk dapat dilihat pada tabel 3.7 untuk kalimat " *They study English at schol* ". Contoh kalimat tersebut terdapat kata yang salah.

Tabel 3.7 Pengelompokan Kata

Kata	Tag
They	PP1
study	VB
English	NN
at	IN
schol	X

3.1.4.2 Pemeriksaan dan Perbaikan Kata dengan Algoritma Levenshtein

Pada tabel 3.7 terdapat kata yang diberi *tag* X. Pada tahap ini kata yang diberi tag X akan diperbaiki dengan cara pencocokan kata dengan kamus. Setiap kata yang keliru dibandingkan dengan setiap kata yang ada pada kamus kemudian dicari nilai *edit distance* yang terkecil dengan algoritma Levenshtein. Langkah-langkah perbaikannya sebagai berikut:

1. Pembentukan Tabel Dua Dimensi

Pada tahap ini kata "schol" akan dibandingkan dengan kata yang ada pada kamus. Contoh kata yang diambil dari kamus untuk dibandingkan yaitu "school". Selanjutnya dibuat tabel dua dimensi seperti tabel 3.8.

Tabel 3.8 Pembentukan tabel dua dimensi Levenshtein

		S	C	H	O	L
	0	1	2	3	4	5
S	1					
C	2					
H	3					
O	4					
O	5					
L	6					

3. Membandingkan Setiap Karakter

Tahap selanjutnya membandingkan setiap karakter pada tabel. Karakter pertama pada *string* pertama dibandingkan dengan karakter pertama pada *string* kedua. Apabila karakter yang dibandingkan sama, pengisian pada blok[1,1] tinggal menurunkan angka yang ada pada blok[0,0] yaitu angka 0. Jika karakter yang dibandingkan tidak sama, angka yang ada pada blok di kiri, kiri atas, dan atas blok yang akan disi, ditambahkan dengan angka 1 (penambahan bersifat sementara tidak permanen sehingga tidak mengubah angka yang disekitarnya) kemudian dicari jumlah mana yang paling minimum. Jumlah minimum dari penambahan dimasukkan ke blok yang akan diisi. Hasil dari perbandingan karakter dapat dilihat pada tabel 3.9.

Tabel 3.9 Hasil tabel dua dimensi Levenshtein

		S	C	H	O	L
	0	1	2	3	4	5
S	1	0	1	2	3	4
C	2	1	0	1	2	3
H	3	2	1	0	1	2
O	4	3	2	1	0	1
O	5	4	3	2	1	1
L	6	5	4	3	2	1

Pada baris dan kolom terakhir adalah jumlah edit distance dari perbandingan kata “schol” dengan “school” adalah 1. Pada tabel 3.10 ditunjukkan beberapa kata yang huruf awalnya sama dengan kata “school”.

Tabel 3.10 Hasil Perbandingan Kata

Hasil Pencarian Levenshtein	Nilai Edit Distance
school	1
schoolar	3
schooling	4

Kata dengan edit distance yang terkecil akan dijadikan rekomendasi perbaikan oleh aplikasi dan dipilih oleh user. Setelah melakukan perbaikan kata akan dilakukan kembali pengelompokkan kata untuk mengetahui tag pada kata yang telah diperbaiki.

Tabel 3.11 Pengelompokan kata setelah diperbaiki

Kata	Tag
They	PP1
study	VB
English	NN
at	IN
school	NN

3.1.4.3 Pengenalan Pola Kalimat dengan Algoritma CYK

Pada tahap ini dilakukan pengecekkan kalimat yang diperiksa sesuai dengan aturan produksi atau tidak. Pemeriksaan pola kalimat menggunakan algoritma CYK. Kalimat yang akan diperiksa yaitu " They study English at school" .

Tabel 3.12 Pembentukan tabel pola kali kalimat dengan algoritma CYK

	they	study	english	at	school
			i →		
	1	2	3	4	5
1					
2					
3					
4					
5					

Tabel di atas digunakan untuk mempermudah dalam menyelesaikan persoalan, i akan menyatakan kolom, j akan menyatakan baris. Baris pertama pada tabel akan diisi sebagai berikut:

- Untuk V11, periksa variable yang bisa menurunkan ‘they’, dari PP1 → they . V11 = [PP1]
- Untuk V21, periksa variable yang bisa menurunkan ‘study’, dari VB → study. V21 = [VB]
- Untuk V31, periksa variable yang bisa menurunkan ‘english’, dari NNP → english dan NN → english. V31 = [NNP, NN]
- Untuk V41, periksa variable yang bisa menurunkan ‘at’, dari IN → at. V41= [IN]
- Untuk V51, periksa variable yang bisa menurunkan ‘school’, dari NN → school. V51 = [NN]

Hasil dari pengisian dari baris pertama dapat dilihat pada tabel 3.13.

Tabel 3.13 Hasil pola kalimat baris pertama algoritma CYK

	they	study	english	at	school
			i →		
	1	2	3	4	5
1	PP1	VB	NNP, NN	IN	NN
2					
3					
4					
5					

Selanjutnya akan diisi baris ke 2 sebagai berikut:

Untuk k=1

- Untuk V12, periksa Vik – Vi+k, j-k, berarti V11 – V21, yaitu PP1 – VB, variabel yang bisa menurunkan PP1 VB adalah S, maka V12 = [S]
- Untuk V22, periksa Vik – Vi+k, j-k, berarti V21 – V31 yaitu VB – NNP, NN, variabel yang bisa menurunkan VB NNP atau VB NN adalah PRE1 maka V22 = [PRE1]

- Untuk V32, periksa Vik – Vi+k, j-k, berarti V31 – V41 yaitu NNP, NN – IN, variabel yang bisa menurunkan NNP IN atau NN IN adalah tidak ada.
- Untuk V42, periksa Vik – Vi+k, j-k, berarti V41 – V51, yaitu IN-NN, variabel yang menurunkan IN NN adalah CONJUNCTION, maka V42 = [CONJUCTION]

Hasil dari pengisian dari baris kedua dapat dilihat pada tabel berikut.

Tabel 3.14 Hasil pola kalimat baris kedua algoritma CYK

	they	*study*	*english*	*at*	*school*
			i →		
	1	2	3	4	5
1	PP1	VB	NNP, NN	IN	NN
2	S	PRE1	-	CONJUCTION	
3					
4					
5					

Pada baris ke 3 (k =1 sampai 2):

- Untuk V13, periksa Vik – Vi+k, j-k , berarti V11- V22 & V12 – V31, yaitu PP1 – PRE1 & S – NNP , NN, variabel yang bisa menurunkan PP1 PRE1, S NNP, atau S NN adalah S, maka V13 = [S]
- Untuk V23, periksa Vik – Vi+k, j-k , berarti V21- V32 & V22 – V41, yaitu VB - - & PRE1 - IN, variabel yang bisa menurunkan PRE1 IN adalah tidak ada.
- Untuk V33, periksa Vik – Vi+k, j-k , berarti V31- V42 & V32 – V51, yaitu NNP, NN – CONJUNCTION & - - NN, variabel yang bisa menurunkan NNP CONJUNTION atau NN CONJUNCTION adalah S, maka V33 [S]

Hasil dari pengisian dari baris ketiga dapat dilihat pada tabel berikut.

Tabel 3.15 Hasil pola kalimat baris ketiga algoritma CYK

	they	*study*	*english*	*at*	*school*
			i →		
	1	2	3	4	5
1	PP1	VB	NNP, NN	IN	NN
2	S	PRE1	-	CONJUNTION	
3	S	-	S		
4					
5					

Pada baris ke 4 (k =1 sampai 3):

- Untuk V14, periksa Vik – Vi+k, j-k, berarti V11 - V23 & V12 – V32 & V13 – V41, yaitu PP1 - - & S - & S – IN, variable yang bisa menurunkan S IN adalah tidak ada
- Untuk V24, periksa Vik – Vi+k, j-k, berarti V21 – V33 & V22 – V42 & V23 – V51, yaitu VB – S & PRE1 - CONJUNCTION & - NN, variable yang bisa menurunkan VB S, atau PRE1 CONJUNTION adalah tidak ada

Hasil dari pengisian dari baris keempat dapat dilihat pada tabel berikut.

Tabel 3.16 Hasil pola kalimat baris keempat algoritma CYK

	they	*study*	*english*	*at*	*school*
			i →		
	1	2	3	4	5
1	PP1	VB	NNP, NN	IN	NN
2	S	PRE1	-	CONJUNCTION	
3	S	-	S		
4	-	-			
5					

Pada baris 5 (k =1 sampai 4)

- Untuk V15, periksa Vik – Vi+k, j-k, berarti V11 – V24 & V12 – V33 & V13 – V42 & V14 – V51 yaitu PP1 - - & S – S & S – CONJUNCTION & - - NN, variable yang bisa menurunkan S S, atau S CONJUNCTION adalah S, maka V15 = [S]

Hasil dari pengisian dari baris kelima dapat dilihat pada berikut.

Tabel 3.17 Hasil pola kalimat baris kelima algoritma CYK

	they	*study*	*english*	*at*	*scho ol*
			i →		
	1	2	3	4	5
1	PP1	VB	NNP, NN	IN	NN
2	S	PRE1	-	CONJUNCTION	
3	S	-	S		
4	-	-			
5	S				

Dapat dilihat dari tabel yang telah dibuat string " *they study english at school*" termasuk ke dalam aturan produksi yang sudah dibuat karena baris dan kolom terakhir tidak menghasilkan tanda "-" yang artinya string tersebut belum dapat dikenali oleh algoritma CYK.

3.1.4.4 Perbaikan Pola Menggunakan Algoritma *Levenshtein*

Pada tahap ini dilakukan perbaikan terhadap pola kalimat. Perbaikan dilakukan agar dapat diketahui pola yang benar. Sebelum memperbaiki kalimat, *rule* dari sebuah kalimat harus diketahui terlebih dahulu. *Rule* ini diambil dari masing-masing *rule* yang ada pada *tenses* khususnya *common tenses* yang telah dibuat aturan produksinya. Hasil perbaikan pola kalimat disusun menjadi *rule* untuk dilakukan pencocokan pola kalimat. Algoritma ini akan mencari pola yang paling mendekati dengan pola yang keliru. Misalnya untuk kalimat " *they study English school at*". Langkah-langkah yang dilakukan dalam perbaikan pola yaitu:

1. Pembuatan tabel dua dimensi

   Pembuatan tabel dua dimensi sesuai dengan jumlah karakter *string*. Pola kalimat yang keliru akan dilakukan pencocokkan dengan kumpulan pola kalimat yang menjadi *rule* untuk perbaikan. Pembentukan tabel dapat dilihat pada gambar berikut.

Tabel 3.18 Pembentukan Tabel Dua Dimensi Levenshtein untuk Pola

		P	P	1		V	B		N	N		N	N		I	N
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P	1															
P	2															
1	3															
	4															
V	5															
B	6															
	7															
N	8															
N	9															
	10															
I	11															
N	12															
	13															
N	14															
N	15															

2. Membandingkan setiap karakter

Tabel 3.19 Hasil Perbaikan Pola Kalimat

		P	P	1		V	B		N	N		N	N		I	N
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
P	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12
	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11
V	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10
B	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9
	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8
N	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7
N	9	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6
	10	9	8	7	6	5	4	3	2	1	0	1	2	3	4	5
I	11	10	9	8	7	6	5	4	3	2	1	1	1	2	3	4
N	12	11	10	9	8	7	6	5	4	3	2	2	1	2	3	4
	13	12	11	10	9	8	7	6	5	4	3	3	2	1	2	3
N	14	13	12	11	10	9	8	7	6	5	4	4	3	2	3	4
N	15	14	13	12	11	10	9	8	7	6	5	5	4	3	3	2

Pada tabel 3.24 nilai *edit distance* dari perbandingan seluruh karakter adalah 2.. Pada tabel berikut ditunjukkan beberapa pola kalimat yang menjadi perbandingan.

Tabel 3.20 Hasil Perbandingan Pola

Hasil Pencarian Levenshtein	Nilai Edit Distance
PP1 VB NN IN NN	2
PP2 VBZ NN IN NN	3
PP1 VB NOT VB NN	4

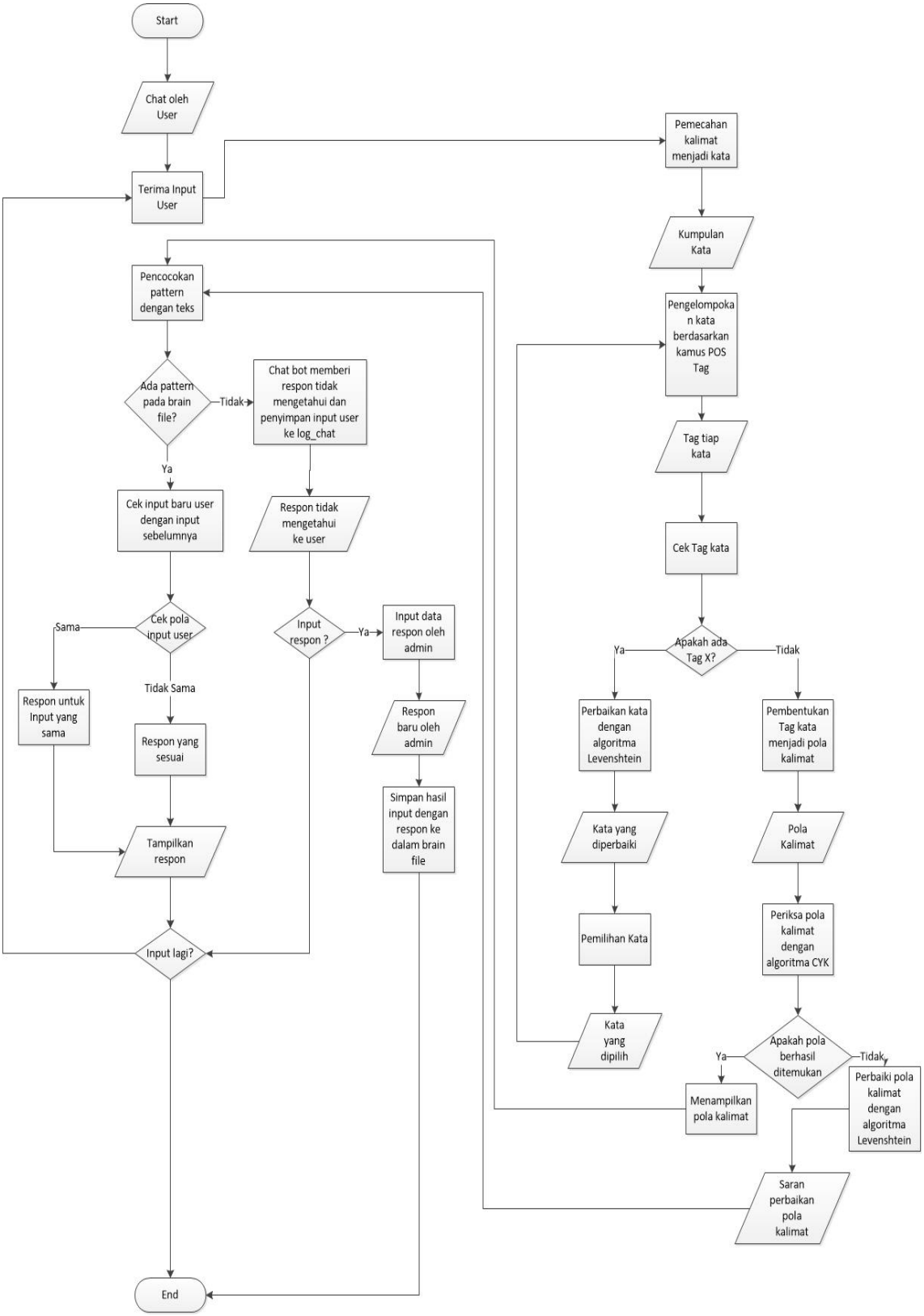
Edit distance dengan nilai terendah akan menjadi rekomendasi perbaikan pola kalimat.

3.1.5 Model Kerja *Chat Bot*

Pada bagian ini merupakan gambaran 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 tidak mengetahui apa yang ditanyakan pengguna, untuk mengalihkan topik pembicaraan.

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



Gambar 3.3. Alur Proses Chat Bot

3.1.5.1 Analisis Pencocokan *Pattern*

Berikut ini merupakan contoh bagaimana sistem berjalan pada aplikasi *chat bot* ini. Dimana isi *brain file* memiliki 3 *pattern* dan teks dari pengguna adalah ”*WHO IS F1 DRIVER FROM INDONESIA* ?”.

Diketahui :

Teks yang diinputkan pengguna : ” *WHO IS F1 DRIVER FROM INDONESIA*”

Daftar *Pattern* dapat dilihat pada table 3.25.

