



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.

BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3.1. Metodologi Penelitian

Dalam melakukan penelitian ini, terdapat tahap-tahap penelitian yang digunakan sebagai berikut.

1. Telaah Literatur

Tahap pertama dalam penelitian ini adalah melakukan telaah literatur mengenai *n-gram model* dan algoritma *sentence generation* dengan referensi dari buku, jurnal, artikel, dan sumber-sumber lainnya.

2. Analisis Kebutuhan

Selanjutnya dilakukan analisis terhadap API, *library*, atau hal-hal lain yang sepertinya dibutuhkan selama proses pemrograman dan *debug*.

3. Desain Sistem

Add-in dirancang dengan memanfaatkan konsep n-gram model untuk memecah informasi berupa kumpulan kalimat, atau disebut corpus, menjadi pasangan kata yang disebut bigram. Setelah seluruh kalimat diolah menjadi bigram dan dimasukan ke dalam database, informasi tersebut akan digunakan untuk membentuk kalimat. Algoritma pembentuk kalimat yang dipilih adalah algoritma markov chain karena selain mendukung penggunaan data model bigram, algoritma tersebut bersifat fleksibel dan dapat diterapkan untuk menyelesaikan berbagai permasalahan yang menggunakan pendekatan statistik.

NUSANTARA

4. Pemrograman sistem dan debug

Pemrograman dilakukan menggunakan VSTO sebagai penghubung antara aplikasi *Microsoft Word* dengan Visual Studio 2017. Selama proses pemrograman sistem, proses *debug* dilakukan menggunakan *corpus* berupa kalimat dari buku anak-anak yang berjudul "*All About Apples*".

5. Uji Coba

Setelah *add-in* dapat membentuk kalimat sederhana dalam proses *debug*, lalu dimasukan salah satu *corpus* dari *Leipzig Corpora Collection*, yaitu *English News Corpus 2016* yang berisi tiga puluh ribu kalimat yang didapat dari berbagai sumber berita bahasa Inggris. Setelah seluruh kalimat tersebut diolah dalam bentuk *bigram*, informasi tersebut digunakan untuk membentuk kalimat menggunakan seratus *bag of words* yang didapatkan dari penelitian yang dilakukan oleh Ritos Panyawang yang berjudul "Implementasi Algoritma *Backtracking* dalam Aplikasi Pembentuk Kalimat Bahasa Inggris Berbasis *Android*".

Hasil kalimat yang dibentuk lalu dicatat ke dalam sebuah tabel yang lalu diserahkan kepada seorang guru TOEFL yang bernama Masayu Sri Hesti Adlianti, S.S. untuk diperiksa apakah kalimat yang dibentuk bersifat *grammatically correct* atau tidak.

3.2. Perancangan Sistem

Perancang sistem dalam penilitian ini dilakukan perancangan awal terlebih dahulu untuk memahami alur kerja dari sistem. Perancangan sistem yang dibuat adalah sebagai berikut.

SANTAR

3.2.1. Use Case Diagram

Use case diagram pada gambar 3.1. menggambarkan relasi antara aktor dengan *use case*. Tabel 3.1. memberikan informasi mengenai fungsionalitas sistem terhadap aktor-aktor yang ada.



Gambar 3.1. Use Case Diagram

Tabel 3.1.	Tabel	Kegiatan	Use	Case
------------	-------	----------	-----	------

No.	Actor	Use Case	Keterangan
1	Pengguna	Melihat <i>Corpora</i>	Pengguna dapat melihat kalimat apa saja yang digunakan referensi untuk merangkai kalimat.
2	Pengguna	Menambah <i>Corpora</i> Baru	Pengguna dapat menambahkan <i>file</i> yang berisi kalimat untuk digunakan sebagai referensi dalam merangkai kalimat.
3	Pengguna	Melihat N-Gram	Pengguna dapat melihat informasi yang digunakan untuk merangkai kalimat
4	Pengguna	Melakukan Generate N-Gram	Pengguna dapat menghapus hasil <i>n-gram</i> sebelumnya dan membuat <i>n-gram</i> baru dari <i>corpora</i> yang ada.
5	Pengguna S	Melakukan Generate Kalimat	Pengguna dapat menggunakan informasi <i>n-gram</i> untuk merangkai kalimat dari

No.	Actor	Use Case	Keterangan
			kumpulan kata acak.
6	Pengguna	Melihat Tutorial	Pengguna dapat melihat paduan penggunaan <i>add-in</i> .

3.2.2. Activity Diagram

A. Activity Melihat Corpora

Activity diagram pada Gambar 3.2. mengambarkan proses ketika pengguna memilih tombol "View Corpora". Setelah tombol dipilih, add-in akan menampilkan daftar seluruh kalimat yang disimpan sebagai corpora supaya dapat dilihat oleh pengguna.



