BAB 3
ANALISIS DAN PERANCANGAN

3.1. Analisis Masalah

Banyak proses NLP yang membutuhkan data masukan kata yang benar. Jika terdapat kesalahan ejaan kata dalam data masukan, proses-proses tersebut tidak akan berjalan dengan baik. Sebagai contoh, dalam *information retrieval*, jika ada ejaan kata yang salah dalam *query* yang dimasukkan, maka informasi yang ditemukan akan berbeda dari yang diharapkan. Dalam *POS-tagging*, ejaan kata yang salah akan menyebabkan salahnya pengelompokan kata tersebut. Adapun dalam *machine translation*, salahnya ejaan kata akan membuat terjemahan kata tersebut salah pula. Oleh karena itu deteksi dan koreksi kesalahan ejaan kata sangat diperlukan dalam bidang NLP.

Ada dua jenis kesalahan ejaan. Yang pertama adalah kesalahan ejaan *non-word*, yaitu kesalahan ejaan kata dimana kata tersebut tidak terdapat dalam kamus. Yang kedua adalah kesalahan ejaan *real-word*, dimana kata yang salah merupakan kata valid yang terdapat dalam kamus, namun secara konteks bukan merupakan kata yang dimaksud. Contohnya, dalam potongan kalimat

"... saya makan basi goreng ...,"

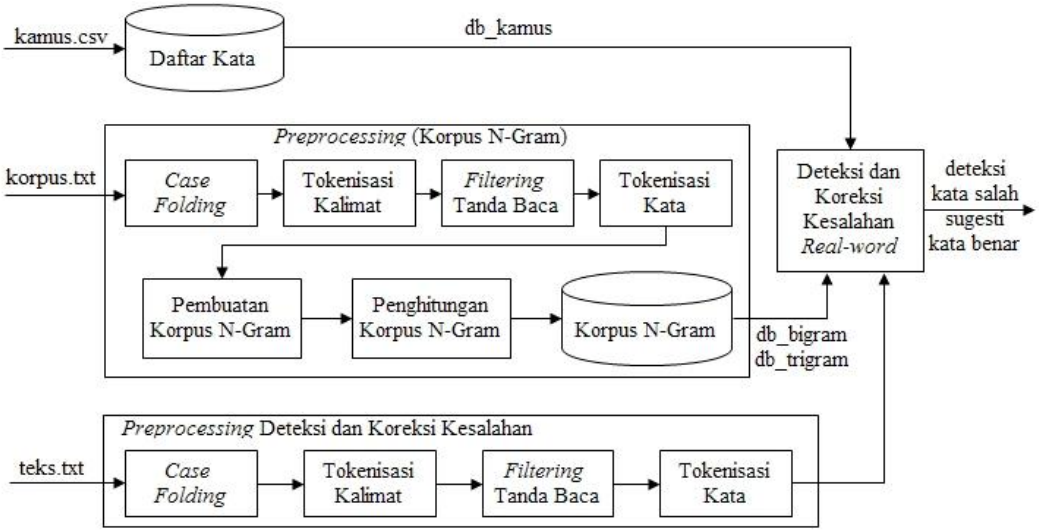
kata "basi" terdapat dalam kamus bahasa Indonesia, namun merupakan kesalahan ejaan dari kata "nasi". Dalam bahasa Indonesia, jenis kesalahan *non-word* sudah banyak diteliti, sedangkan kesalahan *real-word* masih dibutuhkan banyak pengembangan. Oleh karena itu dibutuhkan suatu metode untuk mendeteksi dan mengkoreksi kesalahan kata *real-word* dalam bahasa Indonesia.

Untuk menjawab permasalahan di atas, penulis berniat untuk mengambil suatu metode deteksi dan koreksi kesalahan *real-word* yang sudah ada dan menyesuaikannya untuk bahasa Indonesia. Beberapa metode yang sudah diajukan di antaranya Mays *et al.* [8][1], Hirst dan Budanitsky [3], Peddler [4], dan Samanta dan Chaudhuri [2]. Dari metode-metode di atas, metode bigram dan trigram kata lokal yang diajukan Samanta dan Chaudhuri merupakan metode yang paling menjanjikan, dengan akurasi deteksi yang tinggi (*precision* 71%-79%,

recall 81%-88%) dan kemampuan untuk mendeteksi lebih dari satu kesalahan dalam satu kalimat. Oleh karena itu, dalam penelitian ini akan dilakukan penerapan metode bigram dan trigram kata lokal untuk mendeteksi dan mengkoreksi kesalahan ejaan *real-word* dalam bahasa Indonesia.

3.2. Analisis Solusi

Dalam penelitian ini akan dibangun sebuah sistem untuk mendeteksi dan mengkoreksi kesalahan kata *real-word* dalam bahasa Indonesia dengan menerapkan metode bigram dan trigram kata lokal. Berikut adalah gambaran umum sistem yang akan dibangun.



Gambar 3.1 Gambaran Umum Sistem

3.2.1 Analisis Data Masukan

Data masukan yang digunakan dalam penelitian ini adalah daftar kata, teks korpus dan data uji. Berikut adalah penjelasan masing-masing data masukan.

3.2.1.1 Daftar Kata

Daftar kata dibutuhkan untuk membuat *confusion set*. *Confusion set* adalah himpunan kata-kata yang biasa tertukar satu sama lain [1]. Daftar kata diambil dari *database* Kamus Besar Bahasa Indonesia [21] yang berformat .csv. Berikut adalah contoh potongan *database* Kamus Besar Bahasa Indonesia.

Tabel 3.1 *Database* Kamus Besar Bahasa Indonesia

Kata	Jenis Kata	Bentuk Kata
a	n	Dasar
ab	n	Dasar
aba	n	Dasar
aba-aba	n	Dasar
abad	n	Dasar
abad kekal	n	Majemuk
abad keemasan	n	Majemuk
abad pertengahan	n	Majemuk
berabad-abad	v	Dasar
seabad	n	Dasar
abadi	a	Dasar
.....	.....	.....

Daftar kata dibuat dengan mengambil kolom Kata dari *database* Kamus Besar Bahasa Indonesia di atas. Kata yang diambil hanyalah kata dasar saja, dan dilakukan *filtering* dengan menghapus tanda strip. Berikut adalah potongan daftar kata yang dihasilkan.

Tabel 3.2 Daftar Kata

Daftar Kata
a
ab
aba
abaaba
abad
berabadabad
seabad
abadi
.....

Daftar kata tersebut kemudian disimpan dalam *database*. Daftar kata ini akan dipanggil saat sistem akan membuat *confusion set* pada tahap Deteksi dan Koreksi Kesalahan Kata.

3.2.1.2 Teks Korpus

Teks Korpus digunakan untuk membuat korpus n-gram. Teks korpus diambil dari korpus-korpus *UI Tagged Corpus* [22] dan *PAN-BPPT Localization* [23]. *PAN-BPPT Localization* adalah kumpulan dokumen yang berisi teks-teks berita dalam bidang ekonomi, internasional, iptek, dan olahraga dalam bahasa Indonesia dan bahasa Inggris. Setiap dokumen berformat .txt dan berisi antara 100 ribu – 150 ribu kata. Berikut contoh potongan dokumen *PAN-BPPT Localization*.

Tabel 3.3 Potongan Dokumen *PAN-BPPT Localization*

Menteri Keuangan Sri Mulyani mengatakan koreksi tajam pada Indeks Harga Saham Gabungan IHSG hingga sekitar 4 persen dalam perdagangan Rabu 10/1 hanya efek sesaat dari faktor-faktor regional seperti penurunan harga komoditi perkebunan dan krisis finansial di Thailand. Dalam jumpa pers bersama Dirut Bursa Efek Jakarta BEJ, Erry Firmansyah di gedung Depkeu Jakarta, Rabu malam, Menkeu menjelaskan beberapa faktor ekonomi Indonesia justru membaik. Kita melihat faktor inflasi dari makanan akan segera hilang yang terkonfirmasi dari harga beras di semua propinsi, katanya. Demikian pula dengan harga minyak. .....

Sementara itu, dokumen *UI Tagged Corpus* adalah kumpulan teks berbahasa Indonesia beserta *POS tag* untuk setiap katanya. Dokumen *UI Tagged Corpus* berformat .txt dan berisi 1 juta kata. Berikut adalah contoh potongan dokumen *UI Tagged Corpus*.

Tabel 3.4 Potongan Dokumen *UI Tagged Corpus*

Indeks/nn biaya/nnc tenaga/nnu kerja/nnu sektor/nnc swasta/jj secara/in keseluruhan/nn ,/, termasuk/vbt gaji/nn dan/cc manfaat/nnu -LRB-/nn benefit/nn -RRB-/nn ,/, naik/nn 4.7/nnp %/sym dalam/in 12/cdp bulan/nnc berakhir/vbi pada/in September/nn ,/, dibandingkan/vbi dengan/in 4.5/fw %/sym untuk/in 12/cdp bulan/nnc yang/sc berakhir/vbi pada/in Juni/nn dan/cc 12/cdp bulan/nnc yang/sc berakhir/vbi pada/in September/nn 1988/cdp ./.

Untuk kepentingan penelitian, yang akan digunakan hanyalah *plaintext*-nya saja tanpa *POS tag*. Berikut adalah potongan dokumen *UI Tagged Corpus* yang telah dijadikan *plaintext.*

Tabel 3.5 Potongan Dokumen *Plaintext UI Corpus*

Indeks biaya tenaga kerja sektor swasta secara keseluruhan, termasuk gaji dan manfaat (benefit), naik 4.7% dalam 12 bulan berakhir pada September, dibandingkan dengan 4.5% untuk 12 bulan yang berakhir pada Juni dan 12 bulan yang berakhir pada September 1988.

3.2.1.3 Data Uji

Belum ada korpus uji berbahasa Indonesia yang dapat digunakan untuk menguji metode deteksi dan koreksi kesalahan kata *real-word* dalam bahasa Indonesia, oleh karena itu pengujian akan disimulasikan dengan menggunakan dokumen-dokumen yang telah ditambahkan kesalahan kata secara sintetis [2]. Dalam tiap dokumen, 1 dari 20 kata dipilih secara acak. Untuk setiap kata terpilih, dibuatlah himpunan *string* yang dibuat menggunakan *single edit operation* (*insert, delete, substitute*). Kemudian dari himpunan tersebut, dipilihlah satu kata valid (terdapat dalam kamus) secara acak. Kata ini kemudian menjadi kesalahan ejaan *real-word*. Sebagai contoh, tabel 3.6 dan 3.7 berikut adalah contoh potongan

sebuah dokumen berita sebelum dan sesudah menjadi data uji. Kata yang digarisbawahi adalah kata yang terpilih menjadi kesalahan *real-word.*

Tabel 3.6 Potongan Teks Berita Sebelum Menjadi Dokumen Uji

Kita telah mengetahui segi kimia yang menjadikan kunang-kunang bercahaya, tapi kita kini mendapatkan jawaban dari teka-teki yang selama ini tak terjawab yang menjelaskan bagaimana mereka mampu menghidupkan dan mematikan saklarnya. Cahaya kunang-kunang berperan pula sebagai tanda peringatan, untuk memperingatkan antar sesama jenisnya tentang ancaman bahaya, maupun peringatan bagi serangga dan burung pemangsa agar tidak memakannya.

Tabel 3.7 Potongan Teks Berita Setelah Menjadi Dokumen Uji

Kita telah mengetahui segi kimia yang menjadikan kunang-kunang bercahaya, tapi kita kini mendapatkan jawatan dari teka-teki yang selama ini tak terjawab yang menjelaskan bagaimana mereka mampu menghidupkan dan mematikan saklarnya. Cahaya kunang-kunang berperan pula sebagai tandu peringatan, untuk memperingatkan antar sesama jenisnya tentang ancaman bahaya, maupun peringatan bagi serangga dan burung pemangsa agar tidak memakannya.

Dari pembuatan dokumen uji di atas, terdapat dua buah kesalahan kata *real-word* yaitu kata "jawatan" (harusnya "jawaban") dan kata "tandu" (harusnya "tanda").

3.2.2 Analisis Proses

Analisis proses yang dilakukan adalah *preprocessing* pembuatan korpus n-gram, *preprocessing* deteksi dan koreksi kesalahan, dan proses deteksi dan koreksi kesalahan *real-word.* Berikut penjelasannya.

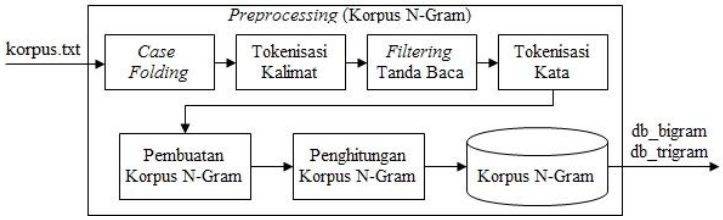
3.2.2.1 *Preprocessing* Pembuatan Korpus N-Gram

Korpus n-gram digunakan untuk menghitung nilai probabilitas suatu kata. Korpus n-gram berisi daftar n-gram beserta jumlah kemunculan n-gram tersebut dalam semua dokumen sumber. Berikut adalah contoh potongan teks yang akan dijadikan korpus n-gram.

Tabel 3.8 Contoh Teks Korpus

Cahaya kunang-kunang berperan pula sebagai tanda peringatan, untuk memperingatkan antar sesama jenisnya tentang ancaman bahaya, maupun peringatan bagi serangga dan burung pemangsa agar tidak memakannya. Para peneliti juga terilhami menggunakan teknologi pencahayaan kunang-kunang untuk berbagai hal, termasuk untuk memantau baik tidaknya teknik pengobatan baru bekerja. Sejak Thomas Edison hingga kini, tak ada teknologi lampu listrik yang menyamai cahaya kunang-kunang yang dingin.

Langkah-langkah membuat korpus n-gram adalah sebagai berikut.



Gambar 3.2 Diagram *Preprocessing* Pembuatan Korpus N-Gram

1. *Case Folding*

Pertama lakukan *case folding* dengan mengubah semua huruf besar menjadi huruf kecil. Hasilnya adalah sebagai berikut.

Tabel 3.9 Proses *Case Folding*

Sebelum	Cahaya kunang-kunang berperan pula sebagai tanda peringatan, untuk memperingatkan antar sesama jenisnya tentang ancaman bahaya, maupun peringatan bagi serangga dan burung pemangsa agar tidak memakannya. Para peneliti juga terilhami menggunakan teknologi pencahayaan kunang-kunang untuk berbagai hal, termasuk untuk memantau baik tidaknya teknik pengobatan baru bekerja. Sejak Thomas Edison hingga kini, tak ada teknologi lampu listrik yang menyamai cahaya kunang-kunang yang dingin.
Sesudah	**c**ahaya kunang-kunang berperan pula sebagai tanda peringatan, untuk memperingatkan antar sesama jenisnya tentang ancaman bahaya, maupun peringatan bagi serangga dan burung pemangsa agar tidak memakannya. **p**ara peneliti juga terilhami menggunakan teknologi pencahayaan kunang-kunang untuk berbagai hal, termasuk untuk memantau baik tidaknya teknik pengobatan baru bekerja. **s**ejak **t**homas **e**dison hingga kini, tak ada teknologi lampu listrik yang menyamai cahaya kunang-kunang yang dingin.

Huruf-huruf yang ditebalkan dan digarisbawahi adalah huruf-huruf yang sudah diubah dari huruf besar ke huruf kecil.

2. Tokenisasi Kalimat

Selanjutnya lakukan tokenisasi kalimat dengan memisahkan tiap-tiap kalimat. Berikut adalah hasilnya.

