BAB 3

PENDAHULUAN

3.1 Analisis Sistem

Analisis metode dilakukan dengan tujuan mendapatkan analisis yang sesuai dengan bidang ilmu yang diteliti. Analisis dijelaskan secara detil mulai dari analisis masalah, analisis data masukkan, analisis sintaksis untuk pembentukan aturan produksi, analisis kebutuhan non-fungsional, dan analisis kebutuhan fungsional dari studi kasus yang diteliti.

3.1.1 Analisis Masalah

Berdasarkan hasil penelitian tentang EYD, algoritma CYK dan algoritma Levenshtein, maka masalah yang dihadapi pada penelitian yang membahas tentang algoritma Levenshtein [2] adalah belum diketahui cara memperbaiki pola kalimat yang keliru. Oleh karena itu, pada penelitian ini akan dilakukan perbaikan pola kalimat yang dimulai dari pemeriksaan kata dengan membagi ke dalam kelompok-kelompok kata, pemeriksaan kalimat menggunakan algoritma CYK, sampai perbaikan pola kalimat menggunakan algoritma Levenshtein.

3.1.2 Analisis Data Masukkan

Data masukan untuk simulator ini berupa dokumen dengan ekstensi .doc atau .docx. Setiap kalimat dalam dokumen diperiksa satu persatu dan setiap kata yang ada pada kalimat digolongkan ke dalam kelompok kata. Pengelompokkan kata diambil dari kamus menggunakan POS Tag Indonesia. Pengelompokkan kata dengan POS Tag dapat dilihat pada tabel 3.1.

Tabel 3.1 Pengelompokkan kata berdasarkan POS Tag Indonesia [6]

No.	Tag	Deskripsi	Contoh
1	CC	Konjungsi koordinatif adalah konjungsi yang menghubungkan dua buah konstituen yang kedudukannya sederajat.	dan, tetapi, namun, lalu, sedangkan.
2	CD	Numeralia kardinal yaitu numeralia yang menjadi jawaban atas pertanyaan "berapa?".	dua juta, enam, 900, 0.25, banyak, ratusan.
3	OD	Numeralia ordinal menyatakan urutan dan menjadi jawaban atas pertanyaan "yang keberapa?"	kedua, ke-3, pertama
4	DT	Artikula bertugas membatasi makna nomina	para, sang, si
5	FW	Kata yang berasal dari bahasa asing yang belum diserap ke dalam bahasa Indonesia. Pada dasarnya, kata bahasa asing adalah kata yang tidak terdapat di dalam kamus bahasa Indonesia. Jika sebuah kata bahasa asing menjadi bagian dari *proper noun* atau nama, kata bahasa asing tersebut dimasukkan ke dalam kelompok kata NNP.	*climate change*, *terms and condition*
6	IN	Preposisi menghubungkan kata atau frasa dengan konstituen di depan preposisi tersebut sehingga terbentuk frasa preposisional.	dalam, dengan, di, oleh, pada
7	JJ	Ajektifa adalah kata yang memberikan keterangan khusus tentang sesuatu yang dinyatakan oleh nomina dalam kalimat.	bersih, marah, lebar, suram, pintar
8	MD	Verba modal atau verba bantu	boleh, harus, sudah, perlu
9	NEG	Kata ingkar	tidak, jangan, belum
10	NN	Nomina yaitu kata yang mengacu pada manusia, binatang, benda, konsep, atau pengertian.	kera, sekarang, bawah
11	NNP	*Proper Noun* adalah nama spesifik dari seseorang, sesuatu, atau sebuah tempat.	Boediono, Indonesia, Bank Mandiri, Lord of the Rings

12	NND	Penggolong atau nomina ukuran. Penggolong menempatkan nomina ke dalam sebuah kelompok tertentu dalam jumlah tertentu. Nomina ukuran merujuk pada ukuran, jarakm volume, kecepatan, berat, atau temperatur.	orang, ton, helai, lembar
13	PR	Demonstrativa atau pronomina penunjuk	ini, itu, disini, disitu
14	PRP	Pronomina persona, yaitu pronomina yang dipakai untuk mengacu pada orang atau kata ganti orang tunggal atau jamak.	saya, kami, dia, mereka
15	RB	Adverbia atau disebut juga kata keterangan.	sangat, hanya, seluruh, cukup
16	RP	Partikel yang digunakan untuk menegaskan kalimat interogatif, imperatif, atau deklaratif.	pun, -lah, -kah
17	SC	Konjungsi subordinatif adalah kkonjungsi yang menghubungkan dua buah konstituen yang kedudukannya tidak sederajat.	sebab, jika, meskipun, supaya
18	SYM	Simbol	IDR, +, %
19	UH	Interjeksi mengungkapkan rasa hati atau perasaan pembicara dan secara sintaksis tidak berhubungan dengan kata-kata lain di dalam kalimat.	oh, aduh, ayo, mari
20	VB	Verba atau kata kerja dapat berupa verba transitif, verba intransitif, verba aktif, verba pasif, dan kopula. Jika kata dasar dari verba berimbuhan berupa kata asing, verba tersebut tetap dimasukkan ke dalam kelompok kata VB, misalnya di-*accept*.	merancang, mengatur, pergi, bekerja, terbangun
21	WH	Pronomina penanya atau disebut juga kata tanya. Pronomina penanya digunakan dalam kalimat interogatif sebagai pemarkah pertanyaan. Klausa tanya yang ada di kalimat deklaratif merupakan klausa subordinatif sehingga pronomina penanya yang menghubungkan kata tanya dengan klausa lain di dalam kalimat deklaratif menjadi konjungsi subordinatif.	siapa, apa, kemana, di mana, berapa
22	X	Kata atau bagian yang tidak diketahui atau belum diketahui secara pasti kategorinya.	Statemen
23	Z	Tanda baca	?, ., "…"

3.1.3 Analisis Sintaksis untuk Aturan Produksi

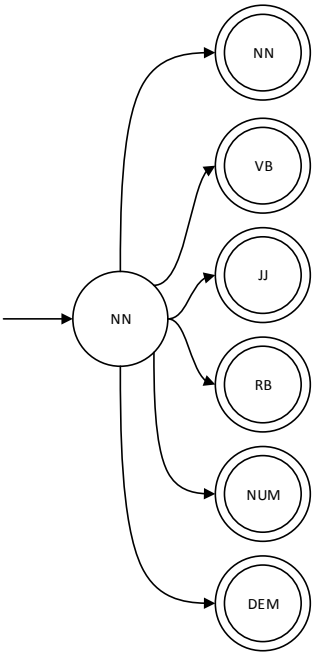
Berdasarkan pembahasan tentang sintaksis Bahasa Indonesia dan pengelompokkan kata dengan POS Tag Indonesia, selanjutnya pada bagian ini akan dijelaskan bagaimana membuat aturan produksi dalam bentuk CNF. Sebelumnya, seluruh penamaan kelompok kata harus diubah ke dalam aturan POS Tag agar tidak sulit dalam mengecek aturan produksinya. Perubahannya dapat dilihat pada tabel 3.2.

Tabel 3.2 Tabel penamaan kelompok kata ke dalam POS Tag

Nama Kelompok Kata	Diubah Menjadi
Noun / Kata Benda (N)	NN, NNP, NND, PRP, atau WH
Adjective / Kata Sifat (A)	JJ
Verb / Kata Kerja (V)	VB
Adverb / Kata Keterangan (Adv)	RB
Numeralia (Num)	OD dan CD
Preposisi (P)	IN
Demonstrative (Dem)	PR

3.1.3.1 Aturan Produksi untuk Frase Nomina (FN)

Pada pembahasan tentang sintaksis Bahasa Indonesia untuk frase nomina (FN) dapat diketahui bahwa susunan kelompok kata yang dapat dibentuk adalah N + N, N + V, N + A, N + Adv, Adv + N, N + Num, Num + N, N + Dem. Bentuk aturan produksinya digambarkan dalam *statechart* dan dapat dilihat pada gambar 3.1.

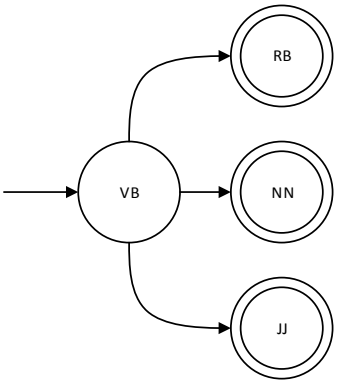


Gambar 3.1 *Statechart* untuk aturan produksi frase nomina

Aturan penamaan kelompok kata disesuaikan dengan *tag* dari POS Tag Indonesia (berdasarkan tabel 3.2) agar tidak perlu lagi ada perubahan pada saat pengecekkan kelompok kata. Pemberian *tag* untuk nomina pada gambar 3.1 digeneralisasi menjadi NN. *Tag* NN bisa disebut sebagai NND, NNP, NND, PRP, atau WH.

3.1.3.2 Aturan Produksi untuk Frase Verba (FV)

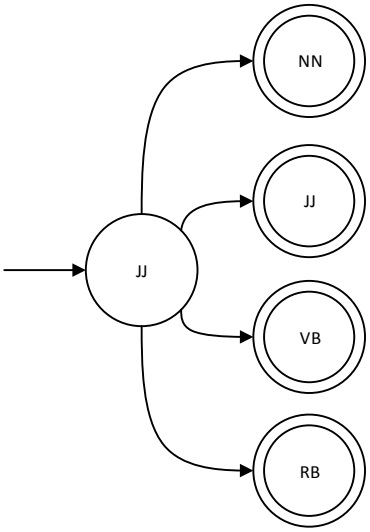
Susunan kelompok kata yang dapat dibentuk dari frase verba adalah V + Adv, Adv + V, V + N, V + A. Aturan produksinya dapat dilihat pada gambar 3.2.



Gambar 3.2 *Statechart* untuk aturan produksi frase verba

3.1.3.3 Aturan Produksi untuk Frase Ajektifa (FA)

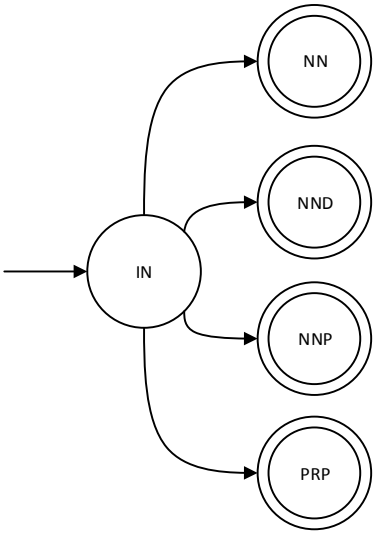
Susunan kelompok kata yang dapat dibentuk dari frase ajektiva adalah A + N, A + A, A + V, Adv + A, A + Adv. Aturan produksinya dapat dilihat pada gambar 3.3.



Gambar 3.3 *Statechart* untuk aturan produksi frase ajektiva

3.1.3.4 Aturan Produksi untuk Frase Preposisional (FP)

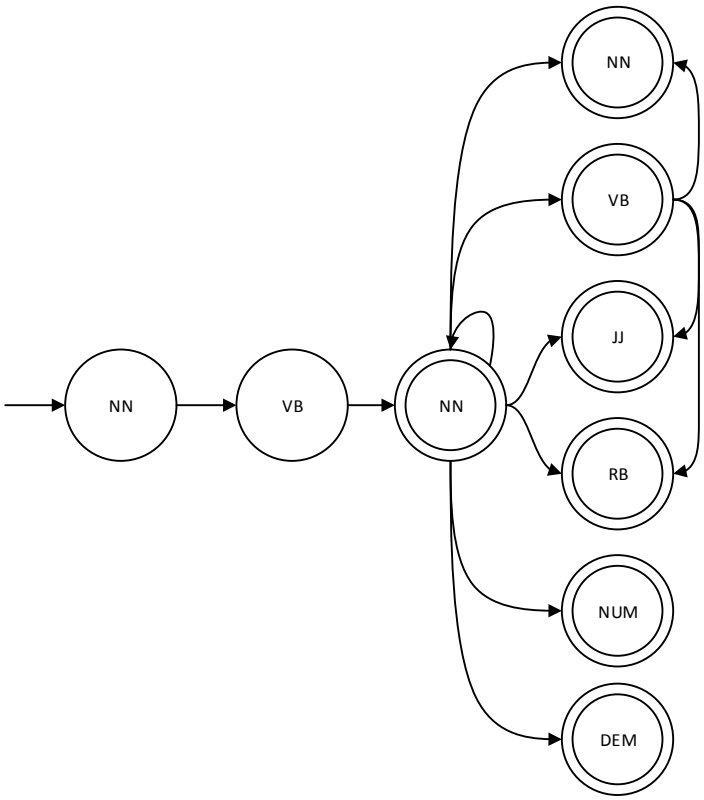
Susunan untuk frase preposisional hanya P + N saja. Nomina bisa berupa kata atau frase. Apabila variabel P disesuaikan dengan *tag* yang ada pada POS Tag, hasilnya akan seperti pada gambar 3.4.



