Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

Tahap-tahap yang dilalui dalam penelitian ini dapat dibagi dalam beberapa bagian yaitu sebagai berikut.

1) Studi Literatur

   Proses ini merupakan awal dari keseluruhan proses penelitian. Pada tahap ini dilakukan pengumpulan materi yang berhubungan dengan sistem temu kembali informasi, metode evaluasi Paice, dan algoritma-algoritma *stemming* yang digunakan dalam penelitian ini.

2) Perancangan Sistem

   Dalam tahap perancangan, dibuat *flow chart* aplikasi, *data flow diagram* (DFD), struktur tabel yang akan ditambahkan ke dalam basis data, dan juga desain antarmuka aplikasi.

3) Implementasi

   Dalam tahap ini, algoritma yang diuji diimplementasikan ke dalam aplikasi menggunakan bahasa pemrograman C# dengan IDE Visual Studio 2010.

4) Pengujian Data

   Tahap pengujian adalah tahap dimana aplikasi yang telah dibuat digunakan untuk menguji performa ketiga algoritma yang digunakan dalam *stemming* dokumen berbahasa Indonesia.

Perbandingan Performa ..., Hadi Hartono, FTI UMN, 2015

5) Penulisan Laporan

Tahap selanjutnya adalah penulisan laporan sebagai dokumentasi atas penelitian yang telah dilakukan, hasil pengujian dan analisa data, serta kesimpulan dan saran untuk penelitian selanjutnya.

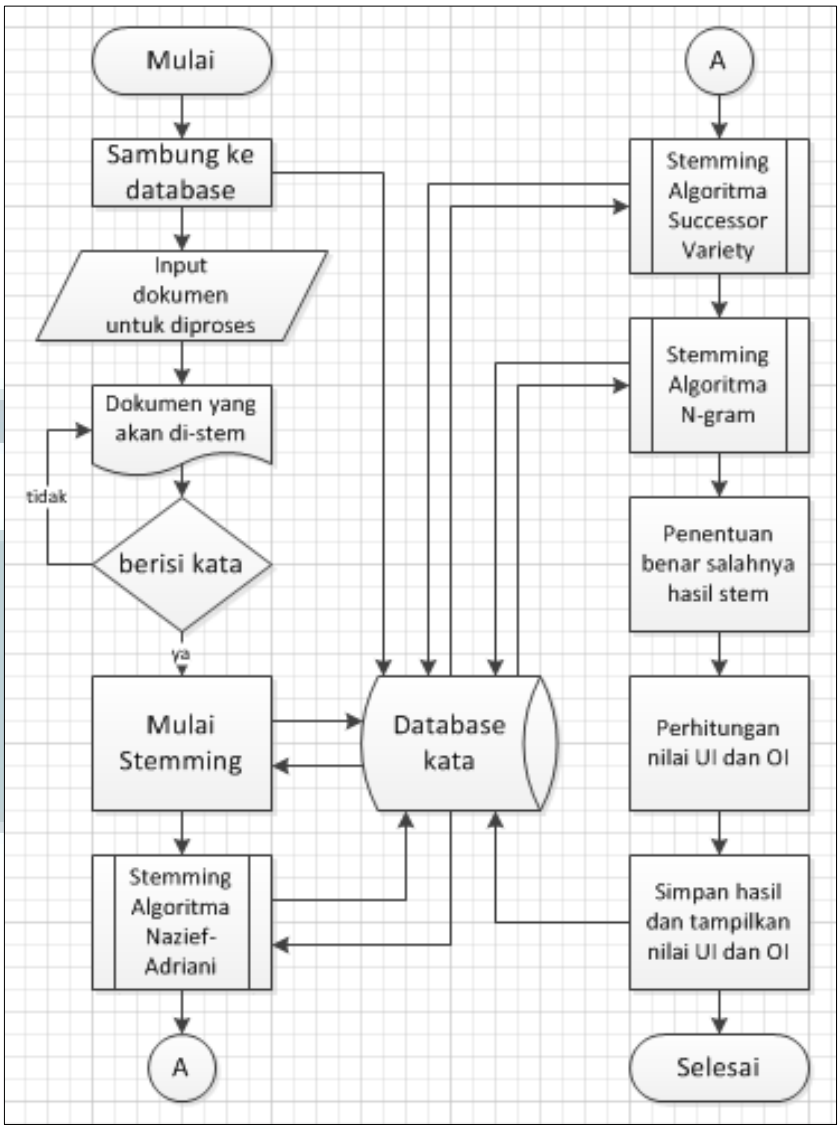
3.2 Perancangan Sistem

Rancangan sistem merupakan proses menyusun atau mengembangkan sistem informasi untuk mendapatkan hasil yang dipenuhi (Reza, 2013). Ada beberapa hal yang dilakukan dalam perancangan sistem, yaitu perancangan *flow chart*, *data flow diagram* (DFD), struktur tabel basis data, dan desain antarmuka aplikasi. Berikut masing-masing penjelasannya.

3.2.1 Flow Chart

Flow chart adalah representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri atas sekumpulan simbol, dimana masing-masing simbol merepresentasikan suatu kegiatan tertentu. *Flow chart* diawali dengan penerimaan *input*, pemrosesan *input*, dan diakhiri dengan *output* (Reza, 2013). Gambar 3.1 menunjukkan *flow chart* yang telah dibuat.

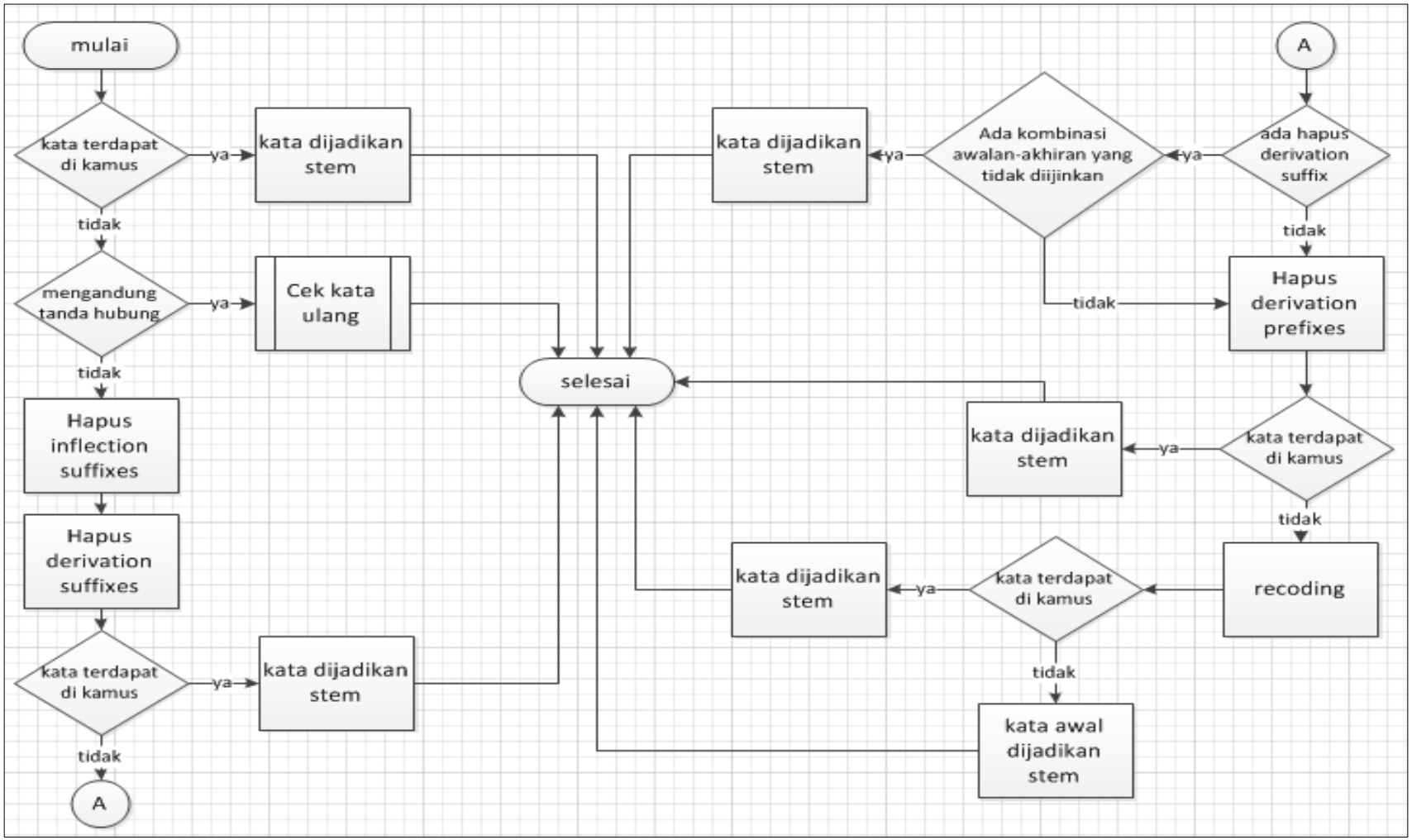
Perbandingan Performa ..., Hadi Hartono, FTI UMN, 2015