Tabel 3.21 Daftar Pattern Yang Ada Dalam Database

NO	*PATTERN*	*RESPON*
1	*WHAT IS F1*	*Formula One (also Formula 1 or F1) is the highest class of single-seat auto racing that is sanctioned by the Fédération Internationale de l'Automobile (FIA)*
2	*WHO IS F1 DRIVER FROM INDONESIA*	*Rio Haryanto (born 22 January 1993) is an Indonesian Formula One driver for Manor. He is Indonesia's first F1 driver*
3	*WHO IS F1 CHAMPIONSHIP 2015*	*Lewis Hamilton was the defending Drivers' Champion after securing his second title at the 2014 Abu Dhabi Grand Prix*

Sistem akan membandingkan semua *pattern* yang ada dalam *database* satu-persatu dengan teks pengguna, pencocokan *pattern* dengan teks dilakukan dengan fungsi LIKE pada *databas*e.

3.1.6 Analisis Kebutuhan Non Fungsional

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 Kebutuhan 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 :

Tabel 3.22 Kebutuhan Perangkat Keras

Pengguna	Pengembang
Processor Pentium 4 1 Ghz	Processor Core i5 2.26Ghz
Harddisk 40 GB	Harddisk 500 GB
VGA 128 MB	VGA 512 MB
RAM 512 MB	RAM 4 GB
Monitor	Monitor
Mouse	Mouse
Keyboard	Keyboard
Lan Card 10/100 Mbps	Lan Card 10/100 Mbps

3.1.6.2 Analisis Kebutuhan Perangkat Lunak

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

Tabel 3.23 Kebutuhan Perangkat Lunak

Pengguna	Pengembang
Windows 7 sebagai Sistem Operasi	Windows 7 sebagai Sistem Operasi
Mozilla Firefox / Chrome sebagai web browser	WAMP Server sebagai aplikasi web server
	Sublime Text sebagai editor php
	MySQL untuk DBMS
	Mozilla Firefox sebagai web browser

3.1.6.3 Analisis Kebutuhan Pengguna

Untuk menjalankan aplikasi ini dibutuhkan spesifikasi *user* yang dapat menjalankan aplikasi ini, baik dalam sisi pengguna maupun admin.

Adapun spesifikasi *user* yang dibutuhkan dapat dilihat sebagai berikut.

Tabel 3.24 Kebutuhan Pengguna

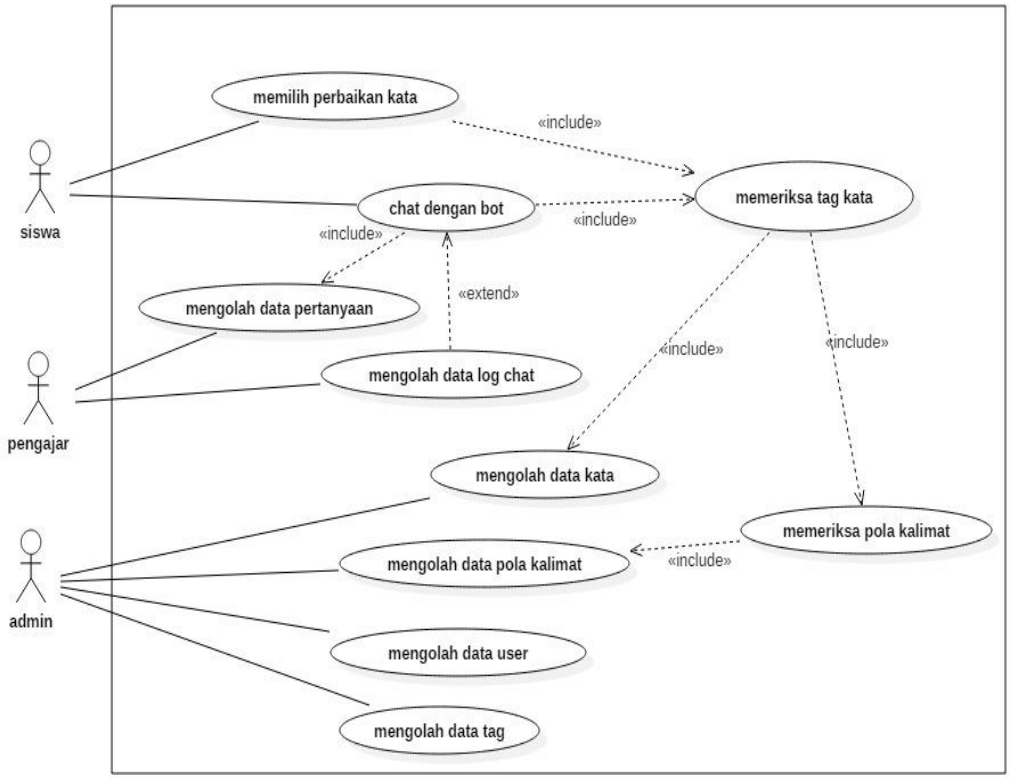
Kemampuan	Pengguna	Admin
Mampu menggunakan komputer	√	√
Mampu mengoperasikan internet	√	√
Mampu Mengoperasikan aplikasi	√	√
Mengetahui pengelolaan data(tambah, ubah, hapus)		√

3.1.7 Analisis Kebutuhan Fungsional

Kebutuhan fungsional meliputi beberapa diagram UML, diantranya: diagran use case, diagran activity, diagram class dan diagram sequence. Diagram tersebut dipergunakan sebagai desain rancangan untuk mempermudah proses pengembangan aplikasi khususnya untuk aplikasi yang dikembangkan menggunakan pemograman berorientasi objek.

3.1.7.1 *Use Case Diagram*

Use Case Diagram merupakan bagian tertinggi dari fungsionalitas yang dimiliki sistem yang akan menggambarkan bagaimana seseorang atau aktor akan menggunakan dan memanfaatkan sistem. Diagram ini juga mendeskripsikan apa yang akan dilakukan oleh sistem. *Use Case* terdiri dari tiga bagian yaitu identifikasi aktor, identifikasi *Use Case* dan skenario *Use Case.*



Gambar 3.4. Use Case Diagram

3.1.7.1.1 Definisi Aktor

Definisi aktor berfungsi untuk menjelaskan aktor yang terdapat pada *Use Case Diagram*. Definisi aktor dijelaskan pada Tabel 3.25

Tabel 3.25 Definisi Aktor

No	Aktor	Deskripsi
1	Admin	Merupakan aktor yang menggunakan fitur pengelolaan data user , data kata, dan data pola kalimat
2	Pengajar	Merupakan aktor yang menggunakan fitur pengelolaan data pertanyaan, dan data log chat
2	Siswa	Merupakan aktor yang menggunakan aplikasi chat dengan bot

3.1.7.1.2 Definisi *Use Case*

Definisi *Use Case* berfungsi untuk menjelaskan fungsi *Use Case* yang terdapat pada *Use Case Diagram*. Definisi *Use Case* diterangkan pada Tabel 3.26

Tabel 3.26 Definisi *Use Case*

No	Use Case	Deskripsi
1	Mengolah Data Kata	Proses untuk pengelolaan data kata dengan fungsi tambah, edit, dan hapus
2	Mengolah Data Pola Kalimat	Proses untuk pengelolaan data pola kalimat dengan fungsi tambah, edit, dan hapus
3	Mengolah Data User	Proses untuk pengelolaan data user dengan fungsi tambah, edit, dan hapus
4	Mengelolah Data Pertanyaan	Proses untuk pengelolaan data brainfile dengan fungsi tambah, edit, dan hapus
5	Mengolah Data Log Chat	Proses untuk pengelolaan data log chat dengan fungsi tambah, edit, dan hapus
6	Chat dengan Bot	Proses dimana siswa chat dengan bot kemudian bot akan membalasnya

7	Memeriksa Tag Kata	Proses pengelompokan kata berdasarkan tag-nya
8	Memilih Perbaikan Kata	Proses siswa memilih rekomendasi pembenaran kata yang salah
9	Memeriksa Pola Kalimat	Proses pemeriksaan pola kalimat yang diinputkan oleh siswa
10	Mengolah Data Tag	Proses untuk pengelolaan data tag dengan fungsi tambah, edit, dan hapus

3.1.7.1.3 Skenario *Use Case*

Skenario *Use Case* menggambarkan alur penggunaan sistem dimana setiap skenario digambarkan dari sudut pandang aktor, seseorang, atau piranti yang berinteraksi dengan perangkat lunak dalam berbagai cara.

Skenario *Use Case* mengolah data kata dapat dilihat pada tabel 3.27 .

Tabel 3.27 Skemario *Use Case* Mengolah Data Kata

Use Case Name	**Mengolah Data Kata**	
Related Requirements	SKPL – F - 01	
Goal In Context	Admin dapat mengelola data kata	
Preconditions	Admin telah berada di menu data kata	
Succesfull End Condition	Data kata yang diolah oleh admin berhasil tersimpan	
Failed End Condition	Admin tidak bisa melakukan proses mengelola data kata	
Primary Actors	Admin	
Secondary Actors	None	
Trigger	Admin memilih menu data kata	
Main Flow	Step	Action
	1	Sistem menampilkan tampilan menu data kata

	2	Admin mengelola data kata (tambah, ubah, atau hapus)
	3	Sistem melakukan aksi(tambah, ubah, atau hapus)
	4	Sistem menampilkan pesan berhasil pada tiap aksi(tambah, ubah, atau hapus)
Extensions	Step	Branching Action
	4.1	Sistem menampilkan pesan kesalahan pada tiap aksi(tambah, edit, atau hapus)

Skenario *Use Case* mengolah data pola kalimat dapat dilihat pada tabel 3.28 .

Tabel 3.28 Skemario *Use Case* Mengola Data Pola Kalimat

Use Case Name	**Mengolah Data Pola Kalimat**	
Related Requirements	SKPL – F - 02	
Goal In Context	Admin dapat mengelola data pola kalimat	
Preconditions	Admin telah berada di menu data pola kalimat	
Succesfull End Condition	Data pola kalimat yang diolah oleh admin berhasil tersimpan	
Failed End Condition	Admin tidak bisa melakukan proses mengelola data pola kalimat	
Primary Actors	Admin	
Secondary Actors	None	
Trigger	Admin memilih menu data pola kalimat	
Main Flow	Step	Action
	1	Sistem menampilkan tampilan menu data pola kalimat
	2	Admin mengelola data pola kalimat (tambah, ubah, atau hapus)

	3	Sistem melakukan aksi(tambah, ubah, atau hapus)
	4	Sistem menampilkan pesan berhasil pada tiap aksi(tambah, ubah, atau hapus)
Extensions	Step	Branching Action
	4.1	Sistem menampilkan pesan kesalahan pada tiap aksi(tambah, edit, atau hapus)

Skenario *Use Case* mengolah data user sistem dapat dilihat pada tabel 3.29.

Tabel 3.29 Skenario *Use Case* Data User

Use Case Name	**Mengolah Data User**	
Related Requirements	SKPL – F - 03	
Goal In Context	Admin dapat mengelola data user	
Preconditions	Admin telah berada di menu data user	
Succesfull End Condition	Admin dapat melakukan proses mengelola data user	
Failed End Condition	Admin tidak bisa melakukan proses mengelola data user	
Primary Actors	Admin	
Secondary Actors	None	
Trigger	Admin memilih menu data user	
Main Flow	Step	Action
	1	Sistem menampilkan tampilan menu data user
	2	Admin mengelola data user (tambah, ubah, atau hapus)

	3	Sistem melakukan aksi(tambah, ubah, atau hapus)
	4	Sistem menampilkan pesan berhasil pada tiap aksi(tambah, ubah, atau hapus)
Extensions	Step	Branching Action
	4.1	Sistem menampilkan pesan kesalahan pada tiap aksi(tambah, edit, atau hapus)

Skenario *Use Case* mengolah data pertanyaan dapat dilihat pada tabel 3.30

.

Tabel 3.30 Skenario *Use Case* Data Pertanyaan

Use Case Name	**Mengolah Data Pertanyaan**	
Related Requirements	SKPL – F - 04	
Goal In Context	Admin dapat mengelola data pertanyaan	
Preconditions	Admin telah berada di menu data pertanyaan	
Succesfull End Condition	Admin dapat melakukan proses mengelola data pertanyaan	
Failed End Condition	Admin tidak bisa melakukan proses mengelola data pertanyaan	
Primary Actors	Admin	
Secondary Actors	None	
Trigger	Admin memilih menu data pertanyaan	
Main Flow	Step	Action
	1	Sistem menampilkan tampilan menu data pertanyaan
	2	Admin mengelola data pertanyaan (tambah, ubah, atau hapus)

	3	Sistem melakukan aksi(tambah, ubah, atau hapus)
	4	Sistem menampilkan pesan berhasil pada tiap aksi(tambah, ubah, atau hapus)
Extensions	Step	Branching Action
	4.1	Sistem menampilkan pesan kesalahan pada tiap aksi(tambah, edit, atau hapus)

Skenario *Use Case* mengolah data log chat dapat dilihat pada tabel 3.31 .

Tabel 3.31 Skenario *Use Case* Data Pertanyaan

Use Case Name	**Mengolah Data Log Chat**	
Related Requirements	SKPL – F - 05	
Goal In Context	Admin dapat mengelola data log chat	
Preconditions	Admin telah berada di menu data log chat	
Succesfull End Condition	Admin dapat melakukan proses mengelola data log chat	
Failed End Condition	Admin tidak bisa melakukan proses mengelola data log chat	
Primary Actors	Admin	
Secondary Actors	None	
Trigger	Admin memilih menu data log chat	
Main Flow	Step	Action
	1	Sistem menampilkan tampilan menu data log chat
	2	Admin mengelola data log chat (tambah, ubah, atau hapus)
	3	Sistem melakukan aksi(tambah, ubah,

		atau hapus)
	4	Sistem menampilkan pesan berhasil pada tiap aksi(tambah, ubah, atau hapus)
Extensions	Step	Branching Action
	4.1	Sistem menampilkan pesan kesalahan pada tiap aksi(tambah, edit, atau hapus)

Skenario *Use Case* Chat dengan Bot dapat dilihat pada tabel 3.32 .

Tabel 3.32 Skenario *Use Case* Chat dengan Bot

Use Case Name	**Chat dengan Bot**	
Related Requirements	SKPL – F - 06	
Goal In Context	Siswa dapat melakukan chat	
Preconditions	Siswa telah menginputkan kalimat	
Succesfull End Condition	Siswa menerima balasan chat dari bot	
Failed End Condition	Siswa tidak menerima balasan chat dari bot	
Primary Actors	Siswa	
Secondary Actors	None	
Trigger	Siswa berada ditampilan chat	
Main Flow	Step	Action
	1	Sistem menampilkan tampilan chat
	2	Siswa menginputkan kalimat untuk chat
	3	Sistem akan memeriksa pattern yang ada di brainfile disesuaikan dengan chat siswa
	4	Sistem akan membalas chat siswa
Extensions	Step	Branching Action
	4.1	Sistem tidak dapat menemukan pattern

		di chat siswa dan menampilkan pesan chat tidak diketahui

Skenario *Use Case* Memeriksa Tag Kata sistem dapat dilihat pada tabel 3.33 .

Tabel 3.33 Skenario *Use Case* Memeriksa Tag Kata

Use Case Name	**Memeriksa Tag Kata**	
Related Requirements	SKPL – F - 07	
Goal In Context	Sistem dapat mengelompokkan kata sesuai tag	
Preconditions	Kalimat telah diinputkan	
Succesfull End Condition	Kata dikelompokkan sesuai tag	
Failed End Condition	Kata tidak dapat dikelompokkan sesuai tag	
Primary Actors	None	
Secondary Actors	None	
Trigger	Setelah siswa menginputkan kalimat chat dengan bot	
Main Flow	Step	Action
	1	Sistem mengelompokkan kata yang diinputkan siswa seusai tag yang ada pada data kata
	2	Sistem melakukan perbaikan kata sesuai dengan pada data kata dengan algoritma Levenshtein jika terdapat tag X pada kalimat
Extensions	Step	Branching Action

	2.1	Sistem tidak dapat melakukan perbaikan kata

Skenario *Use Case* Memilih Perbaikan Kata sistem dapat dilihat pada tabel 3.34 .

Tabel 3.34 Skenario *Use Case* Memilih Perbaikan Kata

Use Case Name	**Memilih Perbaikan Kata**	
Related Requirements	SKPL – F - 08	
Goal In Context	Sistem menampilakan rekomendasi kata yang sesuai dengan kalimat siswa	
Preconditions	Sistem telah memeriksa kata sesuai tag	
Succesfull End Condition	Menampilakan rekomendasi kata	
Failed End Condition	Tidak dapat menampilkan rekomendasi kata	
Primary Actors	Siswa	
Secondary Actors	None	
Trigger	Sistem telah menyesuaikan tag kata	
Main Flow	Step	Action
	1	Sistem menampilkan rekomendasi kata
	2	Siswa memilih rekomendasi kata
	3	Sistem mengelompokkan kata yang diinputkan siswa seusai tag yang ada pada data kata
	4	Sistem melakukan perbaikan kata sesuai dengan pada data kata dengan algoritma Levenshtein jika terdapat tag X pada kalimat

Extensions	Step	Branching Action
	1.1	Sistem tidak dapat menampilkan rekomendasi kata

Skenario *Use Case* Memeriksa Pola Kalimat sistem dapat dilihat pada tabel 3.35.