Gambar 3.4 *Statechart* untuk aturan produksi frase preposisional

3.1.3.5 Aturan Produksi untuk Klausa Verba (KV)

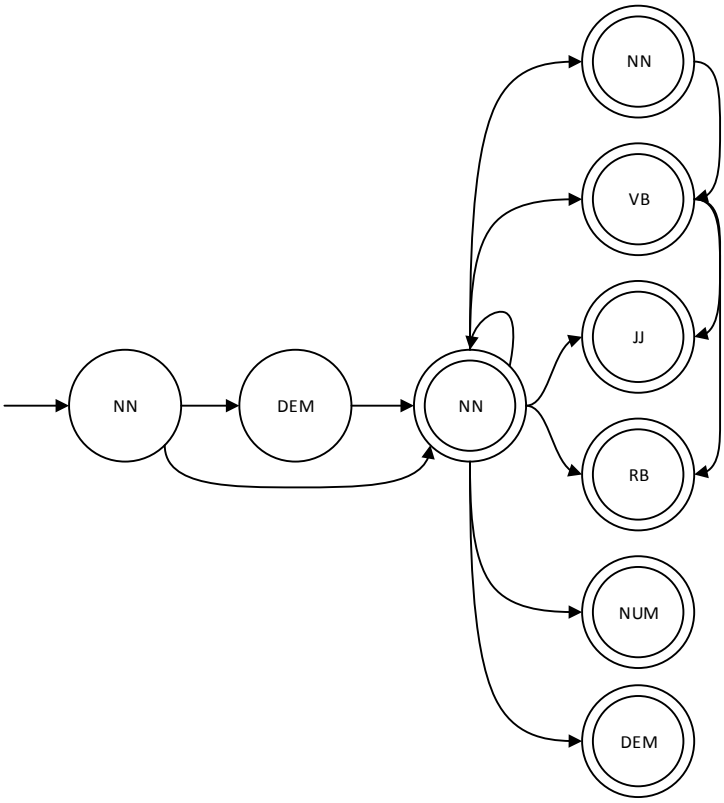
Secara umum aturan produksi untuk klausa verba dapat dilihat pada gambar 3.5.



Gambar 3.5 *Statechart* untuk aturan produksi klausa verba

3.1.3.6 Aturan Produksi untuk Klausa Nomina (KN)

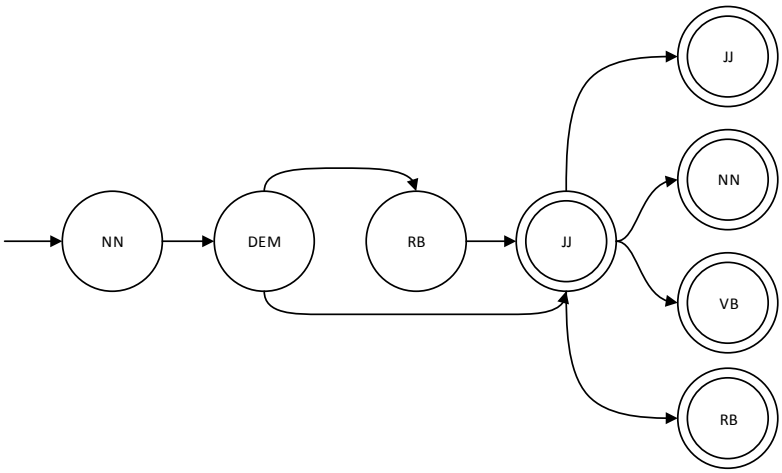
Penyusunan aturan produksi untuk klausa nomina wajib memiliki fungsi S dan P. Fungsi P diisi dengan kategori nomina. Aturan produksinya dapat dilihat pada gambar 3.6.



Gambar 3.6 *Statechart* untuk aturan produksi klausa nomina

3.1.3.7 Aturan Produksi untuk Klausa Ajektiva (KA)

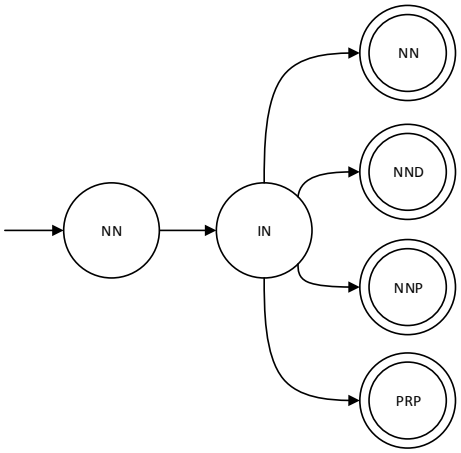
Penyusunan aturan produksi untuk klausa ajektiva wajib memiliki fungsi S dan P. Fungsi N diisi dengan kategori nomina dan P diisi dengan kategori ajektiva. Aturan produksinya dapat dilihat pada gambar 3.7.



Gambar 3.7 *Statechart* untuk aturan produksi klausa ajektiva

3.1.3.8 Aturan Produksi untuk Klausa Preposisional

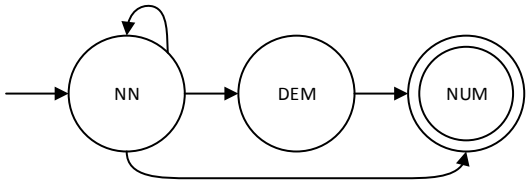
Penyusunan aturan produksi untuk klausa preposisional wajib memiliki fungsi S dan P. Fungsi P diisi oleh frase preposisional. Aturan produksinya dapat dilihat pada gambar 3.8.



Gambar 3.8 *Statechart* untuk aturan produksi klausa preposisional

3.1.3.9 Aturan Produksi untuk Klausa Numeral

Penyusunan aturan produksi untuk klausa nomina wajib memiliki fungsi S dan P. Fungsi P diisi oleh frase numeral. Aturan produksinya dapat dilihat pada gambar 3.9.



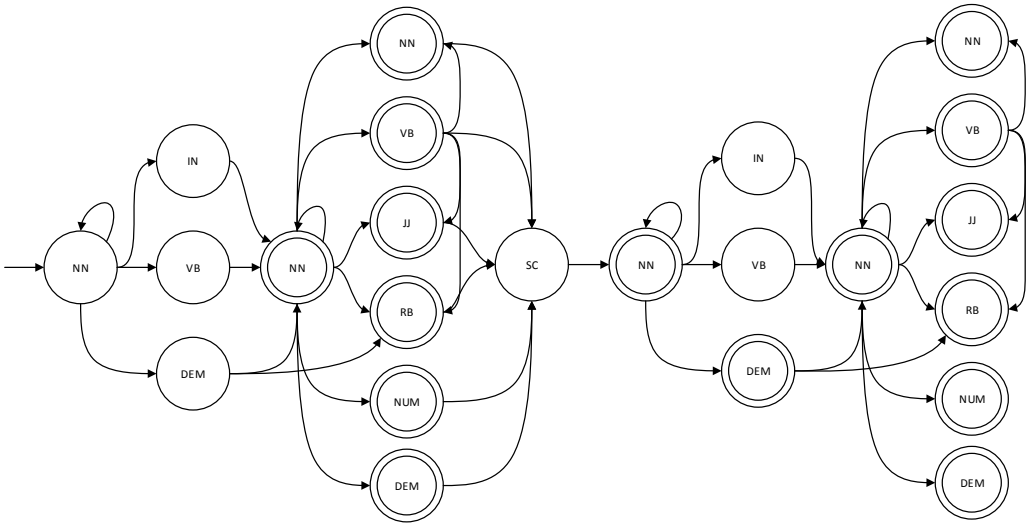
Gambar 3.9 *Statechart* untuk aturan produksi klausa numeral

3.1.3.10 Aturan Produksi untuk Kalimat Sederhana

Secara umum gambaran aturan produksi dari kalimat sederhana sama seperti klausa. Kalimat sederhana dapat terbentuk dari klausa nomina, klausa verba, klausa ajektiva, klausa preposisional, atau klausa numeral.

3.1.3.11 Aturan Produksi untuk Kalimat Luas

Secara umum gambaran aturan produksi dari kalimat luas dalam hal ini adalah gabungan dari kedua klausa dengan menggunakan kata sambung antar klausa.



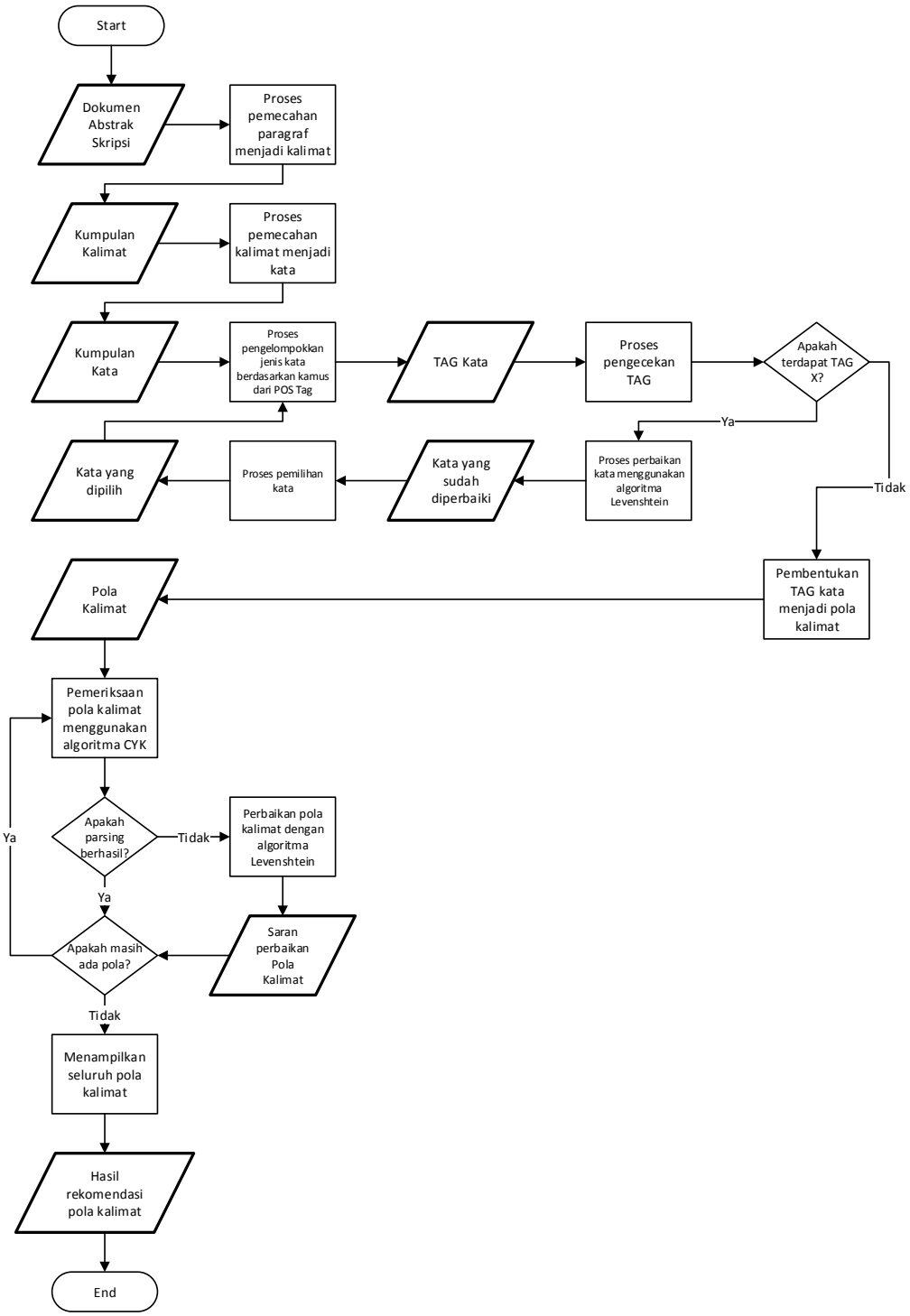
Gambar 3.10 *Statechart* untuk aturan produksi kalimat luas

3.1.3.12 Aturan Produksi yang Terbentuk

Hasil aturan produksi yang terbentuk dari *statechart* yang sudah digambarkan dapat dilihat pada lampiran A.

