# BAB III

# ANALISIS DAN PERANCANGAN

## 3.1 Metodologi Penelitian

Alur kerja penelitian ini dilakukan berdasarkan langkah-langkah berikut.

1. Mempelajari kajian teori yang dibutuhkan

   Merupakan proses mempelajari kajian teori-teori yang dibutuhkan dalam membuat aplikasi *grammar checker* dan cara-cara membangun aplikasi *grammar checker.*

2. Menetapkan atribut-atribut yang diperlukan oleh program *grammar checker*

   Merupakan proses penetapan atribut-atribut yang dibuthkan oleh sistem sesuai dengan pengetahuan yang didapat dari mempelajari teori-teori tentang pembuatan *grammar checker*

3. Merancang aplikasi *grammar checker*

   Merancang antar muka, tampilan input dan output dari program *grammar checker.*

4. Merancang *database* kata-kata

   Merancang struktur database yang berisi dua tabel yaitu tabel daftar kata-kata baku yang diperlukan oleh program *grammar checker* dan tabel yang berisi daftar kata-kata tidak baku.

5. Membuat aplikasi *grammar checker*

   Membuat *coding* dari aplikasi yang dibuat dengan menggunakan bahasa C# dan Visual Studio untuk melakukan pemrograman

6. Melakukan pengujian

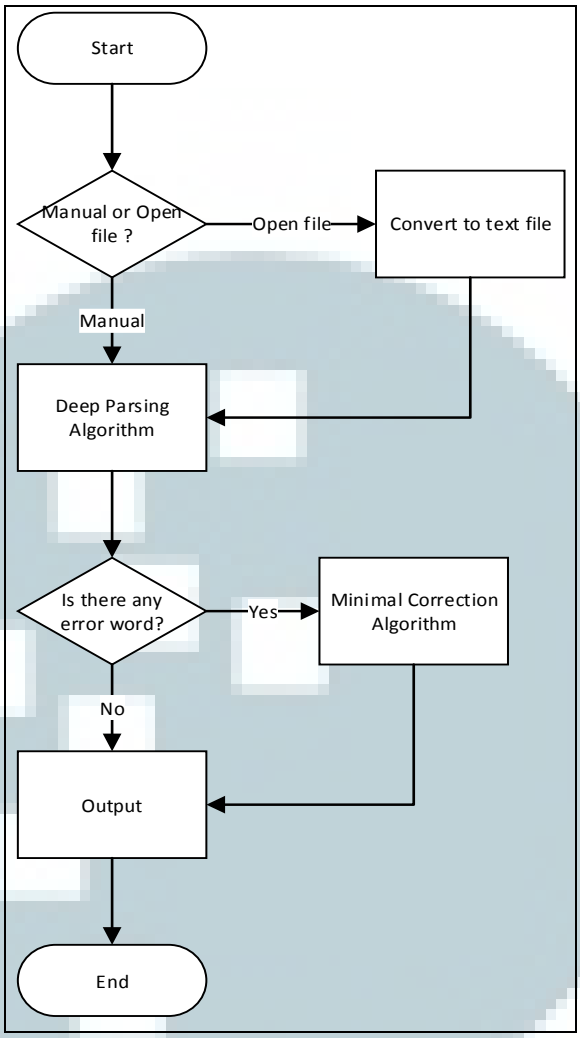
   Melakukan pengujian atas hasil yang dihasilkan oleh aplikasi yang dibuat, dan kuesioner yang akan dibagikan kepada tester untuk menguji hasil dari aplikasi yang dibuat.

7. Analisa

   Melakukan analisis dari data yang didapat selama melakukan pengujian dari aplikasi yang dibuat dan kesimpulan dari data yang didapat berdasarkan hasil uji coba yang dilakukan terhadap program.

### 3.1.1 Diagram Sistem

Prosedur kerja dari sistem dapat dijelaskan dengan menggunakan diagram sebagai berikut. Pada intinya ada dua proses yang harus dilakukan program setiap proses tersebut bersifat *mandatory* atau harus dikerjakan. Penjelasan di atas dijabarkan lebih lanjut oleh Gambar 3.1 berikut.




Gambar 3.1 *Flowchart* Aplikasi *Grammar checker*

### 3.1.2 Fungsionalitas Sistem

Fungsi dari sistem yang dibuat adalah sebagai berikut.

1. Melakukan proses pengubahan dokumen dengan format .docx atau .doc ke dalam format text file jika user memilih untuk memasukan input menggunakan file *word.*
2. Memecah-mecah kalimat menjadi sekelompok kata-kata.

3. Melakukan proses pemeriksaan terhadap kesalahan aturan bahasa menggunakan algorithma deep parsing terhadap setiap kata-kata.
4. Memberitahukan atau menginformasikan jika ditemukan kata-kata yang tidak ditemukan di dalam kamus.
5. Memberikan koreksi atas kesalahan yang dibuat menggunakan algoritma *minimal correction*.

