BAB 4

PENDAHULUAN

4.1 Implementasi

Implementasi merupakan tahap pembangunan simulator yang merupakan hasil penerjemahan dari perancangan berdasarkan analisis yang telah dibuat. Tahap ini juga merupakan tahap dimana aplikasi siap dioperasikan.

4.2 Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam implementasi algoritma CYK dan Levenshtein ini dapat dilihat pada tabel 4.1.

Tabel 4.1 Implementasi Perangkat Keras

No.	Perangkat Keras	Spesifikasi
1	*Processor*	Intel Core i3-3220 @ 3.30GHz
2	RAM	8 GB
3	VGA	1 GB
4	*Harddisk*	500 GB
5	Monitor	Resolusi 1366 x 768 pixels

4.3 Implementasi Perangkat Lunak

Adapun spesifikasi perangkat lunak yang digunakan dalam pembuatan simulator ini dapat dilihat pada tabel 4.2.

Tabel 4.2 Implementasi Perangkat Lunak

No.	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 8
2	*Tools*	Microsoft Visual Studio 2012 dan NET Framework 4

4.4 Implementasi *Class*

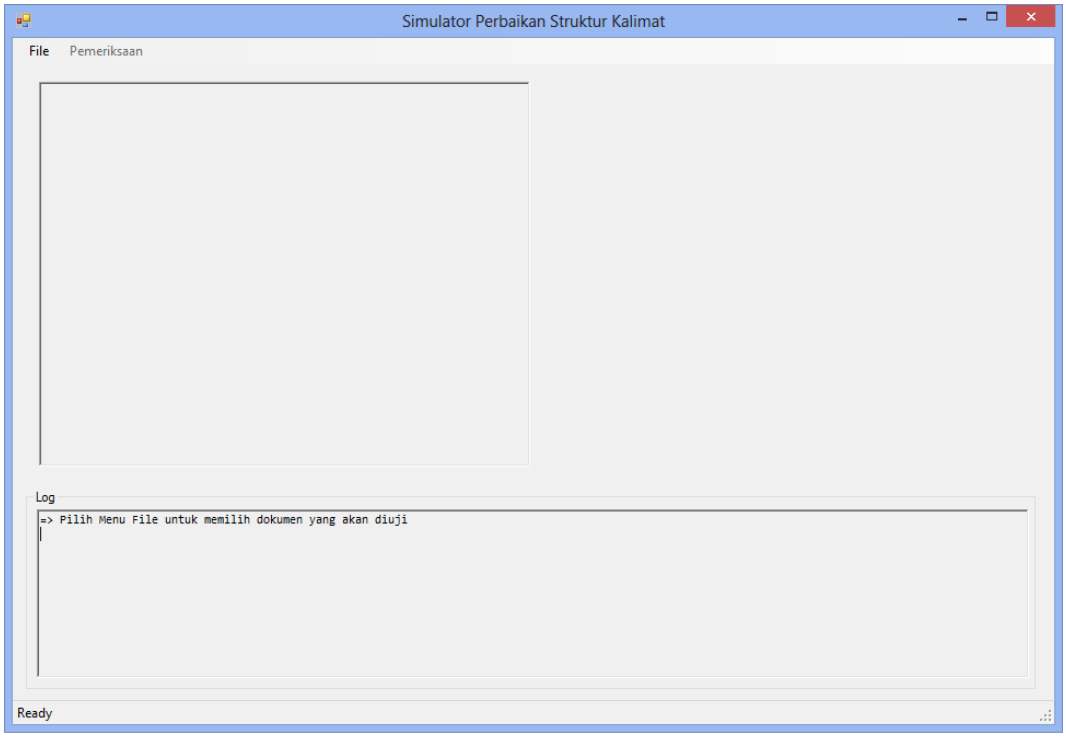
Implementasi *class* merupakan implementasi dari hasil kelas yang ada pada *class diagram*. *Class* yang terbentuk dapat dilihat pada tabel 4.3.