Tabel 3.10 Tokenisasi Kalimat

Sebelum	Sesudah	
cahaya kunang-kunang berperan pula sebagai tanda peringatan, untuk memperingatkan antar sesama jenisnya tentang ancaman bahaya, maupun peringatan bagi serangga dan burung pemangsa agar tidak memakannya. para peneliti juga terilhami menggunakan teknologi pencahayaan kunang-kunang untuk berbagai hal, termasuk untuk memantau baik tidaknya teknik pengobatan baru bekerja. sejak thomas edison hingga kini, tak ada teknologi lampu listrik yang menyamai cahaya kunang-kunang yang dingin.	1	cahaya kunang-kunang berperan pula sebagai tanda peringatan, untuk memperingatkan antar sesama jenisnya tentang ancaman bahaya, maupun peringatan bagi serangga dan burung pemangsa agar tidak memakannya.
	2	para peneliti juga terilhami menggunakan teknologi pencahayaan kunang-kunang untuk berbagai hal, termasuk untuk memantau baik tidaknya teknik pengobatan baru bekerja.
	3	sejak thomas edison hingga kini, tak ada teknologi lampu listrik yang menyamai cahaya kunang-kunang yang dingin.

Pada tabel di atas, teks ditokenisasi dengan memisahkan kalimat-kalimat yang ditandai dengan tanda titik (.). Teks ditokenisasi menjadi tiga kalimat.

3. *Filtering* Tanda Baca

Setelah itu dilakukan penghilangan tanda baca. Di sini tanda baca adalah karakter selain karakter huruf, angka atau spasi. Berikut adalah hasilnya.

Tabel 3.11 *Filtering* Tanda Baca

Sebelum		Sesudah	
1	cahaya kunang-kunang berperan pula sebagai tanda peringatan, untuk memperingatkan antar sesama jenisnya tentang ancaman bahaya, maupun peringatan bagi serangga dan burung pemangsa agar tidak memakannya.	1	cahaya kunangkunang berperan pula sebagai tanda peringatan untuk memperingatkan antar sesama jenisnya tentang ancaman bahaya maupun peringatan bagi serangga dan burung pemangsa agar tidak memakannya
2	para peneliti juga terilhami menggunakan teknologi pencahayaan kunang-kunang untuk berbagai hal, termasuk untuk memantau baik tidaknya teknik pengobatan baru bekerja.	2	para peneliti juga terilhami menggunakan teknologi pencahayaan kunangkunang untuk berbagai hal termasuk untuk memantau baik tidaknya teknik pengobatan baru bekerja
3	sejak thomas edison hingga kini, tak ada teknologi lampu listrik yang menyamai cahaya kunang-kunang yang dingin.	3	sejak thomas edison hingga kini tak ada teknologi lampu listrik yang menyamai cahaya kunangkunang yang dingin

Pada kata yang mengandung strip (kata berulang), seperti “kunang-kunang”, tanda strip dihilangkan dan kata digabung, sehingga menjadi “kunangkunang”.

4. Tokenisasi Kata

Selanjutnya lakukan proses tokenisasi kata. Kalimat yang telah dihilangkan tanda bacanya dipisah-pisah per kata menjadi deretan kata tunggal. Berikut adalah hasilnya.

Tabel 3.12 Tokenisasi Kata

Token		
Kalimat 1	**Kalimat 2**	**Kalimat 3**
_	_	_
cahaya	para	sejak
kunangkunang	peneliti	thomas
berperan	juga	edison
pula	terilhami	hingga
sebagai	menggunakan	kini
tanda	teknologi	tak
peringatan	pencahayaan	ada
untuk	kunangkunang	teknologi
memperingatkan	untuk	lampu
antar	berbagai	listrik
sesama	hal	yang
jenisnya	termasuk	menyamai
tentang	untuk	cahaya
ancaman	memantau	kunangkunang
bahaya	baik	yang
maupun	tidaknya	dingin
peringatan	teknik	_
bagi	pengobatan	
serangga	baru	
dan	bekerja	
burung	_	
pemangsa		
agar		
tidak		
memakannya		
_		

5. Pembuatan Korpus N-Gram

Setelah itu lakukan pembuatan korpus n-gram. Dalam penelitian ini korpus n-gram yang digunakan adalah korpus bigram dan trigram yang dibuat secara terpisah. Korpus bigram dibuat dengan menggabungkan token kata dengan token kata di depannya. Lakukan proses untuk tiap-tiap kata dalam kalimat. Hentikan proses jika di depan token sudah tidak ada token lagi (kata ada di akhir

kalimat). Karena itu, jika jumlah token dalam kalimat adalah n, maka jumlah bigram yang terbentuk adalah n-1. Berikut adalah proses pembuatan bigram.

Tabel 3.13 Pembuatan Bigram

Kalimat	Kata	Bigram
1	_	_ cahaya
	cahaya	cahaya kunangkunang
	kunangkunang	kunangkunang berperan
	berperan	berperan pula
	pula	pula sebagai
	sebagai	sebagai tanda
	tanda	tanda peringatan
	peringatan	peringatan untuk
	untuk	untuk memperingatkan
	memperingatkan	memperingatkan antar
	antar	antar sesama
	sesama	sesama jenisnya
	jenisnya	jenisnya tentang
	tentang	tentang ancaman
	ancaman	ancaman bahaya
	bahaya	bahaya maupun
	maupun	maupun peringatan
	peringatan	peringatan bagi
	bagi	bagi serangga
	serangga	serangga dan
	dan	dan burung
	burung	burung pemangsa
	pemangsa	pemangsa agar
	agar	agar tidak
	tidak	tidak memakannya
	memakannya	memakannya _
	_	
2	_	_ para
	para	para peneliti
	peneliti	peneliti juga
	juga	juga terilhami
	terilhami	terilhami menggunakan
	menggunakan	menggunakan teknologi

Kalimat	Kata	Bigram
2	teknologi	teknologi pencahayaan
	pencahayaan	pencahayaan kunangkunang
	kunangkunang	kunangkunang untuk
	untuk	untuk berbagai
	berbagai	berbagai hal
	hal	hal termasuk
	termasuk	termasuk untuk
	untuk	untuk memantau
	memantau	memantau baik
	baik	baik tidaknya
	tidaknya	tidaknya teknik
	teknik	teknik pengobatan
	pengobatan	pengobatan baru
	baru	baru bekerja
	bekerja	bekerja _
	_	
3	_	_ sejak
	sejak	sejak thomas
	thomas	thomas edison
	edison	edison hingga
	hingga	hingga kini
	kini	kini tak
	tak	tak ada
	ada	ada teknologi
	teknologi	teknologi lampu
	lampu	lampu listrik
	listrik	listrik yang
	yang	yang menyamai
	menyamai	menyamai cahaya
	cahaya	cahaya kunangkunang
	kunangkunang	kunangkunang yang
	yang	yang dingin
	dingin	dingin _
	_	

Sementara itu, proses pembuatan korpus trigram hampir sama dengan pembuatan korpus bigram. Korpus trigram dibuat dengan menggabungkan token kata dengan token kata di depannya. Lakukan proses untuk tiap-tiap kata dalam kalimat. Hentikan proses jika token kata ada di urutan kedua dari terakhir kalimat. Karena itu, jika jumlah token dalam kalimat adalah n, maka jumlah trigram yang terbentuk adalah n-2. Berikut adalah proses pembuatan trigram.

Tabel 3.14 Pembuatan Trigram

Kalimat	Kata	Trigram
1	_	_ cahaya kunangkunang
	cahaya	cahaya kunangkunang berperan
	kunangkunang	kunangkunang berperan pula
	berperan	berperan pula sebagai
	pula	pula sebagai tanda
	sebagai	sebagai tanda peringatan
	tanda	tanda peringatan untuk
	peringatan	peringatan untuk memperingatkan
	untuk	untuk memperingatkan antar
	memperingatkan	memperingatkan antar sesama
	antar	antar sesama jenisnya
	sesama	sesama jenisnya tentang
	jenisnya	jenisnya tentang ancaman
	tentang	tentang ancaman bahaya
	ancaman	ancaman bahaya maupun
	bahaya	bahaya maupun peringatan
	maupun	maupun peringatan bagi
	peringatan	peringatan bagi serangga
	bagi	bagi serangga dan
	serangga	serangga dan burung
	dan	dan burung pemangsa
	burung	burung pemangsa agar
	pemangsa	pemangsa agar tidak
	agar	agar tidak memakannya
	tidak	tidak memakannya _
	memakannya	
	_	

Kalimat	Kata	Trigram
2	_	_ para peneliti
	para	para peneliti juga
	peneliti	peneliti juga terilhami
	juga	juga terilhami menggunakan
	terilhami	terilhami menggunakan teknologi
	menggunakan	menggunakan teknologi pencahayaan
	teknologi	teknologi pencahayaan kunangkunang
	pencahayaan	pencahayaan kunangkunang untuk
	kunangkunang	kunangkunang untuk berbagai
	untuk	untuk berbagai hal
	berbagai	berbagai hal termasuk
	hal	hal termasuk untuk
	termasuk	termasuk untuk memantau
	untuk	untuk memantau baik
	memantau	memantau baik tidaknya
	baik	baik tidaknya teknik
	tidaknya	tidaknya teknik pengobatan
	teknik	teknik pengobatan baru
	pengobatan	pengobatan baru bekerja
	baru	baru bekerja _
	bekerja	
	_	
3	_	_ sejak thomas
	sejak	sejak thomas edison
	thomas	thomas edison hingga
	edison	edison hingga kini
	hingga	hingga kini tak
	kini	kini tak ada
	tak	tak ada teknologi
	ada	ada teknologi lampu
	teknologi	teknologi lampu listrik
	lampu	lampu listrik yang
	listrik	listrik yang menyamai
	yang	yang menyamai cahaya
	menyamai	menyamai cahaya kunangkunang

Kalimat	Kata	Trigram
	cahaya	cahaya kunangkunang yang
	kunangkunang	kunangkunang yang dingin
	yang	yang dingin _
	dingin	
	_	

6. Penghitungan Korpus N-Gram

Setelah itu hitung jumlah kemunculan masing-masing n-gram dalam teks. Contohnya adalah sebagai berikut. Misalkan kita akan menghitung jumlah kemunculan bigram "cahaya kunang-kunang".

Tabel 3.15 Contoh Penghitungan N-Gram

cahaya kunang-kunang berperan pula sebagai tanda peringatan, untuk memperingatkan antar sesama jenisnya tentang ancaman bahaya, maupun peringatan bagi serangga dan burung pemangsa agar tidak memakannya. para peneliti juga terilhami menggunakan teknologi pencahayaan kunang-kunang untuk berbagai hal, termasuk untuk memantau baik tidaknya teknik pengobatan baru bekerja. sejak thomas edison hingga kini, tak ada teknologi lampu listrik yang menyamai **cahaya kunang-kunang** yang dingin.

Dalam teks, ditemukan bahwa bigram "cahaya kunang-kunang" terdapat sebanyak 2 buah. Lakukan untuk semua bigram dan trigram sisanya. Tabel 3.15 dan 3.16 berikut adalah hasil penghitungan bigram dan trigram dari teks tersebut.

Tabel 3.16 Jumlah Kemunculan Tiap Bigram

No	Bigram	Jumlah
1	_ cahaya	1
2	cahaya kunangkunang	2
3	kunangkunang berperan	1
4	berperan pula	1
5	pula sebagai	1
6	sebagai tanda	1
7	tanda peringatan	1
8	peringatan untuk	1

No	Bigram	Jumlah
9	untuk memperingatkan	1
10	memperingatkan antar	1
11	antar sesama	1
12	sesama jenisnya	1
13	jenisnya tentang	1
14	tentang ancaman	1
15	ancaman bahaya	1
16	bahaya maupun	1
17	maupun peringatan	1
18	peringatan bagi	1
19	bagi serangga	1
20	serangga dan	1
21	dan burung	1
22	burung pemangsa	1
23	pemangsa agar	1
24	agar tidak	1
25	tidak memakannya	1
26	memakannya _	1
27	_ para	1
28	para peneliti	1
29	peneliti juga	1
30	juga terilhami	1
31	terilhami menggunakan	1
32	menggunakan teknologi	1
33	teknologi pencahayaan	1
34	pencahayaan kunangkunang	1
35	kunangkunang untuk	1
36	untuk berbagai	1
37	berbagai hal	1
38	hal termasuk	1
39	termasuk untuk	1
40	untuk memantau	1
41	memantau baik	1
42	baik tidaknya	1
43	tidaknya teknik	1

No	Bigram	Jumlah
44	teknik pengobatan	1
45	pengobatan baru	1
46	baru bekerja	1
47	bekerja _	1
48	_ sejak	1
49	sejak thomas	1
50	thomas edison	1
51	edison hingga	1
52	hingga kini	1
53	kini tak	1
54	tak ada	1
55	ada teknologi	1
56	teknologi lampu	1
57	lampu listrik	1
58	listrik yang	1
59	yang menyamai	1
60	menyamai cahaya	1
61	kunangkunang yang	1
62	yang dingin	1
63	dingin _	1

Tabel 3.17 Jumlah Kemunculan Tiap Trigram

No	Trigram	Jumlah
1	_ cahaya kunangkunang	1
2	cahaya kunangkunang berperan	1
3	kunangkunang berperan pula	1
4	berperan pula sebagai	1
5	pula sebagai tanda	1
6	sebagai tanda peringatan	1
7	tanda peringatan untuk	1
8	peringatan untuk memperingatkan	1
9	untuk memperingatkan antar	1
10	memperingatkan antar sesama	1
11	antar sesama jenisnya	1
12	sesama jenisnya tentang	1

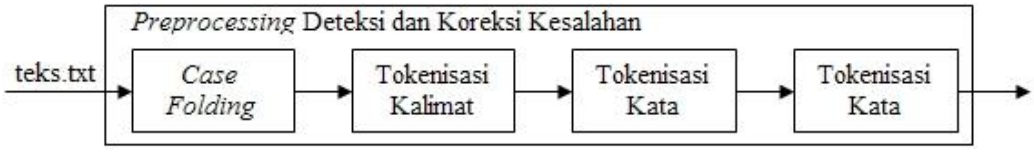
No	Trigram	Jumlah
13	jenisnya tentang ancaman	1
14	tentang ancaman bahaya	1
15	ancaman bahaya maupun	1
16	bahaya maupun peringatan	1
17	maupun peringatan bagi	1
18	peringatan bagi serangga	1
19	bagi serangga dan	1
20	serangga dan burung	1
21	dan burung pemangsa	1
22	burung pemangsa agar	1
23	pemangsa agar tidak	1
24	agar tidak memakannya	1
25	tidak memakannya _	1
26	_ para peneliti	1
27	para peneliti juga	1
28	peneliti juga terilhami	1
29	juga terilhami menggunakan	1
30	terilhami menggunakan teknologi	1
31	menggunakan teknologi pencahayaan	1
32	teknologi pencahayaan kunang-kunang	1
33	pencahayaan kunang-kunang untuk	1
34	kunang-kunang untuk berbagai	1
35	untuk berbagai hal	1
36	berbagai hal termasuk	1
37	hal termasuk untuk	1
38	termasuk untuk memantau	1
39	untuk memantau baik	1
40	memantau baik tidaknya	1
41	baik tidaknya teknik	1
42	tidaknya teknik pengobatan	1
43	teknik pengobatan baru	1
44	pengobatan baru bekerja	1
45	baru bekerja _	1
46	_ sejak thomas	1
47	sejak thomas edison	1

No	Trigram	Jumlah
48	thomas edison hingga	1
49	edison hingga kini	1
50	hingga kini tak	1
51	kini tak ada	1
52	tak ada teknologi	1
53	ada teknologi lampu	1
54	teknologi lampu listrik	1
55	lampu listrik yang	1
56	listrik yang menyamai	1
57	yang menyamai cahaya	1
58	menyamai cahaya kunang-kunang	1
59	cahaya kunang-kunang yang	1
60	kunang-kunang yang dingin	1
61	yang dingin _	1

Hasil penghitungan ini kemudian akan ditambahkan ke dalam korpus bigram dan trigram. Lakukan untuk teks-teks lain yang akan dijadikan korpus n-gram.