Tabel 3.35 Skenario *Use Case* Memeriksa Pola Kalimat

Use Case Name	**Memeriksa Pola Kalimat**	
Related Requirements	SKPL – F - 09	
Goal In Context	Sistem rekomendasi pola kalimat yang benar	
Preconditions	Tag kata sudah sesuai dengan data kata	
Succesfull End Condition	Menampilakan rekomendasi pola kalimat yang benar	
Failed End Condition	Tidak dapat menampilkan rekomendasi pola kalimat	
Primary Actors	None	
Secondary Actors	None	
Trigger	Sistem telah memeriksa tag kata	
Main Flow	Step	Action
	1	Sistem memeriksa pola kalimat dengan algoritma Cocke Younger Kasami
	2	Sistem menampilkan rekomendasi pola kalimat yang benar jika pola kalimat yang diinputkan salah dengan algoritma Levenshtein
Extensions	Step	Branching Action
	2.1	Sistem tidak dapat menampilkan rekomendasi pola kalimat

Skenario *Use Case* mengolah data tag dapat dilihat pada tabel 3.36 .

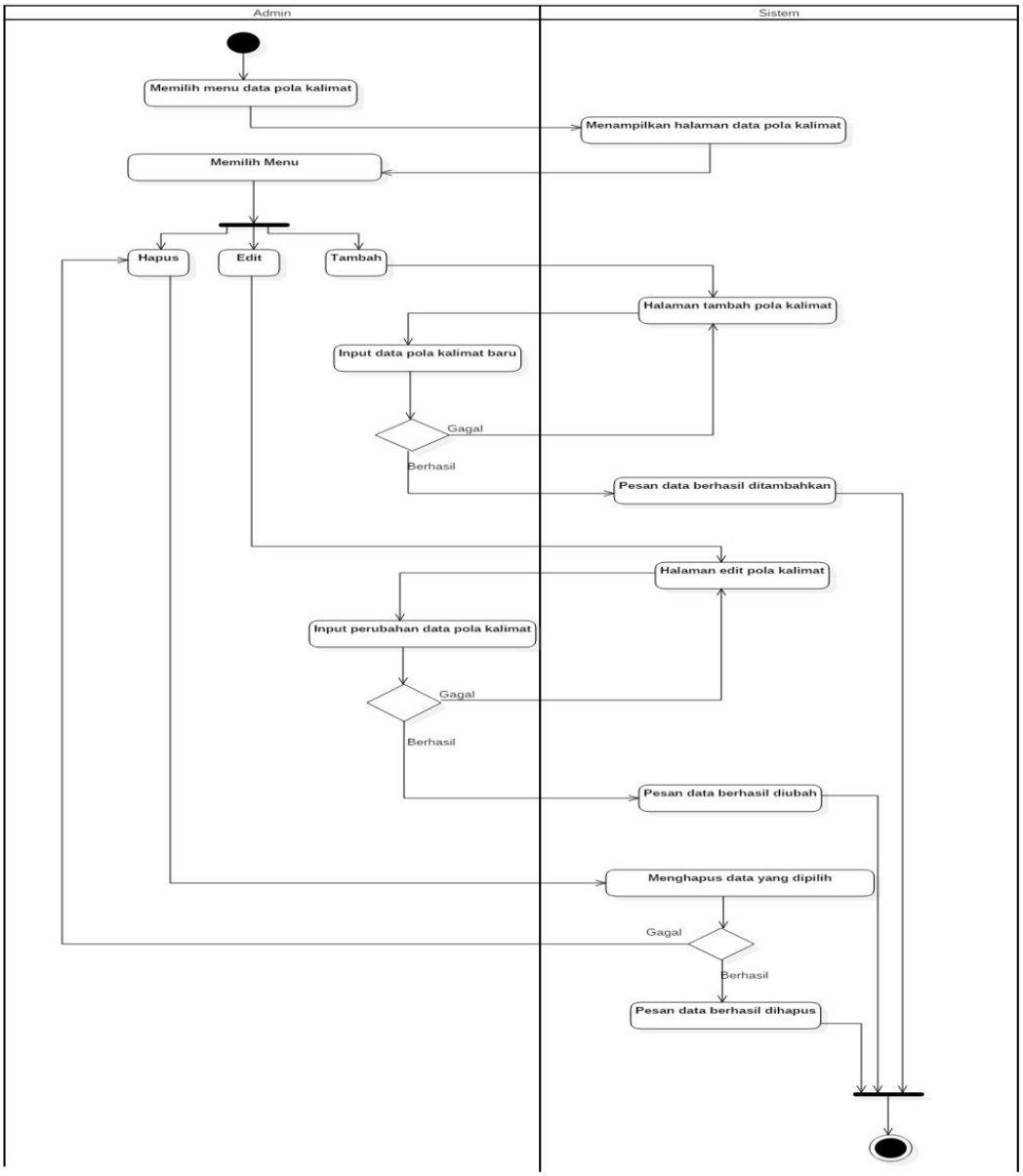
Tabel 3.36 Skemario *Use Case* Mengolah Data Tag

Use Case Name	**Mengolah Data Tag**	
Related Requirements	SKPL – F - 10	
Goal In Context	Admin dapat mengelola data tag	
Preconditions	Admin telah berada di menu data tag	
Succesfull End Condition	Data kata yang diolah oleh admin berhasil tersimpan	
Failed End Condition	Admin tidak bisa melakukan proses mengelola data kata	
Primary Actors	Admin	
Secondary Actors	None	
Trigger	Admin memilih menu data tag	
Main Flow	Step	Action
	1	Sistem menampilkan tampilan menu data tag
	2	Admin mengelola data tag (tambah, ubah, atau hapus)
	3	Sistem melakukan aksi(tambah, ubah, atau hapus)
	4	Sistem menampilkan pesan berhasil pada tiap aksi(tambah, ubah, atau hapus)
Extensions	Step	Branching Action
	4.1	Sistem menampilkan pesan kesalahan pada tiap aksi(tambah, edit, atau hapus)

3.1.7.2 *Actvity Diagram*

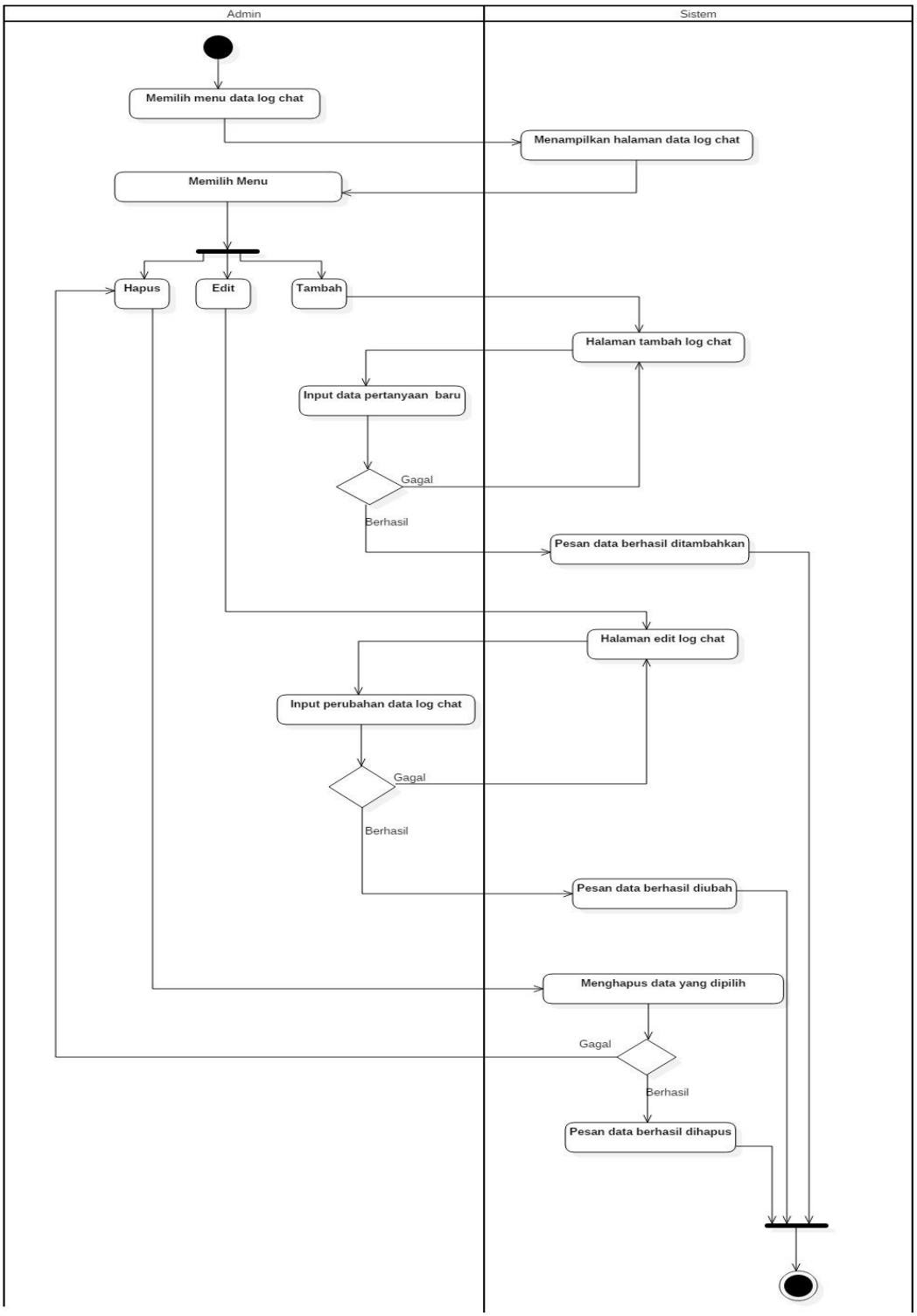
Diagram aktivitas atau *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Penggambaran *activity* diagram memiliki kemiripan dengan *flowchart diagram. Activity diagram* memodelkan *event-event* yang terjadi pada *Use Case* dan digunakan untuk pemodelan aspek dinamis dari sistem.

1. *Activity diagram* mengolah data pola kalimat



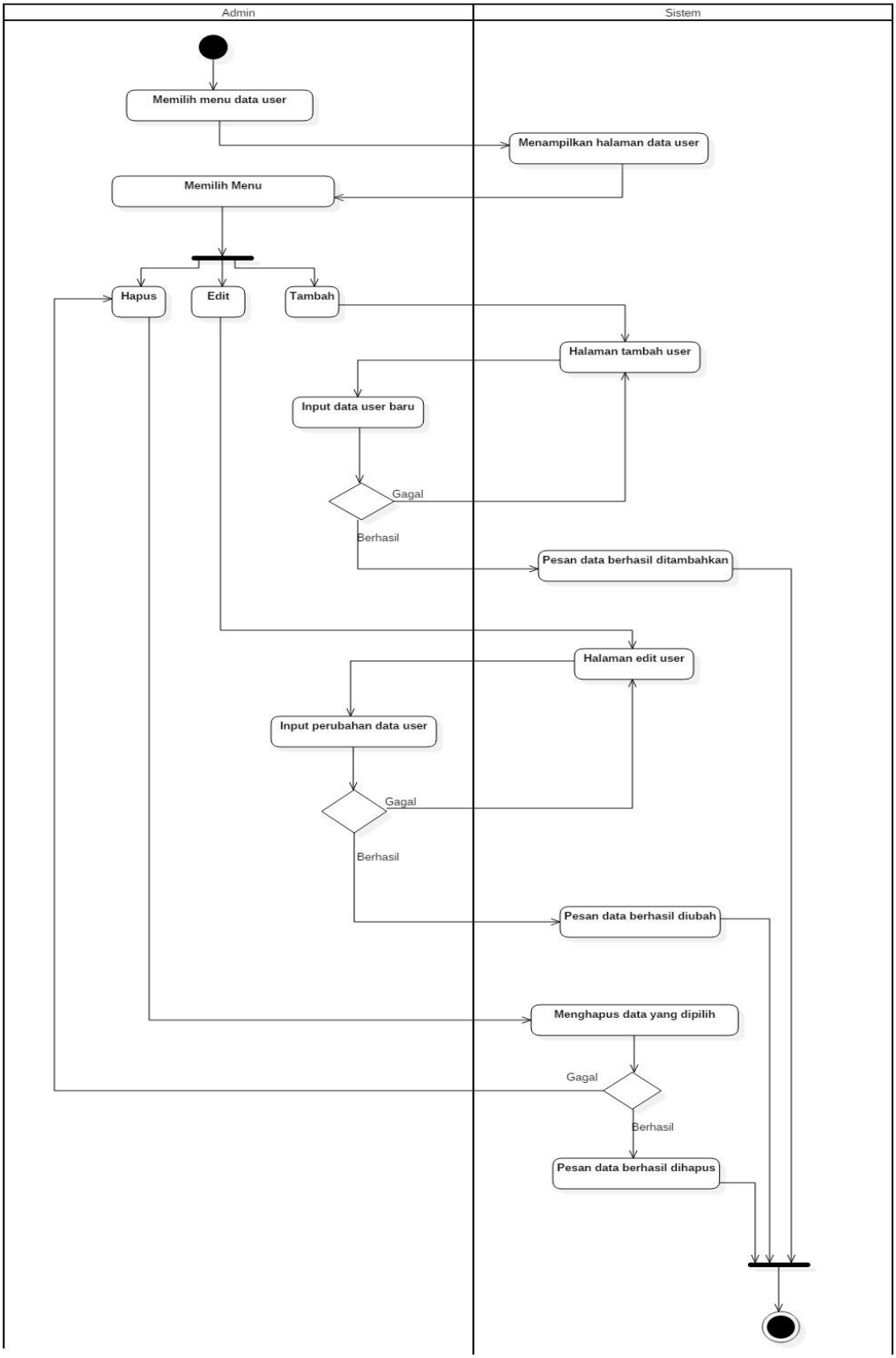
Gambar 3.5. Activity Diagram mengolah data pola kalimat

2. Activity diagram mengolah data log chat



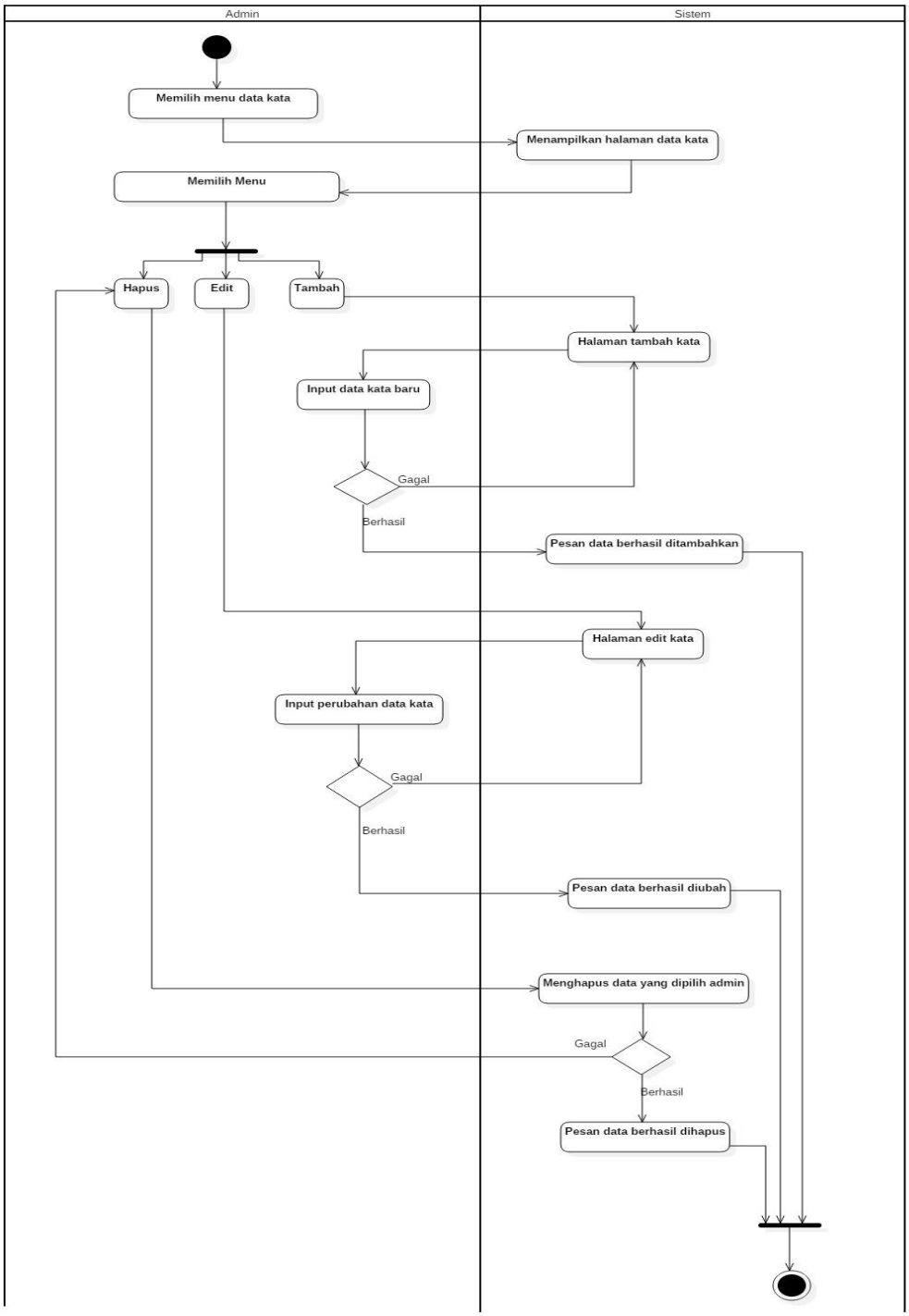
Gambar 3.6. Activity diagram mengolah data log chat