Tabel 4.3 Implementasi *Class*

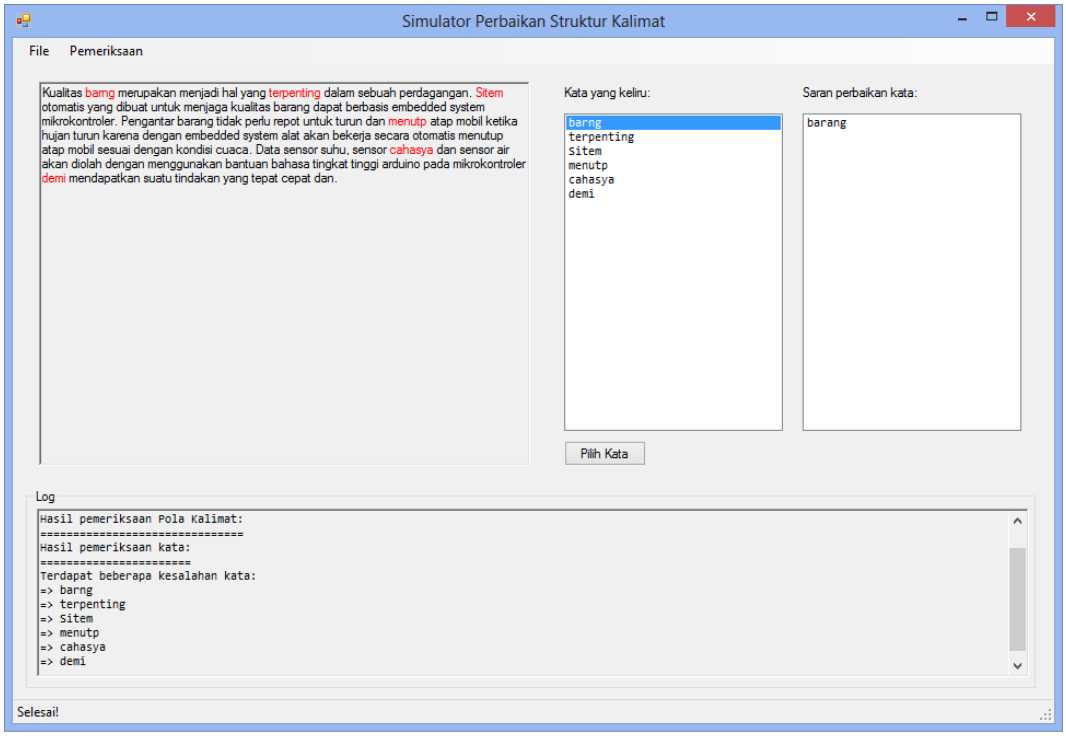
Nama Class	Deskripsi
MainApp.cs	Kelas untuk menampilkan dan mengatur tampilan
WordDocument.cs	Kelas untuk mengolah dokumen abstrak skripsi
Words.cs	Kelas untuk melakukan *tagging* berdasarkan POS Tag
CYK.cs	Kelas untuk melakukan pengenalan kalimat pada abstrak
Levenshtein.cs	Kelas untuk melakukan perbandingan karakter

4.5 Implementasi Antarmuka

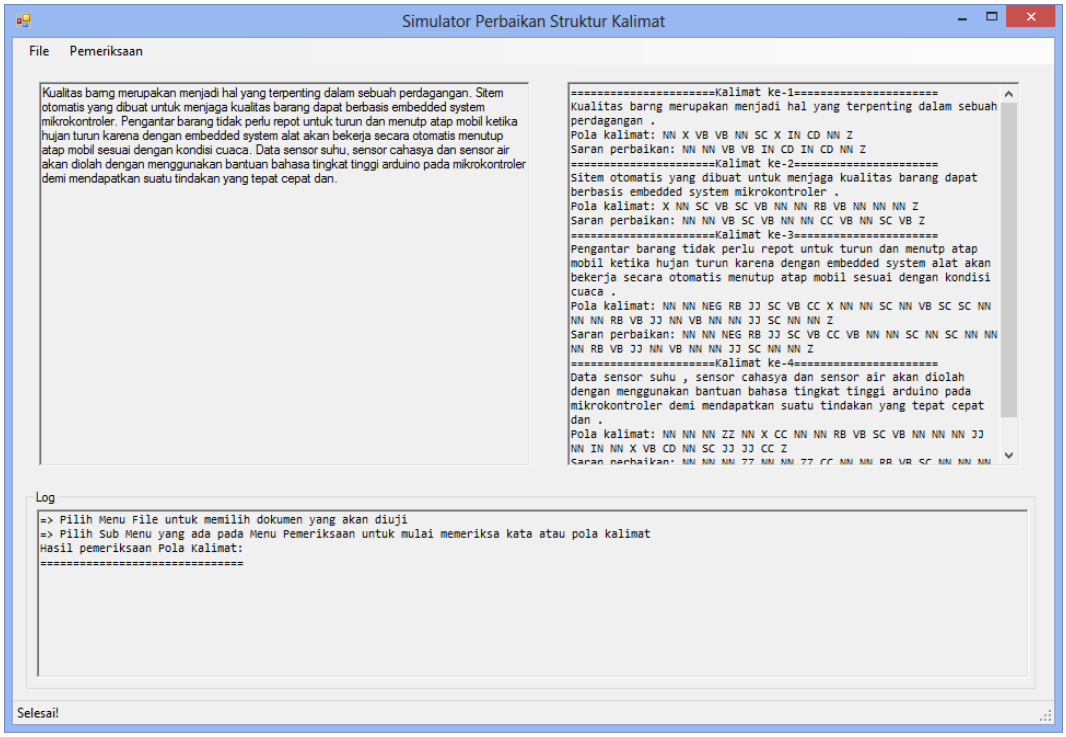
Antarmuka merupakan sarana interaksi antara pengguna dengan sistem yang dibangun. Berikut ini implementasi antarmuka yang dapat dilihat pada gambar 4.1.