3.2.2.2 *Preprocessing* Deteksi dan Koreksi Kesalahan

Preprocessing ini dilakukan untuk menyiapkan data uji agar dapat dilakukan proses deteksi dan koreksi kesalahan ejaan *real-word* pada data uji tersebut. Tahapan *preprocessing* deteksi dan koreksi kesalahan serupa dengan tiga tahapan awal *preprocessing* pembuatan korpus n-gram, yaitu *case folding*, tokenisasi kalimat, dan tokenisasi kata.



Gambar 3.3 Diagram *Preprocessing* Deteksi dan Koreksi Kesalahan

Berikut adalah contoh potongan teks data uji yang akan di-*preprocessing* yang diambil dari Tabel 3.7.

"Cahaya kunang-kunang berperan sebagai tandu peringatan."

Berikut adalah tahapan-tahapan *preprocessing* data uji di atas sehingga data uji siap untuk dideteksi dan dikoreksi.

1. *Case Folding*

Pertama lakukan *case folding* dengan mengubah semua huruf besar menjadi huruf kecil. Hasilnya adalah sebagai berikut. Huruf yang diubah ditandai dengan ditebalkan.

Tabel 3.18 *Case Folding* Data Uji

Sebelum	Sesudah
Cahaya kunang-kunang berperan sebagai tandu peringatan.	**c**ahaya kunang-kunang berperan sebagai tandu peringatan.

2. Tokenisasi Kalimat

Selanjutnya lakukan tokenisasi kalimat dengan memisahkan tiap-tiap kalimat. Pada kasus ini dalam data uji hanya terdapat satu token kalimat.

Tabel 3.19 Tokenisasi Kalimat Data Uji

Sebelum	Sesudah	
cahaya kunang-kunang berperan sebagai tandu peringatan.	1	cahaya kunang-kunang berperan sebagai tandu peringatan.

3. *Filtering* Tanda Baca

Setelah itu dilakukan penghilangan tanda baca. Di sini tanda baca adalah karakter selain karakter huruf, angka atau spasi. Berikut adalah hasilnya.

Tabel 3.20 *Filtering* Tanda Baca Data Uji

Sebelum	Sesudah	
cahaya kunang-kunang berperan sebagai tandu peringatan.	1	cahaya kunangkunang berperan sebagai tandu peringatan

4. Tokenisasi Kata

Selanjutnya lakukan proses tokenisasi kata pada data uji. Berikut adalah hasilnya.

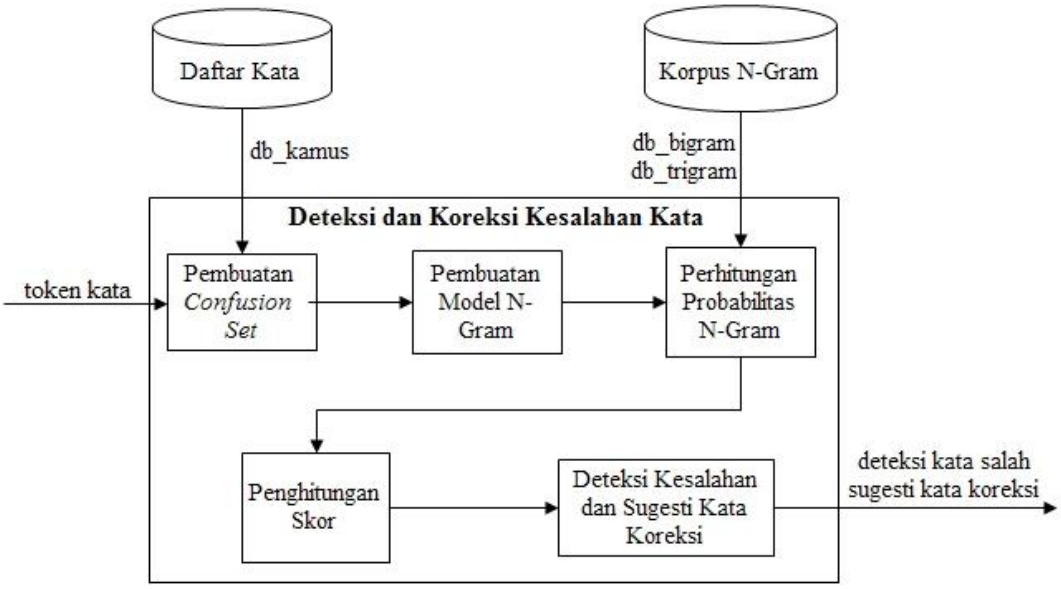
Tabel 3.21 Tokenisasi Kata Data Uji

Token	
_	sebagai
cahaya	tandu
kunangkunang	peringatan
berperan	_

Setelah *preprocessing*, proses deteksi dan koreksi kesalahan ejaan *real-word* pada data uji siap dilakukan.

3.2.2.3 Deteksi dan Koreksi Kesalahan Ejaan *Real-Word*

Proses deteksi dan koreksi kesalahan ejaan *real-word* berawal dengan pembuatan *confusion set* untuk setiap kata kandidat. Setelah itu dilakukan penghitungan peringkat setiap elemen dalam *confusion set* tersebut. Berdasarkan hasil peringkat, kesalahan dideteksi dan diberikan sugesti kata koreksi. Ini dilakukan secara kata per kata.



Gambar 3.4 Diagram Proses Deteksi dan Koreksi Kesalahan Kata

Dalam analisis ini kita akan memakai data uji yang telah di-*preprocessing* pada subbab 3.2.2.2. Agar memudahkan analisis, beri notasi W^i untuk tiap-tiap token sebagai berikut.

Tabel 3.22 Daftar Token Kata Data Uji

W^i	Token
W^0	_
W^1	cahaya
W^2	kunangkunang
W^3	berperan
W^4	sebagai
W^5	tandu
W^6	peringatan
W^7	_

Dari token-token kata di atas, akan dicari kata-kata manakah yang merupakan kesalahan *real-word*. Jika kesalahan terdeteksi, akan diberi sugesti kata yang benar untuk mengoreksi kata yang salah tersebut.

1. Pembuatan Confusion Set

Pertama-tama bangkitkan *confusion set* untuk setiap token kata, yaitu himpunan kata-kata yang merupakan variasi ejaan dari token kata tersebut. Ini didapat dengan menghitung *Levenshtein distance* dari setiap token dalam daftar kata. *Levenshtein distance* adalah jumlah minimum operasi edit yang diperlukan untuk mengubah suatu kata menjadi kata yang lain. Operasi edit merupakan penambahan (*insertion*), penghapusan (*deletion*), atau penggantian (*substitution*) satu karakter dalam suatu kata. Berikut adalah langkah-langkah algoritma *Levenshtein distance* [25].

Tabel 3.23 Algoritma *Levenshtein Distance*

Langkah	Penjelasan
1	Inisialisasi string pertama menjadi *source* (s) dengan panjang m karakter. Inisialisasi string kedua menjadi *target* (t) dengan panjang n karakter. Buat matriks yang mengandung m+1 baris dan n+1 kolom. Matriks dimulai dari baris dan kolom ke-0.
2	Isi baris ke-0 dengan 0..m. Isi kolom ke-0 dengan 0..n.
3	Periksa setiap karakter s (i dari 1 ke n).
4	Periksa setiap karakter t (j dari 1 ke m).
5	Jika s[i] = t[j], *cost* = 0. Jika s[i] ≠ t[j], *cost* = 1.
6	Isi sel matriks d[i,j] dengan nilai minimum dari sel di atasnya + 1, sel di kirinya + 1, dan sel di kiri atasnya + *cost*, dengan rumus: $d[i,j] = min \begin{cases} d[i-1,j]+1 \\ d[i,j-1]+1 \\ d[i-1,j-1]+cost \end{cases}$
7	Iterasi langkah 3, 4, 5, 6. Nilai *Levenshtein distance* ditemukan pada sel terakhir d[n,m].

Berikut adalah contoh penghitungan *Levenshtein distance* untuk token kata “cahaya”. Kata “cahaya” akan dibandingkan dengan kata yang terdapat pada daftar kata kamus (Tabel 3.2). Sebagai contoh, berikut adalah perbandingan kata “cahaya” dengan kata “bahaya” yang diambil dari daftar kata kamus.

Tabel 3.24 Perbandingan Kata "cahaya" Dengan "bahaya"

Daftar Token
cahaya
kunangkunang
berperan
sebagai
tandu
peringatan

Kamus
...
bahasa
bahaya
bahkan
bahtera
...

Inisialisasi s = cahaya sepanjang 6 karakter dan t = bahaya sepanjang 6 karakter. Buat matriks d dengan baris dari 0..6 dan kolom dari 0..6 sebagai berikut

Tabel 3.25 Matriks Awal *Levenshtein Distance d*

		c	a	h	a	y	a
	d[0,0]	d[0,1]	d[0,2]	d[0,3]	d[0,4]	d[0,5]	d[0,6]
b	d[1,0]	d[1,1]	d[1,2]	d[1,3]	d[1,4]	d[1,5]	d[1,6]
a	d[2,0]	d[2,1]	d[2,2]	d[2,3]	d[2,4]	d[2,5]	d[2,6]
h	d[3,0]	d[3,1]	d[3,2]	d[3,3]	d[3,4]	d[3,5]	d[3,6]
a	d[4,0]	d[4,1]	d[4,2]	d[4,3]	d[4,4]	d[4,5]	d[4,6]
y	d[5,0]	d[5,1]	d[5,2]	d[5,3]	d[5,4]	d[5,5]	d[5,6]
a	d[6,0]	d[6,1]	d[6,2]	d[6,3]	d[6,4]	d[6,5]	d[6,6]

Selanjutnya isi baris ke-0 matriks sebagai dengan 0..6 dan kolom ke-0 matriks dengan 0..6.

Tabel 3.26 Pengisian Kolom dan Baris ke-0 Matriks *d*

		c	a	h	a	y	a
	0	1	2	3	4	5	6
b	1						
a	2						
h	3						
a	4						
y	5						
a	6						

Selanjutnya bandingkan setiap karakter pada matriks. Dimulai dari sel d[1,1], bandingkan karakter "c" dengan karakter "b". Pertama tentukan nilai *cost*. Karena "c" ≠ "b", maka *cost* = 1. Lalu hitung nilai minimumnya. Maka didapat

$$d[i,j] = min\begin{cases} d[i-1,j]+1 \\ d[i,j-1]+1 \\ d[i-1,j-1]+cost \end{cases}$$

$$d[1,1] = min\begin{cases} d[1-1,1]+1 = d[0,1]+1 = 1+1 = 2 \\ d[1,1-1]+1 = d[1,0]+1 = 1+1 = 2 \\ d[1-1,1-1]+cost = d[0,0]+1 = 0+1 = \mathbf{1} \end{cases}$$

Tabel 3.27 Contoh 1 Pengisian Matriks d

		c	a	h	a	y	a
	0	1	2	3	4	5	6
b	1	1					
a	2						
h	3						
a	4						
y	5						
A	6						

Selanjutnya untuk sel d[1,2], bandingkan karakter "a" dengan "b". karena "a" ≠ "b", maka *cost* = 1. Lalu hitung nilai minimumnya, didapat

$$d[i,j] = min\begin{cases} d[i-1,j]+1 \\ d[i,j-1]+1 \\ d[i-1,j-1]+cost \end{cases}$$

$$d[1,2] = min\begin{cases} d[1-1,2]+1 = d[0,2]+1 = 2+1 = 3 \\ d[1,2-1]+1 = d[1,1]+1 = 1+1 = 2 \\ d[1-1,2-1]+cost = d[0,1]+1 = 1+1 = 2 \end{cases}$$

$$d[1,1] = 2$$

Tabel 3.28 Contoh 2 Pengisian Matriks d

		c	a	h	a	y	a
	0	1	2	3	4	5	6
b	1	1	2				
a	2						
h	3						
a	4						
y	5						
a	6						

Lakukan untuk setiap sel matriks, hingga didapat hasil penghitungan sebagai berikut.

Tabel 3.29 Hasil Penghitungan Algoritma *Levenshtein Distance*

		c	a	h	a	y	a
	0	1	2	3	4	5	6
b	1	1	2	3	4	5	6
a	2	2	1	2	3	4	5
h	3	3	2	1	2	3	4
a	4	4	3	2	1	2	3
y	5	5	4	3	2	1	2
a	6	6	5	4	3	2	**1**

Hasil penghitungan dapat dilihat pada kolom terakhir baris terakhir matriks. Dari hasil penghitungan, didapat bahwa nilai *Levenshtein distance* dari perbandingan kata “cahaya” dan “bahaya” adalah 1.

Penghitungan *Levenshtein distance* dilakukan untuk semua kata dalam daftar kata kamus. Berikut adalah hasil penghitungan *Levenshtein distance* untuk kata “cahaya”.

Tabel 3.30 Hasil Penghitungan *Levenshtein distance* untuk Kata “cahaya”

Token	Kata	*Levenshtein distance*
cahaya	a	5
cahaya	aba	4
cahaya	abad	4
...	...	...
cahaya	bahaya	1
cahaya	bahkan	4
...	...	...
cahaya	cahar	2
cahaya	cahaya	0
...	...	...
cahaya	sahaya	1
...	...	...

Kata-kata yang dijadikan anggota *confusion set* adalah kata-kata dalam daftar kata yang memiliki nilai *Levenshtein distance* maksimal 1. *Confusion set* dapat direpresentasikan sebagai

$$C(W^i) = \{c_1^i, c_2^i, \dots, c_j^i, \dots, c_{k_i}^i\}$$

dimana W^i adalah kata ke-i di dalam kalimat dan k_i adalah jumlah elemen di dalam $C(W^i)$. Berikut adalah *confusion set* untuk setiap kata dalam data uji.

Tabel 3.31 Hasil Pembuatan Anggota *Confusion Set* untuk Data Uji

W^i	Kata	c_j^i	Anggota	Nilai *Levenshtein*
W^1	cahaya	c_1^1	bahaya	1
		c_2^1	cahaya	0
		c_3^1	sahaya	1
W^2	kunangkunang	c_1^2	kunangkunang	0
W^3	berperan	c_1^3	berperam	1
		c_2^3	berperan	0
		c_3^3	berseran	1
W^4	sebagai	c_1^4	sebagai	0
W^5	tandu	c_1^5	candu	1
		c_2^5	tanda	1
		c_3^5	tandu	0
		c_4^5	tanduk	1
W^6	peringatan	c_1^6	keringatan	1
		c_2^6	peringanan	1
		c_3^6	peringatan	0

2. Pembuatan Model N-Gram

Setelah itu dilakukan pembuatan model n-gram untuk setiap anggota *confusion set.* Untuk setiap anggota $C(W^i)$, bangkitkan bigram kiri, bigram kanan dan trigram dengan cara mengambil kata di kiri, kanan, dan keduanya (untuk trigram) dari W^i (diambil dari tabel 3.20). Maka bigram dan trigram yang terbentuk adalah:

Bigram kiri: $W^{i-1}c_j^i$

Bigram kanan: $c_j^i W^{i+1}$

Trigram: $W^{i-1}c_j^i W^{i+1}$

dimana $1 \leq j \leq k_i$.

Untuk kata di awal dan di akhir kalimat, kata yang kosong ditandai dengan garis bawah (_). Pembuatan model n-gram dilakukan untuk semua anggota *confusion set* untuk setiap kata. Hasil pembuatan model n-gram untuk data uji ditampilkan sebagai berikut.