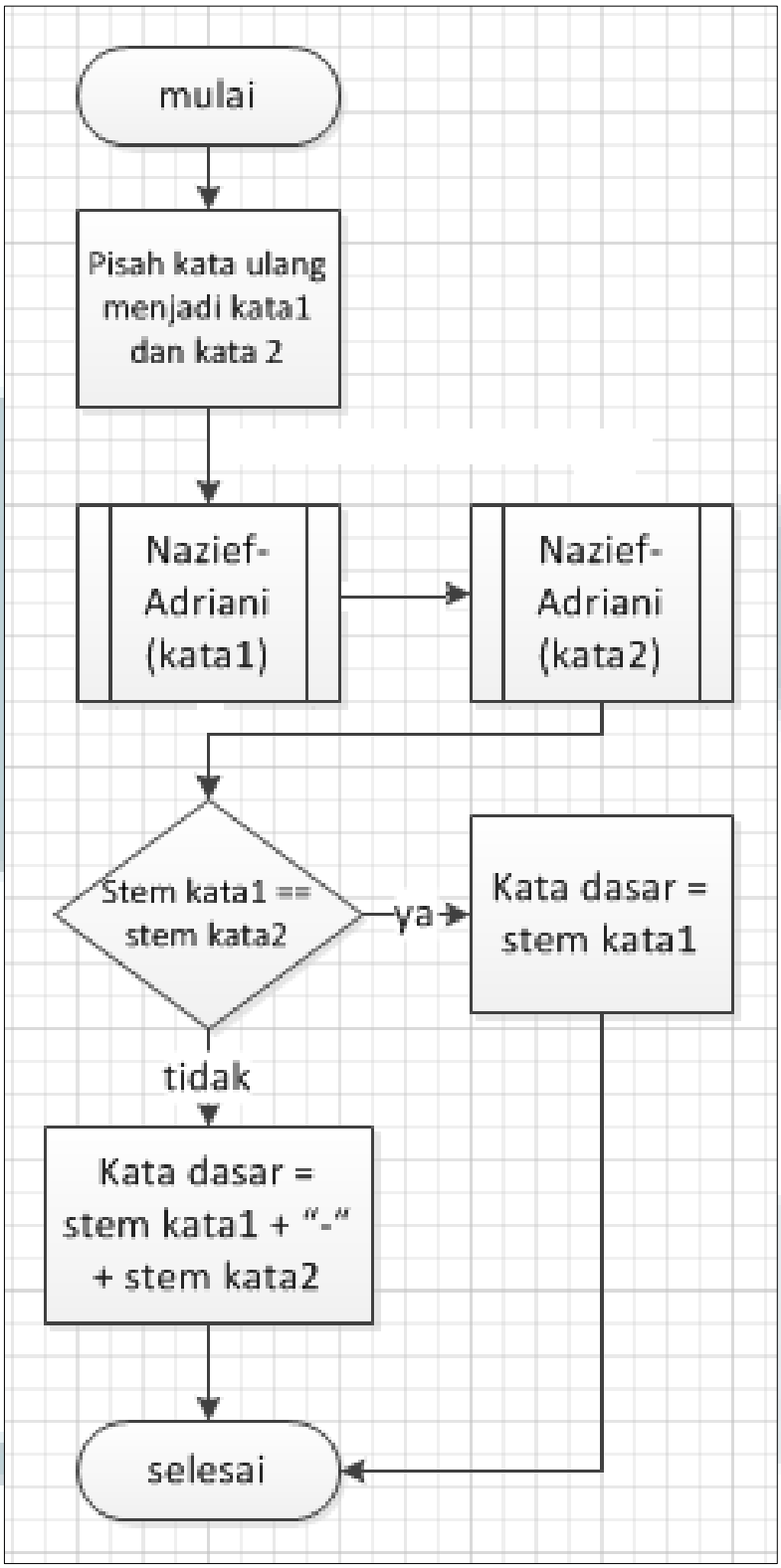
Gambar 3.1 *Flow chart* aplikasi

Setelah aplikasi tersambung pada basis data, pengguna akan diminta untuk memilih dokumen yang akan di-*stem*. Proses berikutnya adalah proses *stemming* oleh masing-masing algoritma terhadap dokumen yang telah ditentukan oleh pengguna.

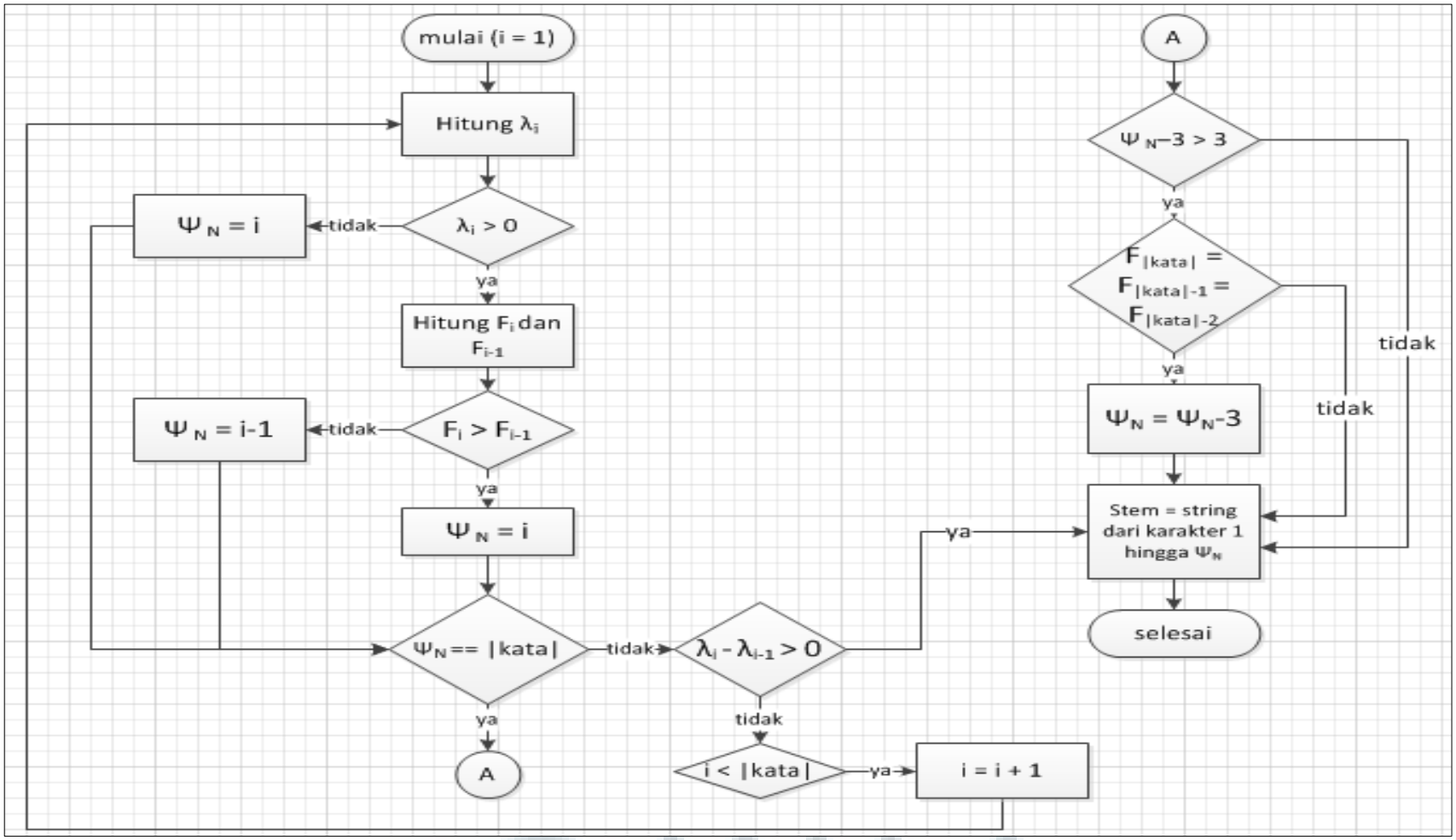
Gambar 3.2, .3.4, dan 3.5 secara berurutan menunjukkan gambar *flow chart* algoritma Nazief-Adriani, N-gram termodifikasi, dan *Successor Variety*.