Gambar 3.2. Activity Melihat Corpora

B. Activity Menambah Corpora Baru



Activity diagram pada gambar 3.3. menggambarkan proses yang terjadi setelah pengguna memilih tombol "Insert New Corpus". Add-in menampilkan sebuah open file dialog di mana pengguna dapat memilih file yang berisi kumpulan kalimat yang akan dijadikan corpus. Kemudian add-in akan memasukan setiap kalimat dalam file tersebut ke dalam database.

C. Activity Melihat N-Gram

Activity diagram pada gambar 3.4. menggambarkan proses yang terjadi ketika pengguna memilih tombol "View N-Gram". Add-in akan menampilkan daftar n-gram (bigram) yang telah diproses kepada pengguna.



D. Activity Melakukan Generate N-Gram

Activity diagram pada gambar 3.5. menggambarkan proses yang terjadi ketika pengguna memilih tombol "Generate N-Gram Table". Oleh karena proses generasi *n-gram* akan menghapus tabel *n-gram* sebelumnya, *add-in* akan memastikan terlebih dahulu apakah pengguna ingin menghapus tabel *n-gram* sebelumnya. Jika pengguna memilih tidak, maka proses generasi *n-gram* akan

dibatalkan. Akan tetapi jike pengguna memilih ya, maka tabel *n-gram* sebelumnya akan dihapus dan proses generasi *n-gram* dari *corpus* akan dimulai.



Gambar 3.5. Activity Melakukan Generate N-Gram

E. Activity Melakukan Generate Kalimat

Activity pada gambar 3.6. menggambarkan proses yang terjadi ketika pengguna akan melakukan generate kalimat dari bag of words. Agar dapat membentuk kalimat, pengguna harus melakukan select terhadap sekumpulan kata yang ada di Microsoft Word. Kumpulan kata tersebut akan disimpan oleh add-in sebagai bag of words. Ketika pengguna memilih tombol "Generate Sentence", add-in akan melakukan query untuk mendapatkan n-gram yang berhubungan dengan setiap kata yang terdapat dalam bag of words.

Proses membentuk kalimat membutuhkan *seed word* sebagai kata pertama dalam kalimat tersebut. Setelah pengguna memilih *seed word* maka *add-in* akan menggunakan kata tersebut sebagai kata pertama dan membuat kalimat menggunakan kata-kata lain dalam *bag of words*.



Gambar 3.6. Activity Melakukan Generate Kalimat

F. Activity Melihat Tutorial

penggunaan add-in kepada pengguna.



3.2.3. Sequence Diagram



A. Sequence Melihat Corpora

Gambar 3.8. Sequence Melihat Corpora

Diagram pada gambar 3.8. menggambarkan alur ketika pengguna melihat corpora. Alur yang ada diawali ketika pengguna memilih tombol "View Corpora" pada ribbon. Ribbon lalu akan memanggil sebuah form khusus yang bertugas untuk melakukan query dan menampilkannya dalam bentuk daftar agar dapat dilihat oleh pengguna.

B. Sequence Menambah Corpora Baru

Diagram pada gambar 3.9. menggambarkan alur ketika pengguna menambah corpora baru. Alur ini diawali ketika pengguna memilih tombol "Insert New Corpus" dan akan memunculkan sebuah open file dialog agar pengguna dapat memilih file yang akan dijadikan sebagai corpus. Setelah itu ribbon akan memanggil sebuah form yang akan membaca isi file tersebut baris per baris dan memasukkannya ke dalam database. Alur ini berakhir ketika seluruh kalimat dalam file yang di-input pengguna telah selesai dimasukkan ke dalam database.



Gambar 3.9. Sequence Menambah Corpora Baru

C. Sequence Melihat N-Gram



Gambar 3.10. Sequence Melihat N-Gram

Diagram pada gambar 3.10. menggambarkan alur ketika pengguna melihat tabel *n-gram*. Alur ini diawali ketika pengguna memilih tombol "*View Corpora*" yang akan memanggil sebuah form khusus yang bertugas untuk melakukan *query* ke *database*. Hasil *query* kemudian akan ditampilkan ke pengguna.



D. Sequence Melakukan Generate N-Gram

Gambar 3.11. Sequence Melakukan Generate N-Gram

Diagram pada gambar 3.11. menggambarkan alur ketika pengguna melakukan generate n-gram. Alur ini diawali ketika pengguna memilih tombol "Generate N-Gram" dan akan memunculkan sebuah message box untuk meminta konfirmasi dari pengguna untuk melakukan truncate terhadap tabel yang menampung informasi hasil n-gram sebelumnya. Apabila pengguna memilih ya, maka add-in akan menghapus isi tabel sebelum mulai mengolah setiap kalimat menjadi bentuk bigram. Setelah setiap kalimat diolah, hasilnya akan disimpan ke dalam database. Apabila pengguna menolak, maka seluruh proses akan dibatalkan.