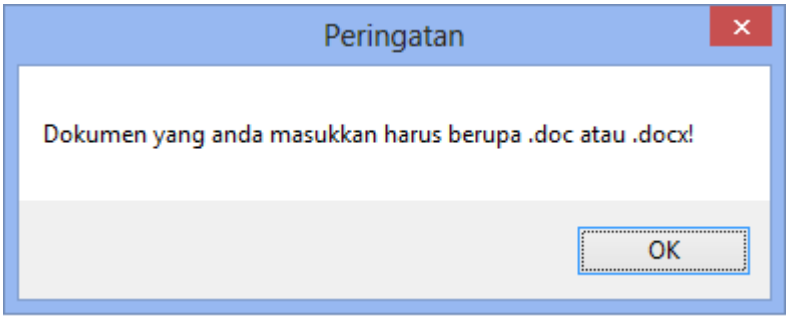
Gambar 4.1 Implementasi antarmuka utama



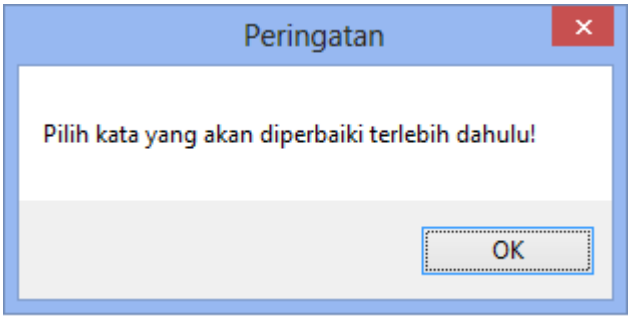
Gambar 4.2 Implementasi antarmuka perbaikan kata



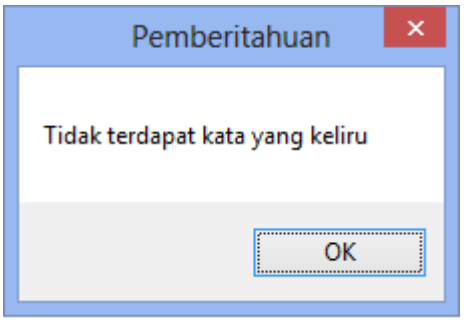
Gambar 4.3 Implementasi antarmuka perbaikan pola kalimat



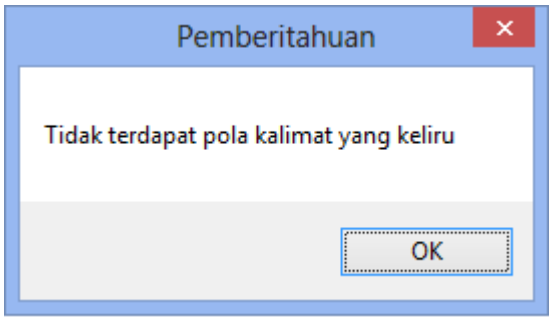
Gambar 4.4 Implementasi antarmuka pesan kesalahan dokumen