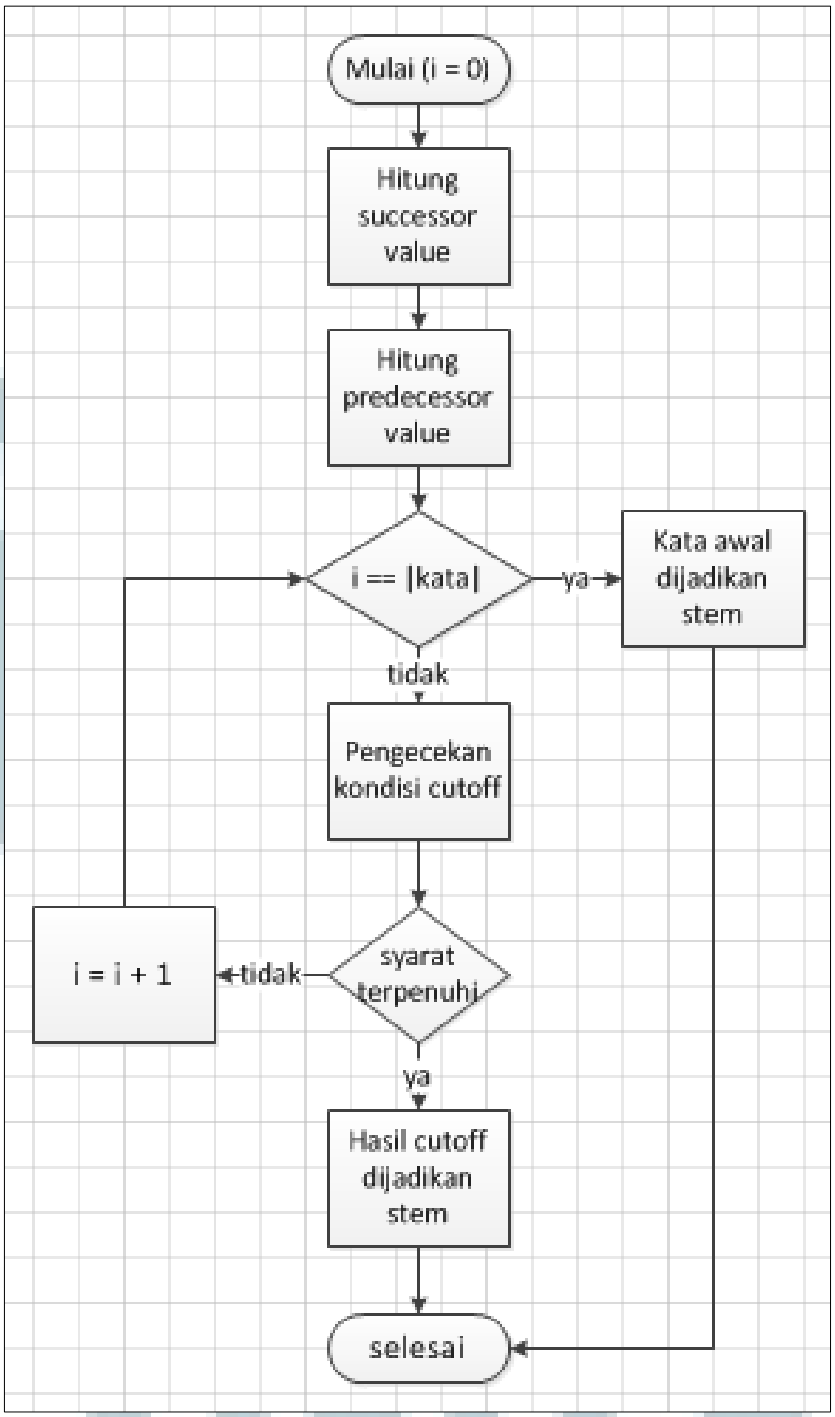
Gambar 3.2 *Flow chart* algoritma Nazief-Adriani



Gambar 3.3 *Flow chart* subproses cek kata ulang algoritma Nazief-Adriani



Gambar 3.4 *Flow chart* algoritma N-gram termodifikasi



Gambar 3.5 *Flow chart* algoritma *Successor Variety* metode *Cutoff*

Setelah proses *stemming*, pengguna akan diminta untuk mengelompokkan kata-kata dalam dokumen berdasarkan kata dasar dari kata-kata tersebut.

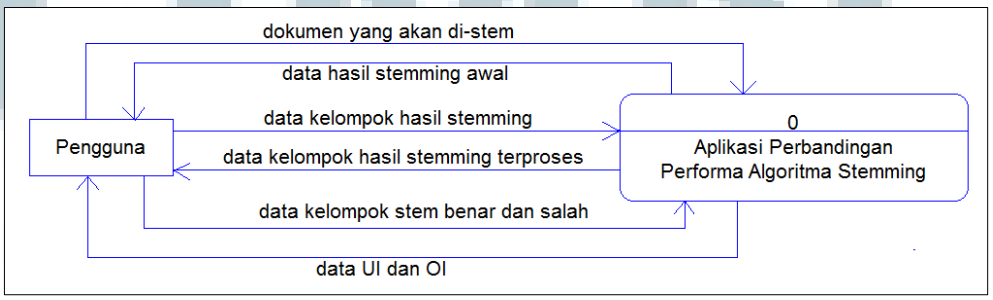
Jika pengguna telah selesai menentukan kelompok-kelompok kata, pada proses berikutnya pengguna akan diminta untuk menentukan kata manakah yang telah di-*stem* dengan benar dan kata manakah yang telah di-*stem* dengan salah.

Berdasarkan kelompok-kelompok yang telah ditentukan sebelumnya, dan juga benar-salahnya *stem* suatu kata, aplikasi akan menghitung nilai *understemming index* (UI) dan *overstemming index* (OI) masing-masing algoritma, menyimpan hasil *stemming*, dan menampilkan nilai UI dan OI pada pengguna.

3.2.2 Data Flow Diagram

Sebagaimana dengan yang telah dijelaskan pada penjelasan *flow chart*, maka aliran-aliran data yang terjadi pada aplikasi dapat digambarkan menjadi sebuah *data flow diagram* (DFD). Gambar 3.6 menggambarkan DFD level 0 atau *context diagram*.

Proses utama pada DFD tersebut adalah proses aplikasi perbandingan performa algoritma *stemming*. Pada DFD level 1, proses utama ini dibagi menjadi 3 subproses, yaitu subproses *stemming*, subproses penentuan kelompok kata, dan subproses penentuan *stem* benar dan salah.



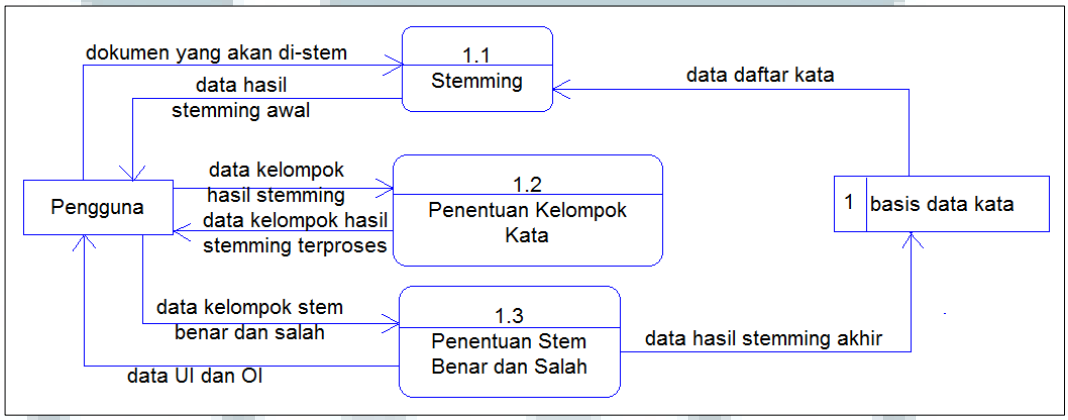
Gambar 3.6 DFD Level 0 atau *Context Diagram* Aplikasi Perbandingan Performa Algoritma *Stemming*

Diagram konteks di atas memiliki satu entitas sebagai berikut.