ΝUSANTARA

E. Sequence Melakukan Generate Kalimat

Diagram pada gambar 3.12. menggambarkan alur ketika pengguna akan membentuk kalimat. Alur ini diawali dengan pengguna melakukan *select* terhadap sekumpulan kata pada *Microsoft Word*. Kumpulan kata ini kemudian akan digunakan sebagai *bag of words*. Setelah itu pengguna akan memilih tombol "Generate Sentence" di mana *ribbon* akan memanggil sebuah form khusus yang akan memilih *n-gram* yang mengandung anggota *bag of words*.

Alur ini berhenti ketika *add-in* menampilkan kalimat yang telah dibentuk kepada pengguna, terlepas apakah pengguna memilih *seed word* atau tidak.



Gambar 3.12. Sequence Melakukan Generate Kalimat

F. Sequence Melihat Tutorial



"Tutorial" dan *add-in* akan menampilkan paduan penggunaan *add-in*.

3.2.4. Struktur Tabel

Tabel : t_sentence

Fungsi : menyimpan setiap kalimat corpus yang di-input pengguna

Tabel 3.1. Struktur Tabel	t	sentence
---------------------------	---	----------

Nama Field	Tipe Data	Keterangan
s_id	int	Primary key
sentence	varchar(200)	Kalimat <i>corpus</i>

Tabel : t_ngram

Fungsi : menyimpan bigram yang diperoleh dari hasil pengolahan kalimat

Tabel 3.2	. Struktur	Tabel t	_ngram
-----------	------------	---------	--------

Nama Field	Tipe Data	Keterangan
ng_id	int V	Primary key
word1	varchar(200)	Kata pertama dalam bigram
word2	varchar(200)	Kata kedua dalam bigram
frequency	int F	Frekuensi kemunculan dalam corpus

Tabel 3.1. dan tabel 3.2. digunakan oleh *add-in* untuk membantu proses pembentukan kalimat. Tabel t_sentence digunakan untuk menyimpan *corpora* yang di-*input* oleh pengguna dengan s_id sebagai *primary key*. Tabel t_ngram digunakan untuk menyimpan informasi *bigram* yang dibuat berdasarkan *corpora*. Kata pertama dalam *bigram* disimpan dalam kolom word1 dan kata kedua dalam *bigram* disimpan dalam kolom word2. Frekuensi kemunculan *bigram* dalam *corpora* disimpan dalam kolom *frequency*.

3.2.5. Class Diagram





Gambar 3.15. Class Diagram Bagian Kedua

Class DBHandler, frmViewCorpus, dan frmInsertCorpus dijabarkan pada gambar 3.17. sedangkan *class* frmViewNGramResult dan frmInsertNGram pada gamabr 3.18.



Gambar 3.16. Class Diagram Bagian Ketiga



Class diagram pada gambar 3.17. di atas menggambarkan hubungan antarkelas dalam add-in yang dirancang.

3.2.6. Flowchart

Flowchart adalah diagram yang menggambarkan proses kerja sebuah algoritma. Berikut adalah *flowchart* yang menggambarkan proses *n-gram* yang digunakan untuk memroses corpus ke dalam bentuk bigram dan markov chain yang digunakan untuk membentuk kalimat dari seed word menggunakan informasi bigram yang ada di database.

A. Flowchart N-Gram



Gambar 3.18. Flowchart Proses Melakukan N-Gram

Flowchart pada gambar 3.18. menggambarkan cara kerja add-in dalam mengubah kalimat corpora menjadi informasi dalam bentuk n-gram. Proses dimulai dengan melakukan query untuk mendapatkan seluruh kalimat dalam

database. Setelah itu setiap kalimat diproses sehingga menjadi *bigram*. Proses mengubah kalimat menjadi *bigram* secara detail dapat dilihat pada gambar 3.19. dan hasilnya akan disimpan ke dalam *database*. Setelah seluruh kalimat selesai diproses sebelum disimpan ke dalam *database*.



Gambar 3.19. *Flowchart* Proses Melakukan *N-Gram* (Lanjutan)

Flowchart pada gambar 3.19. dimulai dengan memisahkan simbol dari setiap kata pada kalimat. Setelah itu kalimat akan dipecah dan setiap kata akan ditampung dalam sebuah *array*. Setiap anggota *array* akan diproses menjadi bentuk *bigram* dalam bentuk pasangan sebuah kata dengan kata yang muncul setelahnya. Setelah setiap kata dibuat menjadi *bigram*, maka *array* penampung akan dikembalikan ke proses pada gambar 3.16. dan akan disimpan ke dalam *database*.