3.1.4 Analisis Algoritma CYK dan Levenshtein pada Perbaikan Kalimat

Pada tahap ini dilakukan langkah-langkah analisis mengenai cara agar sebuah kalimat dapat dilakukan perbaikan pola kalimat berdasarkan analisis aturan produksi yang telah dibuat sebelumnya. Alur dari proses ini dapat dilihat pada gambar 3.1.



Gambar 3.11 Alur proses perbaikan kalimat

3.1.4.1 Proses Pemecahan Paragraf

Pada tahap awal dokumen abstrak yang akan diberi rekomendasi perbaikan akan dipecah paragrafnya menjadi kalimat kemudian kalimat menjadi kata. Proses pemecahan paragraf dibagi menjadi 2 tahap, yaitu:

1. Paragraf menjadi Kalimat

   Pada tahap ini akan diketahui ada berapa kalimat dalam satu paragraf abstrak yang diuji. Cara memecah paragraf menjadi kalimat dapat menggunakan regex. Tanda yang dilihat antar kalimat adalah titik. Titik menjadi penanda akhir kalimat namun tidak semua titik merupakan penanda akhir sebuah kalimat. Pola yang dibentuk dari regex adalah [karakter][tanda_titik][spasi].

   Tahap awal pengecekan dilakukan dengan menelusuri karakter demi karakter yang ada pada paragraf abstrak. Sebagai contoh paragraf yang akan diperiksa dapat dilihat pada tabel 3.3.

Tabel 3.3 Paragraf pada abstrak

Kualitas barng adalah merupakan hal yang terpenting dalam sebuah perdagangan. Sitem otomatis yang dibuat untuk menjaga kualitas barang yaitu berbasis embedded system mikrokontroler. Karena pengantar barang tidak perlu repot untuk turun dan menutp atap mobil truk ktika hujan turun karena dengan embedded system alat akan bekerja secara otomatis menutup atap mobil sesuai dengan kondisi cuaca. Data sensor suhu, sensor cahasya dan sensor air akan diolah dengan bantuan bahasa tingkat tinggi arduino pada mikrokontroler demi mendapatkan suatu tindakan yang tepat dan cepat.

Pada saat penelusuran setiap karakter diperiksa dengan pola regex. Apabila pola tersebut sama dengan pola regex, *string* tersebut merupakan penanda dari akhir kalimat. Pecahan kalimat yang terbentuk dari pemisahan paragraf dapat dilihat pada tabel 3.4.

Tabel 3.4 Pecahan kalimat yang terbentuk

No.	Kalimat
1	*Kualitas barng adalah merupakan hal yang terpenting dalam sebuah perdagangan.*
2	*Sitem otomatis yang dibuat untuk menjaga kualitas barang yaitu berbasis embedded system mikrokontroler.*
3	*Karena pengantar barang tidak perlu repot untuk turun dan menutp atap mobil truk ktika hujan turun karena dengan embedded system alat akan bekerja secara otomatis menutup atap mobil sesuai dengan kondisi cuaca.*
4	*Data sensor suhu, sensor cahasya dan sensor air akan diolah dengan bantuan bahasa tingkat tinggi arduino pada mikrokontroler demi mendapatkan suatu tindakan yang tepat dan cepat.*

2. Kalimat menjadi Kata

Pada tahap ini, kalimat-kalimat yang sudah terbentuk akan dipisahkan berdasarkan spasi. Tanda titik yang ikut dalam pemisahan kalimat akan dibuang untuk menghindari kesalahan pemeriksaan saat melakukan pengelompokkan kata.

3.1.4.2 Pengelompokkan Kata

Pada tahap ini setiap kata dikelompokkan sesuai dengan kamus yang diambil berdasarkan aturan dari POS Tag. Tanda titik pada akhir kalimat harus disertakan juga untuk proses pembentukan pola kalimat. Pengelompokkan kata yang terbentuk dapat dilihat pada tabel 3.5.

Tabel 3.5 Pengelompokkan kata

Kalimat ke	Kata	Tag
Kalimat ke-1	Kualitas	NN
	barng	X
	adalah	VB
	merupakan	VB
	hal	NN
	yang	SC
	terpenting	VB
	dalam	IN
	sebuah	CD

	perdagangan	NN
	.	Z
Kalimat ke-2	Sitem	X
	otomatis	NN
	yang	SC
	dibuat	VB
	untuk	IN
	menjaga	VB
	kualitas	NN
	barang	NN
	yaitu	CC
	berbasis	VB
	embedded	NNP
	system	NNP
	mikrokontroler	NNP
	.	Z
Kalimat ke-3	Karena	SC
	pengantar	NN
	barang	NN
	tidak	NEG
	perlu	RB
	repot	JJ
	untuk	IN
	turun	VB
	dan	CC
	menutp	X
	atap	NN
	mobil	NN
	truk	NN
	ktika	X
	hujan	NN
	turun	VB
	karena	SC
	dengan	SC
	embedded	NNP

	system	NNP
	alat	NN
	akan	RB
	bekerja	VB
	otomatis	NN
	menutup	VB
	atap	NN
	mobil	NN
	sesuai	JJ
	dengan	SC
	kondisi	NN
	cuaca	NN
	.	Z
Kalimat ke-4	Data	NN
	sensor	NN
	suhu	NN
	sensor	NN
	cahasya	X
	dan	CC
	sensor	NN
	air	NN
	akan	RB
	diolah	VB
	dengan	SC
	bantuan	NN
	bahasa	NN
	tingkat	NN
	tinggi	JJ
	arduino	NNP
	pada	IN
	mikrokontroler	NNP
	untuk	SC
	mendapatkan	VB
	suatu	CD
	tindakan	NN

	yang	SC
	tepat	JJ
	cepat	JJ
	dan	CC
	.	Z

3.1.4.3 Pemeriksaan dan Perbaikan Kata dengan Algoritma Levenshtein

Pada tabel 3.5 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 "barng" akan dibandingkan dengan kata yang ada pada kamus. Contoh kata yang diambil dari kamus untuk dibandingkan yaitu "barang". Selanjutnya dibuat tabel dua dimensi seperti gambar 3.2.

		B	A	R	N	G
	0	1	2	3	4	5
B	1					
A	2					
R	3					
A	4					
N	5					
G	6					

Gambar 3.12 Pembentukan tabel dua dimensi Levenshtein

2. 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 gambar 3.13.

		B	A	R	N	G
	0	1	2	3	4	5
B	1	0	1	2	3	4
A	2	1	0	1	2	3
R	3	2	1	0	1	2
A	4	3	2	1	1	2
N	5	4	3	2	1	2
G	6	5	4	3	2	1

Gambar 3.13 Pengisian tabel dua dimensi menggunakan algoritma Levenshtein

Pada baris dan kolom terakhir terlihat jumlah *edit distance* dari perbandingan kata “barng” dan “barang” adalah 2. Pada tabel 3.6 ditunjukkan beberapa kata yang huruf awalan sama dengan kata “barang”.

Tabel 3.6 Perbandingan setiap kata

Saran Perbaikan	*Edit Distance*
bara	2
barang	1
barat	2
beras	2
berang	1
berat	3

Kata dengan *edit distance* paling kecil akan dijadikan rekomendasi perbaikan.

Pseudocode dari algoritma ini dapat dilihat pada gambar 3.14.

```
m ← length(s1)
n ← length(s2)
if m <> 0 and n <> 0 then
    for i ← 0 to m do
        lev_i0 ← 0
        for j ← 0 to n do
            lev_0j ← j
            if i > 0 and j > 0 then
                if s1_i == s2_j then
                    lev_ij ← lev_i-1,j-1
                else
                    lev_ij ← min(lev_i-1,j , lev_i,j-1 , lev_i-1,j-1)
                end if
            end if
        end for
    end for
end if
```

Gambar 3.14 *Pseudocode* Algoritma Levenshtein

3.1.4.4 Pemilihan Kata

Pada tahap ini dilakukan penggantian kata yang keliru dengan kata yang dipilih oleh user. Berdasarkan tabel 3.6 terdapat beberapa kata yang dapat dilakukan perbaikan. Hasil perbaikan akan menggantikan kata yang keliru. Pengelompokkan kata dilakukan kembali untuk mengetahui *tag* pada kata yang telah diperbaiki. Hasil perbaikan dapat dilihat pada tabel 3.7.

Tabel 3.7 Penggantian kata

Kata Awal	Kata Ganti	Tag
barng	barang	NN
sitem	sistem	NN
menutp	menutup	VB
ktika	ketika	SC
cahasya	cahaya	NN

3.1.4.5 Pembentukan Tag menjadi Pola Kalimat

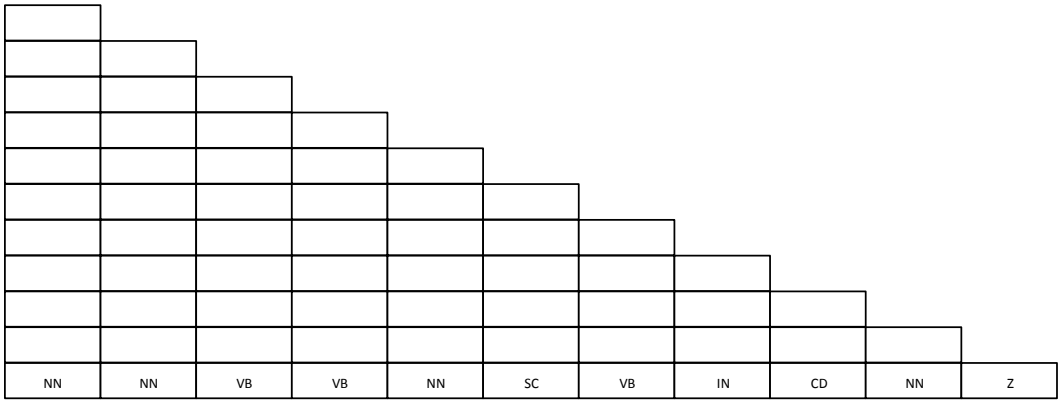
Pada tahap ini dilakukan penyusunan pola kalimat berdasarkan kelompok kata yang sudah diperbaiki. Kumpulan *tag* ditelusuri kembali dari awal. Apabila menemukan *tag* Z, *tag* tersebut merupakan penanda akhir pola kalimat. Pola kalimat yang terbentuk dapat dilihat pada tabel 3.8.

Tabel 3.8 Pola kalimat yang terbentuk

No	Pola Kalimat
1	NN NN VB VB NN SC VB IN CD NN Z
2	NN NN SC VB IN VB NN NN CC VB NNP NNP NNP Z
3	SC NN NN NEG RB JJ IN VB CC VB NN NN NN SC NN VB SC SC NNP NNP NN RB VB NN VB NN NN JJ SC NN NN Z
4	NN NN NN NN NN CC NN NN RB VB SC NN NN NN JJ NNP IN NNP SC VB CD NN SC JJ JJ CC Z

3.1.4.6 Pengenalan Pola Kalimat dengan Algoritma CYK

Pada tahap ini dilakukan pengecekkan apakah kalimat yang diperiksa sesuai dengan aturan produksi atau tidak. Pemeriksaan pola kalimat menggunakan algoritma CYK. Langkah awal pemeriksaan dapat dilihat pada gambar 3.15.