### 3.1.3 Masukan dan Keluaran Sistem

Masukan dari sistem adalah sebagai berikut.

Teks yang ingin dperiksa oleh sistem. Teks yang dapat dimasukkan secara manual melalui *textbox* yang disediakan atau dapat juga berupa *file* dengan format .doc atau docx.

Keluaran dari sistem adalah sebagai berikut.

Teks yang sudah diperiksa dan dibenarkan jika dideteksi terjadi kesalahan grammar dari masukan yang diberikan sebelumnya oleh sistem.

## 3.2 Desain Sistem

Sistem yang dikembangkan akan diimplementasikan sebagai sebuah *desktop application* yang berbasis Windows dan dikembangkan dengan bahasa C# dengan menggunakan Microsoft Visual Studio 2012. Aplikasi ini akan menggunakan dua buah modul yaitu modul *converter* dan modul *checker*.

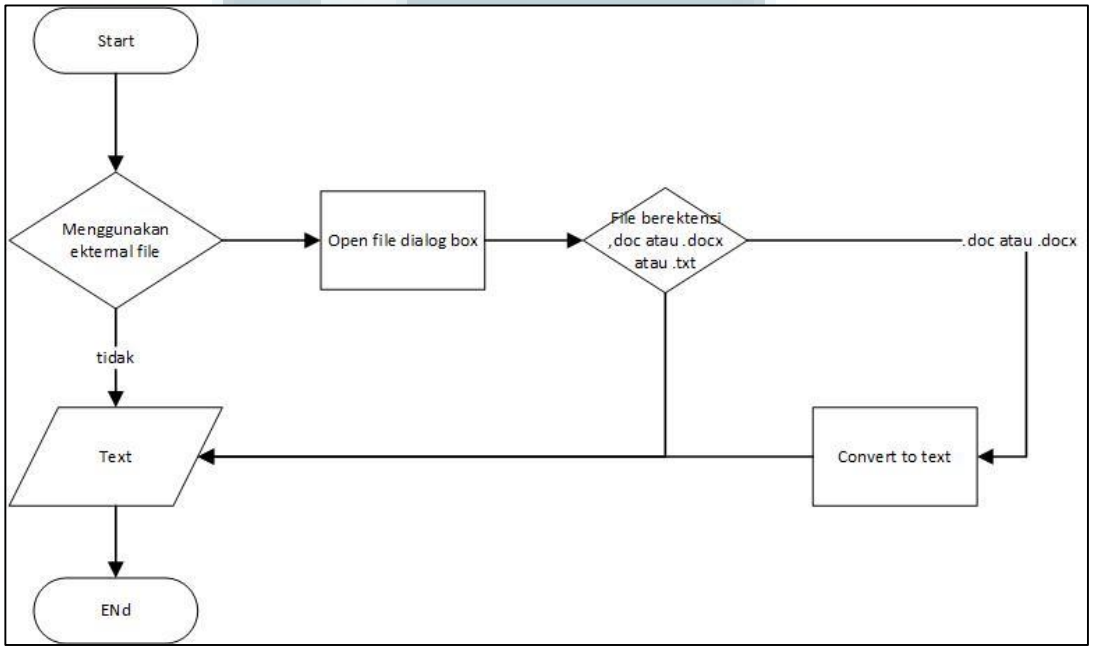
Berikut adalah penjelasan dari modul *grammar checker* yang digambarkan dengan diagram alir, *data flow diagram*, hirarki menu aplikasi, serta tampilan antarmuka aplikasi.

### 3.2.1 Desain Modul dan Subroutine

Modul yang digunakan dalam program ini adalah modul *grammar checker* modul ini terdiri dari beberapa *subroutine*. *Subroutine* yang terdapat dalam modul *grammar checker* adalah sebagai berikut.

*1.* *Subroutine converter*

*Subroutine* ini berfungsi untuk melakukan konversi dari file yang memiliki ektensi .doc atau .docx atau .text menjadi input yang dapat digunakan oleh program. Untuk membuat fungsi ini penulis menggunakan *library* dari code7248 yang dapat diakses di http://sourceforge.net/projects/word-reader/. *Flowchart* dari *subroutine converter* dapat dilihat pada gambar 3.2.