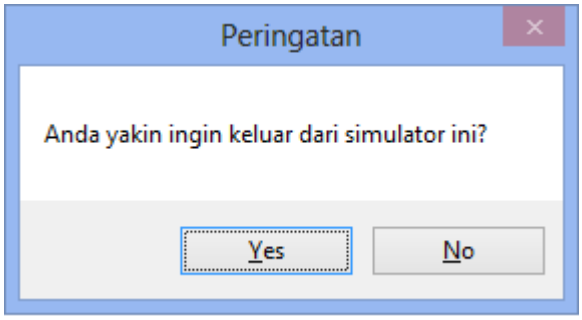
Gambar 4.5 Implementasi antarmuka pesan kesalahan pemilihan kata



Gambar 4.6 Implementasi antarmuka pesan pemeriksaan kata



Gambar 4.7 Implementasi antarmuka pesan perbaikan pola kalimat



Gambar 4.8 Implementasi antarmuka pesan keluar simulator

4.6 Pengujian

Tahap ini ditujukan untuk mengukur tingkat keberhasilan perbaikan pola kalimat menggunakan algoritma Levenshtein dengan bantuan algoritma CYK sebagai pengenal kalimat. Metode pengujian yang digunakan dalam pengujian alpha adalah metode blackbox. Tahap ini dilakukan untuk memastikan simulator ini dapat berjalan sesuai dengan kebutuhan. Pengujian alpha menitikberatkan pada persyaratan fungsional perangkat lunak.

4.6.1 Skenario Pengujian Simulator

Skenario pengujian simulator ini dapat dilihat pada tabel 4.4.

Tabel 4.4 Skenario Pengujian Simulator

No.	Komponen Pengujian	Detail Pengujian	Jenis Pengujian
1	Data masukkan	Mengambil paragraf yang ada pada dokumen	*Blackbox*
2	Pemeriksaan kelas kata	Memeriksa keluaran dari pengelompokkan kata	*Blackbox*
3	Pemilihan perbaikan kata	Memeriksa keluaran dari pemilihan perbaikan kata	*Blackbox*
4	Pemeriksaan pola kalimat	Memeriksa keluaran dari hasil perbaikan pola kalimat	*Blackbox*
5	Keluar simulator	Memeriksa keluaran dari menutup simulator	*Blackbox*

4.6.2 Kasus dan Hasil Pengujian (*Blackbox Testing*)

Pengujian ini dilakukan untuk mengetahui apakah fungsi yang dijalankan sudah sesuai dengan spesifikasi yang dibutuhkan. Kasus dan hasil pengujian dapat dilihat pada tabel 4.5.

Tabel 4.5 Pengujian data masukan

No	Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Hasil Pengujian
1	File dokumen dengan ekstensi .doc / .docx	Memecah paragraf menjadi satuan terkecil dan menampilkan seluruh kata	Memecah paragraf menjadi satuan terkecil dan menampilkan seluruh kata	[√] Diterima [ ] Ditolak
2	File dokumen bukan berekstensi .doc / .docx	Menampilkan pesan kesalahan	Menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

Tabel 4.6 Pengujian Kelas Kata

No	Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Hasil Pengujian
1	Kata	Dapat mengelompokkan kata berdasarkan POS Tag	Dapat mengelompokkan kata berdasarkan POS Tag	[√] Diterima [ ] Ditolak
2	Angka	Dapat mengelompokkan angka berdasarkan POS Tag	Dapat mengelompokkan angka berdasarkan POS Tag	[√] Diterima [ ] Ditolak
3	Simbol	Dapat mengelompokkan simbol berdasarkan POS Tag	Dapat mengelompokkan simbol berdasarkan POS Tag	[√] Diterima [ ] Ditolak

Tabel 4.7 Pengujian Perbaikan Kata

No	Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Hasil Pengujian
1	Perbaikan kata dipilih	Kata yang keliru diganti dengan kata yang diperbaiki	Kata yang keliru diganti dengan kata yang diperbaiki	[√] Diterima [ ] Ditolak
2	Perbaikan kata tidak dipilih	Menampilkan pesan kesalahan	Menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

Tabel 4.8 Pengujian Perbaikan Pola Kalimat

No	Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Hasil Pengujian
1	Pola kalimat keliru	Pola kalimat diperbaiki dan diberikan saran perbaikan	Pola kalimat diperbaiki dan diberikan saran perbaikan	[√] Diterima [ ] Ditolak
2	Pola kalimat tidak keliru	Menampilkan pesan tidak ada kesalahan	Menampilkan pesan tidak ada kesalahan	[√] Diterima [ ] Ditolak