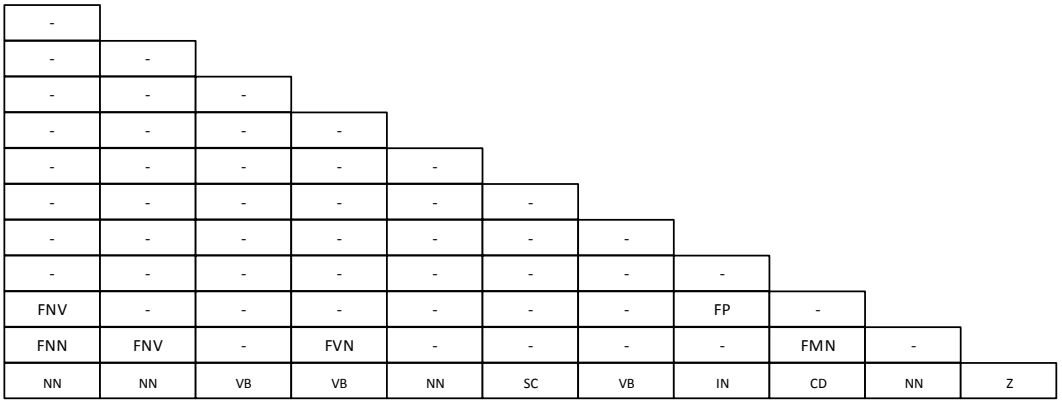
Gambar 3.15 Pemeriksaan pola dengan algoritma CYK

Seperti penjelasan yang telah dijelaskan pada bab 2 tentang langkah-langkah algoritma CYK maka pengisian baris dari tabel dua dimensi algoritma ini dapat dilihat pada tabel 3.9.

Tabel 3.9 Pengisian baris algoritma CYK

Blok	Tag 1	Tag 2	Aturan Produksi
[1,1] & [1,2]	NN	NN	FNN
[1,2] & [1,3]	NN	VB	FNV
[1,3] & [1,4]	VB	VB	-
[1,4] & [1,5]	VB	NN	FVN
[1,5] & [1,6]	NN	SC	-
[1,6] & [1,7]	SC	VB	-
[1,7] & [1,8]	VB	IN	-
[1,8] & [1,9]	IN	CD	-
[1,9] & [1,10]	CD	NN	FMN
[1,10] & [1,11]	NN	Z	-

Hasil akhir yang didapat dari algoritma ini dapat dilihat pada gambar 3.16.



Gambar 3.16 Hasil akhir pemeriksaan pola kalimat.

Pada hasil akhir diketahui bahwa pola kalimat ini tidak dapat dikenali oleh algoritma CYK. Tahap selanjutnya akan dilakukan perbaikan dengan menggunakan algoritma Levenshtein.

Pseudocode dari algoritma CYK dapat dilihat pada gambar 3.17.

```
for i ← 0 to n do
    C0i ← V {V → v aturan produksi dimana simbol ke-i adalah v}
end for
for j ← 1 to n do
    for k ← 0 to n-j do
        Cjk ← ϕ
        for l ← 0 to j do
            if prod(Cj-(l+1),k ∪ Cl,j+k-1) <> ϕ then
                Cjk ← prod(Cj-(l+1),k ∪ Cl,j+k-1) {pengisian diambil dari
aturan produksi}
            end if
        end for
    end for
end for
```

Gambar 3.17 *Pseudocode* algoritma CYK

3.1.4.7 Perbaikan Pola dengan Algoritma Levenshtein

Pada tahap ini dilakukan perbaikan terhadap pola kalimat. Perbaikan dilakukan agar dapat diketahui pola yang benar. Pada gambar 3.16 kalimat tidak dapat dikenali sebagai klausa atau kalimat. Sebelum memperbaiki kalimat, *rule* dari sebuah kalimat harus diketahui terlebih dahulu. *Rule* ini diambil dengan memeriksa terlebih dahulu perbaikan pola kalimat oleh seorang pakar dalam bidang tata bahasa. Hasil perbaikan pola kalimat disusun menjadi *rule* untuk dilakukan pencocokan pola kalimat. Algoritma ini akan mencari pola yang paling mendekati dengan pola yang keliru. 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 3.18.

		N	N		N	N		V	B		V	B		N	N		S	C		V	B		I	N		C	D		N	N		Z
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
N	1																															
N	2																															
	3																															
N	4																															
N	5																															
	6																															
V	7																															
B	8																															
	9																															
N	10																															
N	11																															
	12																															
S	13																															
C	14																															
	15																															
V	16																															
B	17																															
	18																															
I	19																															
N	20																															
	21																															
C	22																															
D	23																															
	24																															
N	25																															
N	26																															
	27																															
Z	28																															

Gambar 3.18 Pembentukan tabel dua dimensi algoritma Levenshtein

2. Membandingkan setiap karakter

Hasil dari perbandingan dari gambar 3.18 dapat dilihat pada gambar 3.19.

		N	N		N	N		V	B		V	B		N	N		S	C		V	B		I	N		C	D		N	N		Z
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
N	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
N	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
N	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
N	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
V	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
B	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
	9	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
N	10	9	8	7	6	5	4	3	2	1	1	2	3	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
N	11	10	9	8	7	6	5	4	3	2	2	2	3	3	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	12	11	10	9	8	7	6	5	4	3	3	3	2	3	4	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
S	13	12	11	10	9	8	7	6	5	4	4	4	3	3	4	4	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
C	14	13	12	11	10	9	8	7	6	5	5	5	4	4	4	5	4	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
	15	14	13	12	11	10	9	8	7	6	6	6	5	5	5	4	5	4	3	4	5	6	7	8	9	10	11	12	13	14	15	16
V	16	15	14	13	12	11	10	9	8	7	6	7	6	6	6	5	5	6	4	3	4	5	6	7	8	9	10	11	12	13	14	15
B	17	16	15	14	13	12	11	10	9	8	7	6	7	7	7	6	6	6	5	4	3	4	5	6	7	8	9	10	11	12	13	14
	18	17	16	15	14	13	12	11	10	9	8	7	6	7	8	7	7	7	6	5	4	3	4	5	6	7	8	9	10	11	12	13
I	19	18	17	16	15	14	13	12	11	10	9	8	7	7	8	8	8	8	7	7	7	6	3	4	5	6	7	8	9	10	11	12
N	20	19	18	17	16	15	14	13	12	11	10	9	8	7	7	8	9	9	8	8	8	7	4	3	4	5	6	7	8	9	10	11
	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	8	9	9	9	9	8	8	5	4	3	4	5	6	7	8	9	10
C	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	8	9	9	10	10	9	9	6	5	4	3	4	5	6	7	8	9
D	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	9	9	10	10	11	10	10	7	6	5	4	3	4	5	6	7	8
	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	10	10	10	11	11	10	8	7	6	5	4	3	4	5	6	7
N	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	10	11	11	12	11	9	8	7	6	5	4	3	4	5	6
N	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	10	11	12	12	10	9	8	7	6	5	4	3	4	5
	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	10	11	12	11	10	9	8	7	6	5	4	3	4
Z	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	10	11	12	12	11	10	9	8	7	6	5	4	3

Gambar 3.19 Perbandingan pola kalimat

Pada gambar 3.19 nilai *edit distance* dari perbandingan seluruh karakter adalah 3. Beberapa pola yang dibandingkan dapat dilihat pada tabel 3.10.

Pola	Edit Distance
NN NN VB NN SC VB IN CD NN Z	3
NN NN VB SC NN IN CD NN Z	5
NN NN VB SC NN VB CD NN Z	7

Edit distance dengan nilai terendah akan dijadikan sebagai rekomendasi perbaikan pola kalimat.

3.2 Analisis Kebutuhan Non-fungsional

Analisis kebutuhan non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan dari sistem yang akan digunakan.

3.2.1 Analisis Kebutuhan Perangkat Keras

Spesifikasi minimum dari perangkat keras yang dibutuhkan untuk menjalankan simulator ini dapat dilihat pada tabel 3.10.

Tabel 3.10 Spesifikasi minimum perangkat keras

No	Nama Perangkat Keras
1	*Processor* 1.2 GHz Dual Core
2	RAM 1 GB
3	*Harddisk* 4 GB
4	NIC (*Network Interface Card*)
5	*Mouse with 2 button*
6	*Keyboard*
7	Monitor 19”

3.2.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan agar simulator ini dapat berjalan dengan baik dapat dilihat pada tabel 3.11.

Tabel 3.11 Kebutuhan perangkat lunak

No	Nama Perangkat Lunak
1	Sistem operasi *Windows* XP
2	NET *Framework* 4.0

3.2.3 Analisis Kebutuhan Pengguna

Selain perangkat keras dan perangkat lunak, pengguna juga harus memenuhi kriteria-kriteria seperti pada tabel 3.12 agar simulator dapat dijalankan dengan baik. Pengguna disini merupakan penguji yang melakukan pengujian terhadap saran yang ditampilkan oleh sistem dalam perbaikan kesalahan kata dan pola kalimat.

Tabel 3.12 Analisis kebutuhan pengguna

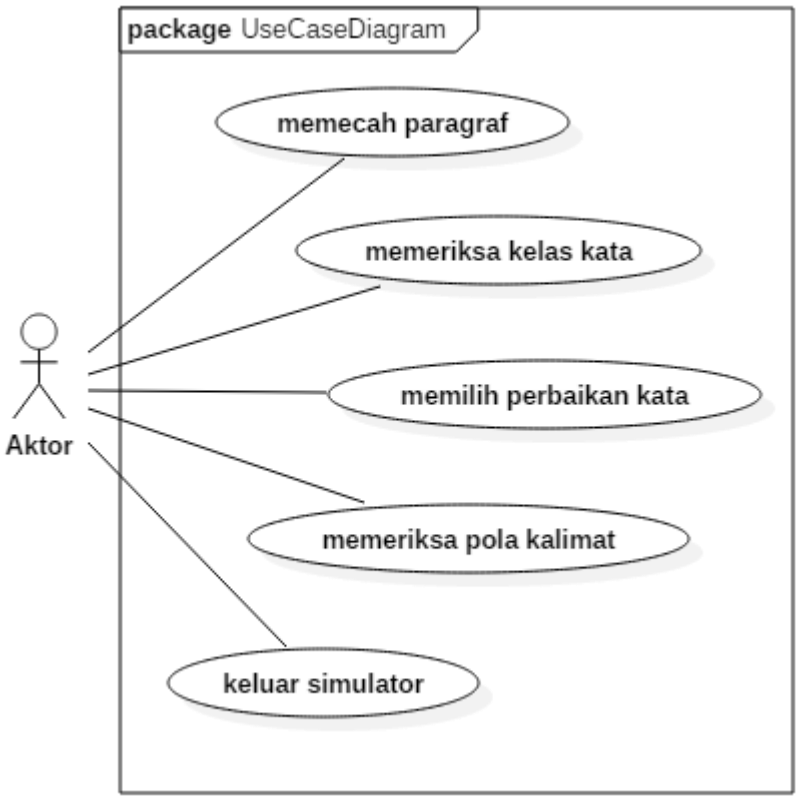
No	Kemampuan
1	Mengoperasikan sistem operasi *Windows*
2	Mampu menggunakan *mouse* dan *keyboard*
3	Tidak buta huruf

3.3 Analisis Kebutuhan Fungsional

Pada tahap ini dilakukan analisis kebutuhan fungsional yang akan digunakan dalam merancang simulator. Perancangan sistem digambarkan dengan alat bantu UML (*Unified Modelling Language*).