3. *Activity diagram* mengolah data user



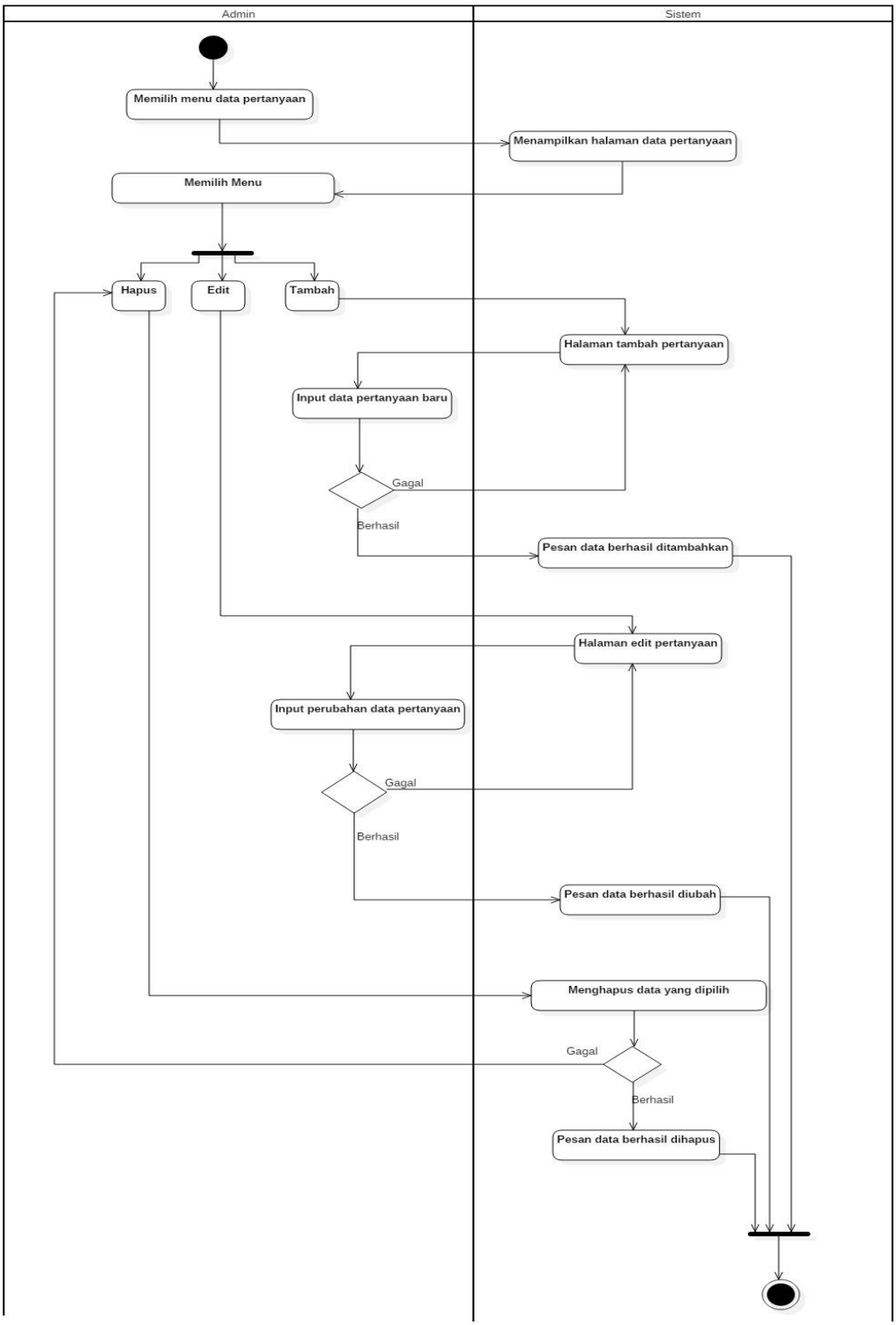
Gambar 3.7. Activity diagram mengolah data user

4. *Activity diagram* data kata



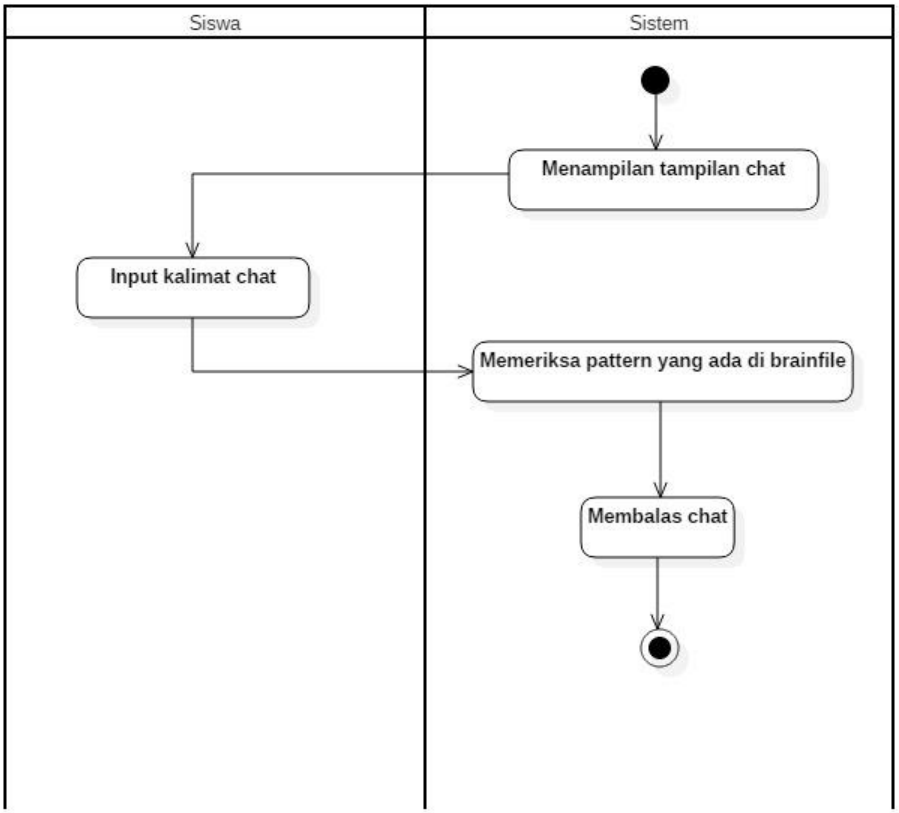
Gambar 3.8. Activity diagram mengolah data kata

5. *Activity diagram* mengolah data pertanyaan



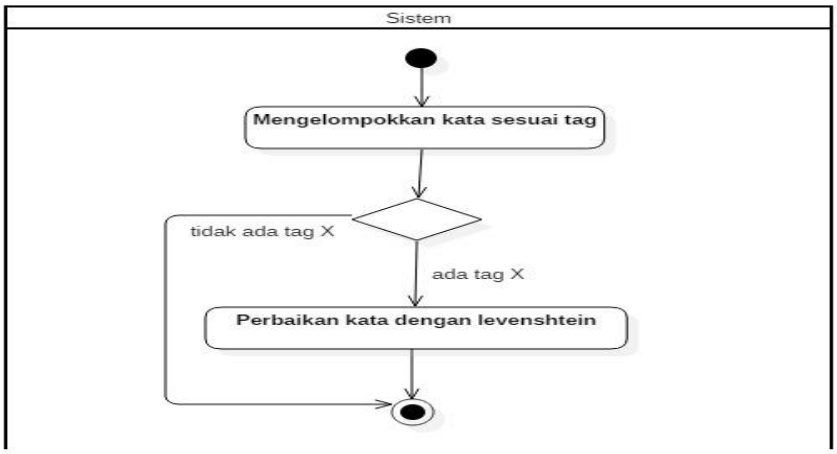
Gambar 3.9. Activity diagram mengolah data pertanyaan

6. *Activity diagram* chat dengan bot



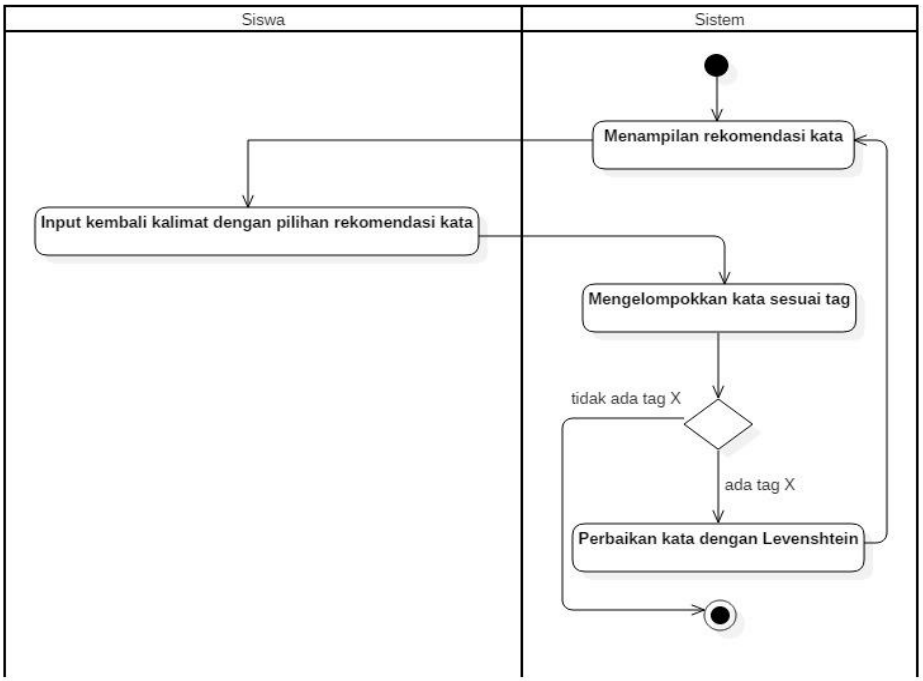
Gambar 3.10. ***Activity diagram*** **Chat dengan Bot**

7. *Activity diagram* memeriksa tag kata



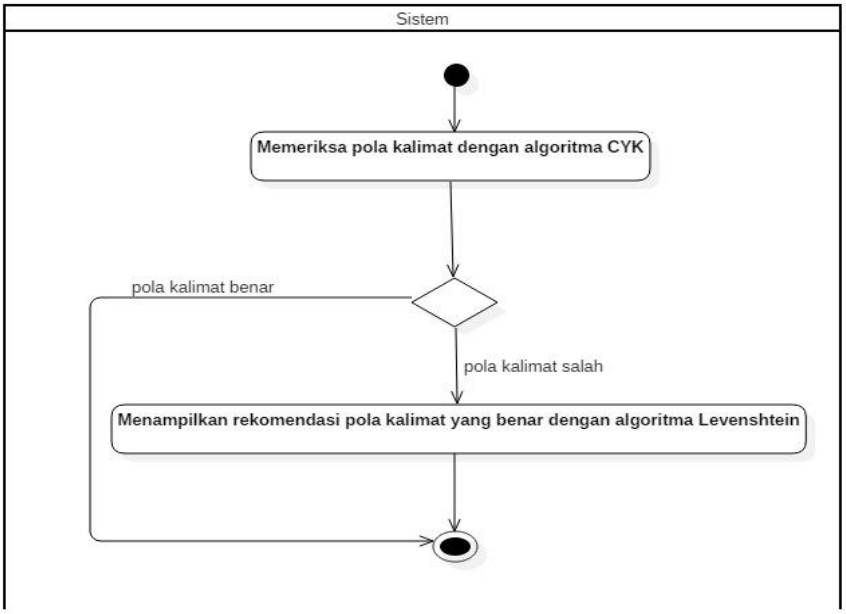
Gambar 3.11. ***Activity diagram*** **Memeriksa Tag Kata**

8. *Activity diagram* memilih perbaikan kata



Gambar 3.12. ***Activity diagram*** **Memilih Perbaikan Kata**

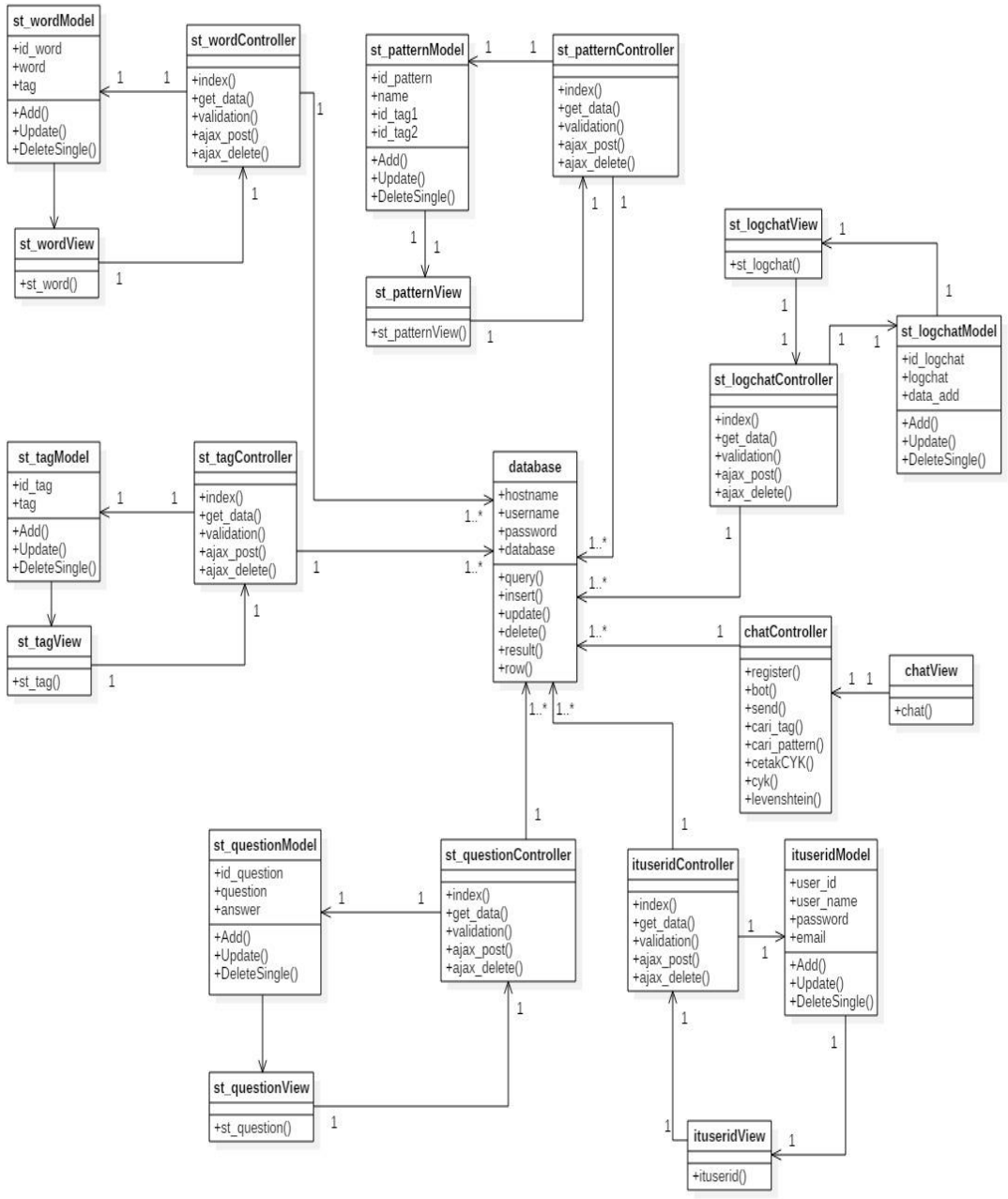
9. *Activity diagram* memeriksa pola kalimat



Gambar 3.13. ***Activity diagram*** **Memeriksa Pola Kalimat**

3.1.7.3 Class Diagram

Class Diagram digunakan untuk menggambarkan secara abstrak struktur dari aplikasi yang akan dibangun, *class-class* yang terlibat, serta hubungan antar *class* untuk saling berkomunikasi satu sama lain. Struktur *class* dari aplikasi yang akan dibangun dapat dilihat sebagai berikut.