Tabel 3.32 Hasil Pembuatan Model N-Gram Untuk Data Uji

c_j^i		W^{i-1}	W^{i+1}	**Bigram Kiri** ($W^{i-1}c_j^i$)	**Bigram Kanan** ($c_j^i W^{i+1}$)	**Trigram** ($W^{i-1}c_j^i W^{i+1}$)
c_1^1	bahaya	W^0	W^2	_ bahaya	bahaya kunangkunang	_ bahaya kunangkunang
c_2^1	cahaya	_	kunangkunang	_ cahaya	cahaya kunangkunang	_ cahaya kunangkunang
c_3^1	sahaya			_ sahaya	sahaya kunangkunang	_ sahaya kunangkunang
c_1^2	kunangkunang	W^1	W^3	cahaya kunangkunang	kunangkunang berperan	cahaya kunangkunang berperan
c_1^3	berperam	cahaya	berperan	kunangkunang berperam	berperam sebagai	kunangkunang berperam sebagai
c_2^3	berperan	W^2	W^4	kunangkunang berperan	berperan sebagai	kunangkunang berperan sebagai
c_3^3	berseran	kunangkunang	sebagai	kunangkunang berseran	berseran sebagai	kunangkunang berseran sebagai
c_1^4	sebagai	W^3	W^5	berperan sebagai	sebagai tandu	berperan sebagai tandu
c_1^5	candu	berperan	tandu	sebagai candu	candu peringatan	sebagai candu peringatan
c_2^5	tanda	W^4	W^6	sebagai tanda	tanda peringatan	sebagai tanda peringatan
c_3^5	tandu	sebagai	peringatan	sebagai tandu	tandu peringatan	sebagai tandu peringatan
c_4^5	tanduk			sebagai tanduk	tanduk peringatan	sebagai tanduk peringatan
c_1^6	keringatan	W^5	W^7	tandu keringatan	keringatan_	tandu keringatan_
c_2^6	peringanan	tandu	_	tandu peringanan	peringanan_	tandu peringanan_
c_3^6	peringatan			tandu peringatan	peringatan_	tandu peringatan_

Setelah itu lihat jumlah kemunculan tiap bigram dan trigram dalam korpus n-gram. Korpus bigram dan trigram yang digunakan untuk mencari jumlah kemunculan adalah berdasarkan dari Tabel 3.15 dan Tabel 3.16. Jumlah kemunculan n-gram dalam korpus dilambangkan dengan

$count(W^{i-1}c_j^i)$ untuk jumlah kemunculan bigram kiri,

$count(c_j^i W^{i+1})$ untuk jumlah kemunculan bigram kanan, dan

$count(W^{i-1}c_j^i W^{i+1})$ untuk jumlah kemunculan trigram.

Tabel-tabel berikut masing-masing menampilkan jumlah kemunculan bigram kiri, bigram kanan dan trigram pada data uji. Jika suatu n-gram tidak ditemukan dalam korpus, maka diberi nilai 0. Berikut adalah daftar kemunculan bigram kiri data uji dalam korpus n-gram. Sebagai contoh, untuk token W^5 = "tandu", untuk anggota *confusion set* c_3^5 = "tanda", jumlah kemunculan bigram kirinya $W^4c_j^5$ = "sebagai tanda" pada korpus n-gram adalah 1.

Tabel 3.33 Daftar Jumlah Kemunculan Bigram Kiri Data Uji

c_j^1		$(W^0c_j^1)$	count$(W^0c_j^1)$
c_1^1	bahaya	_ bahaya	0
c_2^1	cahaya	_ cahaya	1
c_3^1	sahaya	_ sahaya	0
c_j^2		$(W^1c_j^2)$	count$(W^1c_j^2)$
c_1^2	kunangkunang	cahaya kunangkunang	2
c_j^3		$(W^2c_j^3)$	count$(W^2c_j^3)$
c_1^3	berperam	kunangkunang berperam	0
c_2^3	berperan	kunangkunang berperan	1
c_3^3	berseran	kunangkunang berseran	0
c_j^4		$(W^3c_j^4)$	count$(W^3c_j^4)$
c_1^4	sebagai	berperan sebagai	0
c_j^5		$(W^4c_j^5)$	count$(W^4c_j^5)$
c_1^5	candu	sebagai candu	0
c_2^5	tanda	sebagai tanda	1
c_3^5	tandu	sebagai tandu	0
c_4^5	tanduk	sebagai tanduk	0

c_j^6		$(W^5 c_j^6)$	count$(W^5 c_j^6)$
c_1^6	keringatan	tandu keringatan	0
c_2^6	peringanan	tandu peringanan	0
c_3^6	peringatan	tandu peringatan	0

Berikut adalah daftar kemunculan bigram kanan data uji dalam korpus n-gram. Sebagai contoh, untuk token W^5 = "tandu", untuk anggota *confusion set* c_2^5 = "tanda", jumlah kemunculan bigram kanannya $c_2^5 W^6$ = "tanda peringatan" pada korpus n-gram adalah 1.

Tabel 3.34 Daftar Jumlah Kemunculan Bigram Kanan Data Uji

c_j^1		Bigram Kanan $(c_j^1 W^2)$	count$(c_j^1 W^2)$
c_1^1	bahaya	bahaya kunangkunang	0
c_2^1	cahaya	cahaya kunangkunang	2
c_3^1	sahaya	sahaya kunangkunang	0
c_j^2		Bigram Kanan $(c_j^2 W^3)$	count$(c_j^2 W^3)$
c_1^2	kunangkunang	kunangkunang berperan	1
c_j^3		Bigram Kanan $(c_j^3 W^4)$	count$(c_j^3 W^4)$
c_1^3	berperam	berperam sebagai	0
c_2^3	berperan	berperan sebagai	0
c_3^3	berseran	berseran sebagai	0
c_j^4		Bigram Kanan $(c_j^4 W^5)$	count$(c_j^4 W^5)$
c_1^4	sebagai	sebagai tandu	0
c_j^5		Bigram Kanan $(c_j^5 W^6)$	count$(c_j^5 W^6)$
c_1^5	candu	candu peringatan	0
c_2^5	tanda	tanda peringatan	1
c_3^5	tandu	tandu peringatan	0
c_4^5	tanduk	tanduk peringatan	0
c_j^6		Bigram Kanan $(c_j^6 W^7)$	count$(c_j^6 W^7)$
c_1^6	keringatan	keringatan _	0
c_2^6	peringanan	peringanan _	0
c_3^6	peringatan	peringatan _	0

Berikut adalah daftar kemunculan trigram data uji dalam korpus n-gram. Sebagai contoh, untuk token W^5 = "tandu", untuk anggota *confusion set* c_2^5 = "tanda", jumlah kemunculan trigramnya $W^4 c_2^5 W^6$ = "sebagai tanda peringatan" pada korpus n-gram adalah 1.

Tabel 3.35 Daftar Jumlah Kemunculan Trigram Data Uji

c_j^1		**Trigram ($W^0 c_j^1 W^2$)**	**count($W^0 c_j^1 W^2$)**
c_1^1	bahaya	_ bahaya kunangkunang	0
c_2^1	cahaya	_ cahaya kunangkunang	1
c_3^1	sahaya	_ sahaya kunangkunang	0
c_j^2		**Trigram ($W^1 c_j^2 W^3$)**	**count($W^1 c_j^2 W^3$)**
c_1^2	kunangkunang	cahaya kunangkunang berperan	0
c_j^3		**Trigram ($W^2 c_j^3 W^4$)**	**count($W^2 c_j^3 W^4$)**
c_1^3	berperam	kunangkunang berperam sebagai	0
c_2^3	berperan	kunangkunang berperan sebagai	0
c_3^3	berseran	kunangkunang berseran sebagai	0
c_j^4		**Trigram ($W^3 c_j^4 W^5$)**	**count($W^3 c_j^4 W^5$)**
c_1^4	sebagai	berperan sebagai tandu	0
c_j^5		**Trigram ($W^4 c_j^5 W^6$)**	**count($W^4 c_j^5 W^6$)**
c_1^5	candu	sebagai candu peringatan	0
c_2^5	tanda	sebagai tanda peringatan	1
c_3^5	tandu	sebagai tandu peringatan	0
c_4^5	tanduk	sebagai tanduk peringatan	0
c_j^6		**Trigram ($W^5 c_j^6 W^7$)**	**count($W^5 c_j^6 W^7$)**
c_1^6	keringatan	tandu keringatan _	0
c_2^6	peringanan	tandu peringanan _	0
c_3^6	peringatan	tandu peringatan _	0

Lalu hitung juga total jumlah kemunculan n-gram untuk semua anggota *confusion set* untuk setiap kata. Total jumlah kemunculan dilambangkan dengan

$\sum_{r=1}^{k_i} count(W^{i-1} c_r^i)$ untuk total jumlah kemunculan bigram kiri,

$\sum_{r=1}^{k_i} count(W^{i+1} c_r^i)$ untuk total jumlah kemuncula bigram kanan, dan

$\sum_{r=1}^{k_i} count(W^{i-1} c_r^i W^{i+1})$ untuk total jumlah kemunculan trigram,

dengan k adalah jumlah anggota dari *confusion set* suatu kata dan r adalah urutan anggota dari *confusion set* tersebut. Berikut adalah hasil penghitungan total jumlah kemunculan bigram dan trigram untuk data uji.

Tabel 3.36 Total Jumlah Kemunculan Bigram dan Trigram Data Uji

W^i	Token	$\sum_{r=1}^{k_i} count(W^{i-1}c_r^i)$	$\sum_{r=1}^{k_i} count(c_r^i W^{i+1})$	$\sum_{r=1}^{k_i} count(W^{i-1}c_r^i W^{i+1})$
W^1	cahaya	$count(W^0c_1^1) + count(W^0c_2^1) + count(W^0c_3^1) = 0+1+0=1$	$count(c_1^1W^2) + count(c_2^1W^2) + count(c_3^1W^2) = 0+2+0=2$	$count(W^0c_1^1W^2) + count(W^0c_2^1W^2) + count(W^0c_3^1W^2) = 0+1+0=1$
W^2	kunangkunang	$count(W^1c_1^2) = 0$	$count(W_1^2c^3) = 2$	$count(W^1c_1^2W^3) = 0$
W^3	berperan	$count(W^2c_1^3) + count(W^2c_2^1) + count(W^2c_3^1) = 0+1+0=1$	$count(c_1^1W^4) + count(c_2^1W^4) + count(c_3^1W^4) = 0+0+0=0$	$count(W^2c_1^1) + count(W^2c_2^1) + count(W^2c_3^1) = 0+0+0=0$
W^4	sebagai	$count(W^3c_1^4) = 0$	$count(c_1^4W^5) = 0$	$count(W^3c_1^4W^5) = 0$
W^5	tandu	$count(W^4c_1^5) + count(W^4c_2^5) + count(W^4c_3^5) + count(W^4c_4^5) = 0+1+0+0=1$	$count(W_1^5c^6) + count(W_2^5c^6) + count(W_3^5c^6) + count(W_4^5c^6) = 0+1+0+0=1$	$count(W^4c_1^5W^6) + count(W^4c_2^5W^6) + count(W^4c_3^5W^6) + count(W^4c_4^5W^6) = 0+1+0+0=1$
W^6	peringatan	$count(W^5c_1^6) + count(W^5c_2^6) + count(W^5c_3^6) = 0+0+0=0$	$count(c_1^6W^7) + count(c_2^6W^7) + count(c_3^6W^7) = 0+0+0=0$	$count(W^5c_1^6W^7) + count(W^5c_2^6W^7) + count(W^5c_3^6W^7) = 0+0+0=0$

3. Penghitungan Probabilitas N-Gram

Salah satu cara menghitung probabilitas n-gram adalah dengan menggunakan aturan *Markov chain*. Menurut asumsi Markov, kemungkinan suatu kejadian masa depan (kata selanjutnya) berdasarkan hanya kepada sejarah terbatas kejadian-kejadian sebelumnya (kata-kata sebelumnya). Contohnya dalam model bigram untuk kalimat sepanjang m kata $W^1, W^2, \dots, W^m$, dapat dihitung sebagai

$$P(W^1, W^2, \dots, W^m) =$$

$$P(W^1|\text{Ь})P(W^2|W^1)P(W^3|W^2) \dots P(W^m|W^{m-1})P(\text{Ь}|W^m)$$

dimana Ь menandakan kosong (*blank*) [2].

Dalam model ini diambil asumsi bahwa kemunculan kejadian (kata) bergantung pada kejadian (kata) sebelum dan sesudahnya terlepas dari kejadian-kejadian (kata-kata) lain dalam kalimat. Penghitungan probabilitas n-gram dilakukan menggunakan *Maximum Likelihood Estimation* (MLE) [26], yaitu

dengan menghitung jumlah kemunculan n-gram dari korpus lalu menormalisasikannya, dengan cara membaginya dengan suatu nilai total agar nilainya berada di antara 0 dan 1.

Sebagai contoh, untuk menghitung kemungkinan bigram dari suatu kata W^i jika diberikan kata sebelumnya W^{i-1}, maka dapat dihitung jumlah kemunculan bigram *count*($W^{i-1}W^i$) dan menormalisasikannya dengan jumlah semua bigram yang juga memiliki kata pertama W^{i-1}:

$$P(W^i \mid W^{i-1}) = \frac{count(W^{i-1}W^i)}{\sum_m count(W^{i-1}W)}$$

Dengan menggunakan MLE, dihasilkan rumus-rumus [2]:

$$P_1(c_j^i \mid W^{i-1}) = \frac{count(W^{i-1}c_j^i)}{\sum_{r=1}^{k_i} count(W^{i-1}c_r^i)}$$

$$P_2(c_j^i \mid W^{i+1}) = \frac{count(W^{i+1}c_j^i)}{\sum_{r=1}^{k_i} count(W^{i+1}c_r^i)}$$

$$P_3(c_j^i \mid W^{i-1}, W^{i+1}) = \frac{count(W^{i-1}c_j^iW^{i+1})}{\sum_{r=1}^{k_i} count(W^{i-1}c_r^iW^{i+1})}$$

dimana P_1 adalah probabilitas bigram kiri, P_2 probabilitas bigram kanan, dan P_3 probabilitas trigram.

Akan tetapi dengan rumus di atas akan didapat banyak probabilitas yang bernilai 0. Untuk mengatasinya digunakan suatu metode *smoothing* yang menambahkan *pseudocount* pada probabilitas yang bernilai 0. Metode *smoothing* yang digunakan adalah *additive smoothing* [27], yang menambahkan suatu konstanta δ pada *count*. Dalam kasus ini δ yang digunakan adalah 0.01, dengan pertimbangan agar nilai yang dihasilkan tidak terlalu mempengaruhi penghitungan skor pada tahap selanjutnya. Maka rumus-rumus probabilitas n-gram menjadi:

$$P_1'(c_j^i \mid W^{i-1}) = \frac{count(W^{i-1}W_j^i)+\delta}{\sum_{r=1}^{k_i} count(W^{i-1}W_r^i)+\delta*V_j} = \frac{count(W^{i-1}W_j^i)+0.01}{\sum_{r=1}^{k_i} count(W^{i-1}W_r^i)+0.01*V_j}$$

$$P_2'(c_j^i \mid W^{i+1}) = \frac{count(W^{i+1}W_j^i)+\delta}{\sum_{r=1}^{k_i} count(W^{i+1}W_r^i)+\delta*V_j} = \frac{count(W^{i+1}W_j^i)+0.01}{\sum_{r=1}^{k_i} count(W^{i+1}W_r^i)+0.01*V_j}$$

$$P_3{}'(c_j^i|W^{i-1},W^{i+1}) = \frac{count(W^{i-1}W_j^iW^{i+1})+\delta}{\sum_{r=1}^{k_i} count(W^{i-1}W_r^iW^{i+1})+\delta*V_j} = \frac{count(W^{i-1}W_j^iW^{i+1})+0.01}{\sum_{r=1}^{k_i} count(W^{i-1}W_r^iW^{i+1})+0.01*V_j}$$

dimana V_j = jumlah anggota *confusion set*.