Tabel 3.1 Entitas Aplikasi Perbandingan Performa Algoritma *Stemming*

Entitas	**Keterangan**
Pengguna	Entitas ini akan mengirimkan *input* berupa: 1) data dokumen yang akan di-*stem* 2) data kelompok hasil *stemming* 3) data kelompok *stem* benar dan salah Data *output* yang akan diterima entitas ini berupa: 1) data hasil *stemming* awal 2) data kelompok hasil *stemming* terproses 3) data UI dan OI

Berikut Gambar 3.7 adalah gambar DFD level 1 yang menggambarkan dekomposisi proses aplikasi perbandingan performa algoritma *stemming*.



Gambar 3.7 DFD Level 1 Aplikasi Perbandingan Performa Algoritma *Stemming*

Proses-proses yang terdapat pada DFD level 1 aplikasi perbandigan performa algoritma *stemming* sebagai berikut.

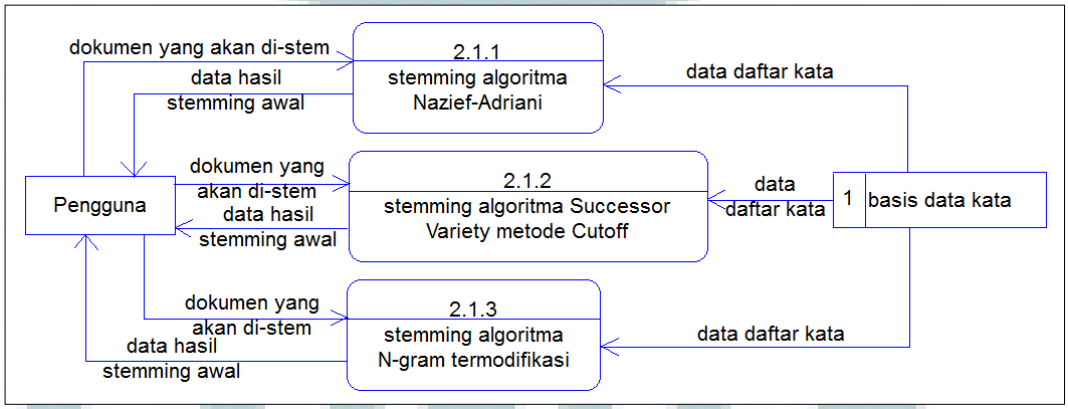
Tabel 3.2 Aliran Data DFD Level 1 Aplikasi Perbandingan Performa Algoritma *Stemming*

N ama Proses	**Input**	**Output**	**Keterangan**
Stemming	• data dokumen yang akan di-	data hasil *stemming* awal	Proses mengambil data daftar kata dari basis

Tabel 3.2 Aliran Data DFD Level 1 Aplikasi Perbandingan Performa Algoritma *Stemming* (lanjutan)

N ama Proses	Input	Output	Keterangan
Stemming (lanjutan)	*stem* • data daftar kata		data kemudian melakukan *stemming* terhadap dokumen yang telah dipilih pengguna.
Penentuan Kelompok Kata	data kelompok hasil *stemming*	data kelompok hasil *stemming* terproses	Pengguna menentukan kelompok kata berdasarkan kata awal (kata sebelum di-stem), proses mengelompokkan kata berdasarkan hasil *stemming*.
Penentuan *Stem* Benar dan Salah	data kelompok *stem* benar dan salah	• data UI dan OI • data hasil *stemming* akhir	Proses menyimpan hasil *stemming* akhir ke basis data, dan menampilkan nilai UI dan OI masing-masing algoritma.

Berikut Gambar 3.8 adalah gambar DFD level 2 yang menggambarkan dekomposisi proses *stemming* yang terdapat pada DFD level 1.



Gambar 3.8 DFD Level 2 Proses *Stemming*

Proses-proses yang terdapat pada DFD level 2 proses *stemming* sebagai berikut.

Tabel 3.3 Aliran Data DFD Level 2 Proses *Stemming*

Nama Proses	Input	Output	Keterangan
stemming algoritma Nazief-Adriani	• data dokumen yang akan di-*stem* • data daftar kata	data hasil *stemming* awal	Proses melakukan *stemming* menggunakan algoritma Nazief-Adriani
stemming algoritma *Successor Variety* metode *Cutoff*	• data dokumen yang akan di-*stem* • data daftar kata	data hasil *stemming* awal	Proses melakukan *stemming* menggunakan algoritma *Successor Variety* metode *Cutoff*
stemming algoritma N-gram termodifikasi	• data dokumen yang akan di-*stem* • data daftar kata	data hasil *stemming* awal	Proses melakukan *stemming* menggunakan algoritma N-gram termodifikasi

3.2.3 Struktur Tabel Basis Data

Sebuah daftar kata dasar diperlukan dalam proses *stemming*. Oleh karena itu, diunduh daftar kata-kata dasar bahasa Indonesia dari kamus Bahtera, yang dapat diunduh dari situs bahtera.org. Daftar kata-kata tersebut dimuat dalam sebuah tabel dengan struktur sebagai berikut.

Nama tabel : tb_katadasar
Fungsi : menyimpan kata dasar bahasa Indonesia beserta tipenya
Jumlah *record* : 28256

Tabel 3.4 Struktur tabel tb_katadasar awal

Field Name	Type	Length	Information
id_katadasar	integer	10	*Primary key, auto increment*
katadasar	varchar	70	
tipe_katadasar	varchar	25	Nomina, adjektiva, verba, adverbia, preposisi, interjeksi, konjungsi, pronomina, numeralia, lain-lain

Algoritma *Successor Variety* yang digunakan memerlukan daftar kata yang dibalik (*reversed word*) untuk pencarian nilai *predecessor value*. Oleh karena itu, tabel tb_katadasar tersebut ditambahkan satu *field* baru yaitu rev_katadasar. Berikut adalah struktur tabel tb_katadasar setelah ditambahkan *field* baru.

Tabel 3.5 Struktur tabel tb_katadasar akhir

Field Name	**Type**	**Length**	**Information**
id_katadasar	integer	10	*Primary key, auto increment*
katadasar	varchar	70	
rev_katadasar	varchar	70	katadasar yang telah dibalik
tipe_katadasar	varchar	25	Nomina, adjektiva, verba, adverbia, preposisi, interjeksi, konjungsi, pronomina, numeralia, lain-lain

Hasil *stemming* yang telah didapatkan setelah proses aplikasi perbandingan performa algoritma *stemming* selesai disimpan dalam basis data yang dibagi dalam lima tabel sebagai berikut.

Nama tabel : tb_dokumen
Fungsi : - menyimpan nama dokumen yang telah di-*stem*

Tabel 3.6 Struktur tabel tb_dokumen

Field Name	**Type**	**Length**	**Information**
id_dokumen	integer	11	*Primary key, auto increment*
nama_dokumen	varchar	50	nama dokumen yang telah di-*stem*

Nama tabel : tb_hasil
Fungsi : - menyimpan kata asal dan *stem* dari kata tersebut
- menyimpan inisial algoritma yang digunakan
- menyimpan jumlah kemunculan kata dalam dokumen

Tabel 3.7 Struktur tabel tb_hasil

Field Name	**Type**	**Length**	**Information**
id_hasil	integer	11	*Primary key, auto increment*
id_dokumen	integer	11	*Foreign key* pada ‘tb_dokumen’.’id_dokumen’
kata_asal	varchar	50	Kata sebelum di-*stem*
hasil_stem	varchar	50	Kata setelah di-*stem*
algoritma	varchar	2	Algoritma yang digunakan dalam *stemming* (na/sv/ng)
frekuensi	integer	11	Frekuensi kata dalam dokumen

Nama tabel : tb_kelompok
Fungsi : - menyimpan nama kelompok
- dasar pengelompokkan yang digunakan (berdasarkan kata asal atau *stem*)
- menyimpan inisial algoritma yang digunakan
- menyimpan jumlah kata yang di-*stem* dengan benar
- menyimpan jumlah kata dalam kelompok

Tabel 3.8 Struktur tabel tb_kelompok

Field Name	**Type**	**Length**	**Information**
id_kelompok	integer	11	*Primary key, auto increment*
id_dokumen	integer	11	*Foreign key* pada ‘tb_dokumen’.’id_dokumen’
dasar_pengelompokan	varchar	2	Penanda kelompok dibentuk berdasarkan kata asal (ka) atau hasil *stem* (hs)
nama_kelompok	varchar	15	
algoritma	varchar	2	Algoritma yang digunakan
nilai_ng	double		Jumlah kata yang di-*stem* dengan benar dalam kelompok
jumlah_kata	integer	11	Jumlah kata dalam kelompok

Nama tabel : tb_isi_kelompok
Fungsi : - menyimpan kata-kata yang terdapat dalam kelompok yang ditentukan saat proses penentuan kelompok.