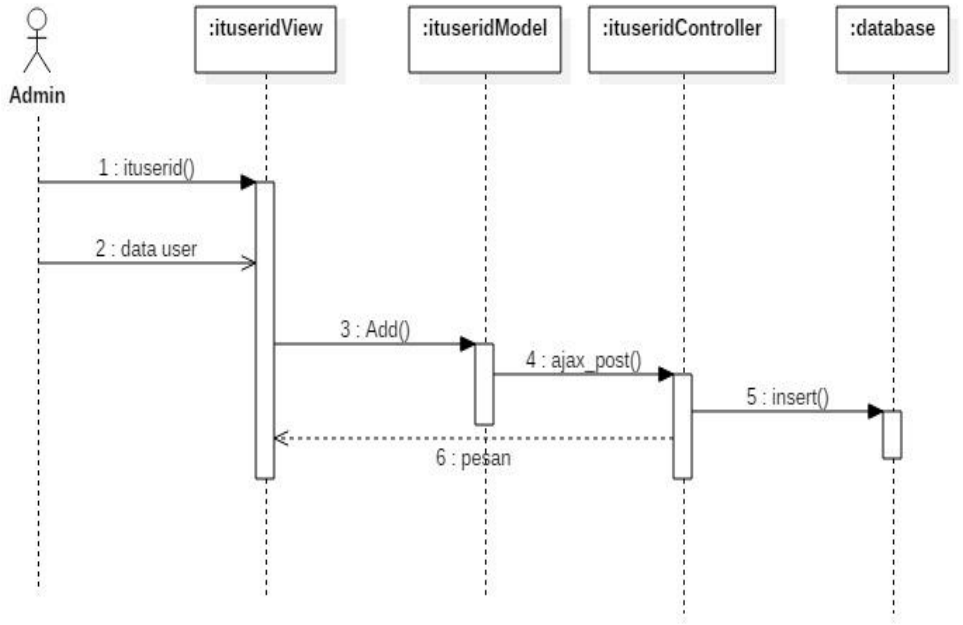


Gambar 3.14. Class Diagram

3.1.7.4 *Sequence Diagram*

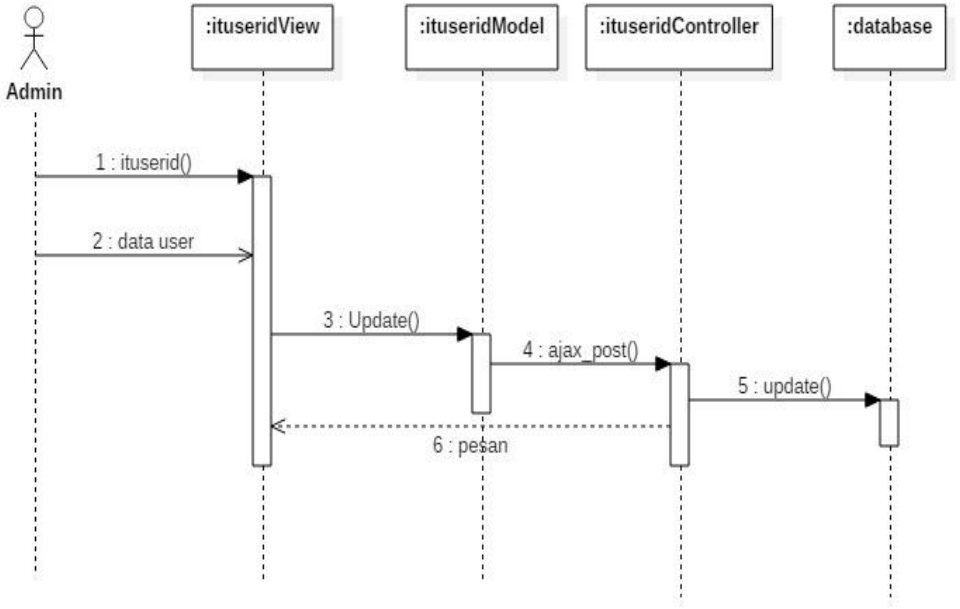
Sequence Diagram menjabarkan urutan aktifitas yang terjadi di dalam sistem. Diagram ini menunjukkan pengiriman pesan-pesan yang melewati objek yang terlibat di dalam sistem serta tahapan langkah yang harus dilalui untuk mencapai *use case* tertentu.

1. Sequence Diagram Tambah User



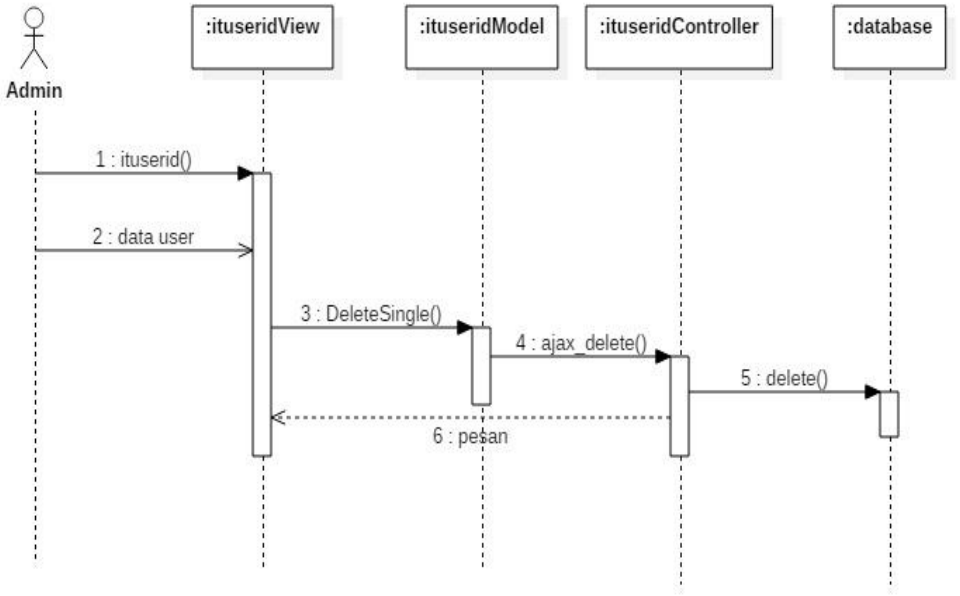
Gambar 3.15. Sequence Diagram Tambah User

2. Sequence Diagram Edit User



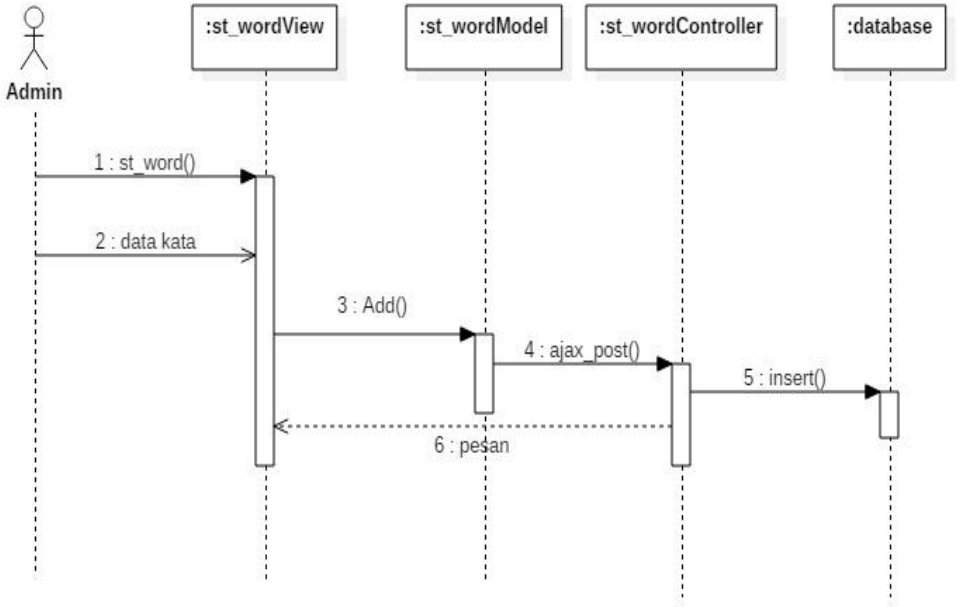
Gambar 3.16. Sequence Diagram Edit User

3. Sequence Diagram Hapus User



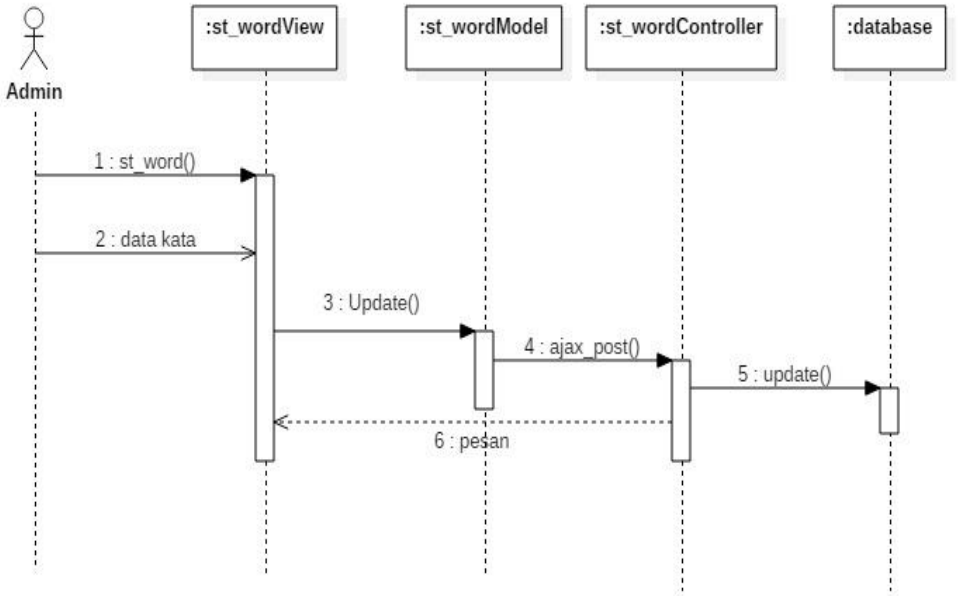
Gambar 3.17. Sequence Diagram Hapus User

4. Sequence Diagram Tambah Kata



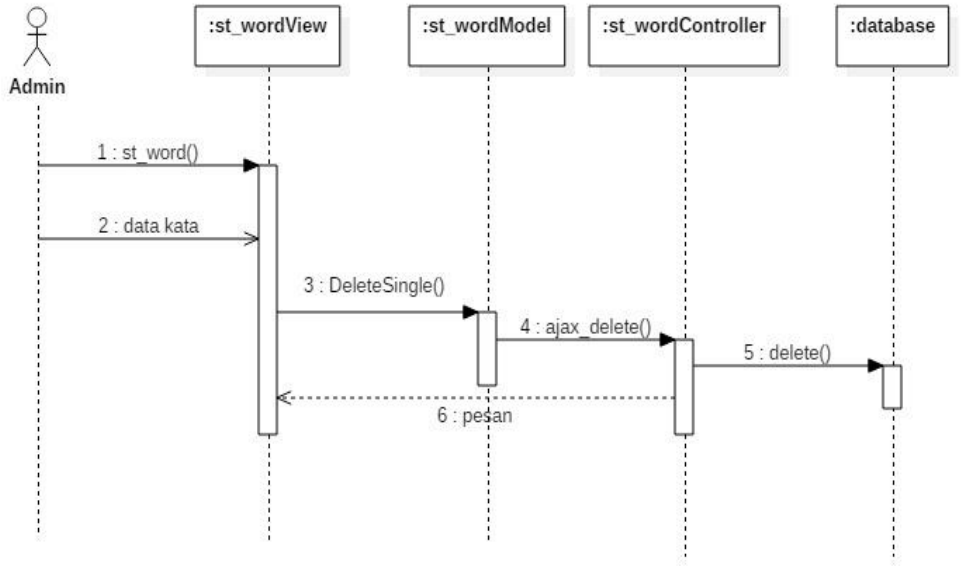
Gambar 3.18. Sequence Diagram Tambah Kata

5. Sequence Diagram Edit Kata



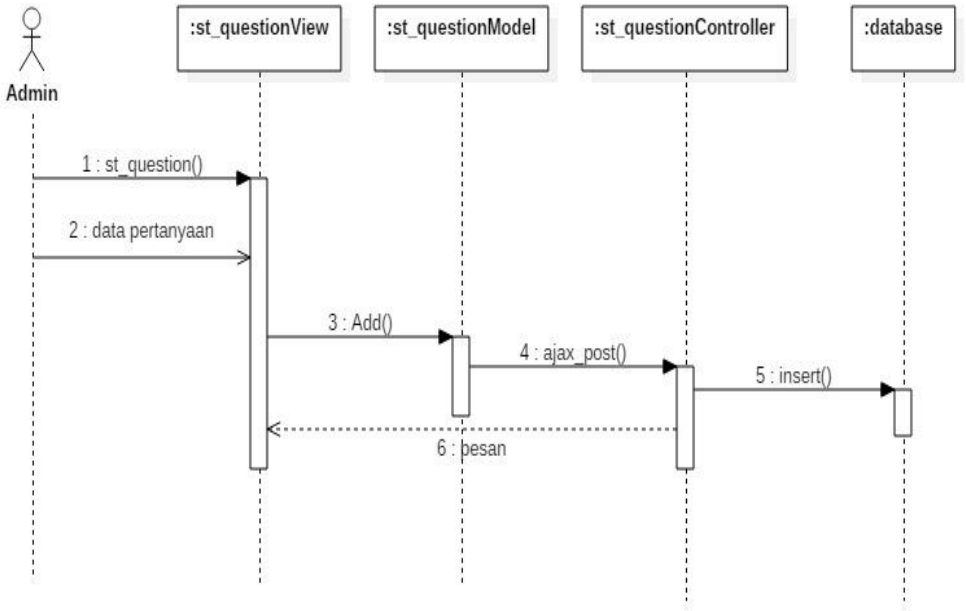
Gambar 3.19. Sequence Diagram Edit Kata

6. Sequence Diagram Hapus Kata



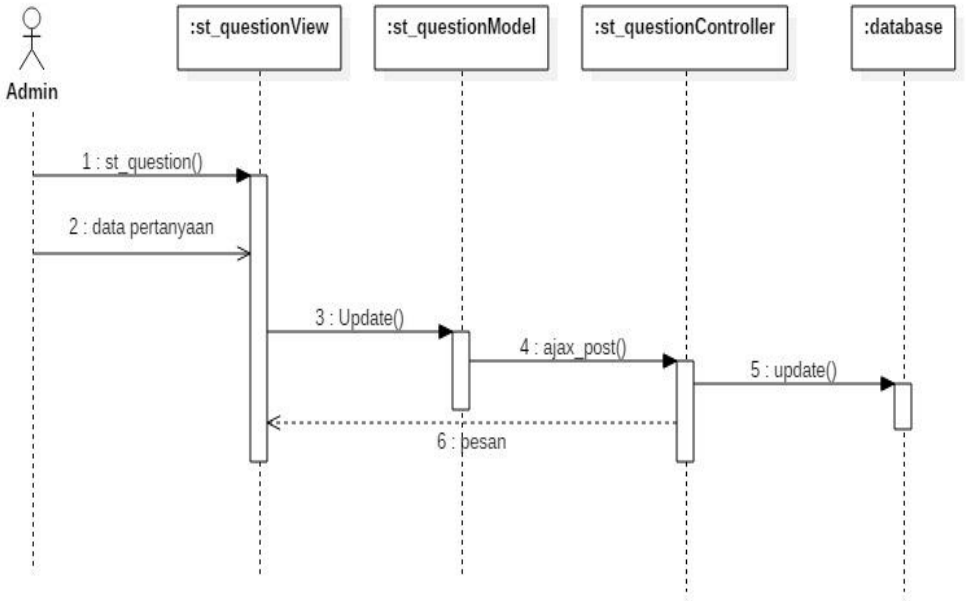
Gambar 3.20. Sequence Diagram Hapus Kata

7. Sequence Diagram Tambah Pertanyaan



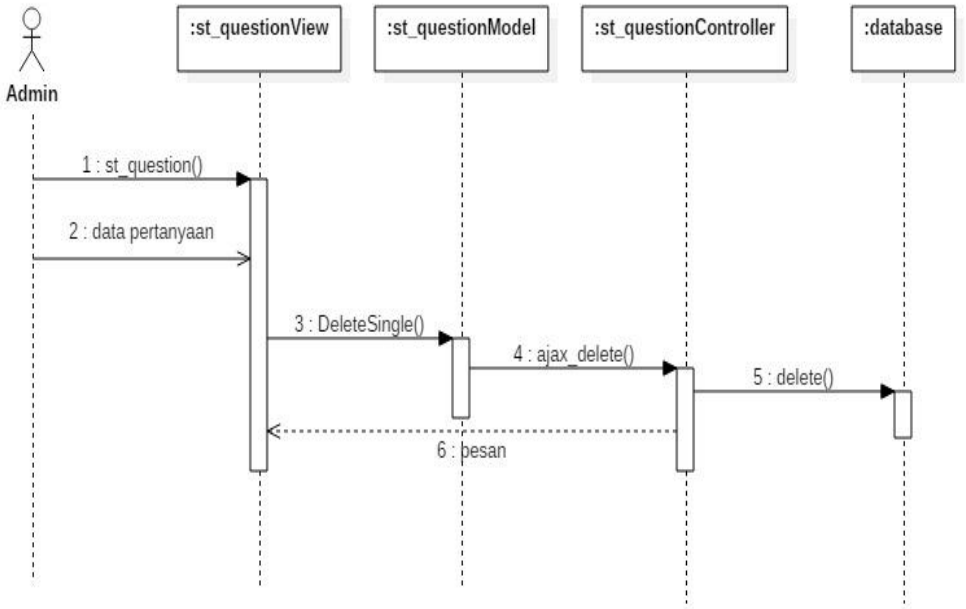
Gambar 3.21. Sequence Diagram Tambah Pertanyaan