Lakukan perhitungan untuk semua anggota di setiap *confusion set* untuk setiap kata. Berikut adalah hasil perhitungan probabilitas bigram dan trigram untuk setiap anggota *confusion set* pada tiap kata.

Tabel 3.37 Hasil Perhitungan Probabilitas Bigram dan Trigram Data Uji

Anggota (c_j^i)		**Bigram dan Trigram**	**Probabilitas** $P_1{}'(c_j^i\|W^{i-1})$ $P_2{}'(c_j^i\|W^{i+1})$ $P_3{}'(c_j^i\|W^{i-1},W^{i+1})$
c_1^1	bahaya	_ bahaya	$P_1(c_1^1\|W^0) = \frac{count(W^0c_1^1)+0.01}{\sum_{r=1}^{k_1} count(W^0c_r^1)+0.01*3}$ = 0.01/1.03 = 0.0097
		bahaya kunangkunang	$P_2(c_1^1\|W^2) = \frac{count(c_1^1W^2)+0.01}{\sum_{r=1}^{k_1} count(c_r^1W^2)+0.01*3}$ = 0.01/2.03 = 0.0049
		_ bahaya kunangkunang	$P_3(c_1^1\|W^0,W^2) = \frac{count(W^0c_1^1W^2)+0.01}{\sum_{r=1}^{k_1} count(W^0c_r^1W^2)+0.01*3}$ = 0.01/1.03 = 0.0097
c_2^1	cahaya	_ cahaya	$P_1(c_2^1\|W^0) = \frac{count(W^0c_2^1)+0.01}{\sum_{r=1}^{k_1} count(W^0c_r^1)+0.01*3}$= 0.01/1.03 = 0.9806
		cahaya kunangkunang	$P_2(c_2^1\|W^2) = \frac{count(c_2^1W^2)+0.01}{\sum_{r=1}^{k_1} count(c_r^1W^2)+0.01*3}$= 2.01 / 2.03 = 0.9901
		_ cahaya kunangkunang	$P_3(c_2^1\|W^0,W^2) = \frac{count(W^0c_2^1W^2)+0.01}{\sum_{r=1}^{k_1} count(W^0c_r^1W^2)+0.01*3}$ = 1.01 / 1.03 = 0.9806
c_3^1	sahaya	_ sahaya	$P_1(c_3^1\|W^0) = \frac{count(W^0c_3^1)+0.01}{\sum_{r=1}^{k_1} count(W^0c_r^1)+0.01*3}$ = 0.01/1.03 = 0.0097
		sahaya kunangkunang	$P_2(c_3^1\|W^2) = \frac{count(c_3^1W^2)+0.01}{\sum_{r=1}^{k_1} count(c_r^1W^2)+0.01*3}$ = 0.01 / 2.03 = 0.0049
		_ sahaya kunangkunang	$P_3(c_3^1\|W^0,W^2) = \frac{count(W^0c_3^1W^2)+0.01}{\sum_{r=1}^{k_1} count(W^0c_r^1W^2)+0.01*3}$ = 0.01/1.03 = 0.0097
c_1^2	kunangkunang	cahaya kunangkunang	$P_1(c_1^2\|W^1) = \frac{count(W^1c_1^2)+0.01}{\sum_{r=1}^{k_2} count(W^1c_r^2)+0.01*1}$= 2.01 / 2.01 = 1.0000
		kunangkunang berperan	$P_2(c_1^2\|W^3) = \frac{count(c_1^2W^3)+0.01}{\sum_{r=1}^{k_2} count(c_r^2W^3)+0.01*1}$ = 1.01/1.01 = 1.0000
		cahaya kunangkunang berperan	$P_3(c_1^2\|W^1,W^3) = \frac{count(W^1c_1^2W^3)+0.01}{\sum_{r=1}^{k_2} count(W^1c_r^2W^3)+0.01*1}$ = 1.01 / 1.01 = 1.0000

Anggota (c_j^i)		Bigram dan Trigram	Probabilitas $P_1{}'(c_j^i \mid W^{i-1})$ $P_2{}'(c_j^i \mid W^{i+1})$ $P_3{}'(c_j^i \mid W^{i-1}, W^{i+1})$
c_1^3	berperam	kunangkunang berperam	$P_1(c_1^3 \mid W^2) = \frac{count(W^2 c_1^3)+0.01}{\sum_{r=1}^{k_3} count(W^2 c_r^3)+0.01*3}$ = 0.01 / 1.03 = 0.0097
		berperam sebagai	$P_2(c_1^3 \mid W^4) = \frac{count(c_1^3 W^4)+0.01}{\sum_{r=1}^{k_3} count(c_r^3 W^4)+0.01*3}$ = 0.01 / 0.03 = 0.3333
		kunangkunang berperam sebagai	$P_3(c_1^3 \mid W^2, W^4) = \frac{count(W^2 c_1^3 W^4)+0.01}{\sum_{r=1}^{k_3} count(W^2 c_r^3 W^4)+0.01*3}$ = 0.01 / 0.03 = 0.3333
c_2^3	berperan	kunangkunang berperan	$P_1(c_2^3 \mid W^2) = \frac{count(W^2 c_2^3)+0.01}{\sum_{r=1}^{k_3} count(W^2 c_r^3)+0.01*3}$ = 1.01 / 1.03 = 0.9806
		berperan sebagai	$P_2(c_2^3 \mid W^4) = \frac{count(c_2^3 W^4)+0.01}{\sum_{r=1}^{k_3} count(c_r^3 W^4)+0.01*3}$ = 0.01 / 0.03 = 0.3333
		kunangkunang berperan sebagai	$P_3(c_2^3 \mid W^2, W^4) = \frac{count(W^2 c_2^3 W^4)+0.01}{\sum_{r=1}^{k_3} count(W^2 c_r^3 W^4)+0.01*3}$ = 0.01 / 0.03 = 0.3333
c_3^3	berseran	kunangkunang berseran	$P_1(c_3^3 \mid W^2) = \frac{count(W^2 c_3^3)+0.01}{\sum_{r=1}^{k_3} count(W^2 c_r^3)+0.01*3}$ = 0.01 / 1.03 = 0.0097
		berseran sebagai	$P_2(c_3^3 \mid W^4) = \frac{count(c_3^3 W^4)+0.01}{\sum_{r=1}^{k_3} count(c_r^3 W^4)+0.01*3}$ = 0.01 / 0.03 = 0.3333
		kunangkunang berseran sebagai	$P_3(c_3^3 \mid W^2, W^4) = \frac{count(W^2 c_3^3 W^4)+0.01}{\sum_{r=1}^{k_3} count(W^2 c_r^3 W^4)+0.01*3}$ = 0.01 / 0.03 = 0.3333
c_1^4	sebagai	berperan sebagai	$P_1(c_1^4 \mid W^3) = \frac{count(W^3 c_1^4)+0.01}{\sum_{r=1}^{k_4} count(W^3 c_r^4)+0.01*1}$ = 0.01/0.01 = 1.0000
		sebagai tandu	$P_2(c_1^4 \mid W^5) = \frac{count(c_1^4 W^5)+0.01}{\sum_{r=1}^{k_4} count(c_r^4 W^5)+0.01*1}$ = 0.01/0.01 = 1.0000
		berperan sebagai tandu	$P_3(c_1^4 \mid W^3, W^5) = \frac{count(W^3 c_1^4 W^5)+0.01}{\sum_{r=1}^{k_4} count(W^3 c_r^4 W^5)+0.01*1}$ = 0.01/0.01 = 1.0000
c_1^5	candu	sebagai candu	$P_1(c_1^5 \mid W^4) = \frac{count(W^4 c_1^5)+0.01}{\sum_{r=1}^{k_5} count(W^4 c_r^5)+0.01*4}$ = 0.01 / 1.04 = 0.0096
		candu peringatan	$P_2(c_1^5 \mid W^6) = \frac{count(c_1^5 W^6)+0.01}{\sum_{r=1}^{k_5} count(c_r^5 W^6)+0.01*4}$ = 0.01/1.04 = 0.0096
		sebagai candu peringatan	$P_3(c_1^5 \mid W^4, W^6) = \frac{count(W^4 c_1^5 W^6)+0.01}{\sum_{r=1}^{k_5} count(W^4 c_r^5 W^6)+0.01*4}$ = 0.01/1.04 = 0.0096

Anggota (c_j^i)		Bigram dan Trigram	Probabilitas $P_1{}'(c_j^i \mid W^{i-1})$ $P_2{}'(c_j^i \mid W^{i+1})$ $P_3{}'(c_j^i \mid W^{i-1}, W^{i+1})$
c_2^5	tanda	sebagai tanda	$P_1(c_2^5 \mid W^4) = \frac{count(W^4 c_2^5)+0.01}{\sum_{r=1}^{k_5} count(W^4 c_r^5)+0.01*4}$ = 1.01/1.04 = 0.9712
		tanda peringatan	$P_2(W_2^5 \mid W^6) = \frac{count(c_2^5 W^6)+0.01}{\sum_{r=1}^{k_5} count(c_r^5 W^6)+0.01*4}$ = 1.01/1.04 = 0.9712
		sebagai tanda peringatan	$P_3(c_2^5 \mid W^4, W^6) = \frac{count(W^4 c_2^5 W^6)+0.01}{\sum_{r=1}^{k_5} count(W^4 c_r^5 W^6)+0.01*4}$ = 1.01/1.04 = 0.9712
c_3^5	tandu	sebagai tandu	$P_1(c_3^5 \mid W^4) = \frac{count(W^4 c_3^5)+0.01}{\sum_{r=1}^{k_5} count(W^4 c_r^5)+0.01*4}$ = 0.01/1.04 = 0.0096
		tandu peringatan	$P_2(c_3^5 \mid W^6) = \frac{count(c_3^5 W^6)+0.01}{\sum_{r=1}^{k_5} count(c_r^5 W^6)+0.01*4}$ = 0.01/1.04 = 0.0096
		sebagai tandu peringatan	$P_3(c_3^5 \mid W^4, W^6) = \frac{count(W^4 c_3^5 W^6)+0.01}{\sum_{r=1}^{k_5} count(W^4 c_r^5 W^6)+0.01*4}$ = 0.01/1.04 = 0.0096
c_4^5	tanduk	sebagai tanduk	$P_1(c_4^5 \mid W^4) = \frac{count(W^4 c_4^5)+0.01}{\sum_{r=1}^{k_5} count(W^4 c_r^5)+0.01*4}$ = 0.01/1.04 = 0.0096
		tanduk peringatan	$P_2(c_4^5 \mid W^6) = \frac{count(c_4^5 W^6)+0.01}{\sum_{r=1}^{k_5} count(c_r^5 W^6)+0.01*4}$ = 0.01/1.04 = 0.0096
		sebagai tanduk peringatan	$P_3(c_4^5 \mid W^4, W^6) = \frac{count(W^4 c_4^5 W^6)+0.01}{\sum_{r=1}^{k_5} count(W^4 c_r^5 W^6)+0.01*4}$ = 0.01/1.04 = 0.0096
c_1^6	keringatan	tandu keringatan	$P_1(c_1^6 \mid W^5) = \frac{count(W^5 c_1^6)+0.01}{\sum_{r=1}^{k_6} count(W^5 c_r^6)+0.01*3}$ = 0.01/0.03 = 0.33
		keringatan _	$P_2(c_1^6 \mid W^7) = \frac{count(c_1^6 W^7)+0.01}{\sum_{r=1}^{k_6} count(c_r^6 W^7)+0.01*3}$ = 0.01/0.03 = 0.3333
		tandu keringatan _	$P_3(c_1^6 \mid W^5, W^7) = \frac{count(W^5 c_1^6 W^7)+0.01}{\sum_{r=1}^{k_6} count(W^5 c_r^6 W^7)+0.01*3}$ = 0.01/0.03 = 0.3333
c_2^6	peringanan	tandu peringanan	$P_1(c_2^6 \mid W^5) = \frac{count(W^5 c_2^6)+0.01}{\sum_{r=1}^{k_6} count(W^5 c_r^6)+0.01*3}$ = 0.01/0.03 = 0.3333
		peringanan _	$P_2(c_2^6 \mid W^7) = \frac{count(c_2^6 W^7)+0.01}{\sum_{r=1}^{k_6} count(c_r^6 W^7)+0.01*3}$ = 0.01/0.03 = 0.3333
		tandu peringanan _	$P_3(c_2^6 \mid W^5, W^7) = \frac{count(W^5 c_2^6 W^7)+0.01}{\sum_{r=1}^{k_6} count(W^5 c_r^6 W^7)+0.01*3}$ = 0.01/0.03 = 0.3333

Anggota (c_j^i)		Bigram dan Trigram	Probabilitas $P_1'(c_j^i \mid W^{i-1})$ $P_2'(c_j^i \mid W^{i+1})$ $P_3'(c_j^i \mid W^{i-1}, W^{i+1})$
c_3^6	peringatan	tandu peringatan	$P_1(c_3^6 \mid W^5) = \frac{count(W^5 c_3^6)+0.01}{\sum_{r=1}^{k_6} count(W^5 c_r^6)+0.01*3}$ = 0.01/0.03 = 0.3333
		peringatan _	$P_2(c_3^6 \mid W^7) = \frac{count(c_3^6 W^7)+0.01}{\sum_{r=1}^{k_6} count(c_r^6 W^7)+0.01*3}$ = 0.01/0.03 = 0.3333
		tandu peringatan _	$P_3(c_3^6 \mid W^5, W^7) = \frac{count(W^5 c_3^6 W^7)+0.01}{\sum_{r=1}^{k_6} count(W^5 c_r^6 W^7)+0.01*3}$ = 0.01/0.03 = 0.3333

Tanda *blank* ("_") dibutuhkan di awal dan akhir n-gram untuk memberikan konteks n-gram pada kata pertama n-gram dan agar n-gram berdistribusi probabilitas sejati (*true probability distribution*). [26]

4. Penghitungan Skor

Selanjutnya hitung skor masing-masing anggota dengan menjumlahkan skor bigram dan trigram untuk masing-masing anggota dengan menggunakan *weighted combination score*, dengan rumus

$$Score(c_j^i) = \lambda_1 P_1(c_j^i \mid W^{i-1}) + \lambda_2 P_2(c_j^i \mid W^{i+1}) + \lambda_3 P_3(c_j^i \mid W^{i-1}, W^{i+1})$$

λ_1 adalah bobot untuk probabilitas bigram kiri, λ_2 adalah bobot untuk probabilitas bigram kanan, dan λ_3 adalah bobot untuk probabilitas trigram. Nilai λ_1, λ_2 dan λ_3 dapat dihitung dengan mengoptimasi akurasi data masukan. Berdasarkan [2], nilai terbaik adalah $\lambda_1 = 0.25$, $\lambda_2 = 0.25$ dan $\lambda_3 = 0.5$. Maka rumus perhitungan skor adalah:

$$Score(c_j^i) = 0.25 P_1(c_j^i \mid W^{i-1}) + 0.25 P_2(c_j^i \mid W^{i+1}) + 0.5 P_3(c_j^i \mid W^{i-1}, W^{i+1})$$

Berikut hasil perhitungan skornya.