3.3.1 *Use Case Diagram*

Use case digunakan untuk memodelkan alur atau proses bisnis yang menggambarkan interaksi antara aktor dengan sistem. Pengguna akan berperan sebagai aktor. Diagram use case dapat dilihat pada gambar 3.20.



Gambar 3.20 *Use Case Diagram*

3.3.1.1 Definisi Aktor

Pendefinisian aktor pada simulator pengujian algoritma CYK dan Levenshtein ini dapat dilihat pada tabel 3.13.

Tabel 3.13 Definisi aktor

No.	Aktor	Deskripsi
1	Aktor	Pengguna yang melakukan pengujian perbaikan kesalahan kata dan poa kalimat dokumen abstrak

3.3.1.2 Definisi Use Case

Pendefinisian *use case* pada simulator pengujian algoritma CYK dan Levenshtein ini dapat dilihat pada tabel 3.14.

Tabel 3.14 Definisi *use case*

No.	Use Case	Deskripsi
1	memecah paragraf	Proses memecah paragraf menjadi kalimat, kalimat menjadi kata, dan menampilkan keseluruhan hasil pemecahan.
2	memeriksa kelas kata	Proses mengelompokkan kata untuk mengenali kelas kata dan memperbaikinya.
3	memilih perbaikan kata	Proses memilih kata dari perbaikan yang disarankan.
4	memeriksa pola kalimat	Proses memeriksa pola kalimat dan memperbaiki pola kalimat dari paragraf yang diperiksa untuk dijadikan saran perbaikan.
6	keluar simulator	Proses keluar dari simulator yang dibuat.

3.3.2 *Use Case Scenario*

Use case scenario digunakan untuk menunjukkan skenario proses yang ada pada *use case*. Pengguna memberikan perintah dan sistem akan merespon aksi dari pengguna. Tabel skenario dapat dilihat pada tabel 3.15.

Tabel 3.15 *Use Case Scenario* memecah paragraf

Nama Use Case	Memecah paragraf	
Aktor	Aktor	
Deskripsi	Proses mengunggah dokumen dan memecah paragraf yang ada pada dokumen menjadi satuan terkecil yaitu kata	
Pre-condition	File dokumen belum dimasukkan	
Post-condition	Menampilkan kata-kata yang sudah dipecah	
Skenario Utama		
Aksi Aktor		Reaksi Sistem
1. Memasukkan file dokumen		
		2. Memeriksa dokumen
		3. Memecah paragraf menjadi kalimat
		4. Memecah kalimat menjadi kata
		5. Menampilkan seluruh kata yang sudah dipecah
Aksi Aktor		Reaksi Sistem
		3a. Menampilkan pesan ekstensi dokumen tidak valid

Tabel 3.16 *Use Case Scenario* memeriksa kelas kata

Nama Use Case	Memeriksa kelas kata
Aktor	Aktor
Deskripsi	Proses memeriksa kelas kata (*tagging*)
Pre-condition	Kumpulan kata belum dimasukkan
Post-condition	Menampilkan perbaikan kata
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Memasukkan kumpulan kata	
	2. Mengelompokkan kata berdasarkan POS Tag
	3. Memeriksa kelas kata
	4. Memperbaiki kata yang mendapatkan *tag* X dengan cara membandingkannya dengan kata yang ada pada kamus
	5. Menghitung nilai *edit distance* dari kata yang dibandingkan
	6. Menampilkan kata dengan perbaikan kata yang memiliki nilai *edit distance* terendah
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
	4a. Menampilkan pesan bahwa tidak ada kata yang diperbaiki

Tabel 3.17 *Use Case Scenario* memilih perbaikan kata

Nama Use Case	Memilih perbaikan kata	
Aktor	Aktor	
Deskripsi	Proses memilih kata dari perbaikan yang disarankan	
Pre-condition	Daftar perbaikan kata ditampilkan	
Post-condition	Mengganti kata yang keliru dengan kata yang diperbaiki	
Skenario Utama		
Aksi Aktor		Reaksi Sistem
1. Memilih perbaikan kata		
		2. Memeriksa kata yang dipilih
		3. Mengganti kata yang sudah dipilih untuk perbaikan.
Skenario Alternatif		
Aksi Aktor		Reaksi Sistem
		3a. Menampilkan pesan *error* bahwa kata harus dipilih terlebih dahulu

Tabel 3.18 *Use Case Scenario* memeriksa pola kalimat

Nama Use Case	Memeriksa pola kalimat	
Aktor	Aktor	
Deskripsi	Proses memeriksa pola kalimat dan memberikan saran perbaikan apabila terdapat kesalahan	
Pre-condition	Kelas kata belum dimasukkan	
Post-condition	Menampilkan saran perbaikan pola kalimat	
Skenario Utama		
Aksi Aktor		Reaksi Sistem
1. Memasukkan kelas kata		
		2. Menyusun kelas kata menjadi pola kalimat
		3. Memeriksa susunan pola kalimat
		4. Memperbaiki pola dengan cara membandingkan susunan pola kalimat dengan daftar pola kalimat
		5. Menghitung nilai *edit distance* dari pola kalimat yang dibandingkan
		6. Menampilkan saran perbaikan pola kalimat yang memiliki *edit distance* terendah
Skenario Alternatif		
Aksi Aktor		Reaksi Sistem
		4a. Menampilkan pesan bahwa pola kalimat sudah benar

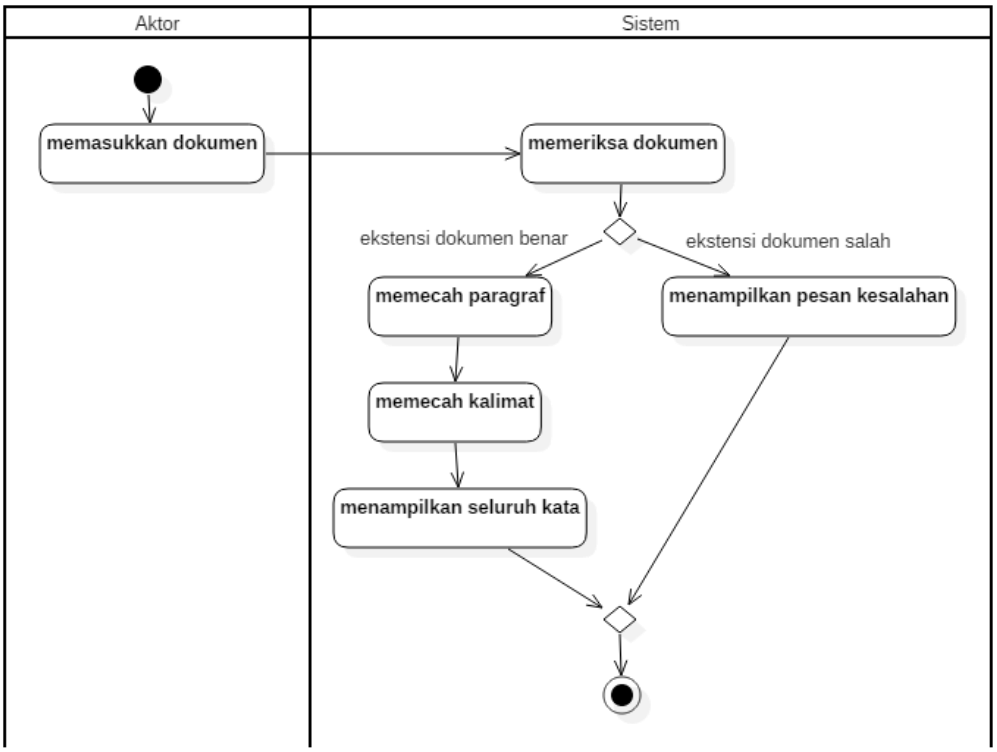
Tabel 3.19 *Use Case Scenario* keluar simulator

Nama Use Case	Keluar Simulator	
Aktor	Aktor	
Deskripsi	Proses menutup simulator	
Pre-condition	Tampilan simulator sudah dibuka	
Post-condition	Menutup simulator	
Skenario Utama		
Aksi Aktor		Reaksi Sistem
1. Keluar simulator		
		2. Menampilkan pertanyaan untuk konfirmasi keluar dari simulator
		3. Simulator ditutup
Skenario Alternatif		
Aksi Aktor		Reaksi Sistem
		3a. Menutup tampilan konfirmasi dan mengembalikan user ke tampilan utama simulator

3.3.3 *Activity Diagram*

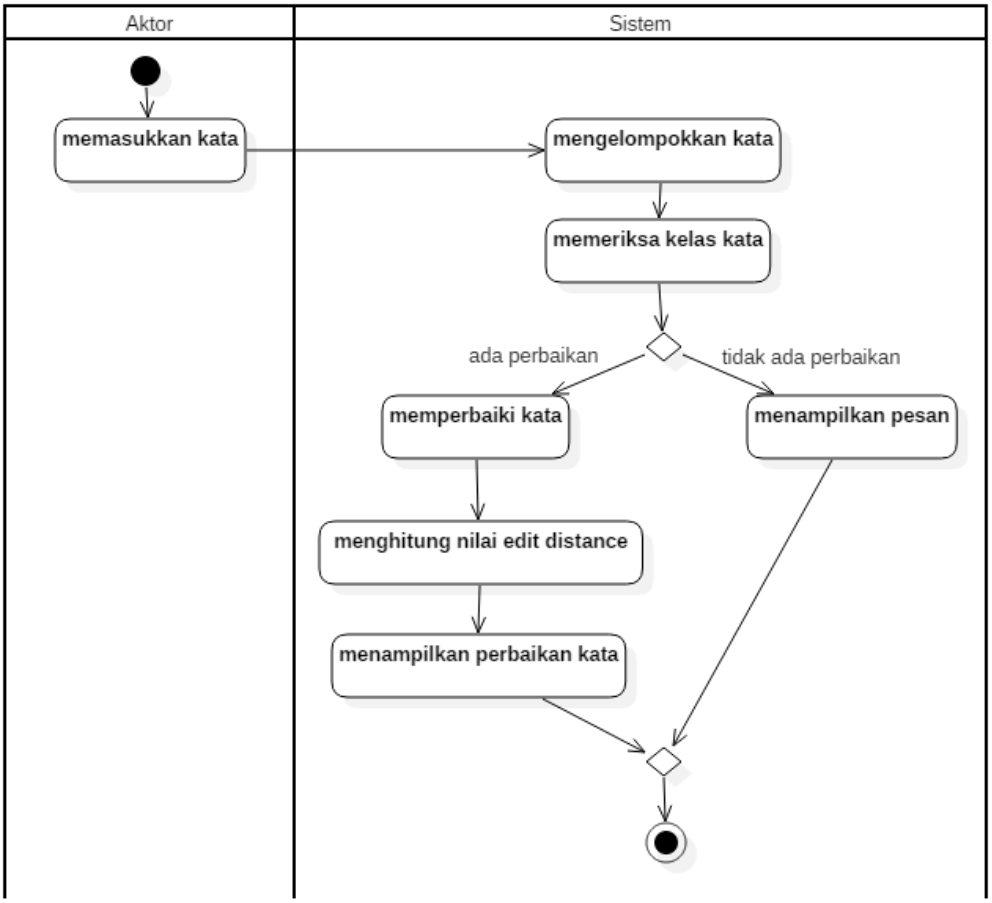
Berikut ini *activity diagram* berdasarkan *use case* yang telah digambarkan pada gambar 3.21.