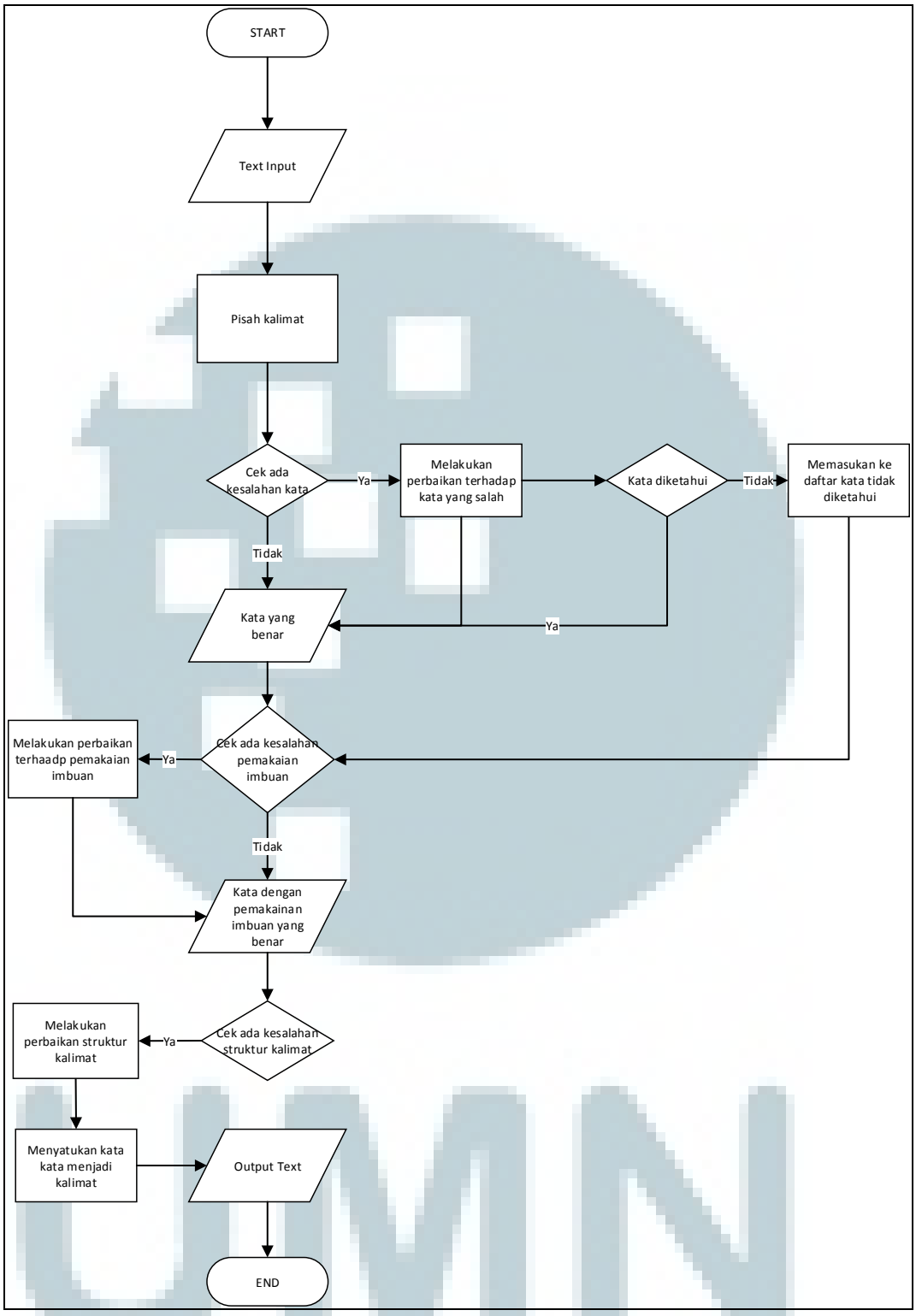

*Gambar 3.2 Flowchart Subroutine Converter*

Modul ini akan digunakan jika pengguna memilih untuk memasukan *input* menggunakan file yang berformat .docx atau .doc. Untuk melakukan hal tersebut maka aplikasi menggunakan *open file dialog* agar pengguna dapat memilih file yang akan digunakan menjadi *input* dari aplikasi ini. Subroutine ini lalu akan merubah file yang berformat .docx atau .doc tersebut ke dalam format text dan menampilkannya di *textbox input* dari aplikasi.

Modul ini hanya dapat merubah file .docx atau .doc yang hanya berisi text saja, modul ini tidak akan bisa merubah file .docx atau .doc yang mempunyai gambar, grafik, tabel di dalamnya.

2. *Subroutine checker*

*Subroutine* ini berfungsi untuk melakukan pengecekan terhadap kesalahan *grammar* terhadap masukan *text* yang dimasukkan oleh pengguna dan juga berfungsi untuk melakukan perbaikan atas kesalahan yang dilakukan oleh pengguna dan juga memberitahu pengguna jika ada kata-kata yang tidak dapat ditemukan di dalam database kamus yang dimiliki oleh aplikasi sehingga pengguna dapat menambah database dari aplikasi sehingga jika terjadi kasus yang sama aplikasi akan dapat memperbaikinya.. Flowchart dari subroutine c*hecker* dapat dilihat pada gambar 3.3.




*Gambar 3.3 Flowchart Subroutine Checker*

Proses akan dimulai dengan melakukan pemisahan kalimat yang dimasukan oelh pengguna menjadi kata-kata fungsi ini dilakukan oleh fungsi pemisah kata. Setelah dipisahkan aplikasi akan melakukan pengecekan apakah adanya penggunaan kalimat yang tidak baku.