Tabel 3.9 Struktur tabel tb_isi_kelompok

Field Name	**Type**	**Length**	**Information**
id_isi	integer	11	*Primary key, auto increment*
id_kelompok	integer	11	*Foreign key* pada ‘tb_kelompok’.’id_kelompok’
id_kata	integer	11	*Foreign key* pada ‘tb_hasil’.’id_hasil’

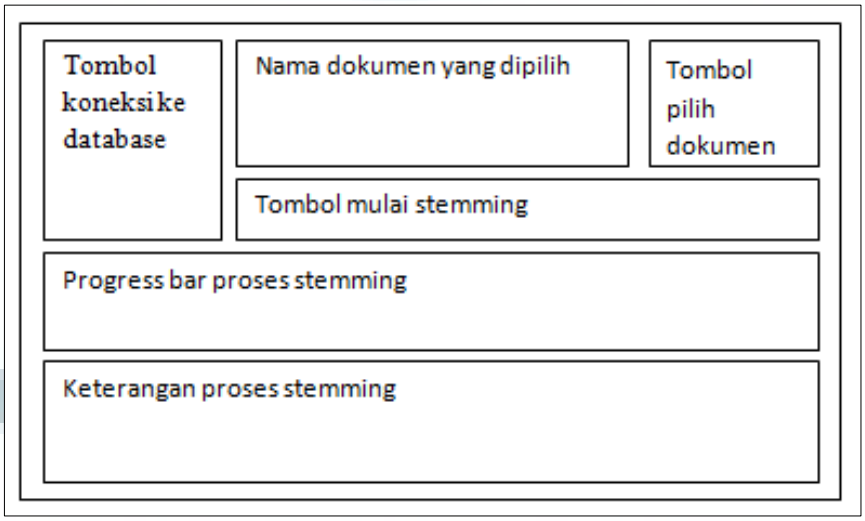
Nama tabel : tb_nilai
Fungsi : - menyimpan nilai-nilai metode evaluasi Paice masing-masing algoritma

Tabel 3.10 Struktur tabel tb_nilai

Field Name	**Type**	**Length**	**Information**
id_nilai	integer	11	*Primary key, auto increment*
id_dokumen	integer	11	*Foreign key* pada 'tb_dokumen'.'id_dokumen'
algoritma	varchar	2	
nilai_UI	double		*Understemming Index*
nilai_OI	double		*Overstemming Index*
nilai_GDMT	double		*Gross Desired Merge Total*
nilai_GWMT	double		*Gross Wrongly-Merge Total*
nilai_GDNT	double		*Gross Desired Non-merged Total*
nilai_GUMT	double		*Gross Unachieved Merge Total*

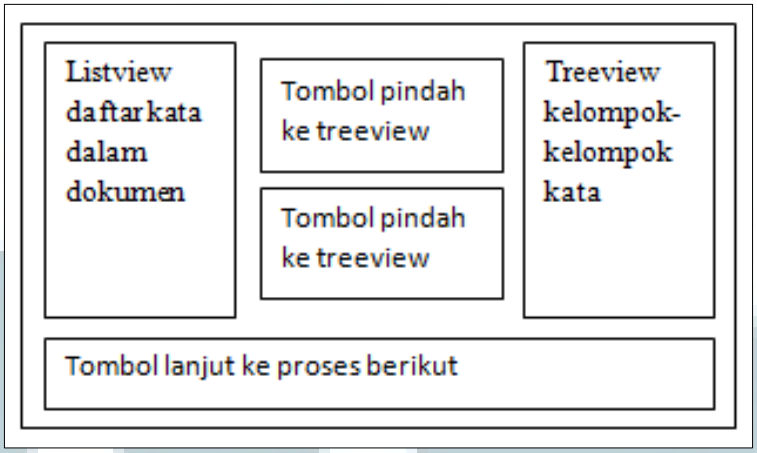
3.2.4 Desain Antarmuka Aplikasi

Berikut Gambar 3.9 adalah desain antarmuka untuk *form* pemilihan dokumen.



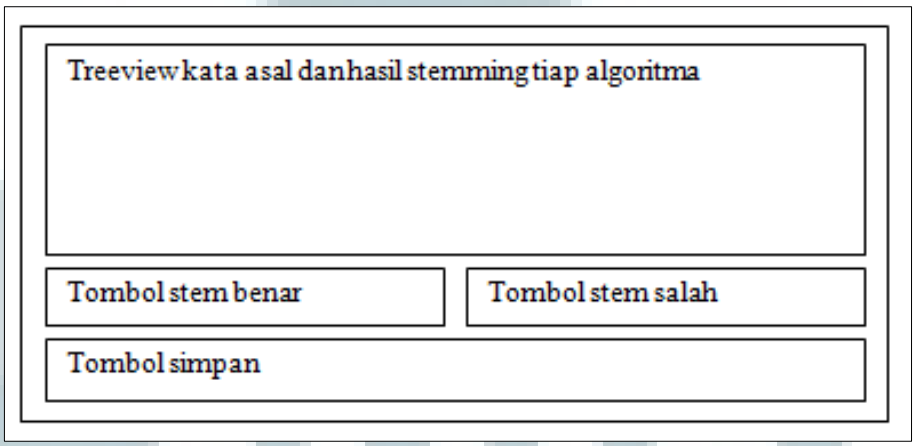
Gambar 3.9 Desain antarmuka *form* pemilihan dokumen

Gambar 3.10 menggambarkan desain antarmuka tampilan saat pengguna diminta untuk menentukan kelompok kata.



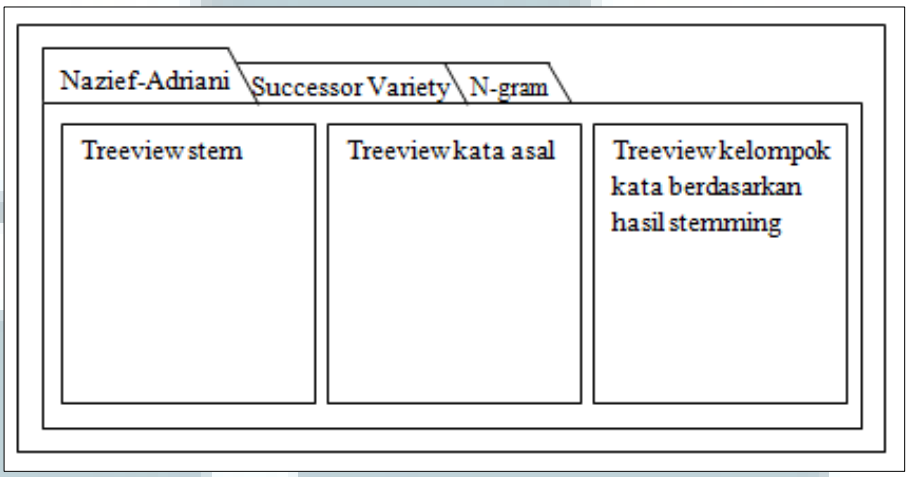
Gambar 3.10 Desain antarmuka *form* penentuan kelompok kata

Desain antarmuka berikutnya adalah antarmuka untuk *form* penentuan benar-salahnya *stem* yang dihasilkan dari proses *stemming*. Gambar 3.11 merupakan desain untuk *form* tersebut.