Tabel 3.38 Hasil Perhitungan Skor Anggota *Confusion Set* Data Uji

Anggota (c_j^i)		*score*(c_j^i)
c_1^1	bahaya	$0.25P_1(c_1^1 \mid W^0) + 0.25P_2(c_1^1 \mid W^2) + 0.5P_3(c_1^1 \mid W^0, W^2)$ = 0.25*0.0097 + 0.25*0.0049 + 0.5*0.0097 = 0.0085
c_2^1	cahaya	$0.25P_1(c_2^1 \mid W^0) + 0.25P_2(c_2^1 \mid W^2) + 0.5P_3(c_2^1 \mid W^0, W^2)$ = 0.25*0.9806 + 0.25*0.9901 + 0.5*0.9806 = 0.9830
c_3^1	sahaya	$0.25P_1(c_3^1 \mid W^0) + 0.25P_2(c_3^1 \mid W^2) + 0.5P_3(c_3^1 \mid W^0, W^2)$ = 0.25*0.0097 + 0.25*0.0049 + 0.5*0.0097 = 0.0085
c_1^2	kunangkunang	$0.25P_1(c_1^2 \mid W^1) + 0.25P_2(c_1^2 \mid W^3) + 0.5P_3(c_1^2 \mid W^1, W^3)$ = 0.25*1.0000 + 0.25*1.0000 + 0.5*1.0000 = 1.0000

Anggota (c_j^i)		*score*(c_j^i)
c_1^3	berperam	$0.25P_1(c_1^3\|W^2) + 0.25P_2(c_1^3\|W^4) + 0.5P_3(c_1^3\|W^2, W^4)$ = 0.25*0.0097 + 0.25*0.3333 + 0.5*0.3333 = 0.2524
c_2^3	berperan	$0.25P_1(c_2^3\|W^2) + 0.25P_2(c_2^3\|W^4) + 0.5P_3(c_2^3\|W^2, W^4)$ = 0.25*0.9806 + 0.25*0.3333 + 0.5*0.3333 = 0.4951
c_3^3	berseran	$0.25P_1(c_3^3\|W^2) + 0.25P_2(c_3^3\|W^4) + 0.5P_3(c_3^3\|W^2, W^4)$ = 0.25*0.0097 + 0.25*0.3333 + 0.5*0.3333 = 0.2524
c_1^4	sebagai	$0.25P_1(c_1^4\|W^3) + 0.25P_2(c_1^4\|W^5) + 0.5P_3(c_1^4\|W^3, W^5)$ = 0.25*1.0000 + 0.25*1.0000 + 0.5*1.0000 = 1.0000
c_1^5	candu	$0.25P_1(c_1^5\|W^4) + 0.25P_2(c_1^5\|W^6) + 0.5P_3(c_1^5\|W^4, W^6)$ = 0.25*0.0096 + 0.25*0.0096 + 0.5*0.0096 = 0.0096
c_2^5	tanda	$0.25P_1(c_2^5\|W^4) + 0.25P_2(c_2^5\|W^6) + 0.5P_3(c_2^5\|W^4, W^6)$ = 0.25*0.9712 + 0.25*0.9712 + 0.5*0.9712 = 0.9712
c_3^5	tandu	$0.25P_1(c_3^5\|W^4) + 0.25P_2(c_3^5\|W^6) + 0.5P_3(c_3^5\|W^4, W^6)$ = 0.25*0.0096 + 0.25*0.0096 + 0.5*0.0096 = 0.0096
c_4^5	tanduk	$0.25P_1(c_4^5\|W^4) + 0.25P_2(c_4^5\|W^6) + 0.5P_3(c_4^5\|W^4, W^6)$ = 0.25*0.0096 + 0.25*0.0096 + 0.5*0.0096 = 0.0096
c_1^6	keringatan	$0.25P_1(c_1^6\|W^5) + 0.25P_2(c_1^6\|W^7) + 0.5P_3(c_1^6\|W^5, W^7)$ = 0.25*0.3333 + 0.25*0.3333 + 0.5*0.3333 = 0.3333
c_2^6	peringanan	$0.25P_1(c_2^6\|W^5) + 0.25P_2(c_2^6\|W^7) + 0.5P_3(c_2^6\|W^5, W^7)$ = 0.25*0.3333 + 0.25*0.3333 + 0.5*0.3333 = 0.3333
c_3^6	peringatan	$0.25P_1(c_3^6\|W^5) + 0.25P_2(c_3^6\|W^7) + 0.5P_3(c_3^6\|W^5, W^7)$ = 0.25*0.3333 + 0.25*0.3333 + 0.5*0.3333 = 0.3333

5. Deteksi Kesalahan dan Sugesti Kata Koreksi

Untuk mendeteksi kesalahan kata, kita lihat dari urutan peringkat anggota *confusion set*-nya. Pertama urutkan anggota-anggota *confusion set* berdasarkan skornya dari yang terbesar sampai terkecil. Dari urutan peringkat ini, dapat ditentukan apakah sebuah kata merupakan kesalahan *real-word* atau tidak. Jika kata salah, diberikan sugesti kata koreksi yang didapat dari urutan peringkat *confusion set* kata tersebut berdasarkan urutan peringkat *confusion set*. Peringkat teratas berarti sugesti kata yang paling cocok secara konteks.

Hasil dari proses deteksi dan koreksi adalah sebagai berikut. Anggota *confusion set* yang juga merupakan kata yang diuji diberi garis bawah. Jika anggota itu skornya 0 atau kurang dari 1% skor anggota peringkat tertinggi *confusion set* [2], maka kata yang diuji dianggap salah.

Tabel 3.39 Urutan Peringkat *Confusion Set* Data Uji

W^i	anggota (c_j^i)		score(c_j^i)	Status	Sugesti
cahaya	c_2^1	**cahaya**	0.9830	Benar	-
	c_1^1	bahaya	0.0085		
	c_3^1	sahaya	0.0085		
kunangkunang	c_1^2	**kunangkunang**	1.0000	Benar	-
berperan	c_3^3	**berperan**	0.4951	Benar	-
	c_1^3	berperam	0.2524		
	c_2^3	berseran	0.2524		
sebagai	c_1^4	**sebagai**	1.0000	Benar	-
tandu	c_2^5	**tanda**	0.9712	Salah	tanda
	c_1^5	candu	0.0096		
	c_3^5	tandu	0.0096		
	c_4^5	tanduk	0.0096		
peringatan	c_1^6	**keringatan**	0.3333	Benar	-
	c_2^6	peringanan	0.3333		
	c_3^6	peringatan	0.3333		

Pada data uji ini, skor kata yang memenuhi syarat menjadi kesalahan adalah skor kata "tandu", yang berskor 0. Ini berarti kata "tandu" tersebut dideteksi sebagai kesalahan *real-word*. Dalam *confusion set* kata "tandu", anggota dengan peringkat teratas adalah kata "tanda". Ini berarti dalam konteksnya kata "tanda" merupakan kata yang paling cocok, dan kemungkinan besar merupakan kata yang dimaksud.

3.2.3 Analisis Rencana Pengujian

Kinerja dari metode deteksi dan koreksi kesalahan ejaan *real-word* ini diukur dengan melakukan pengukuran akurasi deteksi dan akurasi koreksi. Pengukuran akurasi deteksi dilakukan dengan persamaan

$$akurasi\ deteksi = \frac{ds}{dm} \times 100\%$$

dimana

ds : jumlah deteksi yang dihasilkan sistem

dm : jumlah deteksi yang dihasilkan secara manual

Sedangkaran pengukuran akurasi koreksi dilakukan dengan persamaan

$$akurasi\ koreksi = \frac{kb}{kb + ks} \times 100\%$$

dimana

kb : jumlah koreksi benar yang dihasilkan sistem

ks : jumlah koreksi salah yang dihasilkan sistem

Berikut adalah contoh penghitungan pengujian. Hasil penghitungan diambil dari analisis proses deteksi dan koreksi di atas, yaitu sebagai berikut.

Tabel 3.40 Hasil Deteksi dan Koreksi Kesalahan Kata Data Uji

Kata	Hasil Deteksi		Sugesti Koreksi	
	Sistem	Manual	Sistem	Manual
cahaya	Benar	Benar	-	-
kunangkunang	Benar	Benar	-	-
berperan	Benar	Benar	-	-
sebagai	Benar	Benar	-	-
tandu	Salah	Salah	tanda	tanda
peringatan	Benar	Benar	-	-

Dari hasil proses deteksi dan koreksi yang dilakukan pada Tabel 3.21 di atas, didapat nilai parameter sebagai berikut.

$$ds = 1$$

$$dm = 0$$

$$kb = 1$$

$$ks = 0$$

Maka akurasi hasil pengukuran adalah

$$akurasi\ deteksi = \frac{ds}{dm} \times 100\% = \frac{1}{1} \times 100\% = 100\%$$

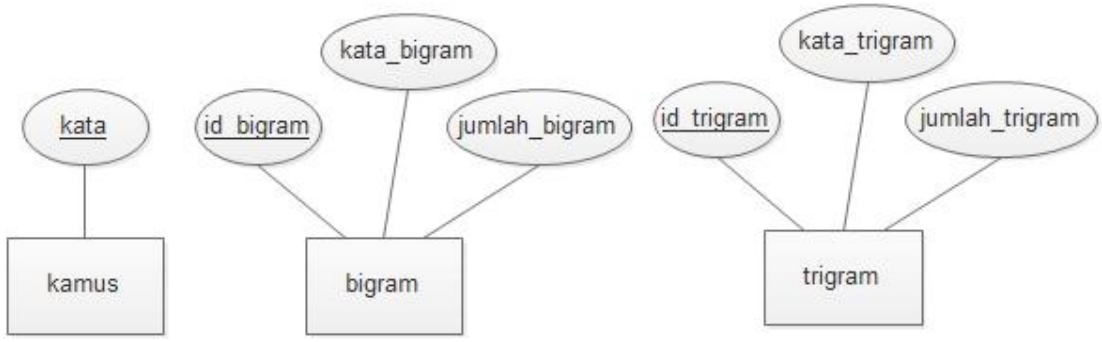
$$akurasi\ deteksi = \frac{kb}{kb + ks} \times 100\% = \frac{1}{1 + 0} \times 100\% = 100\%$$

3.3 Analisis Basis Data

Analisis basis data adalah pendekatan untuk menggambarkan data yang digunakan pada sistem yang akan dibangun yang digambarkan dengan ERD dan dijelaskan dalam kamus data ERD.

3.3.1 *Entity Relationship Diagram*

Sistem yang akan dibangun dapat dilihat pada gambar skema ERD berikut di bawah ini.



Gambar 3.5 Entity Relationship Diagram (ERD)

3.3.2 Kamus Data ERD

Penjelasan mengenai gambar ERD di atas terdapat pada Kamus Data ERD sebagai berikut.

Tabel 3.41 Kamus Data ERD

Entitas	Atribut	Uraian	Tipe	PK/FK
t_kamus	Kata	Kata dalam kamus	Varchar(100)	PK
t_bigram	id_bigram, kata_bigram, jumlah_bigram	No id bigram kata bigram jumlah kemunculan kata bigram	Int(11) Varchar(100) Int(5)	PK
t_trigram	id_trigram, kata_trigram, jumlah_trigram	No id trigram kata trigram jumlah kemunculan kata trigram	Int(11) Varchar(100) Int(5)	PK

3.4 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan Non Fungsional menggambarkan kebutuhan di luar sistem yang diperlukan untuk menperingatankan sistem yang dibangun. Analisis yang dilakukan diantaranya Analisis Kebutuhan Pengguna, Analisis Kebutuhan Perangkat Keras (*Hardware*) dan Analisis Kebutuhan Perangkat Lunak (*Software*).

3.4.1 Analisis Kebutuhan Pengguna

Pengguna sistem ini adalah penguji yang melakukan pengujian terhadap hasil deteksi dan sugesti koreksi yang ditampilkan oleh sistem dalam deteksi dan koreksi kesalahan ejaan *real-word*. Agar sistem dapat diperingatankan dengan baik pengguna harus memenuhi kriteria-kriteria sebagai berikut.

Tabel 3.42 Analisis Kemampuan Pengguna

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

3.4.2 Analisis Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang dibutuhkan untuk menjalankan sistem adalah sebagai berikut.

Tabel 3.43 Spesifikasi Minimum Perangkat Keras

Nama Perangkat Keras	Kebutuhan Minimum
Prosesor	Intel atau AMD setara Pentium IV 3.0 Ghz
RAM	Kapasitas RAM minimal 1 GB
Monitor	Resolusi minimal monitor adalah 1024 x 768 pixel

3.4.3 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan agar sistem ini dapat berjalan dengan baik adalah sebagai berikut.

Tabel 3.44 Spesifikasi Minimum Perangkat Lunak

Nama Perangkat Lunak	Versi Minimum Perangkat Lunak
Sistem Operasi	Intel atau AMD setara Pentium IV 3.0 Ghz
Web Browser	Google Chrome versi 47

3.5 Analisis Kebutuhan Fungsional

Tujuan dari analisis ini adalah untuk menganalisis proses yang akan diterapkan dalam aplikasi deteksi dan koreksi kesalahan ejaan *real-word* yang akan dibangun menggunakan Diagram Konteks, Pembuatan DFD, dan Spesifikasi Proses.

3.5.1 Diagram Konteks

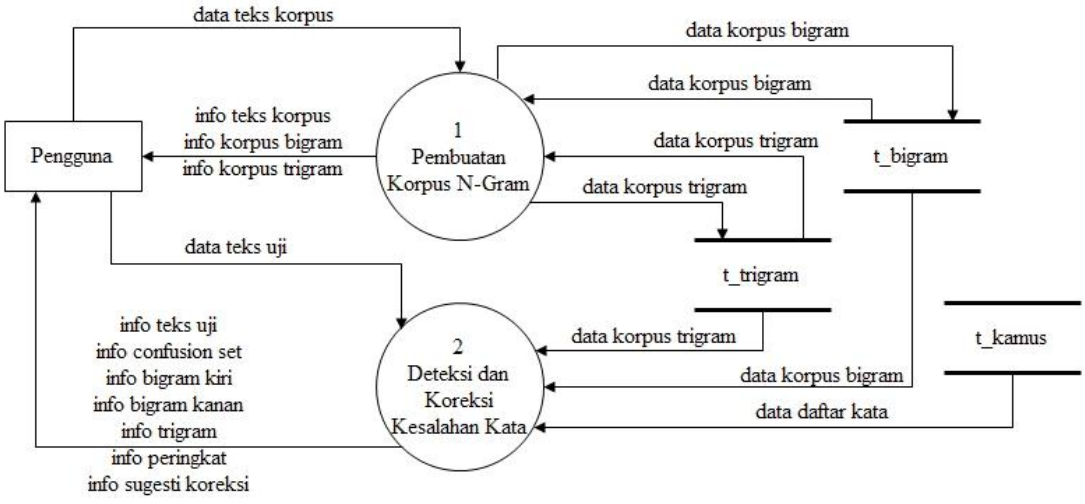
Diagram Konteks menggambarkan ruang lingkup pembangunan aplikasi deteksi dan koreksi kesalahan kata *real-word* dengan Pengguna sebagai entitas luar atau pengguna sistem. Diagram konteks digambarkan sebagai berikut.



Gambar 3.6 Diagram Konteks

3.5.2 *Data Flow Diagram* (DFD) Level 1

Data flow diagram level 1 berisi pemecahan proses dari diagram konteks. Di dalam diagram ini memuat penyimpanan data yang terdiri dari t_kamus, t_bigram dan t_trigram, dan menjelaskan aliran data yang terdapat pada aplikasi deteksi dan koreksi kesalahan kata diantaranya terdapat dua proses, yaitu proses 1 Pembuatan Korpus N-Gram dan 2 Deteksi dan Koreksi Kesalahan Kata. DFD Level 1 ditunjukkan pada gambar berikut.