3.3.3.1 *Activity* Memecah Paragraf



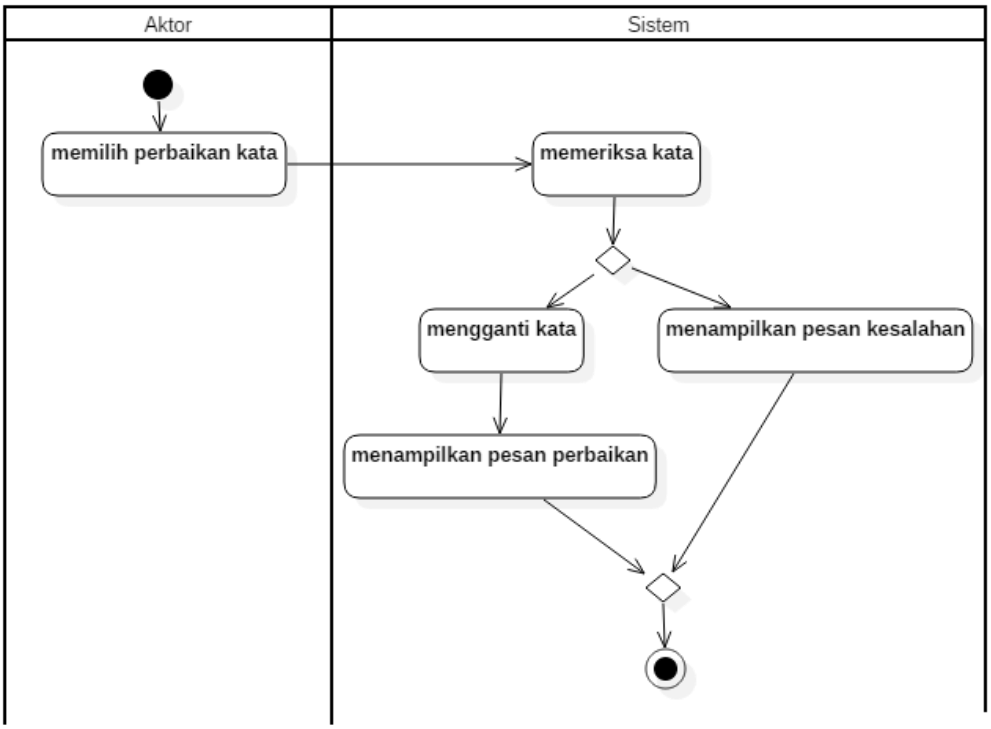
Gambar 3.21 *Activity* memecah paragraf

3.3.3.2 *Activity* Memeriksa Kelas Kata



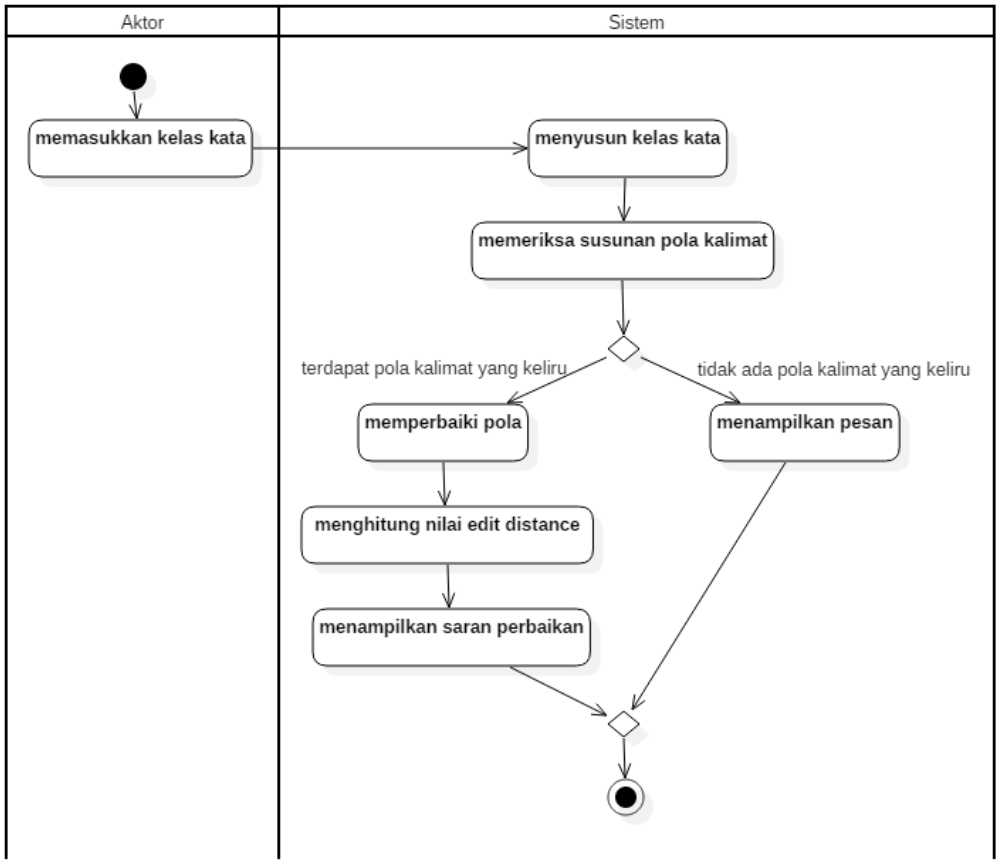
Gambar 3.22 *Activity* memeriksa kelas kata

3.3.3.3 *Activity* Memilih Perbaikan Kata



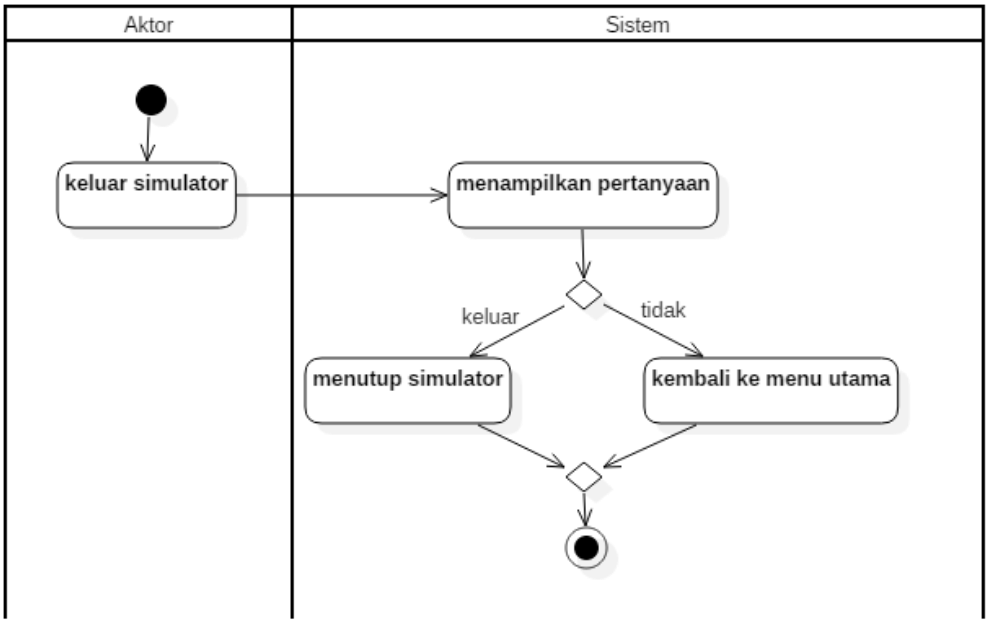
Gambar 3.23 *Activity* memilih perbaikan kata

3.3.3.4 *Activity* Memeriksa Pola Kalimat



Gambar 3.24 *Activity* memeriksa pola kalimat

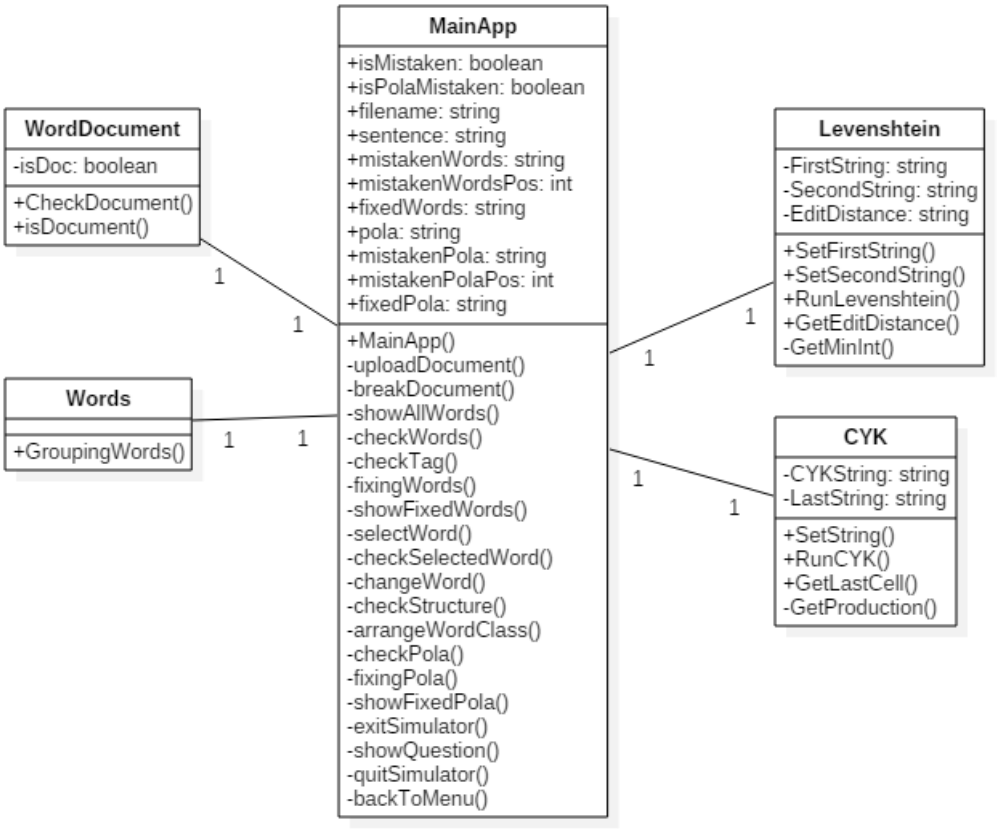
3.3.3.6 Activity Keluar Simulator



Gambar 3.25 *Activity* keluar simulator

3.3.4 *Class Diagram*

Berikut ini *class diagram* yang dapat digambarkan berdasarkan *use case* yang sudah dibuat. *Class diagram* dapat dilihat pada gambar 3.26.



Gambar 3.26 *Class diagram*

Keterangan dari *class diagram* pada gambar 3.33 dapat dilihat pada tabel 3.20.

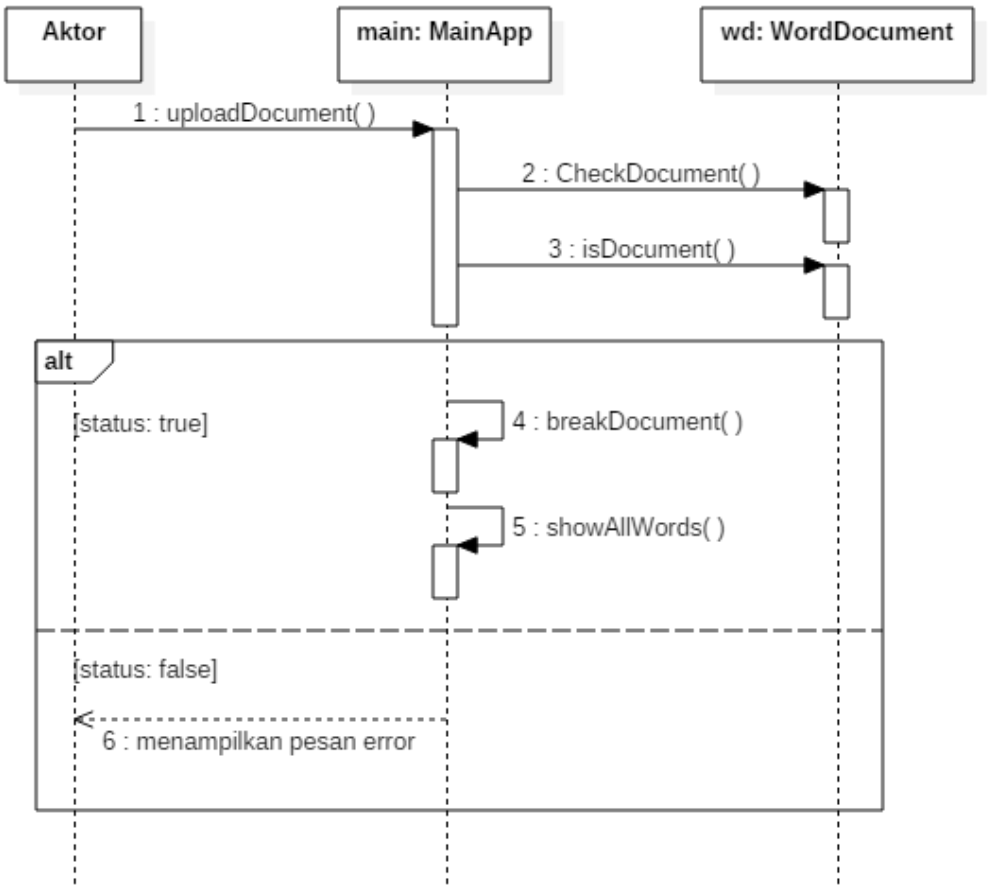
Tabel 3.20 Keterangan *class diagram*

Nama Kelas	Keterangan
MainApp	merupakan kelas utama yang mengatur alur dari perbaikan kata dan kalimat pada dokumen
WordDocument	merupakan kelas yang menangani proses pengolahan dokumen
Words	merupakan kelas yang menangani proses pembagian kelompok kata
Levenshtein	merupakan kelas yang menangani perbaikan kata dan pola kalimat
CYK	merupakan kelas yang menangani pengenalan kalimat