8. Sequence Diagram Edit Pertanyaan



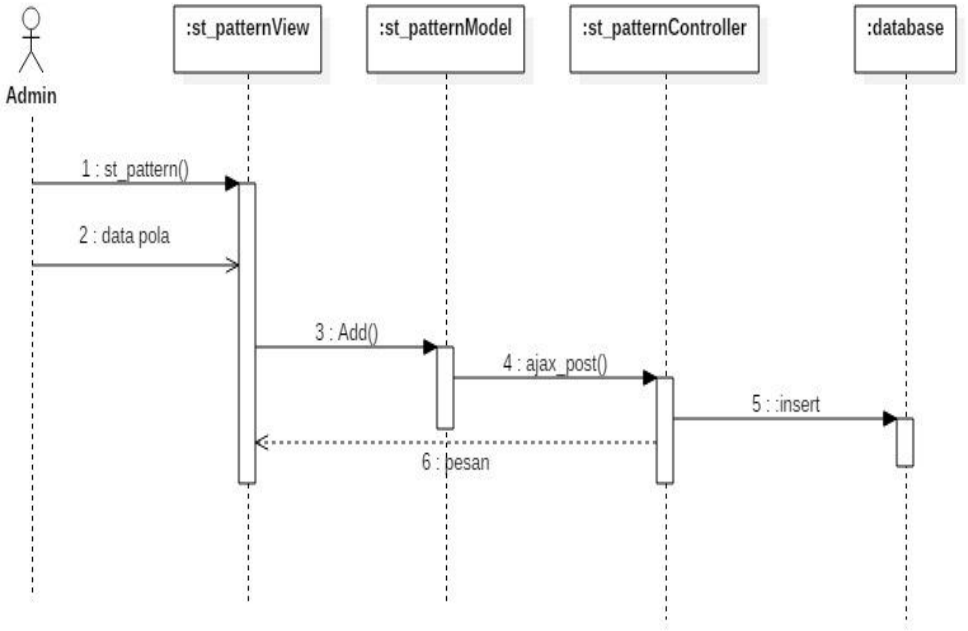
Gambar 3.22. Sequence Diagram Edit Pertanyaan

9. Sequence Diagram Hapus Pertanyaan



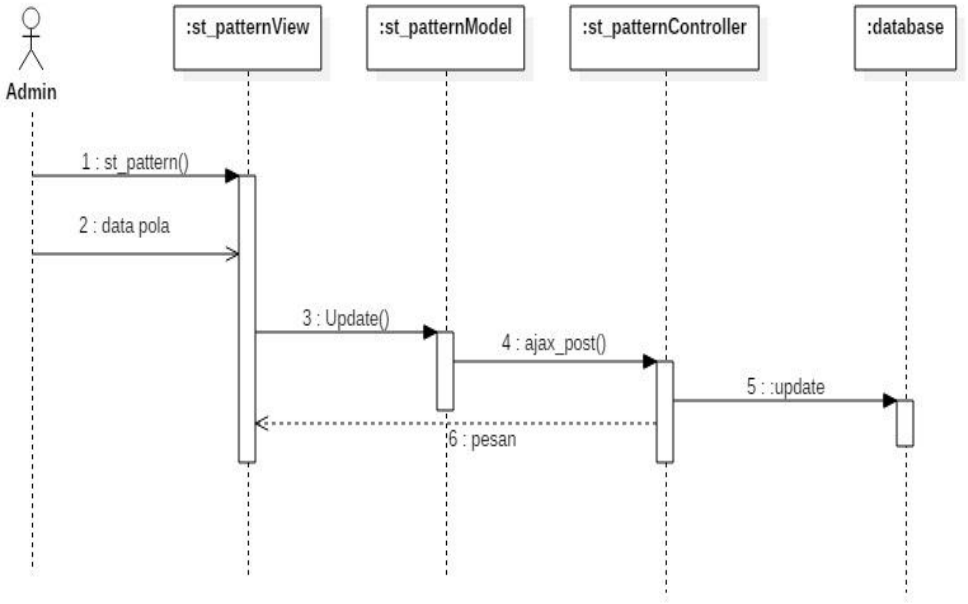
Gambar 3.23. Sequence Diagram Hapus Pertanyaan

10. Sequence Diagram Tambah Pola Kalimat



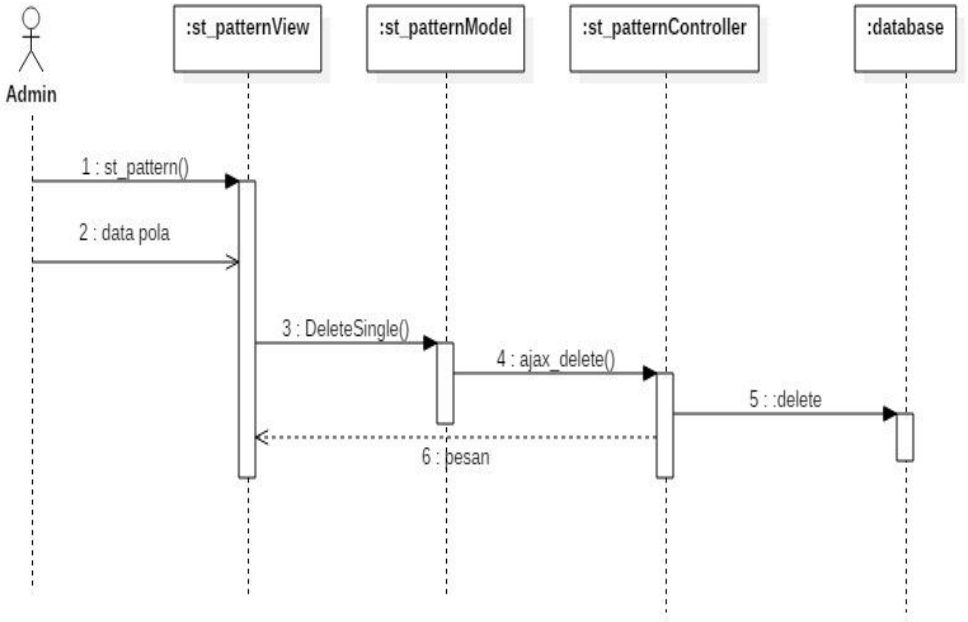
Gambar 3.24. Sequence Diagram Tambah Pola Kalimat

11. Sequence Diagram Edit Pola Kalimat



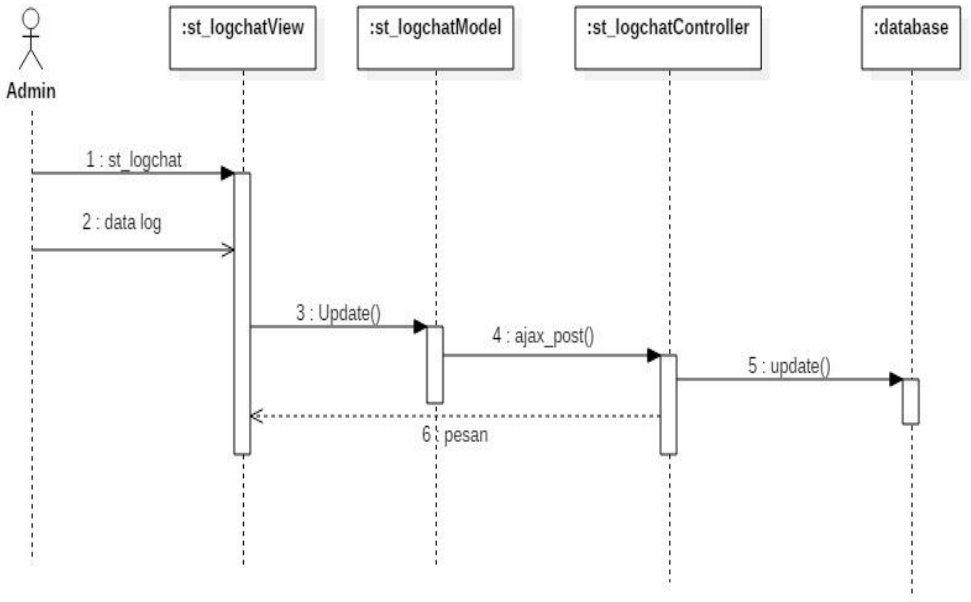
Gambar 3.25. Sequence Diagram Edit Pola Kalimat

12. Sequence Diagram Hapus Pola Kalimat



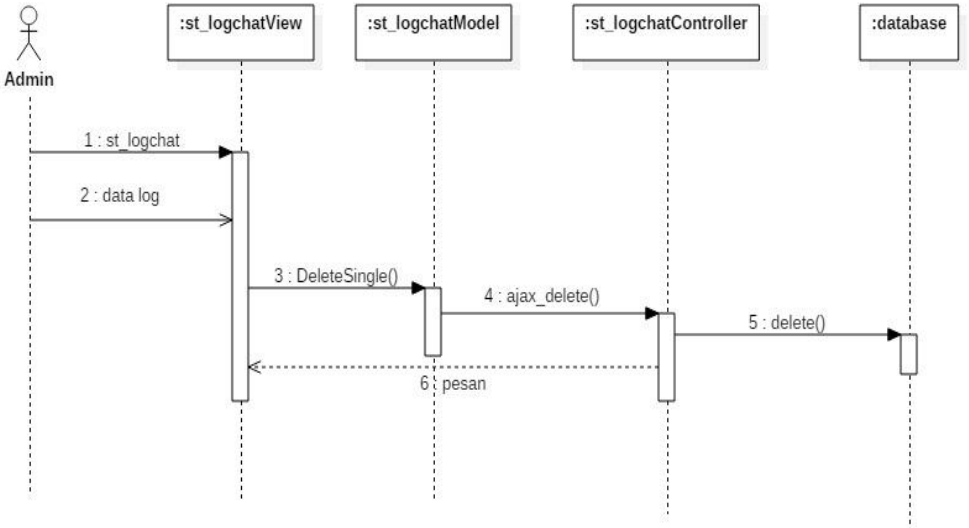
Gambar 3.26. Sequence Diagram Hapus Pola Kalimat

13. Sequence Diagram Edit Log Chat



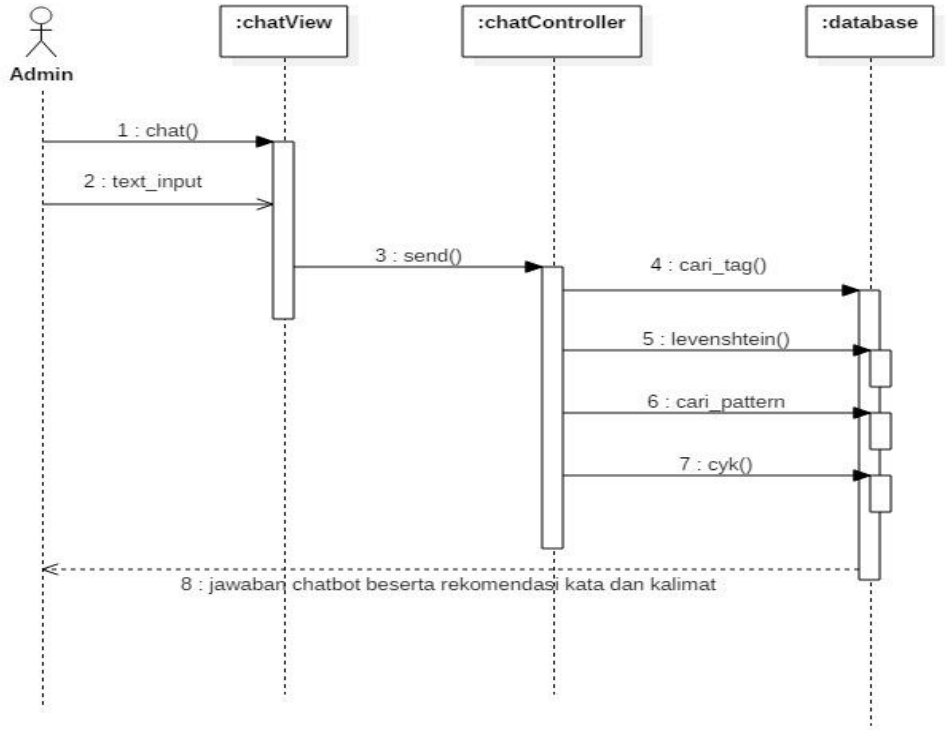
Gambar 3.27. Sequence Diagram Edit Log Chat

14. Sequence Diagram Hapus Log Chat



Gambar 3.28. Sequence Diagram Hapus Log Chat

15. Sequence Diagram Chat Bot



Gambar 3.29. Sequence Diagram Chat Bot

3.2 Perancangan Sistem

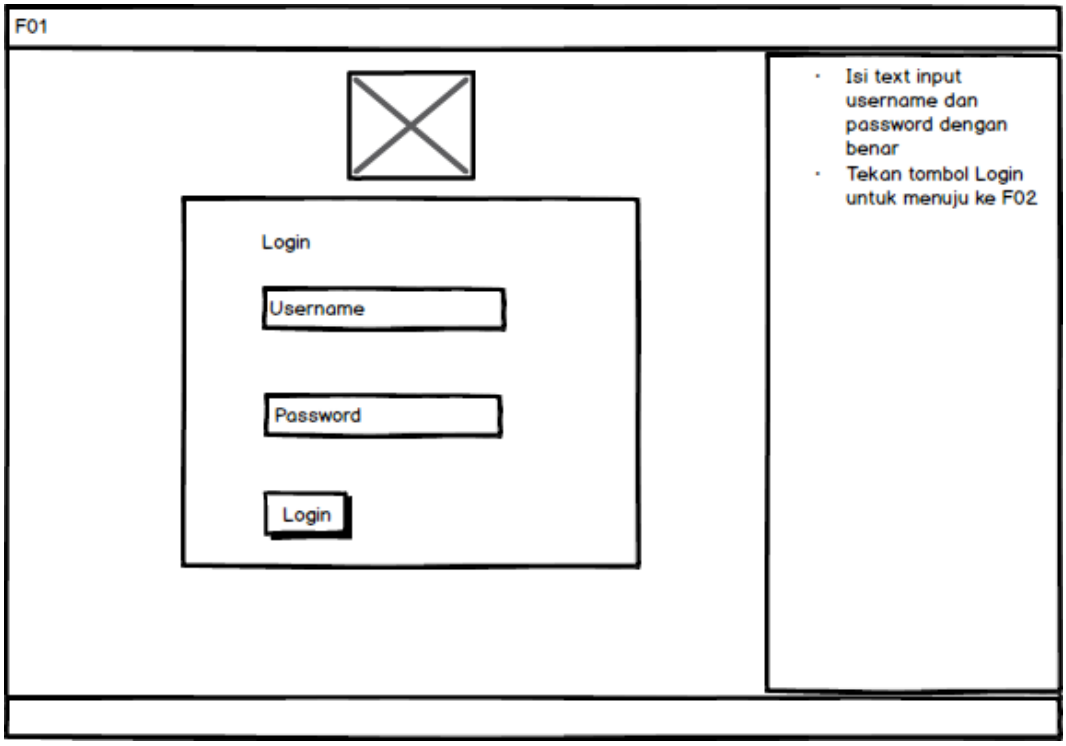
Perancangan sistem merupakan penggambaran dan perencanaan dari beberapa elemen sistem yang terpisah ke dalam suatu kesatuan model yang utuh. Tahapan ini meliputi perancangan antarmuka dan jaringan semantik.

3.2.1 Perancangan Antarmuka

Perancangan antarmuka mendeskripsikan rencana tampilan dari setiap tampilan yang akan digunakan.