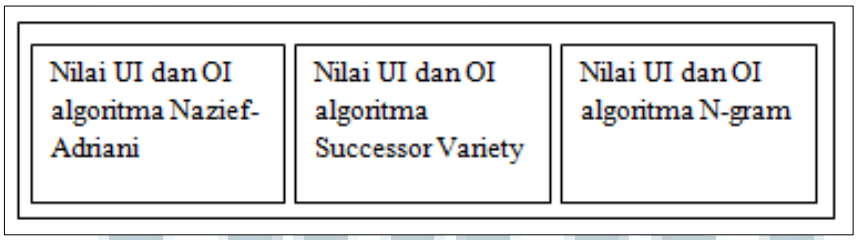
Gambar 3.11 Desain antarmuka *form* penentuan benar-salahnya *stem*

Dalam proses perhitungan nilai UI dan OI, kata-kata dikelompokkan berdasarkan kata dasar dan hasil *stemming*. Gambar 3.12 menggambarkan desain antarmuka untuk *form* kelompok kata berdasarkan hasil *stemming*.



Gambar 3.12 Desain antarmuka *form* kelompok kata berdasarkan hasil *stemming*

Berikut Gambar 3.13 adalah desain antarmuka untuk *form* hasil perbandingan, yang menampilkan nilai UI dan OI masing-masing algoritma kepada pengguna.



Gambar 3.13 Desain antarmuka *form* nilai UI dan OI

3.3 Pengembangan Sistem

Pengembangan sistem meliputi *tools* apa saja yang digunakan dan fitur-fitur yang dibangun untuk memenuhi tujuan penelitian. Berikut masing-masing penjelasannya.

3.3.1 Alat dan Bahan

Dalam rancang bangun aplikasi, dibutuhkan perangkat pendukung pembuatan aplikasi yaitu *hardware* dan *software*. Berikut adalah konfigurasi perangkat yang digunakan selama proses pembuatan aplikasi.

1) **Hardware**

   a) *Processor* : Intel Core i5, 2.4 GHz

   b) Memori : 4GB RAM

   c) *Harddisk* : 500 GB

   d) Koneksi ke internet melalui WiFi

2) **Software**

   a) Sistem operasi : Microsoft Windows 7, 64-bit

   b) *Web server* : Apache 2.4.4

   c) *Database server* : MySQL 5.6.11

   d) *Browser* : Google Chrome 35.0.1916.153 m

   e) XAMPP 3.2.1 yang terdiri atas *web server* dan *database server*

   f) Bahasa pemrograman C#

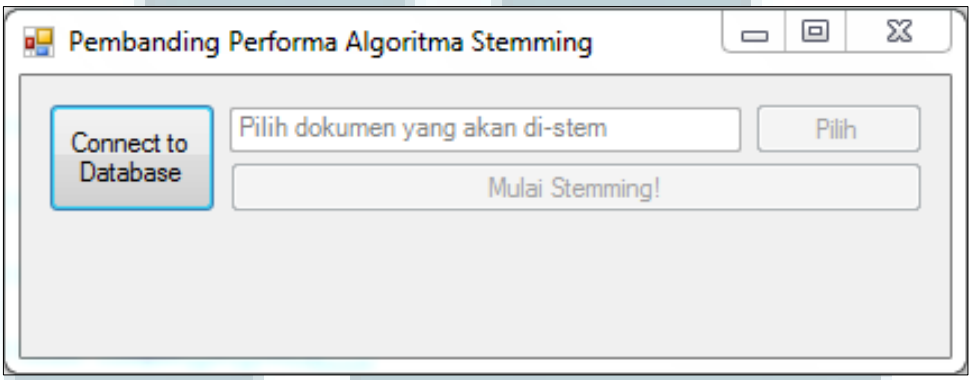
   g) Visual Studio 2010, sebagai IDE penulisan skrip C#

   h) Microsoft Visio 2010, sebagai *tool* untuk membuat *flow chart*

   i) PowerDesigner 6, sebagai *tool* untuk membuat DFD

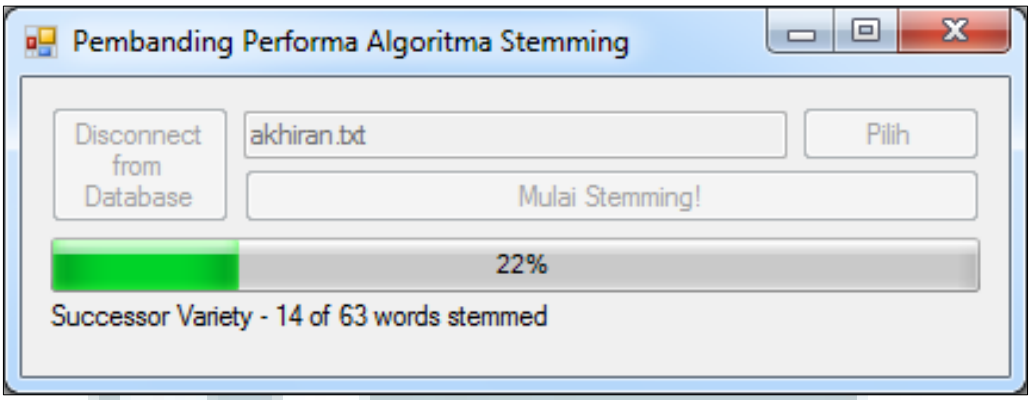
3.3.2 Implementasi

Aplikasi diawali dengan penampilan *form* pemilihan dokumen seperti yang ditunjukkan pada Gambar 3.14.



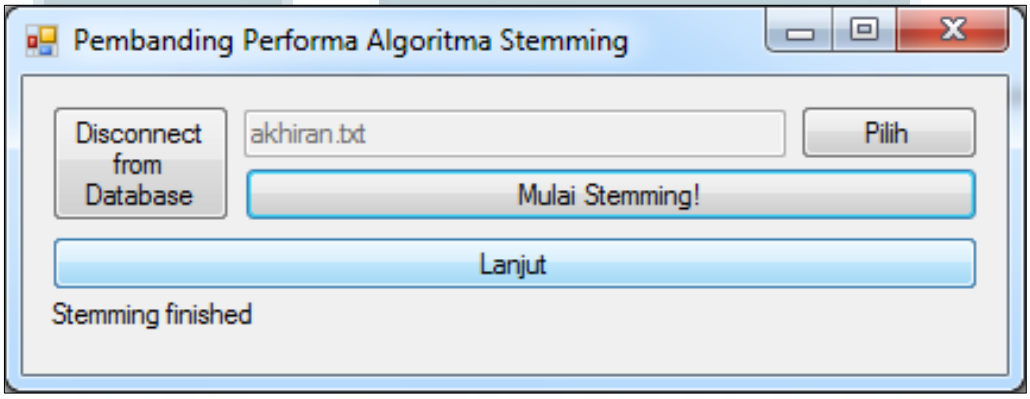
Gambar 3.14 Tampilan awal *form* pemilihan dokumen

Untuk dapat memilih sebuah dokumen untuk di-*stem*, aplikasi harus terhubung dengan basis data terlebih dahulu yang dapat dilakukan dengan menekan tombol "*Connect to Database*" dengan syarat koneksi tersebut memang ada. Setelah terhubung dengan basis data, memilih dokumen yang akan di-*stem*, pengguna dapat memulai proses *stemming* dengan menekan tombol "Mulai *Stemming*!". Gambar 3.15 menggambarkan proses *stemming* yang sedang berlangsung.



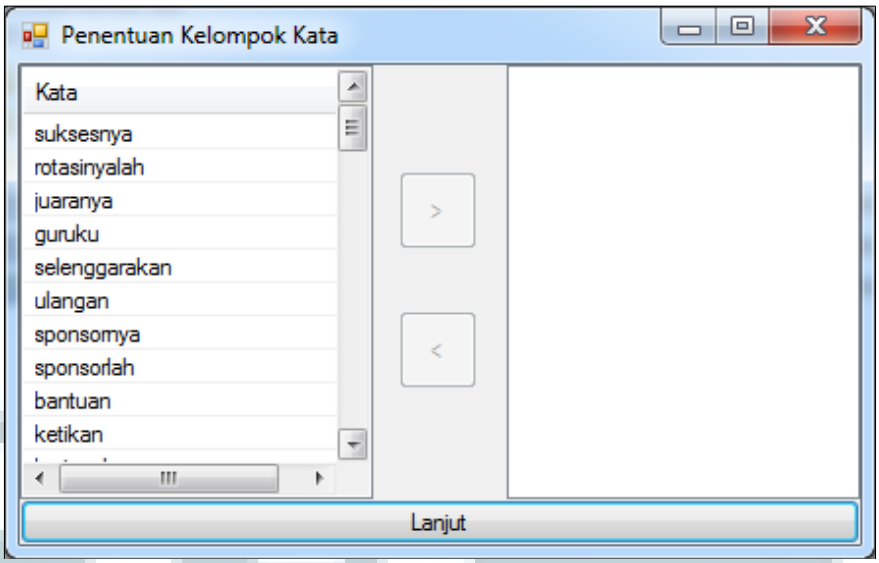
Gambar 3.15 Tampilan ketika proses *stemming* sedang berlangung

Berikut Gambar 3.16 adalah tampilan ketika proses *stemming* telah selesai.