3.3.5 *Sequence Diagram*

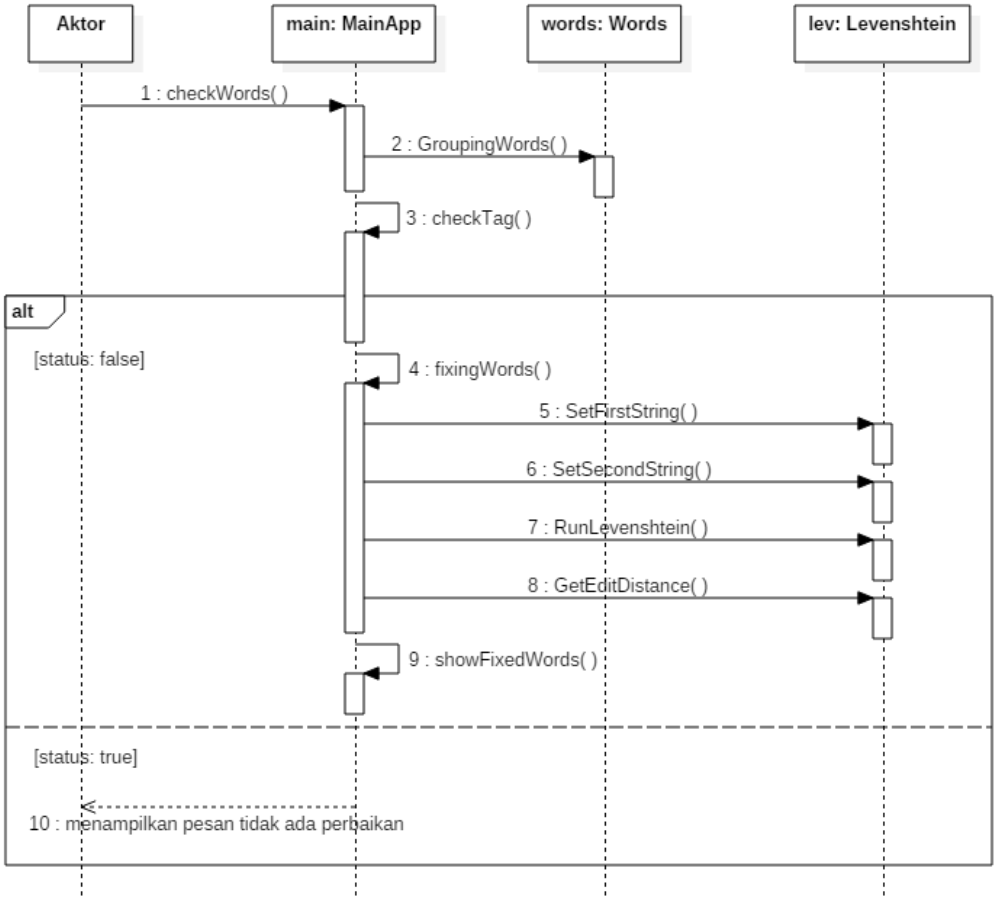
Sequence diagram menggambarkan alur dari tingkah laku objek dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Berdasarkan *use case* dan *class diagram*, *sequence diagram* dapat digambarkan pada gambar 3.27.

3.3.5.1 *Sequence Diagram* Memecah Paragraf



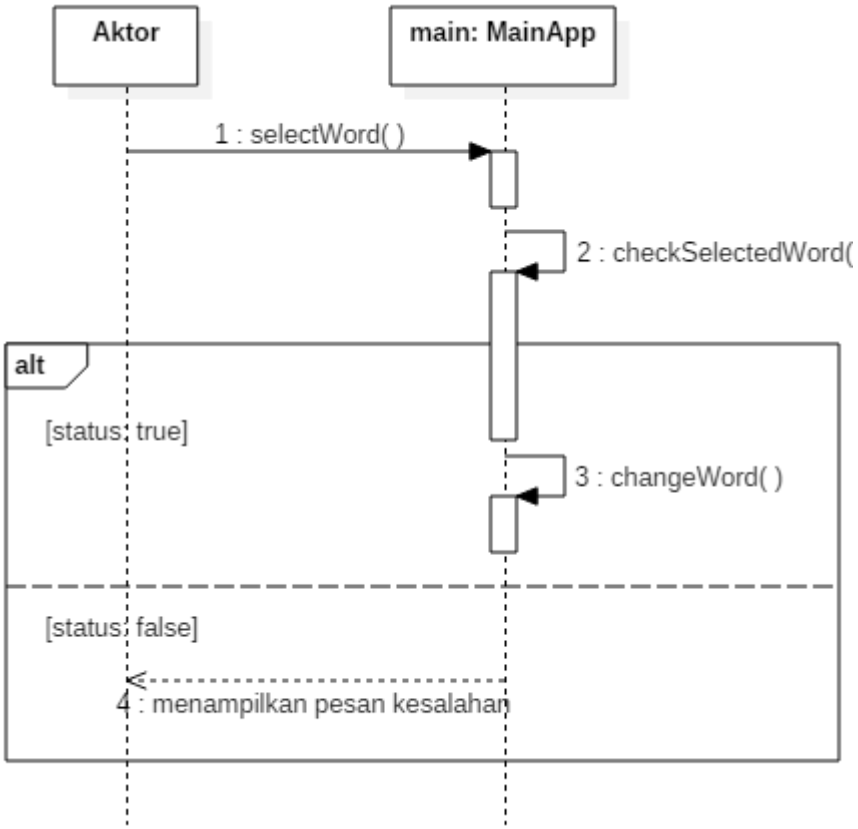
Gambar 3.27 *Sequence diagram* memecah paragraf

3.3.5.2 *Sequence Diagram* Memeriksa Kelas Kata



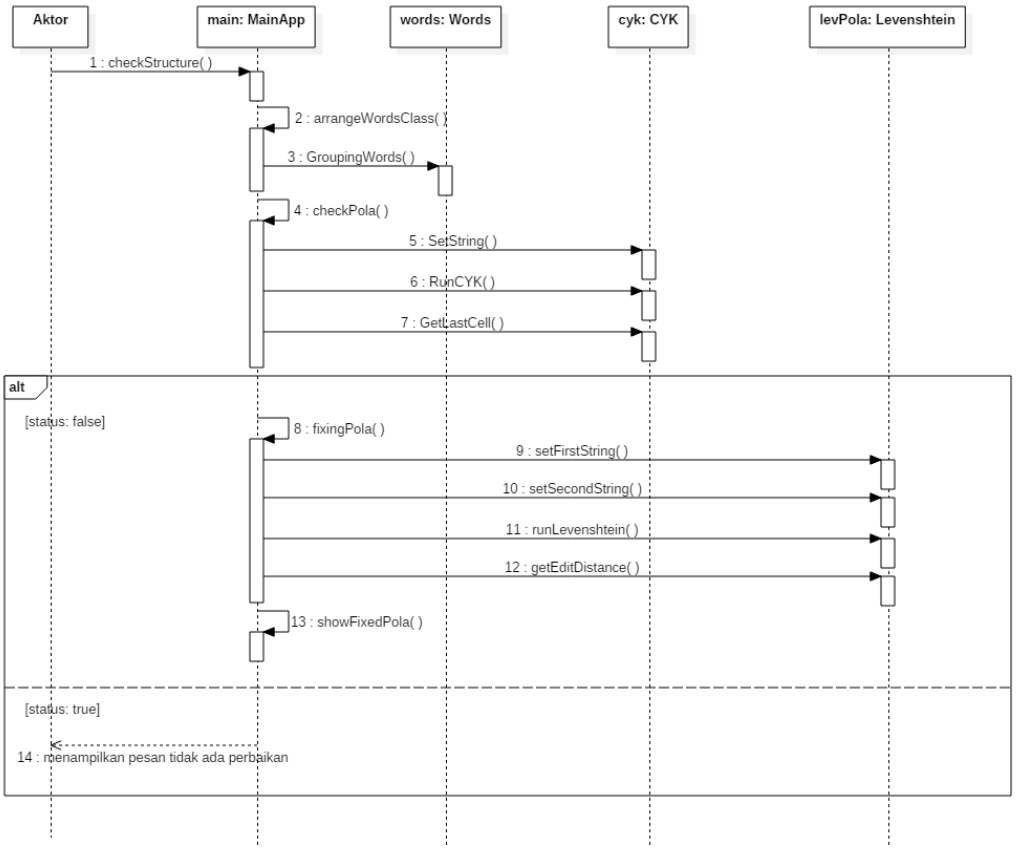
Gambar 3.28 *Sequence diagram* memeriksa kelas kata

3.3.5.3 *Sequence Diagram* Memilih Perbaikan Kata



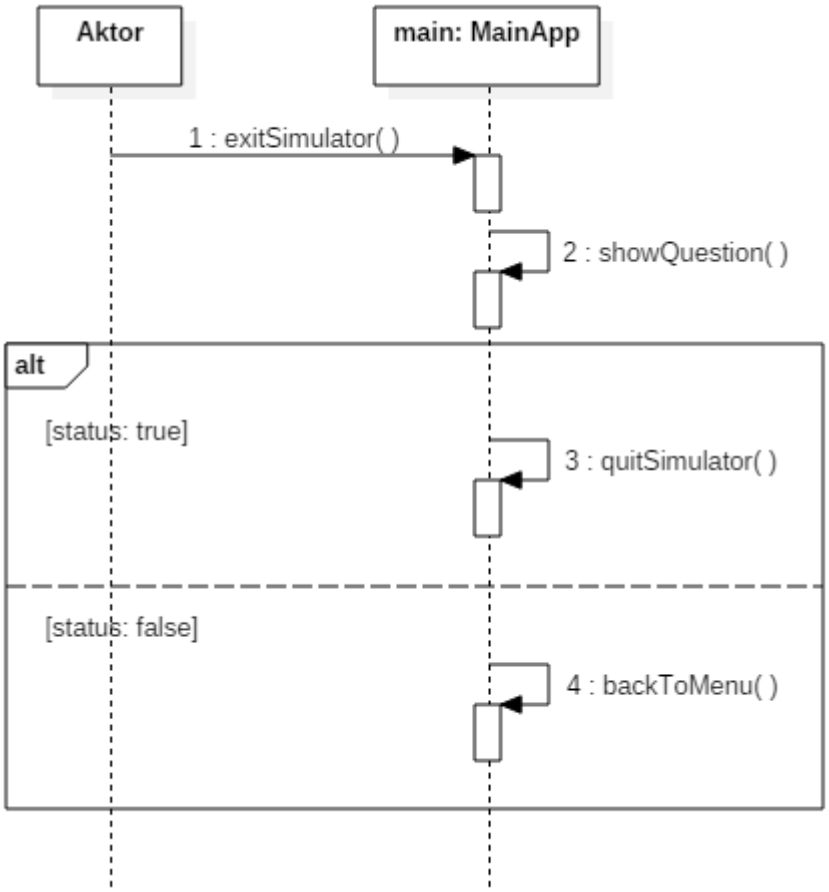
Gambar 3.29 *Sequence diagram* memilih perbaikan kata