Kemudian aplikasi akan melakuakan pengecekan terhadap kata-kata tersebut untuk menentukan apakah kalimat tersebut baku atau tidak. Untuk melakukan proses ini aplikasi menggunakan fungsi cek kata akan mencocokan kata yang ada dengan *database* yang dimiliki oleh program yang berisi dua tabel yaitu tabel kata baku dan tabel kata tidak baku. Proses pengecekan yang dilakukan oleh fungsi cek kata terdiri dari dua tahap. Tahap pertama yaitu melakukan pengecekan kata yang dimasukan pengguna terhadap kata yang ada dalam tabel kata baku yang dimiliki oleh aplikasi.Tabel kata baku ini berisi kata-kata baku yang terdapat dalam Kamus Besar Bahasa Indonesia beserta jenis dari kata tersebut. Untuk isi dari tabel kata baku ini penulis mengambil *database* dari Gerry Eka yang dapat diakses di http://www.mediafire.com/download/83pvaajjjzf91ra /tb_katadasar.sql. Jika ada ditemukan kata yang tidak ada dalam *database* kata baku maka fungsi cek kata akan melakukan tahap kedua yaitu melakukan pengecekan ke tabel kata non baku yang dimiliki program. Jika ditemukan kata yang sama maka aplikasi akan melanjutkan ke proses berikutnya yaitu proses cek imbuan. Tahap kedua yang dilakukan oleh fungsi cek kata adalah melakukan pengecekan kata yang dimasukan oleh pengguna terhadap kata yang terdapt di  dalam tabel kata non baku. Tabel kata non baku ini berisi kata-kata non baku beserta kata baku dari kata tersebut. Jika ditemukan di dalam tabel tersebut maka aplikasi akan melakukan perubahan terhadap kata non

baku tersebut menjadi kata baku untuk kata tersebut yang di dapat dari tabel kata non baku tersebut. Jika tidak ditemukan adanya kesmaan terhadap kata yang ada dalam kata non baku maka sistem akan memberitahukan kata yang tidak dikenal tersebut ke pengguna agar pengguna dapat menambahkan kata tersebut ke dalam tabel kata non baku. Aplikasi melakukan pemberitahuan tersebut ke pengguna dengan cara memunculkan kata tersebut ke dalam *dropdown list* kata tidak dikenal sehingga dapat dimasukkan oleh pengguna suatu persamaan dari kata tidak baku tersebut sehingga aplikasi dapat memperbaiki secara otomatis jika terjadi kesalahan yang sama.

Setelah dilakukan pengecekan kata tidak baku maka aplikasi akan melakukan perngecekan pemakaian imbuhan yang dilakukan oleh fungsi cek imbuhan. Fungsi ini akan medeteksi adanya penggunaan imbuan dan melakukan pemeriksaan apakah pemakaian imbuan yang digunakan sudah sesuai aturan atau tidak. Jika ditemukan adanya kesalahan aturan dalam penggunaaan imbuan maka aplikasi akan melakukan proses perbaikan terhadap kata tersebut sehingga kata tersebut menggunakan imbuan yang sesuai aturan. Jika tidak ditemukan adanya kesalahan yang terjadi terhadap aturan pemakaian imbuan maka proses akan langsung dilanjutkan ke proses cek struktur kata.

Proses cek struktur kalimat yang menggunakan fungsi cek struktur yang berfungsi untuk melakukan pemeriksaan terhadap struktur kalimat yang dimasukan oleh pengguna. Jika ditemukan bahwa kalimat yang dimasukan oleh penggna memiliki struktur kalimat yang salah maka aplikasi akan melakukan penyusunan ulang terhadap kalimat tersebut. Kalimat yang memilik kesalahan struktur tersebut

itu akan disusun ulang dengan menggunakan struktur kalimat yang benar tetapi proses penyusunan ulang kalimat tersebut akan hanya menggunakan struktur kalimat yang benar tetapi tidak melakukan banyak perubahan terhadap kalimat tersebut. Jika tidak ditemukan adanya kesalahan maka aplikasi akan mengambungkan kembali kata-kata tersebut.

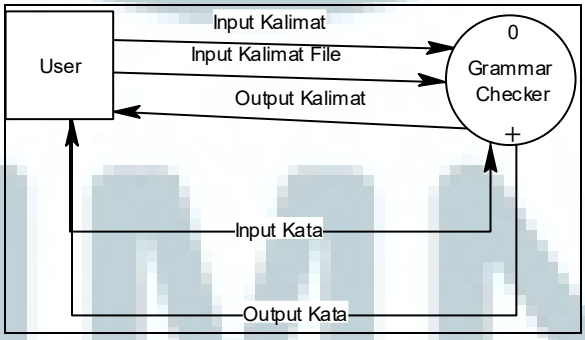
Setelah proses-proses tersebut dilakukan maka aplikasi akan menampikan hasil *output* dari proses yang telah dilakukan ke dalam *textbox output.*

### 3.2.2 Data Flow Diagram

Arus data yang terjadi dalam aplikasi *grammar checker* ini dapat digambarkan melalui *data flow diagram. Data flow diagram* dari aplikasi *grammar checker* ini adalah sebagai berikut.