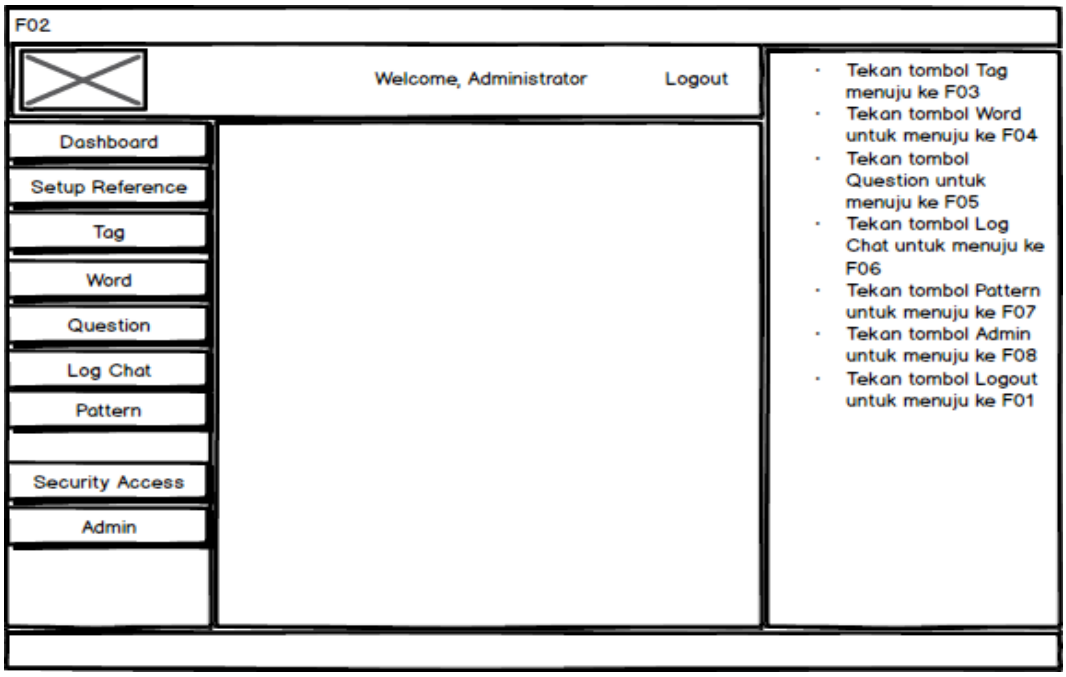
3.2.1.1 Perancangan Antarmuka *Backend*

1. Perancangan Form Login



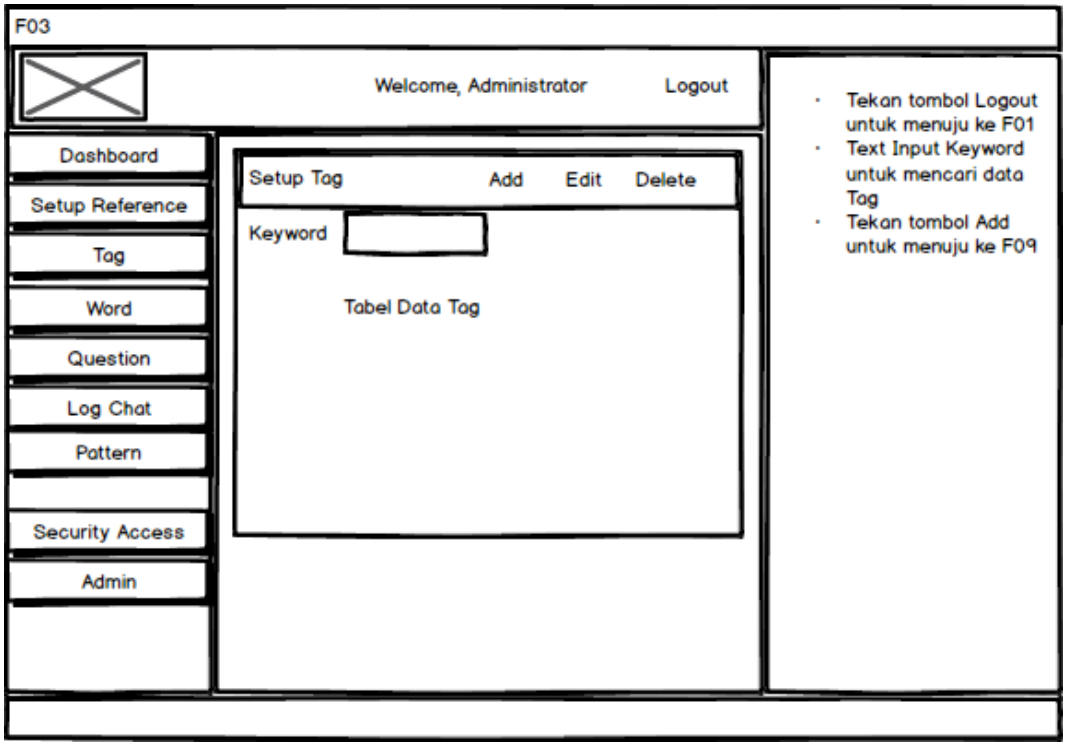
Gambar 3.30. Perancangan Form Login

2. Perancangan Form Menu Awal Admin



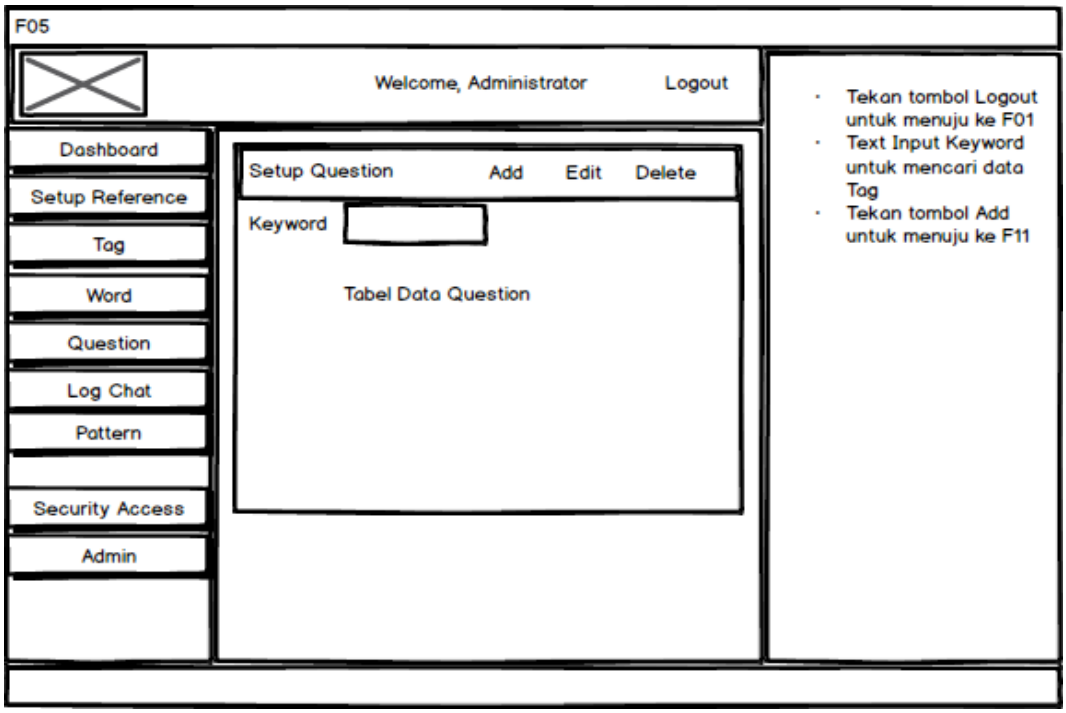
Gambar 3.31. Perancangan Form Menu Awal Admin

3. Perancangan Form Data Tag



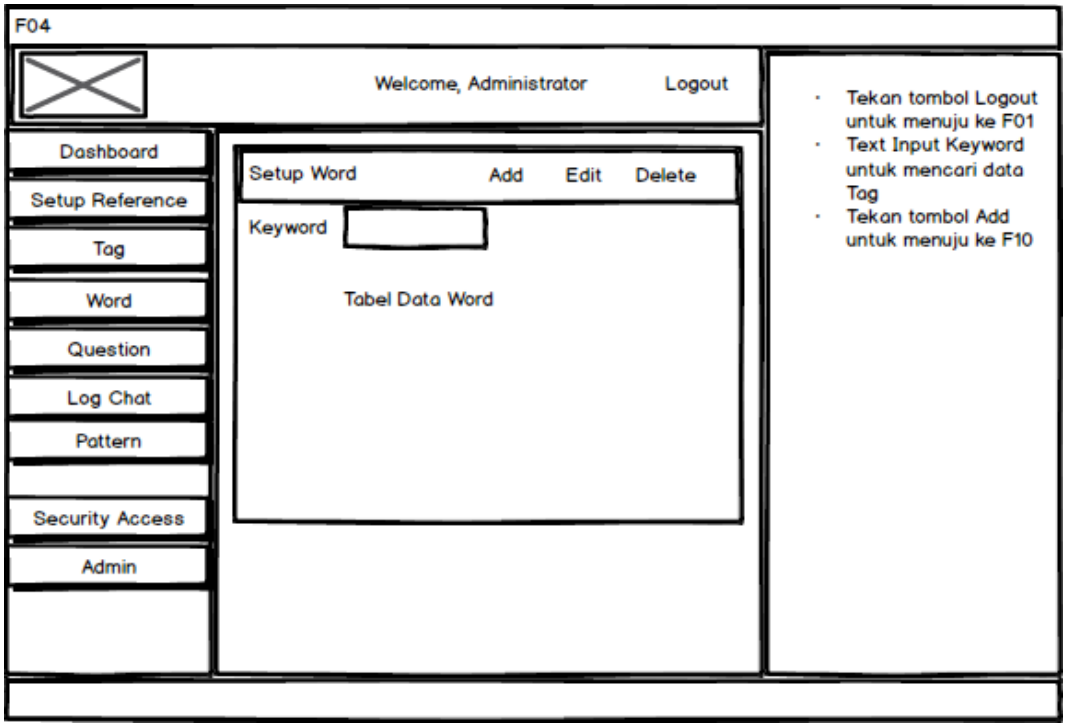
Gambar 3.32. Perancangan Form Data Tag

4. Perancangan Form Data Pertanyaan



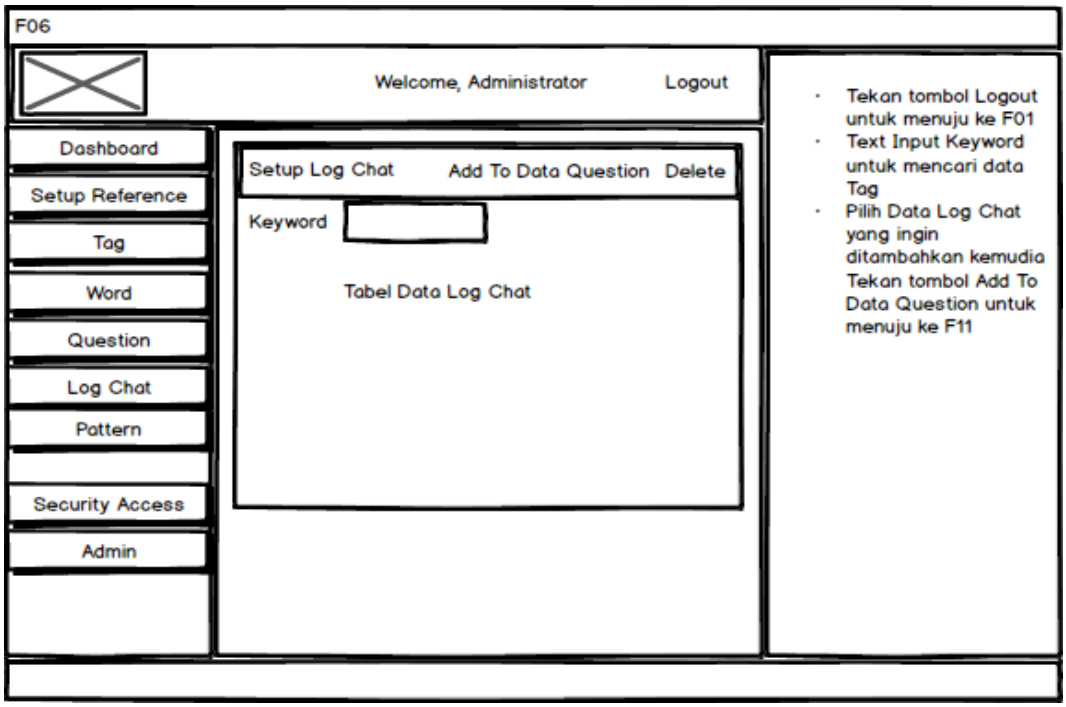
Gambar 3.33. Perancangan Form Data Brainfile

5. Perancangan Form Data Kata



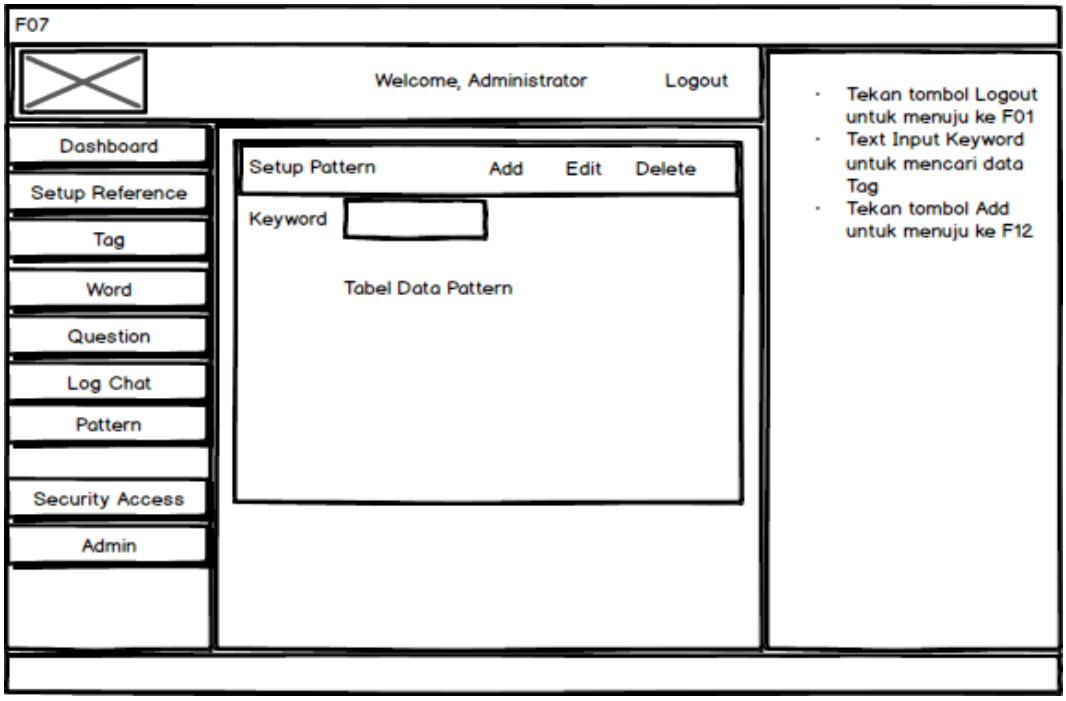
Gambar 3.34. Perancangan Form Data Kata

6. Perancangan Form Data Log Chat



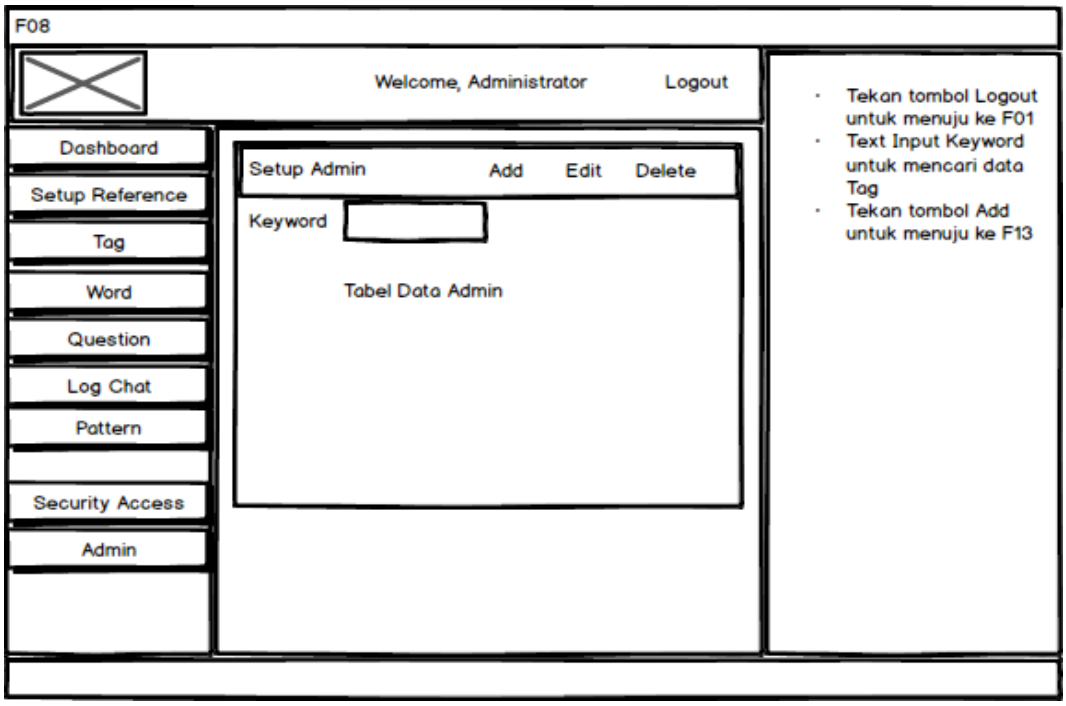
Gambar 3.35. Perancangan Form Data Log Chat