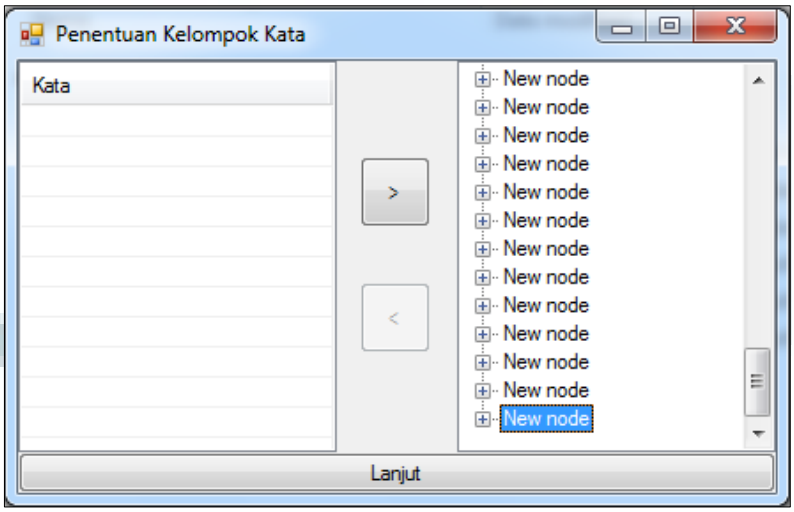
Gambar 3.16 Tampilan ketika proses *stemming* selesai

Tombol "Lanjut" digunakan untuk berpindah ke proses berikutnya, yaitu proses penentuan kelompok kata. Tampilan *form* penentuan kelompok kata ditunjukkan pada Gambar 3.17.



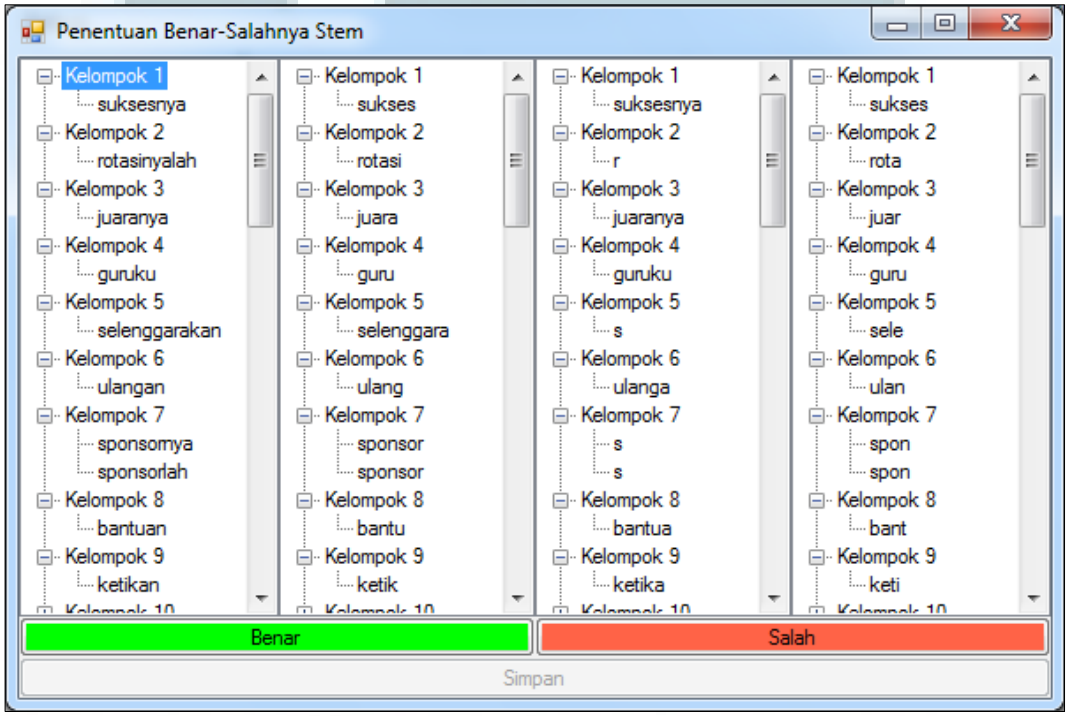
Gambar 3.17 Tampilan awal *form* penentuan kelompok kata

Untuk membuat kelompok kata baru pada *TreeView* (kolom pada sebelah kanan), klik kanan pada kolom tersebut, kemudian akan muncul pilihan "Tambah Kelompok". Tombol ">" digunakan untuk memindahkan *item* pada *ListView* (kolom pada sebelah kiri) ke *TreeView*. Sebaliknya, tombol "<" digunakan untuk memindahkan kembali *item* pada *TreeView* menuju *ListView*.



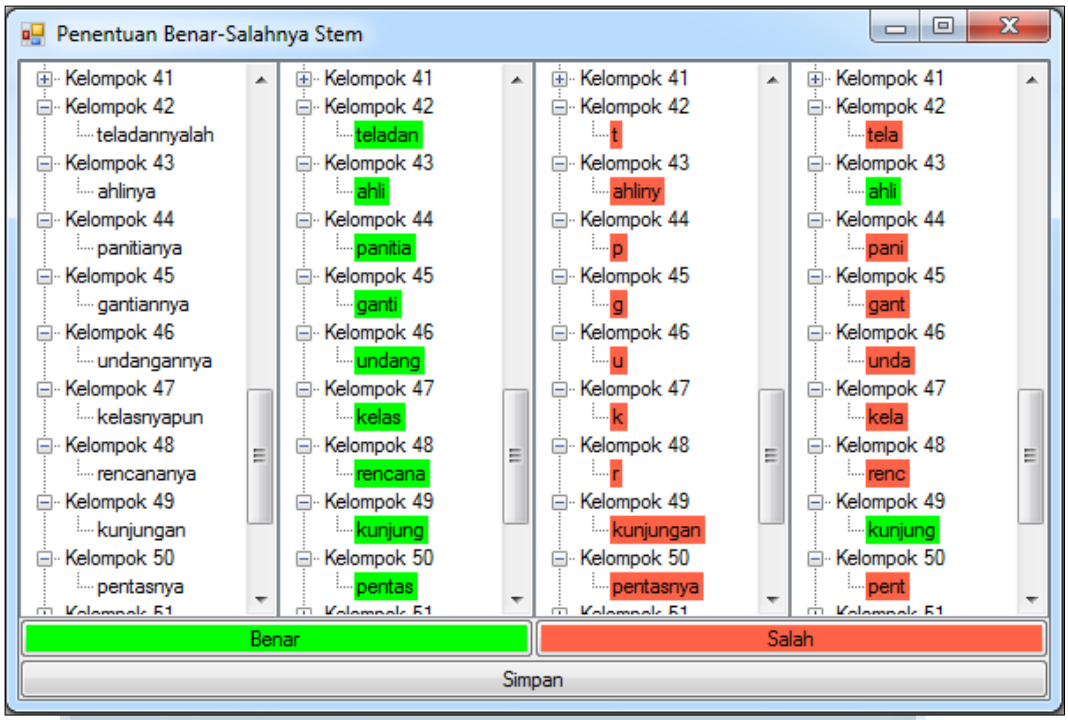
Gambar 3.18 Tampilan ketika pengguna selesai mengelompokkan kata

Gambar 3.19 menunjukkan tampilan awal *form* penentuan benar-salahnya *stem*. Untuk menunjukkan bahwa sebuah kata telah di-*stem* dengan benar, pilih kata tersebut dan tekan tombol “Benar” yang tersedia pada *form*. Sedangkan untuk menunjukkan bahwa sebuah kata telah di-*stem* dengan salah, pilih kata tersebut dan tekan tombol “Salah”.