1. *Data flow diagram level 0*

Berikut adalah *data flow diagram level 0 atau context diagram* dari aplikasi *grammar checker.*

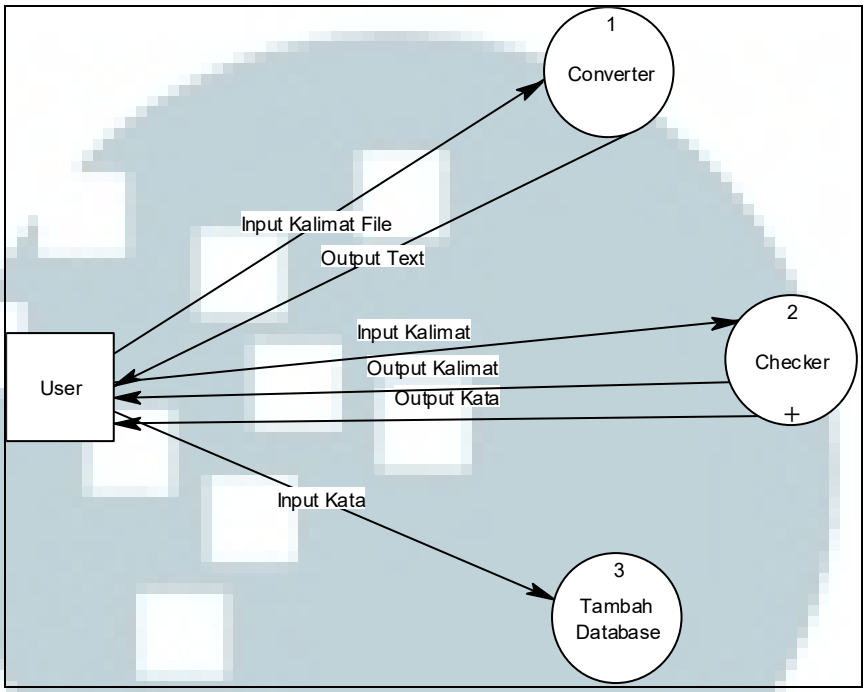



Gambar 3.4 *Data Flow Diagram level 0*

Sistem dari aplikasi *grammar checker* ini memiliki satu entity yaitu *user* yang mempunyai tiga masukan yaitu input kalimat, input kalimat file, dan input kata. Entity *user* juga memiliki 2 keluaran yaitu output kalimat dan output kata.

2. *Data flow diagram level 1*

Berikut adalah *data flow diagram level 1* dari aplikasi *grammar checker.*




Dalam aplikasi *grammar checker* ini terdapat dua subroutine yaitu subroutine converter dan subroutine checker.

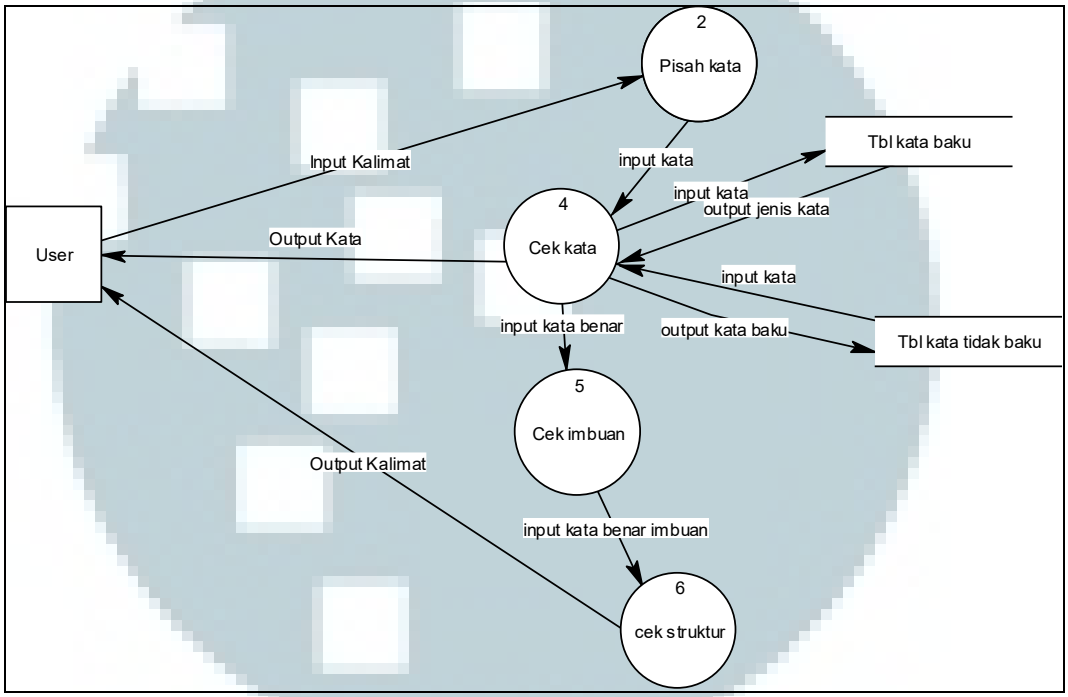
Subroutine *converter* ini akan berfungsi untuk merubah file yang berformat .docx atau .doc yang dimasukan oleh pengguna aplikasi menjadi bentuk text dan menampilkannya ke dalam *textbox input. File* yang dapat dimasukan oleh pengguna aplikasi mempunyai dua syarat yaitu pertama file tersebut berformat .docx atau .doc dan yang kedua adalah file tersebut hanya mengandung text tidak boleh mengandung gambar, grafik ataupun tabel.