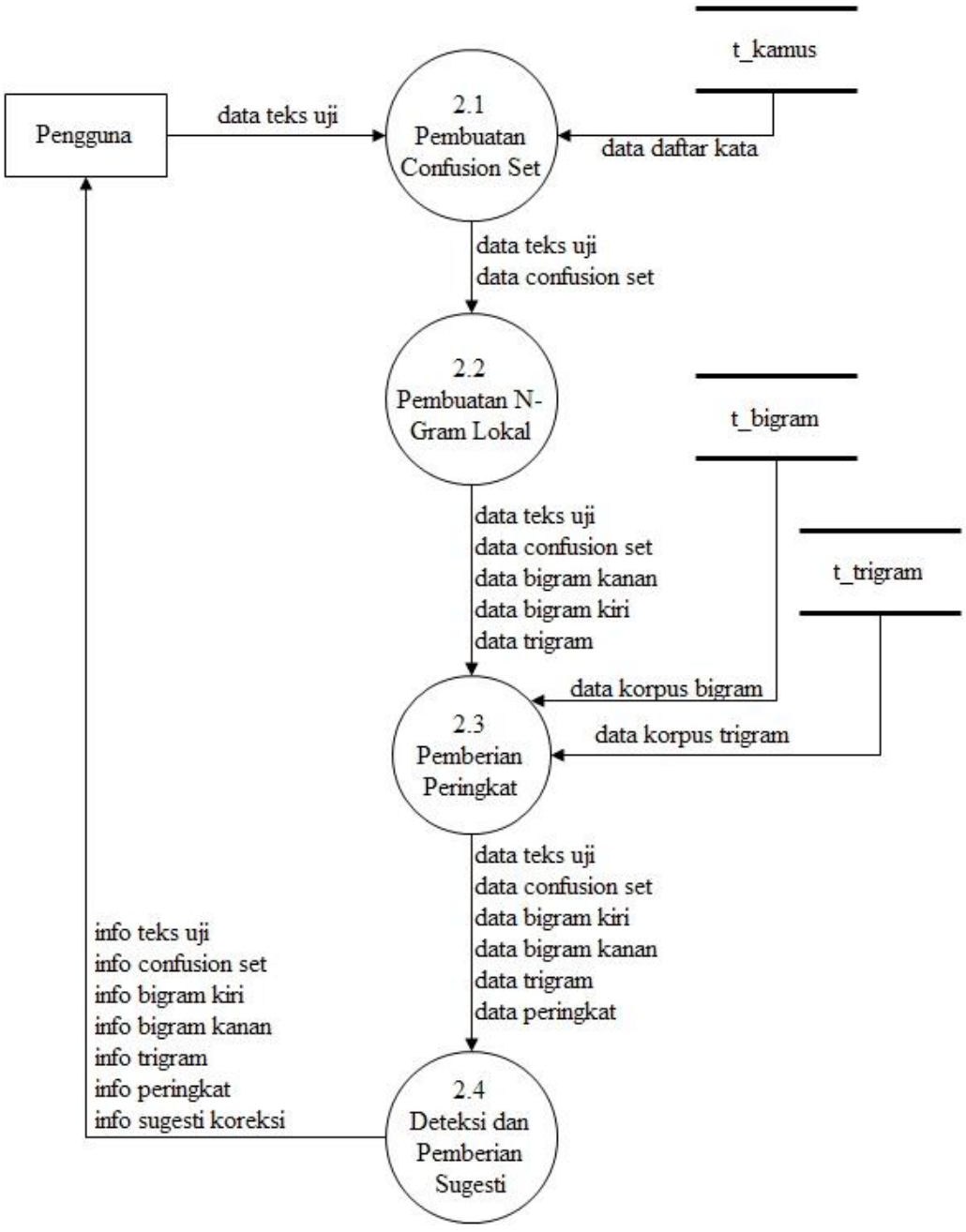


Gambar 3.7 DFD Level 1

3.5.3 *Data Flow Diagram* (DFD) Level 2 Deteksi dan Koreksi Kesalahan

DFD ini menggambarkan proses deteksi dan koreksi kesalahan ejaan *real-word*. Data masukan yang diproses adalah teks yang berisi kesalahan ejaan

real-word berformat .txt. Dalam tiap tahapannya teks diolah sehingga menghasilkan daftar sugesti koreksi kata. Data keluaran adalah daftar kata-kata yang dianggap salah serta sugesti kata untuk mengoreksinya. Berikut di bawah ini merupakan DFD dari deteksi dan koreksi kesalahan.



Gambar 3.8 DFD Level 2 Deteksi dan Koreksi Kesalahan

3.5.4 Spesifikasi Proses

Spesifikasi proses digunakan untuk menggambarkan proses model aliran yang terdapat pada DFD. Spesifikasi proses dari DFD level 2 di atas akan dijelaskan pada tabel dibawah ini.

Tabel 3.45 Spesifikasi Proses

No	Proses	Keterangan
1.	No. Proses	1
	Nama Proses	Pembuatan Korpus N-Gram
	Source (Sumber)	Pengguna
	Input	Data teks korpus
	Output	Info teks korpus, info korpus bigram, info korpus trigram
	Logika Proses	1. Pengguna memasukkan data teks korpus ke dalam sistem. 2. Sistem mengubah data teks korpus menjadi data korpus bigram dan data korpus trigram dan menyimpannya di *database*. 3. Keluaran berupa info teks korpus, info korpus bigram, dan info korpus trigram.
2.	No. Proses	2.1
	Nama Proses	Pembuatan *Confusion Set*
	Source (Sumber)	Pengguna
	Input	Data teks uji
	Output	Data teks uji, data *confusion set*
	Logika Proses	1. Pengguna memasukkan data teks uji ke dalam sistem. 2. Sistem membuat *confusion set* untuk setiap kata dalam teks berdasarkan data daftar kata dalam t_kamus. 3. Keluaran berupa data teks uji beserta data *confusion set* untuk setiap katanya.
3.	No. Proses	2.2
	Nama Proses	Pembuatan N-Gram Lokal
	Source (Sumber)	Proses Pembuatan *Confusion Set*
	Input	Data teks uji, data *confusion set*
	Output	Data teks uji, data *confusion set*, data bigram kanan, data bigram kiri, data trigram
	Logika Proses	1. Proses pembuatan *confusion set* memasukkan data teks uji dan data *confusion set* ke dalam proses pembuatan n-gram lokal. 2. Sistem membuat bigram kanan, bigram kiri, dan trigram untuk setiap anggota *confusion set* untuk setiap kata. 3. Keluaran berupa data teks uji, data *confusion set* untuk setiap kata, berserta data bigram kiri, data bigram kanan, dan data trigram untuk tiap-tiap anggota *confusion set*.
4.	No. Proses	2.3
	Nama Proses	Pemberian Peringkat
	Source (Sumber)	Proses Pembuatan N-Gram Lokal
	Input	Data teks uji, data *confusion set*, data bigram kanan, data bigram kiri, data trigram
	Output	Data teks uji, data *confusion set*, data bigram kanan, data bigram kiri, data trigram, data peringkat
	Logika Proses	1. Proses pembuatan n-gram lokal memasukkan data teks uji, data *confusion set*, data bigram kanan, data bigram kiri, dan

No	Proses	Keterangan
		data trigram ke dalam proses pemberian peringkat. 2. Sistem memberikan skor peringkat untuk setiap anggota *confusion set* untuk setiap kata berdasarkan data korpus pada t_bigram dan t_trigram. 3. Keluaran berupa data teks uji, data *confusion set* untuk setiap kata, berserta data bigram kiri, data bigram kanan, dan data trigram untuk tiap-tiap anggota *confusion set*, dengan data peringkat untuk setiap anggotanya.
5.	No. Proses	2.4
	Nama Proses	Deteksi dan Pemberian Sugesti
	Source (Sumber)	Proses Pemberian Peringkat
	Input	Data teks uji, data *confusion set*, data bigram kanan, data bigram kiri, data trigram, data peringkat
	Output	Info teks uji, info *confusion set*, info bigram kiri, info bigram kanan, info trigram, info peringkat, info sugesti koreksi
	Logika Proses	1. Proses pemberian peringkat memasukkan data teks uji, data *confusion set*, data bigram kanan, data bigram kiri, data trigram, dan data peringkat ke dalam proses deteksi dan pemberian sugesti. 2. Sistem mendeteksi kata-kata yang salah berdasarkan peringkat skor tiap *confusion set* dan memberikan sugesti koreksi kata. 3. Keluaran berupa info teks uji, info *confusion set*, info bigram kiri, info bigram kanan, info trigram, dan info peringkat, beserta info sugesti koreksi untuk setiap kata yang salah kepada Pengguna.

3.5.5 Kamus Data DFD

Kamus data merupakan deskripsi formal dari seluruh elemen atau aliran data yang terdapat dalam data flow diagram. Kamus data yang terdapat pada aplikasi deteksi dan koreksi kesalahan ejaan *real-word* adalah sebagai berikut.

Tabel 3.46 Kamus Data DFD

No	Detail	Keterangan
1	Nama	Data Teks Korpus
	Digunakan pada saat	Proses 1 Pembuatan Korpus N-Gram
	Deskripsi	Dokumen yang akan dijadikan korpus bigram dan trigram
	Struktur data	Teks
	Penjelasan struktur data: Teks	 [0..9 | A..Z | a..z | symbol]
2	Nama	Data Korpus Bigram
	Digunakan pada saat	Proses 1 Pembuatan Korpus N-Gram Proses 2.3 Pemberian Peringkat
	Deskripsi	Data yang berisi daftar bigram kata beserta jumlah kemunculannya
	Struktur data	Id Kata Jumlah
	Penjelasan struktur data: Id	 [0..9]

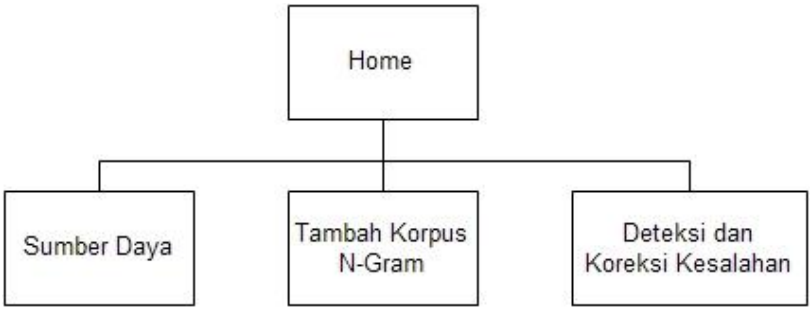
No	Detail	Keterangan
	Kata Jumlah	[a..z | space] [0..9]
3	Nama	Data Korpus Trigram
	Digunakan pada saat	Proses 1 Pembuatan Korpus N-Gram Proses 2.3 Pemberian Peringkat
	Deskripsi	Data yang berisi daftar trigram kata beserta jumlah kemunculannya
	Struktur data	Id Kata Jumlah
	Penjelasan struktur data: Id Kata Jumlah	 [0..9] [a..z | space] [0..9]
4	Nama	Info Teks Korpus
	Digunakan pada saat	Proses 1 Pembuatan Korpus N-Gram Proses 2.3 Pemberian Peringkat
	Deskripsi	Data teks yang dipilih untuk dijadikan korpus
	Struktur data	Teks
	Penjelasan struktur data: Teks	 [0..9 | A..Z | a..z | symbol]
5	Nama	Info Korpus Bigram
	Digunakan pada saat	Proses 1 Pembuatan Korpus N-Gram
	Deskripsi	Data hasil pembuatan korpus bigram yang berisi daftar bigram kata beserta jumlah kemunculannya
	Struktur data	Id Kata Jumlah
	Penjelasan struktur data: Id Kata Jumlah	 [0..9] [a..z | space] [0..9]
6	Nama	Info Korpus Trigram
	Digunakan pada saat	Proses 1 Pembuatan Korpus N-Gram
	Deskripsi	Data hasil pembuatan korpus trigram yang berisi daftar trigram kata beserta jumlah kemunculannya
	Struktur data	Id Kata Jumlah
	Penjelasan struktur data: Id Kata Jumlah	 [0..9] [a..z | space] [0..9]
7	Nama	Data Teks Uji
	Digunakan pada saat	Proses 2.1 Pembuatan *Confusion Set* Proses 2.2 Pembuatan N-Gram Lokal Proses 2.3 Pemberian Peringkat Proses 2.4 Deteksi dan Pemberian Sugesti
	Deskripsi	Dokumen teks yang akan diperiksa akan adanya kesalahan ejaan *real-word*
	Struktur data	Teks
	Penjelasan struktur data: Teks	 [0..9 | A..Z | a..z | symbol]

No	Detail	Keterangan
8	Nama	Data Daftar Kata
	Digunakan pada saat	Proses 2.1 Pembuatan *Confusion Set*
	Deskripsi	Data daftar kata valid dari KBBI
	Struktur data	Kata
	Penjelasan struktur data: Kata	 [a..z]
9	Nama	Data *Confusion Set*
	Digunakan pada saat	Proses 2.1 Pembuatan *Confusion Set* Proses 2.2 Pembuatan N-Gram Lokal Proses 2.3 Pemberian Peringkat Proses 2.4 Deteksi dan Pemberian Sugesti
	Deskripsi	Himpunan kata yang dapat dibingungkan dengan kata yang diperiksa
	Struktur data	Kata
	Penjelasan struktur data: Kata	 [a..z]
10	Nama	Data Bigram Kiri
	Digunakan pada saat	Proses 2.2 Pembuatan N-Gram Lokal Proses 2.3 Pemberian Peringkat Proses 2.4 Deteksi dan Pemberian Sugesti
	Deskripsi	Bigram yang dibuat dari anggota *confusion set* dengan kata di sebelah kiri kata yang diperiksa
	Struktur data	Kata
	Penjelasan struktur data: Kata	 [a..z | space]
11	Nama	Data Bigram Kanan
	Digunakan pada saat	Proses 2.2 Pembuatan N-Gram Lokal Proses 2.3 Pemberian Peringkat Proses 2.4 Deteksi dan Pemberian Sugesti
	Deskripsi	Bigram yang dibuat dari anggota *confusion set* dengan kata di sebelah kanan kata yang diperiksa
	Struktur data	Kata
	Penjelasan struktur data: Kata	 [a..z | space]
12	Nama	Data Trigram
	Digunakan pada saat	Proses 2.2 Pembuatan N-Gram Lokal Proses 2.3 Pemberian Peringkat Proses 2.4 Deteksi dan Pemberian Sugesti
	Deskripsi	Trigram yang dibuat dari anggota *confusion set* dengan kata di sebelah kiri dan kanan kata yang diperiksa
	Struktur data	Kata
	Penjelasan struktur data: Kata	 [a..z | space]
13	Nama	Data Peringkat
	Digunakan pada saat	Proses 2.3 Pemberian Peringkat Proses 2.4 Deteksi dan Pemberian Sugesti
	Deskripsi	*Confusion set* yang anggotanya telah diberi peringkat dan diurutkan dari peringkat tertinggi hingga terendah
	Struktur data	Kata Skor
	Penjelasan struktur data: Kata Skor	 [a..z] [0..9]

No	Detail	Keterangan
14	Nama	Info Teks Uji
	Digunakan pada saat	Proses 2.4 Deteksi dan Pemberian Sugesti
	Deskripsi	Data teks yang dipilih untuk diperiksa akan adanya kesalahan ejaan *real-word*
	Struktur data	Teks
	Penjelasan struktur data: Teks	 [0..9 | A..Z | a..z | symbol]
15	Nama	Info *Confusion Set*
	Digunakan pada saat	Proses 2.4 Deteksi dan Pemberian Sugesti
	Deskripsi	Hasil pembuatan *confusion set*
	Struktur data	Kata
	Penjelasan struktur data: Kata	 [a..z]
16	Nama	Info Bigram Kiri
	Digunakan pada saat	Proses 2.4 Deteksi dan Pemberian Sugesti
	Deskripsi	Hasil pembuatan bigram kiri
	Struktur data	Kata
	Penjelasan struktur data: Kata	 [a..z | space]
17	Nama	Info Bigram Kanan
	Digunakan pada saat	Proses 2.4 Deteksi dan Pemberian Sugesti
	Deskripsi	Hasil pembuatan bigram kanan
	Struktur data	Kata
	Penjelasan struktur data: Kata	 [a..z | space]
18	Nama	Info Trigram
	Digunakan pada saat	Proses 2.4 Deteksi dan Pemberian Sugesti
	Deskripsi	Hasil pembuatan trigram
	Struktur data	Kata
	Penjelasan struktur data: Kata	 [a..z | space]
19	Nama	Info Peringkat
	Digunakan pada saat	Proses 2.4 Deteksi dan Pemberian Sugesti
	Deskripsi	Hasil pemberian peringkat dan pengurutan *confusion set*
	Struktur data	Kata Skor
	Penjelasan struktur data: Kata Skor	 [a..z] [0..9]
20	Nama	Info Sugesti Koreksi
	Digunakan pada saat	Proses 2.4 Deteksi dan Pemberian Sugesti
	Deskripsi	Daftar kata yang terdeteksi salah beserta sugesti kata koreksinya
	Struktur data	Kata Salah Sugesti Kata
	Penjelasan struktur data: Kata Salah Sugesti Kata	 [a..z] [a..z]

3.6 Perancangan Struktur Menu

Perancangan struktur menu digunakan untuk menggambarkan perancangan menu proses yang dapat digunakan dalam menggunakan aplikasi deteksi dan koreksi kesalahan kata *real-word.*. Struktur menu yang akan dibuat pada sistem yang akan dibangun terlihat pada gambar berikut di bawah ini.