B. Flowchart Markov Chain



Gambar 3.20. Flowchart Markov Chain

Flowchart pada gambar 3.20. menggambarkan cara kerja add-in dalam membentuk kalimat bahasa Inggris dari bag of words yang telah ditentukan oleh pengguna. Seluruh proses pada flowchart ini dilakukan dalam fungsi rekursif dan akan berhenti sepenuhnya dalam kondisi tertentu. Sebelum proses ini dimulai, pengguna telah melakukan input bag of words dan telah memilih kata untuk dijadikan seed word. Selain itu add-in telah melakukan query dan menyimpan seluruh n-gram yang berhubungan dengan kata dalam bag of words ke dalam sebuah variabel.

Langkah pertama dalam proses ini adalah mengubah seed word yang dipilih pengguna ke lowercase. Setelah itu akan dilakukan validasi informasi *n*-gram untuk mencari *n*-gram yang diawali dengan seed word yang dipilih pengguna. Proses validasi tersebut digambarkan pada gambar 3.21. dan hasilnya dikumpulkan ke dalam sebuah array penampung. Bila sebagian *n*-gram dinyatakan valid dan array penampung memiliki isi, maka akan dipilih *n*-gram acak. Melalui bigram tersebut akan ditentukan kata yang muncul setelah seed word. Seed word akan direturn dan sekaligus memanggil method yang akan melakukan markov chain menggunakan kata kedua sebagai seed word. Akan tetapi bila array tersebut kosong, maka seed word akan ditampilkan dan seluruh proses akan diakhiri tanpa memanggil fungsi rekursif lain.



Gambar 3.21. Flowchart Validasi N-Gram untuk Markov Chain

Proses validasi dimulai dengan mengubah setiap kata dalam *bigram* menjadi *lowercase*. Apabila *bigram* yang diuji memenuhi 4 kriteria, maka *bigram* tersebut dinyatakan valid. Bila 4 kriteria tidak terpenuhi, maka dinyatakan tidak valid.

3.2.7. Rancangan User Interface

Pada awal proses perancangan diperlukan sebuah tampilan sederhana yang memenuhi seluruh fungsi utama yang ada.Berikut merupakan rancangan awal seluruh *user interface* dari *add-in* yang akan dibuat.

1. Ribbon Utama

	T		
View Corpora	Tutorial		
Insert New Corpus	1000000		
View N-Gram			
Generate N-Gram	ж. Ж.		

Gambar 3.22. Rancangan Ribbon Utama

Ribbon utama pada gambar 3.20. digunakan untuk menghubungkan

pengguna dengan setiap fitur yang ada dalam add-in.

2. Windows Form View Corpora

F	View	Corpora 🖸 🗡	
	No.	Sentence Corpora	
	1 2 3	Lorem lpsum Lorem lpsum Lorem lpsum	
UNI		Close	AS
Gambar	3.23	3. Rancangan Form View C	orpora A
Form View Corpo	ora p	pada gambar 3.23. digunal	kan untuk menampilkan
kalimat yang disimpan di c	lalar	n <i>database</i> .	

3. Windows Form View N-Gram

View N-Gram No. Word 1 Word 2 Frequency 1 Lorem Ipsum 8 2 Ipsum Lorem 3 Image: Second structure Image: Second structure Image: Second structure Close Second structure Second structure					
No. Word 1 Word 2 Frequency 1 Lorem Ipsum 8 2 Ipsum Lorem 3 1 Lorem 3 1 Lorem 3 1 Lorem 1 1 Lorem 1 2 Ipsum Lorem 3 1 Lorem 1 Lorem 1 1 Lorem 1 1 Lorem 1 1 Lorem 1 1 Lorem 1 1 1 Lorem 1 Lorem 3 1 <	 Viev	v N-Gram		回×	
1 Lorem Ipsum 8 2 Ipsum Lorem 3	No.	Word 1	Word 2	Frequency	
Close	1 2	Lorem Ipsum	lpsum Lorem	8 3	
Close					
		C	Close		

Gambar 3.24. Rancangan Form View N-Gram

Form View N-Gram pada gambar 3.24. digunakan untuk melihat seluruh informasi *n-gram* yang disimpan dalam *database*.

4. Windows Form Generate Sentence



Tutorial	
Tutorial:	
1. step 1	
2. step 2	
3. finally step 3	
(clara	
ciose	

Gambar 3.26. Rancangan Form Tutorial

Form Tutorial pada gambar 3.26. digunakan untuk memberikan paduan

penggunaan *add-in* kepada pengguna.