Subroutine *checker* berfungsi untuk menjalankan fungsi pemeriksaan kesalahan pada kalimat dengan menggunakan algoritma *deep parsing* untuk

melakukan pemerksaan tersebut dan menggunakan algoritma *minimal correction* untuk melakukan pemilihan perbaikan yang dilakukan oleh aplikasi.

*3. Data flow diagram level 2.1 – Subroutine checker*

Berikut adalah *data flow digram level 2.1* dari aplikasi grammar checker.




Gambar 3.5 *Data Flow Diagram level 2.1 – Subroutine Checker*

Di dalam subroutine checker terdapat beberapa proses yang dilakukan oleh aplikasi *grammar checker* proses-proses, proses-proses itu adalah sebagai berikut.

1. Proses pisah kata

Proses ini akan memisah-misah kalimat yang dimasukan oleh user menjadi kata-kata sehingga dapat diproses oleh proses yang lainnya. Hasil dari proses ini adalah *array* kata-kata dari kalimat yang dimasukan oleh pengguna.

2. Proses cek kata

Proses ini akan melakukan pengecekan terhadap kata-kata yang terdapat dapat di dalam *array* tersebut. Pemeriksaan itu dilakukan dengan melakukan komparasi dengan tabel kata baku dan kata non baku, Dan jika dideteksi terjadi kesalahan maka akan di lakukan perbaikan yang diperlukan. Hasil dari proses ini adalah kata-kata yang baku dan sesuai dengan kamus besar bahasa indonesia.

3. Proses cek imbuan

Proses ini akan melakukan pengecekan kata-kata terhadap aturan imbuan yang ada dalam bahasa Indonesia dan melakukan perbaikan yang diperlukan jika dideteksi terjadinya kesalahan pemakaian aturan imbuan. Hasil dari proses ini adalah kata-kata yang menggunakan imbuan sesuai dengan aturan.

4. Proses cek struktur

Proses ini akan melakukan pengecekan struktur dari kalimat yang dimasukan. Jika dideteksi adanya kesalahan struktur kalimat maka proses ini akan melakukan penyusuan ulang terhadap kalimat tersebut sehingga kalimat tersebut mempunyai struktur kalimat yang benar. Hasil dari proses ini adalah kalimat yang mempunyai struktur kalimat yang benar.

### 3.2.3 Struktur Tabel

Terdapat dua tabel yang digunakan pada aplikasi *grammar checker*. Rincian dari tabel yang digunakan adalah sebagai berikut.

Nama tabel : kata baku

Keterangan : berisi kata-kata baku yang ada pada Bahasa Indonesia

Tabel 3.1 Struktur tabel kata baku

| Nama Kolom | Tipe | Panjang Data | PK | FK | Keterangan |
|---|---|---|---|---|---|
| ID | INT | - | Y | N | ID dari record kata |
| Kata | String | 255 | N | N | kata baku |
| Jenis Kata | String | 255 | N | N | jenis kata baku |

Nama tabel : kata non baku

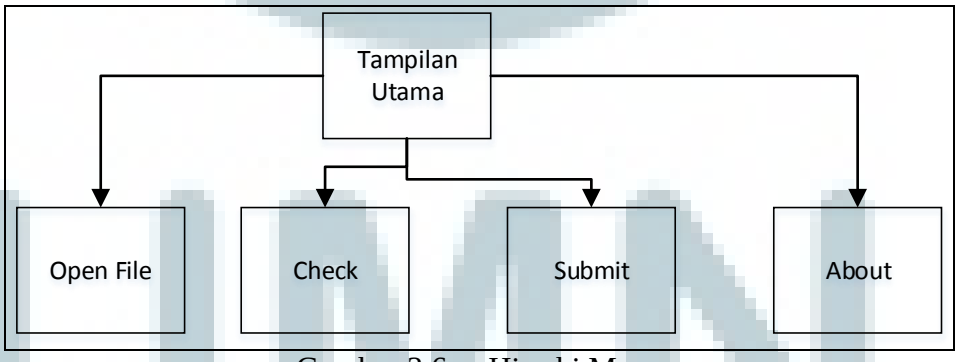
Keterangan : berisi kata-kata non baku dan persamaan kata bakunya

Tabel 3.2 Struktur tabel kata non baku

| Nama Kolom | Tipe | Panjang Data | PK | FK | Keterangan |
|---|---|---|---|---|---|
| ID | INT | - | Y | N | ID dari record kata |
| Kata baku | String | 255 | N | N | kata non baku |
| Kata non baku | String | 255 | N | N | kata baku |

### 3.2.4 Hirarki Menu

Diagram berikut merupakan hirarki dari menu aplikasi *grammar checker* yang dikembangkan.