3.3.5.4 *Sequence Diagram* Memeriksa Pola Kalimat



Gambar 3.30 *Sequence diagram* memeriksa pola kalimat

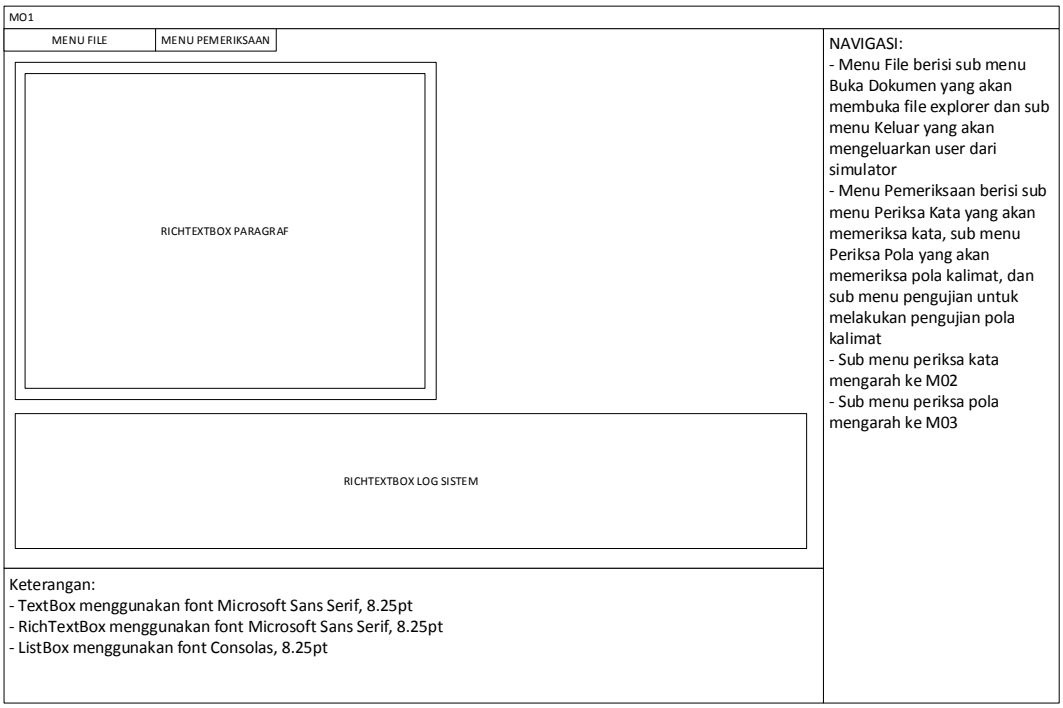
3.3.5.6 *Sequence Diagram* Keluar Simulator



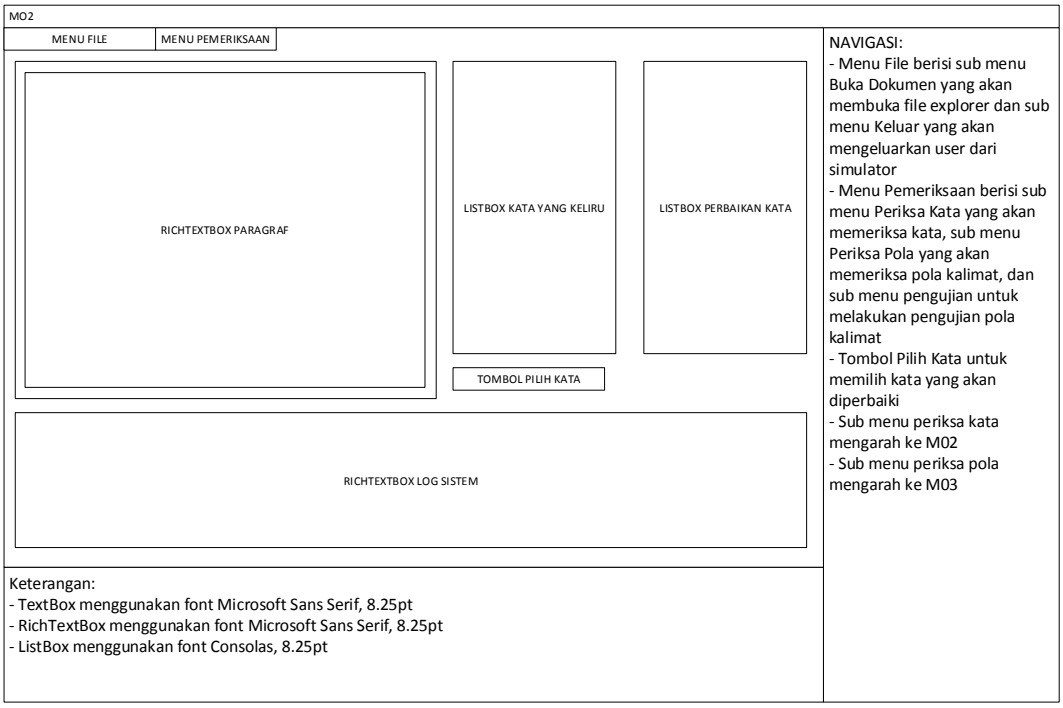
Gambar 3.31 *Sequence Diagram* Keluar Simulator

3.3.6 Perancangan Antarmuka

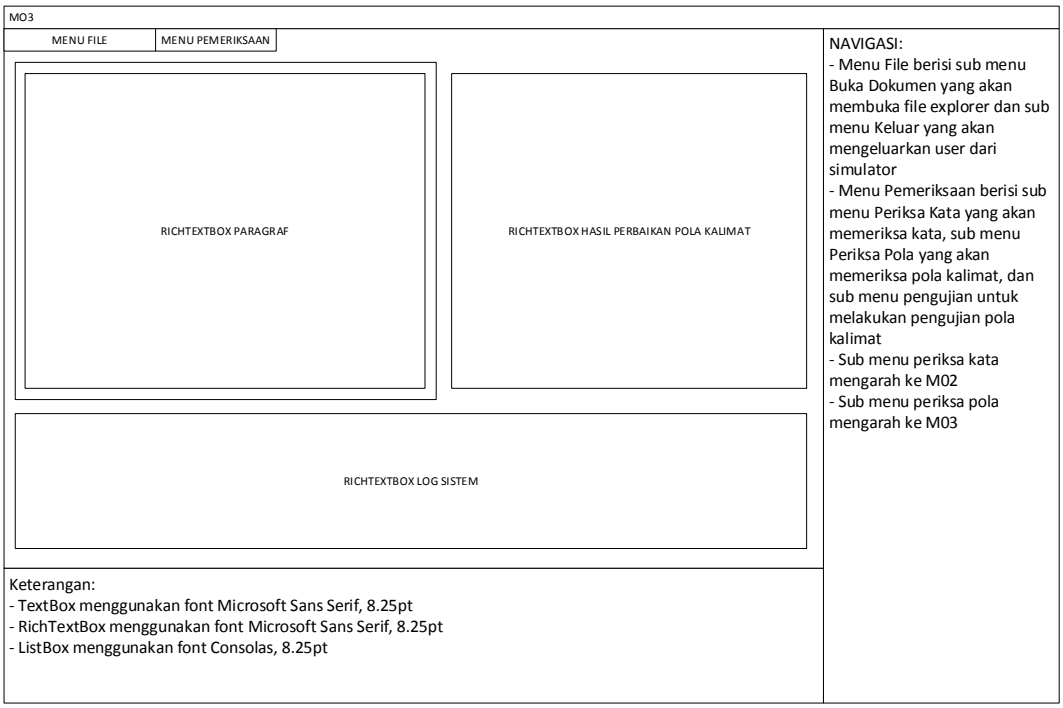
Berikut ini perancangan antarmuka dari simulator yang akan dibangun. Gambar perancangan antarmuka dapat dilihat pada gambar 3.32.



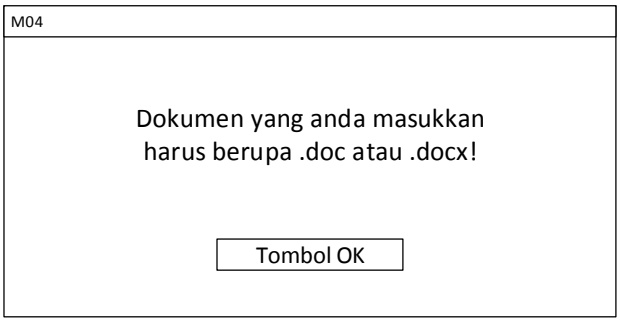
Gambar 3.32 Perancangan antarmuka tampilan utama



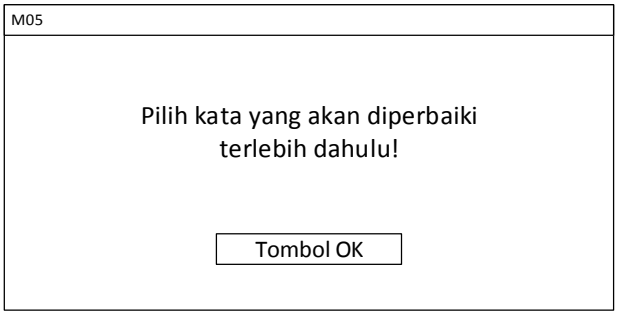
Gambar 3.33 Perancangan antarmuka perbaikan kata



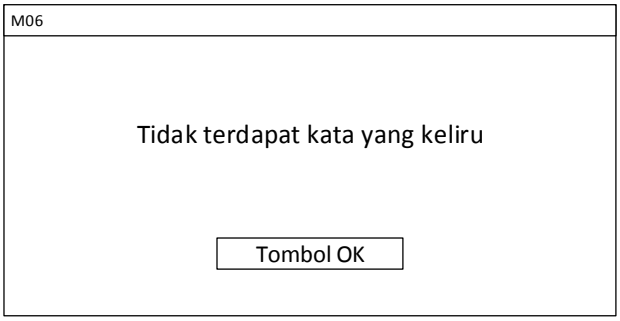
Gambar 3.34 Perancangan antarmuka perbaikan pola



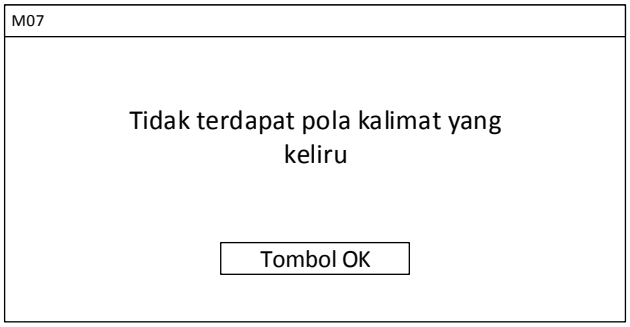
Gambar 3.35 Pesan kesalahan dokumen



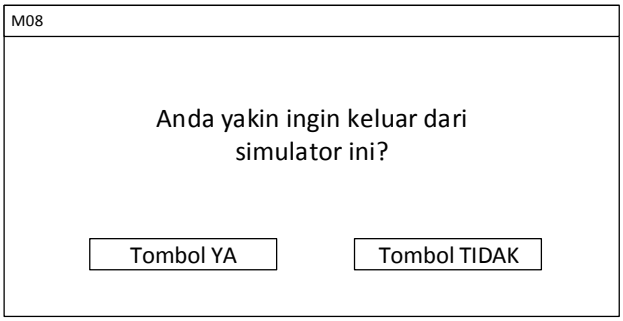
Gambar 3.36 Pesan kesalahan pemilihan kata



Gambar 3.37 Pesan pemeriksaan kata



Gambar 3.38 Pesan perbaikan pola kalimat



Gambar 3.39 Pesan peringatan keluar simulator