7. Perancangan Form Data Pattern



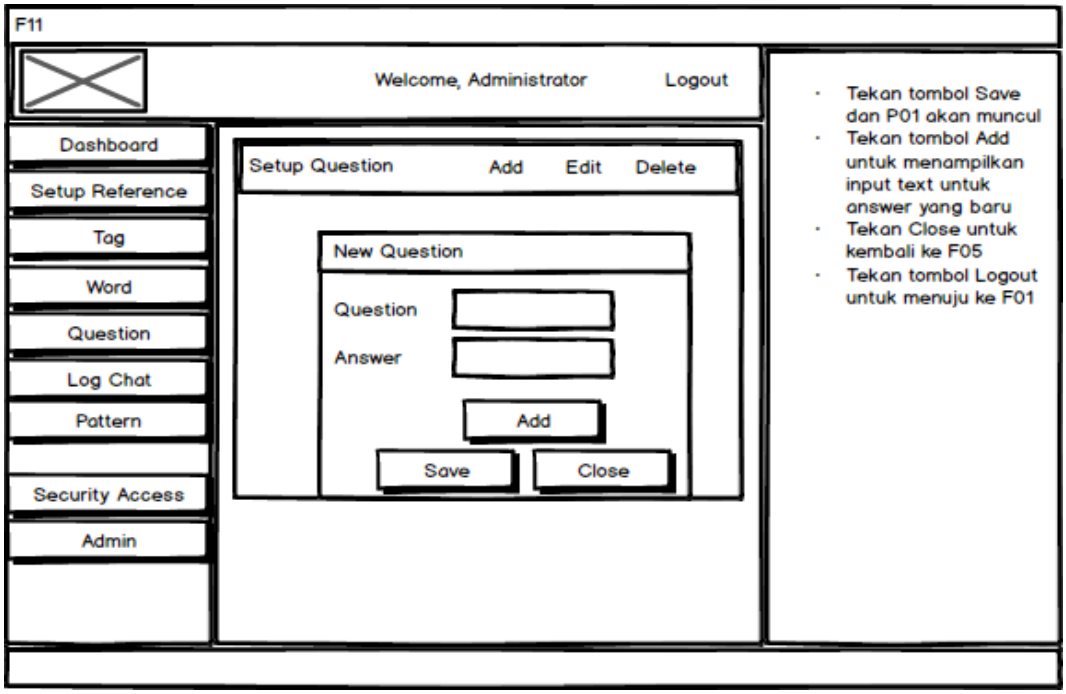
Gambar 3.36. Perancangan Form Data Pattern

8. Perancangan Form Data Admin



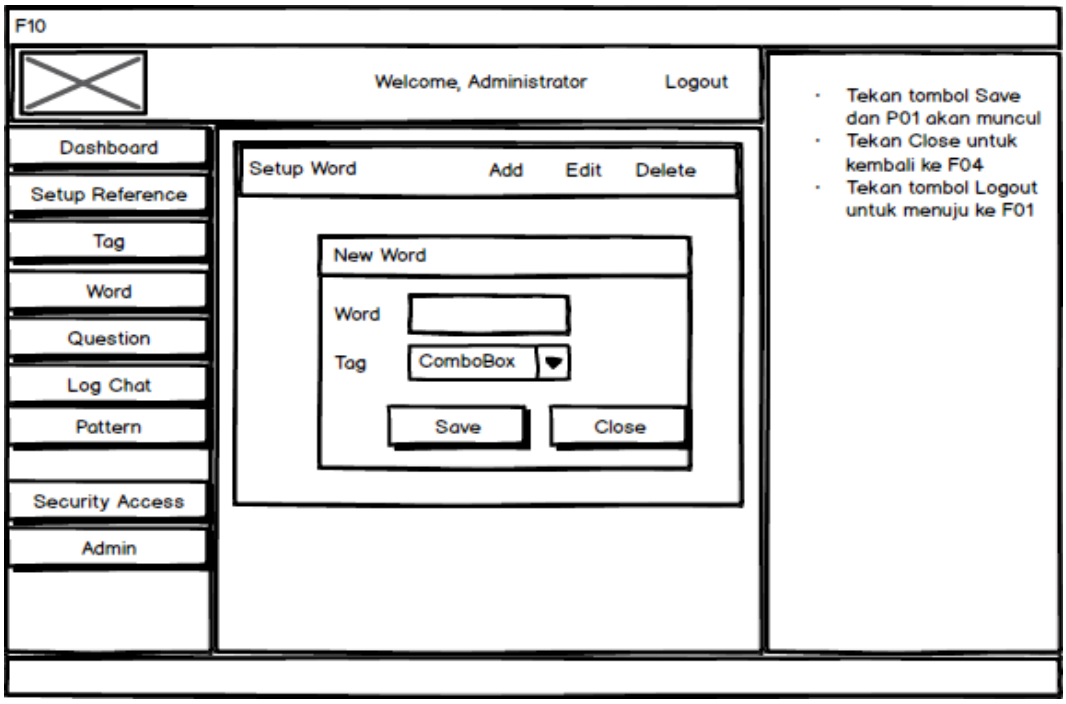
Gambar 3.37. Perancangan Form Data Admin

9. Perancangan Form Tambah Brainfile



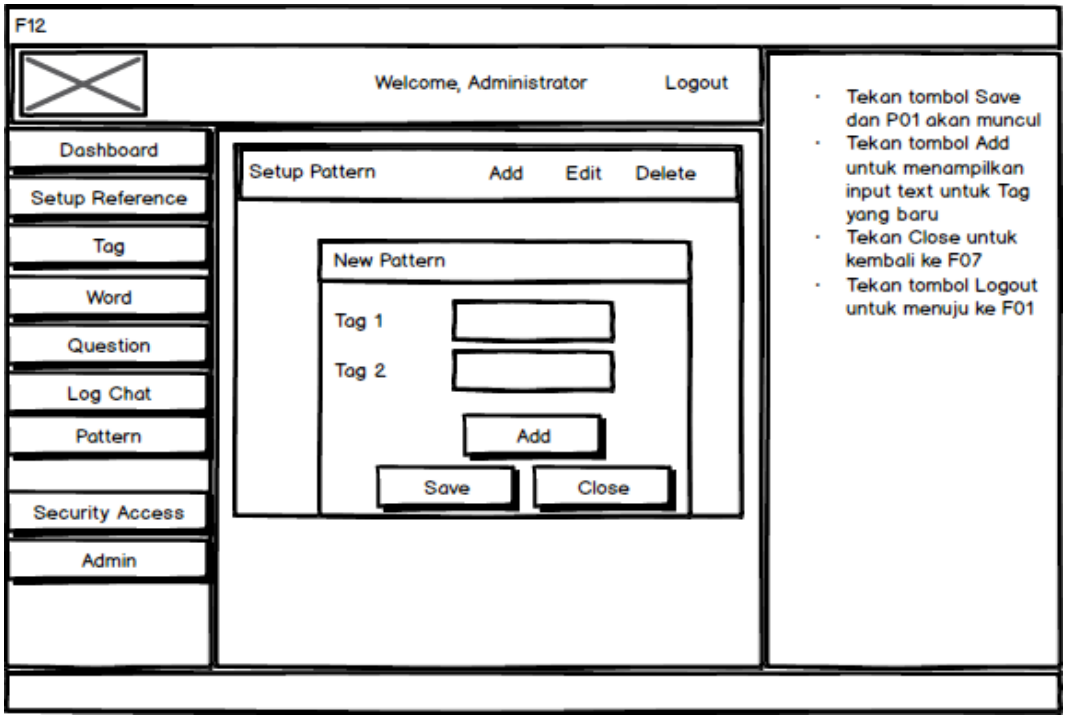
Gambar 3.38. Perancangan Form Tambah Brainfile

10. Perancangan Form Tambah Kata



Gambar 3.39. Perancangan Form Tambah Kata

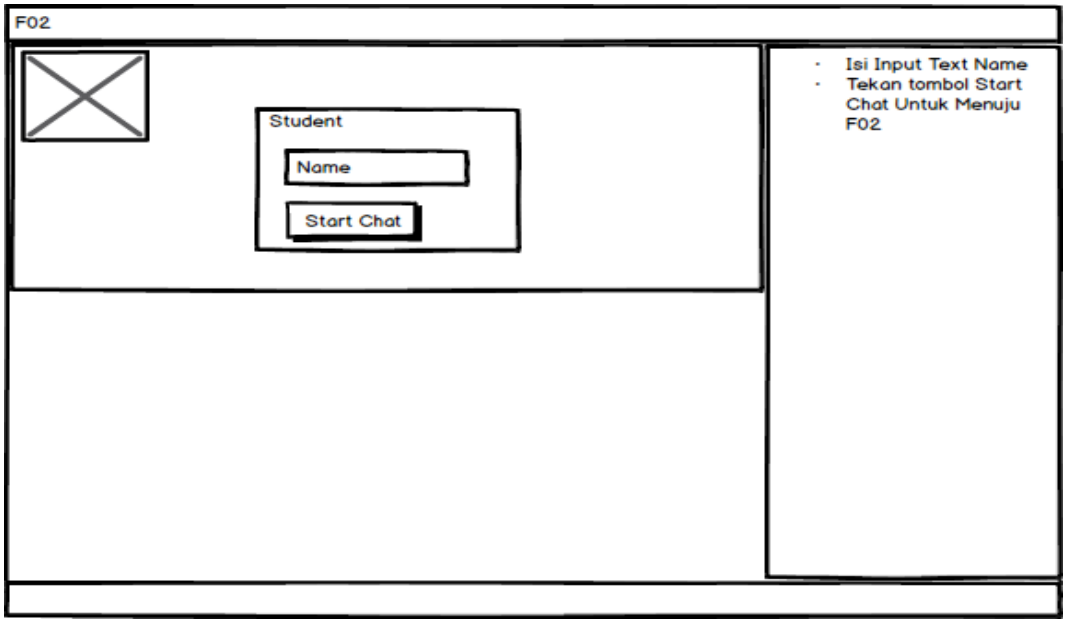
11. Perancangan Form Tambah Pattern



Gambar 3.40. Perancangan Form Tambah Pattern

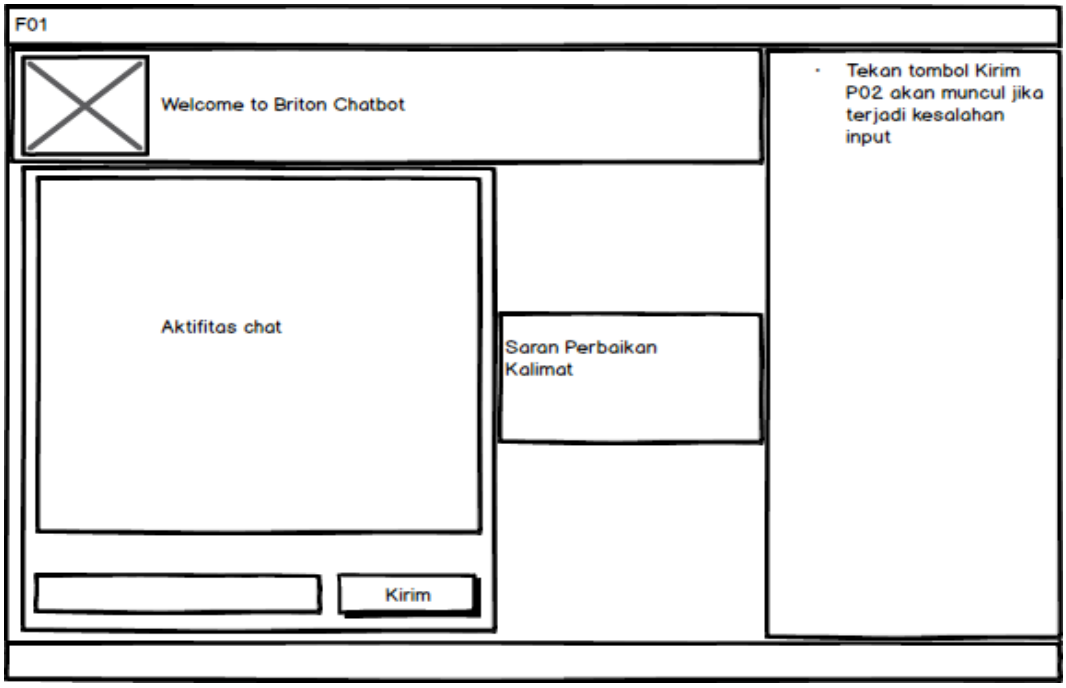
3.2.1.2 Perancangan Antarmuka *Frontend*

1. Perancangan Form Chatbot



Gambar 3.41. Perancangan Form Chatbot

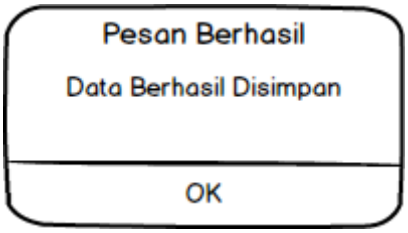
2. Perancangan Form Chatbot



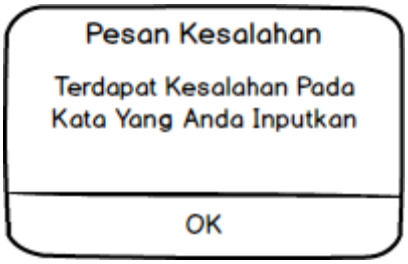
Gambar 3.42. Perancangan Form Chatbot

3.2.2 Perancangan Pesan

Perancangan pesan adalah rancangan antarmuka dari pesan yang akan ditampilkan apabila terjadi kesalahan dan pemberitahuan dalam proses yang dilakukan.



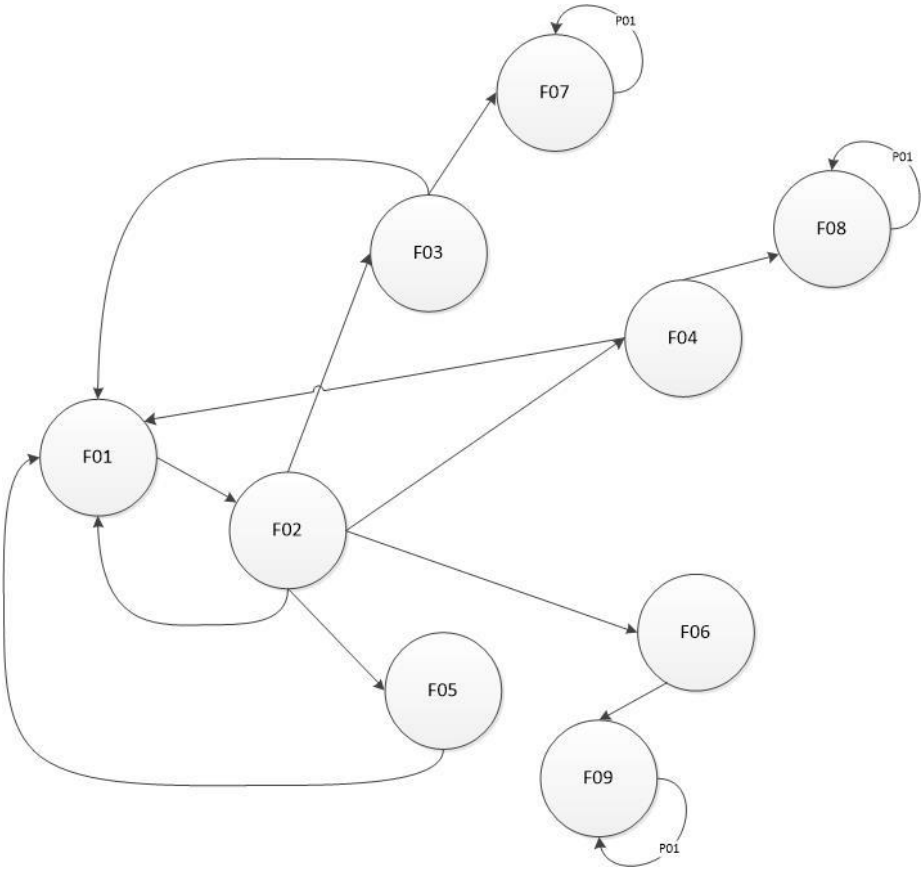
Gambar 3.43. Pesan Berhasil Menyimpan Data



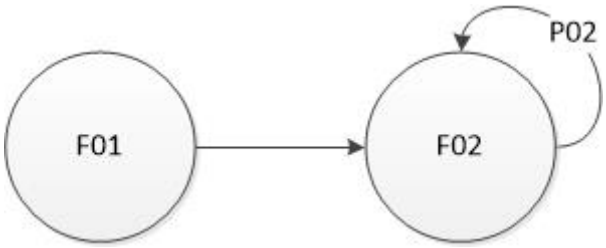
Gambar 3.44. Pesan Kesalahan Input Chat

3.2.3 Jaringan Semantik

Jaringan semantik menggambarkan keterhubungan navigasi menu dari satu halaman ke halaman lainnya. Jaringan semantik perancangan ini dapat dilihat sebagai berikut.



Gambar 3.45. Jaringan Semantik Backend



Gambar 3.46. Jaringan Semantik Frontend