Gambar 3.6 Hirarki Menu

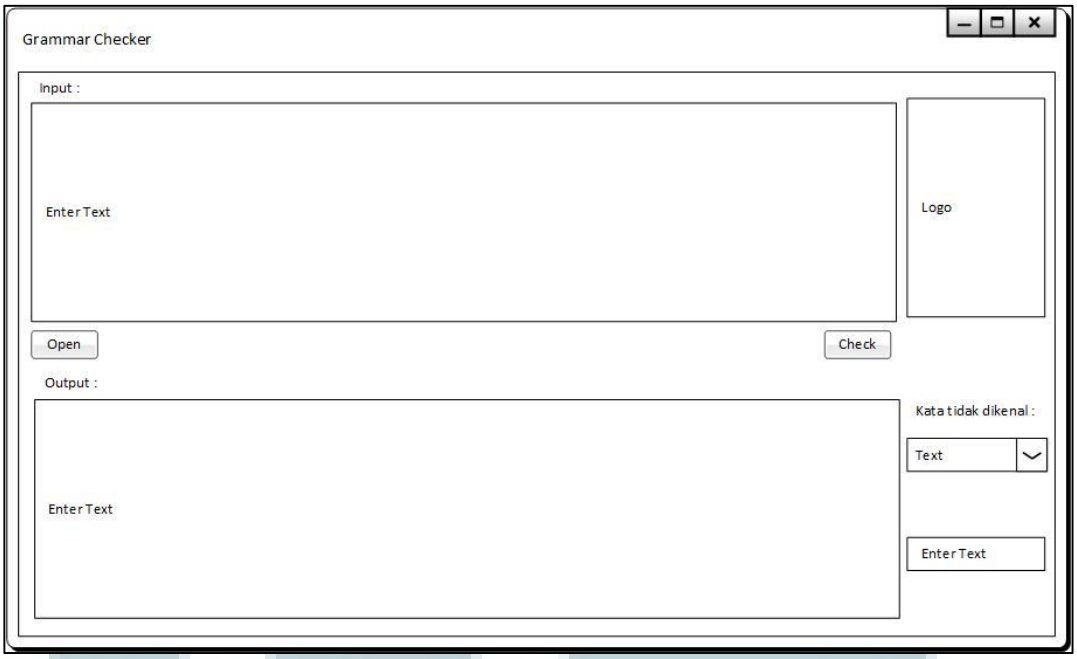
Pada aplikasi tampilan akan memiliki beberapa menu yaitu menu *open file*, menu *check*, menu *submit* dan menu *about*. Menu open file akan membuka *open file dialog* yang berguna agar pengguna aplikasi dapat memilih file dokumen yang

ingin di masukan. Menu *check* akan memulai proses pemeriksaan kesalahan pada masukan yang dimasukan oleh pengguna aplikasi. Menu *submit* digunakan untuk melakukan penambahan data terhadap tabel kata non baku yang terdapat di dalama database yang dimiliki aplikasi . Menu submit hanya dapat digunakan jika aplikasi mendeteksi adanya kata-kata yang tidak dikenal oleh aplikasi dan pengguna telah memberikan *input* kata pengganti untuk kata tersebut. Menu *about* akan membuka jendela baru yang akan memberikan informasi pembuat dari aplikasi dan informasi sumber-sumber lain yang digunakan oleh pengembang aplikasi dalam membuat aplikasi ini.

### 3.2.5 Tampilan Antarmuka

1. Tampilan Utama

Tampian Utama merupakan halaman utama yang berisi tombol-tombol untuk membuka *file* untuk *file* input yaitu *open file* dan tombol *check* untuk melakukan proses pengecekan terhadap masukan yang diberikan dan *textbox* untuk memberikan masukan secara manual. Tombol-tombol tersebut berguna untuk menjalankan aplikasi dan modul-modul lainnya. Desain dari tampilan utama dapat dilihat pada gambar 3.7.




Gambar 3.7 Desain Tampilan Utama

Untuk memasukkan input pengguna dapat menggunakan dua cara yaitu melakukan *manual input* ke dalam *textbox input* atau pengguna dapat menggunakan file berformat .docx atau .doc, dengan syarat isi dari file tersebut hanya text saja tidak mengandung gambar, grafik, atau tabel. Untuk melakukan hal tersebut pengguna aplikasi dapat menggunakan tombol open yang jika ditekan akan membuka *open file dialog* sehingga pengguna dapat memilih *file* mana yang ingin dijadikan masukan untuk aplikasi.