Tabel 4.9 Pengujian Keluar Simulator

No	Data Masukan	Hasil yang Diharapkan	Hasil Pengamatan	Hasil Pengujian
1	Memilih keluar simulator	Simulator ditutup	Simulator ditutup	[√] Diterima [ ] Ditolak
2	Tidak memilih keluar dari simulator	Kembali ke simulator	Kembali ke simulator	[√] Diterima [ ] Ditolak

4.6.3 Hasil Pengujian Perbaikan Kata

Pada pengujian ini digunakan sebanyak 10 abstrak dengan total 82 kalimat. Dokumen-dokumen yang diuji merupakan dokumen yang sudah diperiksa oleh ahli tata bahasa. Pola kalimat yang benar disimpan dalam sebuah dokumen untuk dibandingkan hasilnya dengan sistem. Rekomendasi perbaikan diambil dari pola-pola yang sudah diperbaiki. Pada tahap awal dilakukan pengecekan pola kalimat menggunakan algoritma CYK. Algoritma ini akan mengenali seberapa jauh kalimat yang ada pada dokumen dapat diketahui letak kekeliruannya.

Pada pengujian ini akan diuji kata yang diketahui terdapat kekeliruan. Tujuan dari pengujian ini untuk melihat berapa banyak kata yang rekomendasinya tepat. Berdasarkan skenario yang dibuat maka hasil pengujian perbaikan kata dapat dilihat pada tabel 4.10.

Tabel 4.10 Pengujian perbaikan kata

Kata	Perbaikan Kata	Perbaikan Kata yang Seharusnya	Kesimpulan
kecnderungan	kecenderungan	kecenderungan	Perbaikan dapat diterima
dikarenakan	digunakan diharapkan	dikarenakan	Perbaikan tidak terdapat dalam kamus
kepda	kepada	kepada	Perbaikan dapat diterima
metde	metode	metode	Perbaikan dapat diterima
waancara	wawancara	wawancara	Perbaikan dapat diterima
cntoh	contoh	contoh	Perbaikan dapat diterima
diatas	atas data batas	di atas	Perbaikan tidak dapat diterima
berbais	berbasis	berbasis	Perbaikan dapat diterima
luyout	layout	layout	Perbaikan dapat diterima
scrift	script	script	Perbaikan dapat diterima
memerikan	memberikan	memberikan	Perbaikan dapat diterima
diakibatkan	mengakibatkan didapatkan	diakibatkan	Perbaikan tidak terdapat dalam kamus
saaat	saat	saat	Perbaikan dapat diterima
terlambat	terhambat	terlambat	Perbaikan tidak terdapat dalam kamus
analisi	analisis	analsis	Perbaikan dapat diterima
memerikan	memberikan	memberikan	Perbaikan dapat diterima
pengerajan	pengerjaan pengenalan penerapan	pengerjaan	Terdapat perbaikan yang dapat diterima

berakhir	akhir	berakhir	Perbaikan tidak terdapat dalam kamus
keatas	atas kelas batas kata	ke atas	Perbaikan tidak dapat diterima
serfis	servis	servis	Perbaikan dapat diterima
penilaiaian	penilaian	penilaian	Perbaikan dapat diterima
popular	populer	populer	Perbaikan dapat diterima
metoda	metode	metode	Perbaikan dapat diterima
merpakan	merupakan	merupakan	Perbaikan dapat diterima
sitem	sistem	sistem	Perbaikan dapat diterima
karateristik	karakteristik	karakteristik	Perbaikan dapat diterima
barng	barang	barang	Perbaikan dapat diterima
menutp	menutup	menutup	Perbaikan dapat diterima
cahasya	cahaya	cahaya	Perbaikan dapat diterima
demi	di dari segi	demi	Perbaikan dapat diterima
plaigat	plagiat	plagiat	Perbaikan dapat diterima
samua	semua sama	semua	Terdapat perbaikan yang dapat diterima
gambr	gambar	gambar	Perbaikan dapat diterima
memprioriatskan	memprioritaskan	memprioritaskan	Perbaikan dapat diterima
maslah	salah masalah	masalah	Terdapat perbaikan yang dapat diterima

memerbaiki	memperbaiki	memperbaiki	Perbaikan dapat diterima
hubngan	hubungan	hubungan	Perbaikan dapat diterima