Gambar 3.9 Struktur Menu

Aplikasi deteksi dan koreksi kesalahan ejaan *real-word* dalam bahasa Indonesia memiliki 4 halaman. Halaman *Home* merupakan halaman utama aplikasi. Halaman Sumber Daya memuat daftar kata kamus, korpus bigram, dan korpus trigram yang digunakan. Halaman Tambah Korpus N-Gram memuat proses untuk menambah korpus bigram dan trigram. Terakhir pada halaman Deteksi dan Koreksi Kesalahan terdapat proses untuk deteksi dan koreksi kesalahan kata *real-word*.

3.7 Perancangan Antarmuka

Perancangan antarmuka dibuat untuk menggambarkan tampilan program yang akan digunakan oleh pengguna untuk berinteraksi dengan sistem yang akan dibuat. Perancangan dibuat berdasarkan tampilan antarmuka yang akan dihasilkan saat program diimplementasikan. Perancangan antarmuka dalam aplikasi yang akan dibangun akan dibahas sebagai berikut.

Nama Tampilan: T01



Keterangan:

Klik Home untuk menuju ke T01.

Klik Sumberdaya untuk menuju ke T02.

Klik Tambah Korpus N-Gram untuk menuju ke T03.

Klik Deteksi dan Koreksi Kesalahan untuk menuju ke T04.

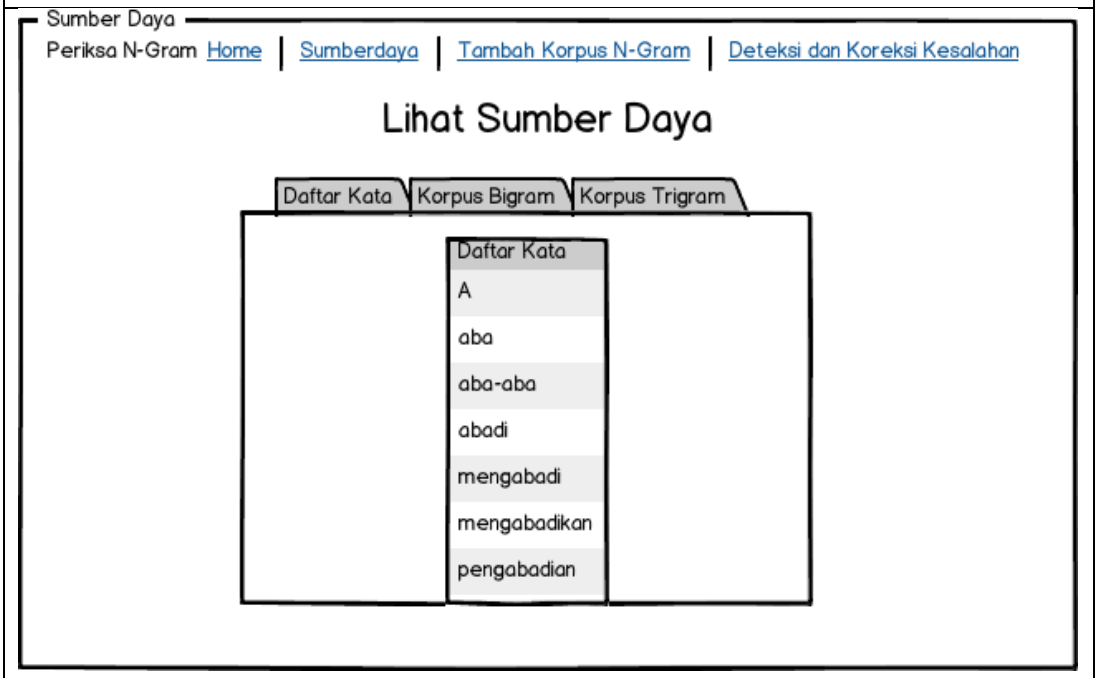
Ukuran Layar = 1366 x 768

Font = Helvetica Neue

Warna Background = Putih

Gambar 3.10 Antarmuka Halaman Home

Nama Tampilan: T02



Keterangan:

Klik Home untuk menuju ke T01.

Klik Sumberdaya untuk menuju ke T02.

Klik Tambah Korpus N-Gram untuk menuju ke T03.

Klik Deteksi dan Koreksi Kesalahan untuk menuju ke T04.

Klik Daftar Kata untuk melihat daftar kata kamus.

Klik Korpus Bigram untuk melihat korpus bigram.

Klik Korpus Trigram untuk melihat korpus trigram.

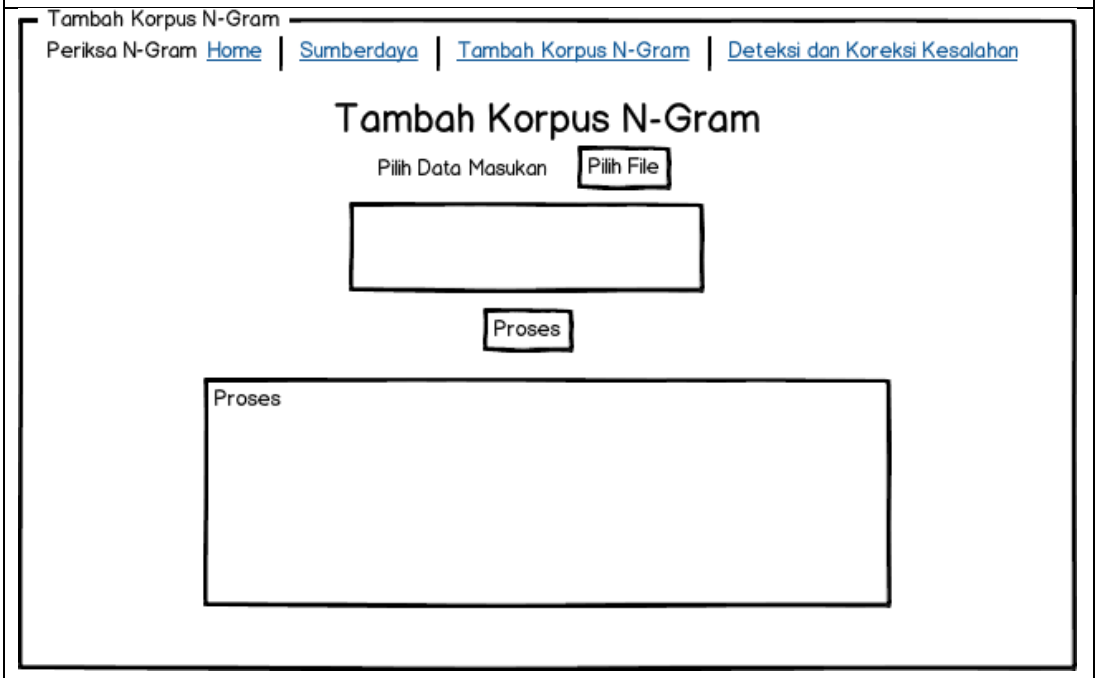
Ukuran Layar = 1366 x 768

Font = Helvetica Neue

Warna Background = Putih

Gambar 3.11 Antarmuka Halaman Sumber Daya

Nama Tampilan: T03



Keterangan:

Klik Home untuk menuju ke T01.

Klik Sumberdaya untuk menuju ke T02.

Klik Tambah Korpus N-Gram untuk menuju ke T03.

Klik Deteksi dan Koreksi Kesalahan untuk menuju ke T04.

Klik Pilih File untuk memilih data masukan yang akan dijadikan korpus n-gram.

Klik Proses untuk memulai proses penambahan korpus n-gram.

Bagian Proses menampilkan proses penambahan korpus n-gram.

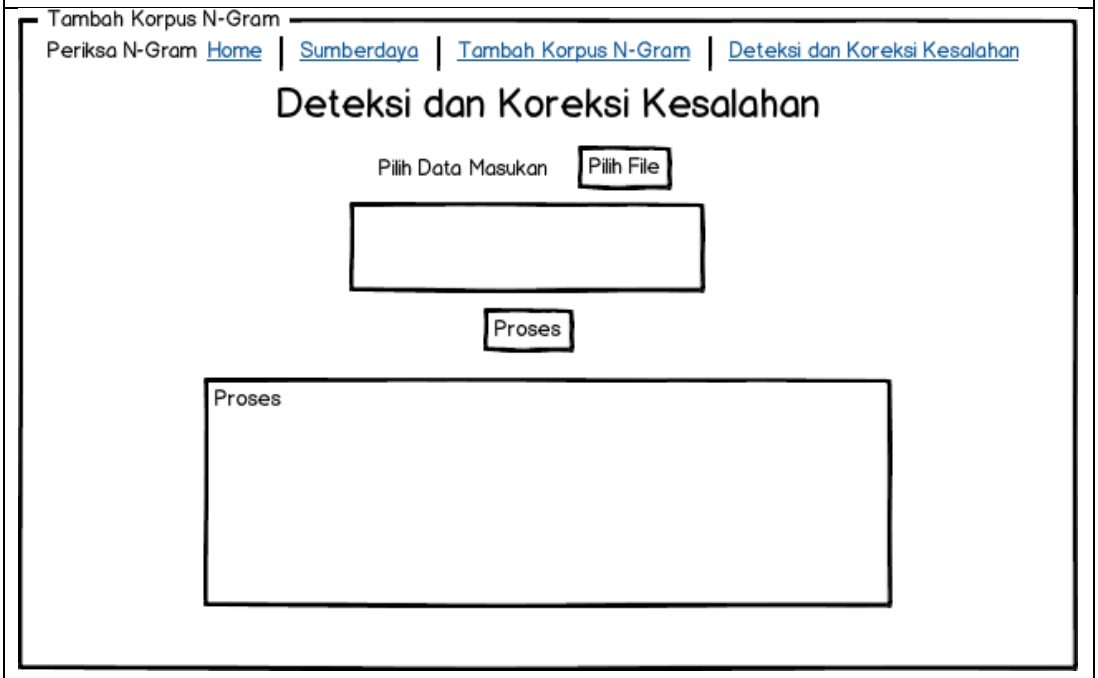
Ukuran Layar = 1366 x 768

Font = Helvetica Neue

Warna Background = Putih

Gambar 3.12 Antarmuka Halaman Tambah Korpus N-Gram

Nama Tampilan: T04



Keterangan:

Klik Home untuk menuju ke T01.

Klik Sumberdaya untuk menuju ke T02.

Klik Tambah Korpus N-Gram untuk menuju ke T03.

Klik Deteksi dan Koreksi Kesalahan untuk menuju ke T04.

Klik Pilih File untuk memilih data uji yang akan diperiksa.

Klik Proses untuk memulai proses deteksi dan koreksi kesalahan kata.

Bagian Proses menampilkan proses deteksi dan koreksi kesalahan kata.

Ukuran Layar = 1366 x 768

Font = Helvetica Neue

Warna Background = Putih

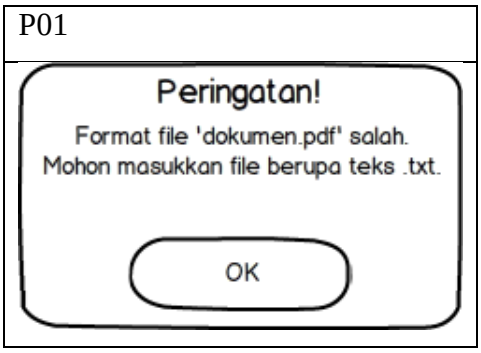
Gambar 3.13 Antarmuka Halaman Deteksi dan Koreksi Kesalahan

3.8 Perancangan Pesan

Perancangan pesan adalah keluaran dalam proses yang dijalankan dalam pengoperasian sebuah proses.

1. Pesan proses *file* masukan

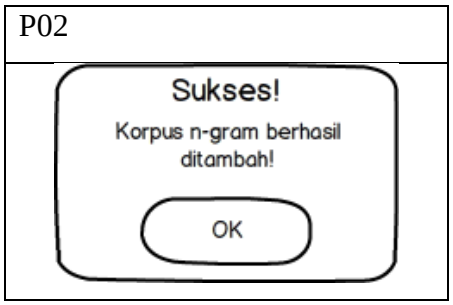
   Pesan yang ditampilkan ketika pengguna memasukkan *file* yang formatnya bukan merupakan .txt. Contoh, pengguna memasukan *file* dengan tipe .pdf.



Gambar 3.14 Perancangan Pesan (P01)

2. Pesan tambah korpus berhasil

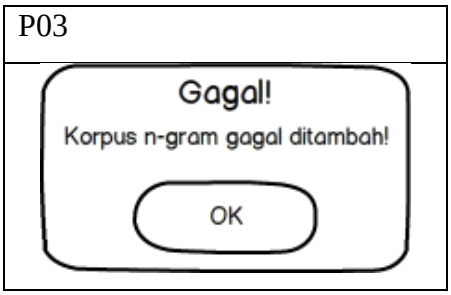
   Pesan yang ditampilkan ketika aplikasi selesai menambah data korpus n-gram ke *database*.



Gambar 3.15 Perancangan Pesan (P02)

3. Pesan tambah korpus gagal

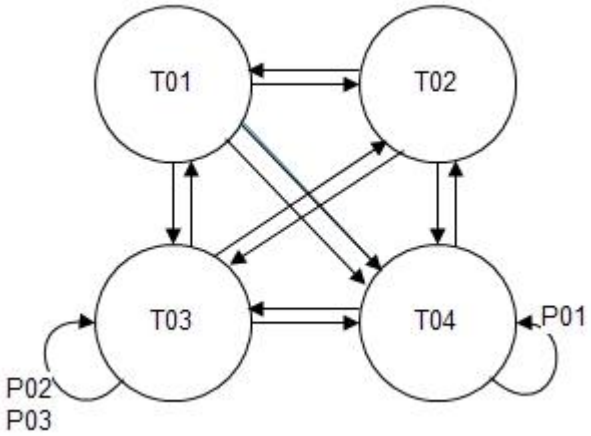
   Pesan yang ditampilkan ketika aplikasi tidak bisa menambah data korpus n-gram ke *database*.



Gambar 3.16 Perancangan Pesan (P03)

3.9 Perancangan Jaringan Semantik

Perancangan jaringan semantik adalah perancangan jaringan penghubung menu-menu yang dapat diakses oleh pengguna dengan pesan kesalahan yang terjadi pada setiap proses. Perancangan jaringan semantik dalam aplikasi deteksi dan koreksi kesalahan ejaan *real-word* yang akan dibangun adalah sebagai berikut.



Gambar 3.17 Jaringan Semantik