Untuk melakukan proses pengecekan *grammar* pengguna dapat menekan tombol *check* yang akan memulai proses pemeriksan terhadap masukan yang telah dimasukan oleh pengguna. Hasil dari proses pemeriksaan tersebut kemudian akan ditampilkan pada *textbox output.*

Jika ada kata tidak baku yang tidak dikenal oleh aplikasi maka pada *dropdown list* akan muncul kata yang tidak dikenal tersebut yang berguna untuk memasukan

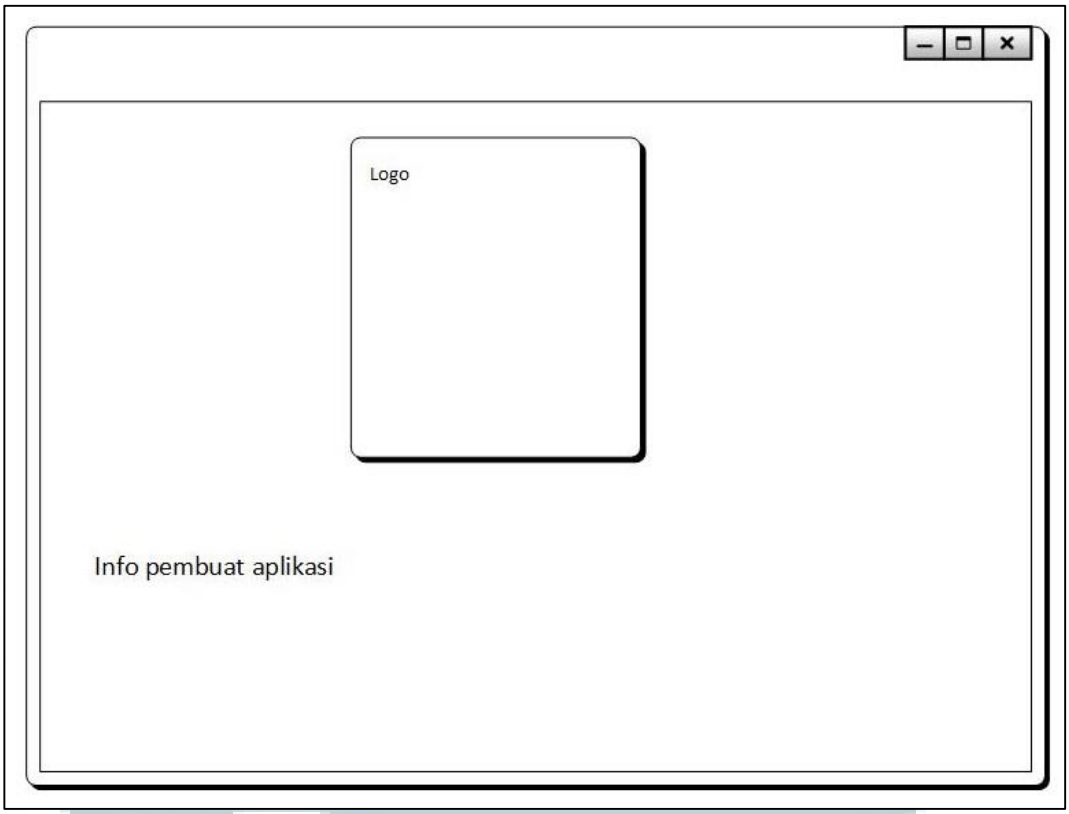
kata baku dari kata yang tidak baku tersebut sehingga jika terjadi kasus yang sama aplikasi akan dapat langsung melakukan perbaikan. Untuk memasukan kata tersebut pengguna harus memilih kata yang terdapat dalam *dropdownlist* dan menuliskan persamaan kata baku dari kata yang dia pilih di *dropdownlist* tersebut lalu menekan tombol *submit*, yang akan membuat aplikasi untuk menyimpan kata tersebut ke dalam tabel kata non baku yang terdapat di dalam database yang dimiliki aplikasi.

Untuk memunculkan informasi tentang pembuat dari aplikasi pengguna dapat mengklik logo aplikasi yang terdapat di samping *textbox input*, yang jika ditekan oleh pengguna akan menampilkan jendela baru berisi informasi peembuat dari aplikasi dan sumber-sumber lain yang digunakan pembuat aplikasi untuk membuat aplikasi tersebut.

Untuk keluar dari aplikasi pengguna dapat menekan tombol x yang ada di seblah kanan atas dari program.

2. Tampilan *About*

Tampilan *about* merupakan halaman yang berisi gambar logo dari aplikasi dan informasi tentang pembuat aplikasi dan sumber-sumber lain yang digunakan untuk membuat aplikasi. Desain dari tampilan about dapat dilihat pada gambar 3.8.




Gambar 3.8 Desain Tampilan *About*