Dari hasil pengujian terdapat 37 kata yang mendapat perbaikan. Terdapat 5 kata yang tidak dapat diperbaiki. Tiga kata diantaranya tidak terdapat dalam kamus sehingga hasilnya tidak optimal dan dua diantaranya merupakan kata yang seharusnya dipisah. Jumlah kata yang mendapat perbaikan dengan benar ada 32. Nilai keberhasilan dari pengujian perbaikan kata dapat diukur dengan cara:

$$\frac{jumlah\ perbaikan\ yang\ sesuai}{jumlah\ kata\ yang\ keliru} x\ 100$$

$$\frac{32}{37} x100 = \ 86{,}4\%$$

Hasil pengujian didapatkan nilai sebesar 86,4% untuk pengujian algoritma Levenshtein dalam memperbaiki kata.

4.6.4 Hasil Pengujian Perbaikan Pola Kalimat

Pengujian ini bertujuan untuk melihat berapa banyak pola kalimat yang dapat diberikan rekomendasi. Tabel 4.11 menunjukkan berapa banyak jumlah kalimat yang diuji.

Tabel 4.11 Jumlah Pengujian Kalimat pada Dokumen

No	Nama Dokumen	Jumlah Kalimat	Jumlah Kalimat yang dianggap keliru
1.	Abstrak01	12	7
2.	Abstrak02	10	7
3.	Abstrak03	6	2
4.	Abstrak04	11	8
5.	Abstrak05	6	5
6.	Abstrak06	11	8
7.	Abstrak07	4	4
8.	Abstrak08	7	4
9.	Abstrak09	8	4
10.	Abstrak10	7	3
Total		82	52

Selanjutnya pengujian dilakukan dengan pembandingan pola kalimat yang diperbaiki oleh sistem dengan pola kalimat yang diperbaiki oleh ahli tata bahasa. Tahap ini juga merupakan pengujian sejauh mana algoritma Levenshtein ini dapat menemukan pola kalimat yang benar untuk perbaikan. Nilai *edit distance* yang digunakan adalah nilai yang terkecil dari perbandingan pola kalimat. Pengujian pola kalimat yang diperbaiki dapat dilihat pada lampiran B.

Tabel 4.12 Pengujian Perbaikan Pola Kalimat

No	Nama Dokumen	Jumlah Kalimat Keliru	Jumlah Kalimat yang Sesuai Perbaikannya
1.	Abstrak01	7	6
2.	Abstrak02	7	6
3.	Abstrak03	2	1
4.	Abstrak04	8	8
5.	Abstrak05	5	5
6.	Abstrak06	8	8
7.	Abstrak07	4	4
8.	Abstrak08	4	3
9.	Abstrak09	4	4
10.	Abstrak10	3	3
Total		52	48

Pada tabel 4.12 dapat dilihat bahwa jumlah kalimat yang sesuai perbaikannya dengan pola kalimat yang diperiksa oleh ahli tata bahasa berjumlah 33 kalimat dari 52 kalimat yang keliru.

Dari hasil pengujian pada tabel 4.12 dapat diukur nilai keberhasilannya menggunakan persentase dengan cara:

$$\frac{jumlah\ perbaikan\ yang\ sesuai}{jumlah\ kalimat\ keliru} x\ 100$$

$$\frac{48}{52} x100 =\ 92{,}3\%$$

Hasil pengujian didapatkan nilai sebesar 92,3% untuk pengujian algoritma Levenshtein dalam mencocokan pola kalimat yang digunakan untuk rekomendasi perbaikan.

4.7 Kesimpulan Pengujian

Dari hasil pengujian dapat disimpulkan bahwa algoritma CYK dapat mengenali pola kalimat dengan baik sesuai dengan aturan produksi yang dibentuk. Perbaikan kata yang dilakukan oleh algoritma Levenshtein sudah cukup baik.

Algoritma ini dapat memberikan rekomendasi berdasarkan nilai *edit distance* yang terendah namun algoritma ini tidak dapat melihat kata mana yang berdempet dan tidak berdempet. Hasil rekomendasi dapat dilakukan terhadap 32 dari 37 kata. Algoritma ini juga cukup baik dalam memberikan rekomendasi untuk perbaikan pola kalimat namun algoritma ini hanya sekedar melakukan perbandingan saja. Algoritma Levenshtein tidak dapat mengenali bagaimana makna dari sebuah kalimat. Hasil rekomendasi yang mendapat perbaikan sesuai dengan pola kalimat yang dimaksud ada 48 dari 52.