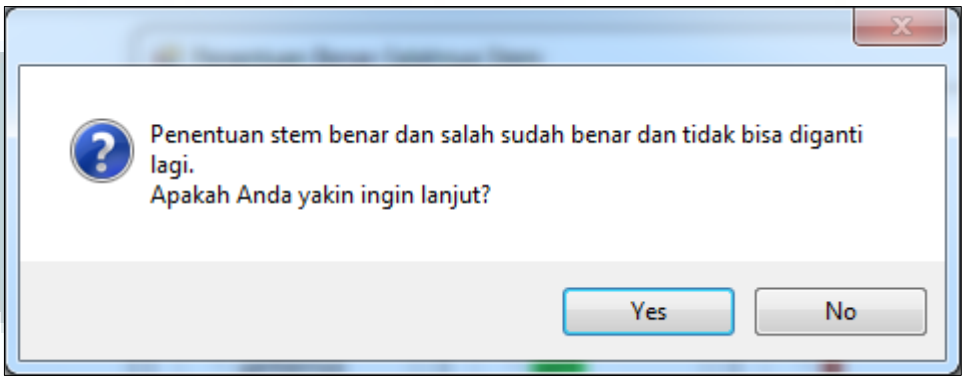
Gambar 3.19 Tampilan awal *form* penentuan benar-salahnya *stem*

Gambar 3.20 menunjukkan tampilan *form* ketika pengguna telah selesai menentukan *stem* yang benar dan salah pada ketiga algoritma.



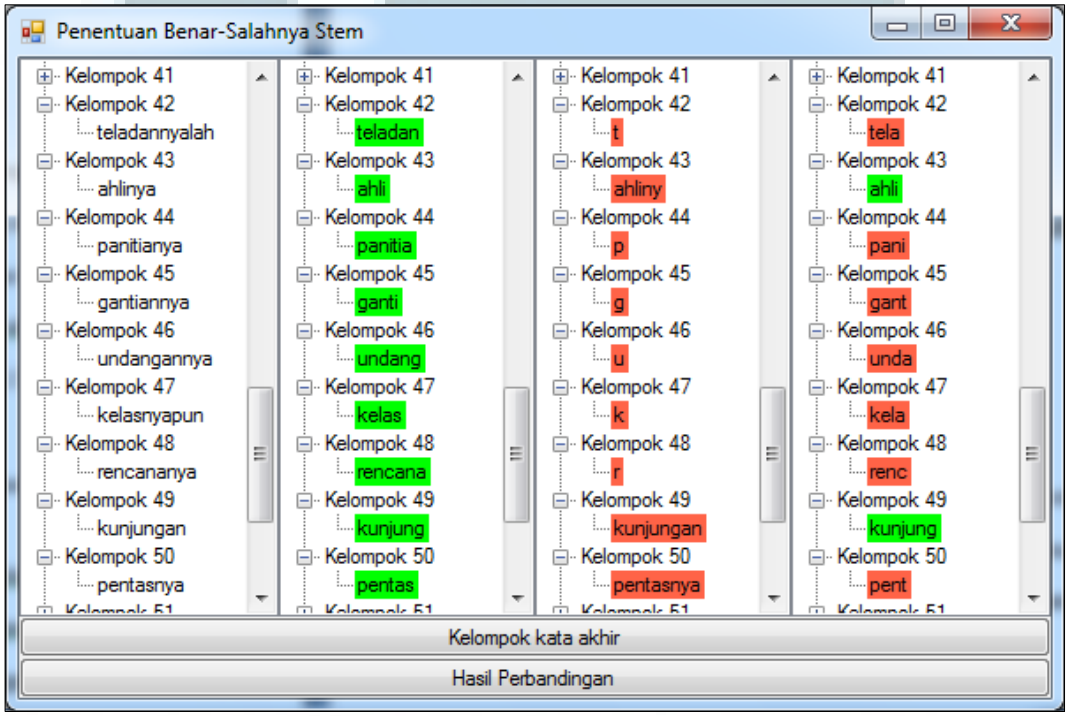
Gambar 3.20 Tampilan ketika selesai menentukan benar-salahnya *stem*

Jika pengguna menekan tombol "Simpan" maka akan muncul sebuah pertanyaan yang menanyakan pada pengguna apakah telah menentukan *stem* yang benar dan salah dengan benar, seperti yang ditunjukkan pada Gambar 3.21.



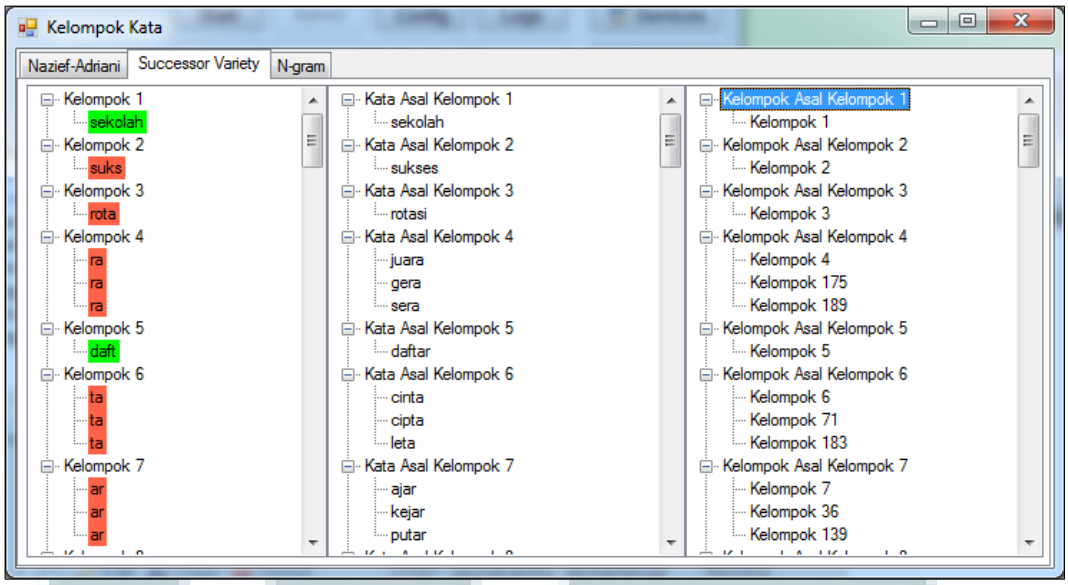
Gambar 3.21 Dialog pemastian data telah diisi dengan benar

Dengan menjawab "Iya", aplikasi akan mulai menghitung nilai *understemming index* (UI) dan *overstemming index* (OI) dan menyimpan daftar kata yang telah diproses, kelompok-kelompok kata, nilai UI dan OI ke dalam basis data. Kemudian *form* penentuan benar-salahnya *stem* akan berubah tampilannya menjadi seperti yang ditunjukkan pada Gambar 3.22.



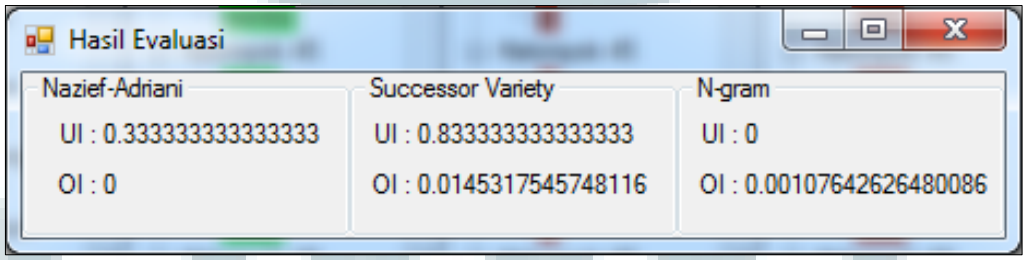
Gambar 3.22 Tampilan setelah aplikasi selesai menyimpan data

Pengguna dapat memilih untuk melihat kelompok kata berdasarkan hasil *stemming* ataupun melihat nilai UI dan OI dari hasil perhitungan yang telah dilakukan. Gambar 3.23 menggambarkan tampilan jika pengguna memilih "Kelompok kata akhir".



Gambar 3.23 Tampilan *form* kelompok kata berdasarkan hasil *stemming*

Gambar 3.24 adalah tampilan *form* yang menampilkan nilai *understemming index* dan *overstemming index* masing-masing algoritma terhadap dokumen yang telah di-*stem*.



Gambar 3.24 Tampilan *form* hasil evaluasi