

PENERAPAN ALGORITMA *BOYER MOORE* DAN METODE *N-GRAM* PADA APLIKASI PENYUNTING NASKAH TEKS BAHASA INDONESIA BERBASIS *WEB*

^[1]Dini SURIANTO, ^[2]Dedi TRIYANTO, ^[3]Uray RISTIAN

^[1]^[2]^[3]Jurusan Rekayasa Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jalan Prof. Dr. H. Hadari Nawawi, Pontianak

Telp./Fax.: (0561) 577963

e-mail : ^[1]dinisurianto@gmail.com_^[2]deditriyanto@siskom.untan.ac.id

^[3]eristian@siskom.untan.ac.id

ABSTRAK

Literatur yang saat ini sering digunakan adalah dalam bentuk digital dengan format pdf. Untuk dapat mengolah perlu dilakukan konversi kedalam bentuk teks yang editable. Hasil konversi yang dihasilkan kebanyakan dalam bentuk yang tidak teratur format penulisannya dan perlu dilakukan pengeditan yang memerlukan waktu dan tenaga yang tidak sedikit. Untuk melakukan pengeditan yang lebih efisien diperlukanlah sebuah aplikasi yang dapat mengoreksi teks secara otomatis. Pencarian dan perbaikan kesalahan ejaan menggunakan algoritma Boyer Moore dan metode N-Gram. Dari hasil pengujian, Penerapan Algoritma Boyer Moore pada aplikasi untuk mencari kata yang salah dalam sebuah teks mendapatkan keberhasilan 100%. Kata-kata pada teks di bandingkan dengan kata-kata dalam database dan dengan Algoritma Boyer Moore kata yang tidak sesuai dengan database berhasil dikenali dan dianggap sebagai kata yang salah. Penerapan Metode N-Gram pada aplikasi pengoreksi teks, dari hasil pengujian mendapatkan rata-rata tingkat keberhasilan mencapai 89,68%. Masih ada beberapa kata yang diperbaiki tidak sesuai dengan kata yang seharusnya karena perhitungan dengan Metode N-Gram mendapatkan hasil yang lebih tinggi dengan kata lain pada database..

Kata Kunci: *Boyer Moore, N-Gram, Kata Bahasa Indonesia*

1. PENDAHULUAN

Literatur dapat diartikan sebagai sumber ataupun acuan yang digunakan dalam berbagai macam aktivitas di dunia pendidikan ataupun aktivitas lainnya. Literatur juga dapat diartikan sebagai rujukan yang digunakan untuk mendapatkan informasi tertentu. Literatur dapat berupa buku ataupun berbagai macam tulisan lainnya [1]. Literatur yang mulai banyak digunakan saat ini adalah literatur dalam bentuk digital. Untuk mendapat sumber pengetahuan bisa dari internet. Banyak sekali literatur-literatur yang mulai didigitalisasi, seperti buku. Buku-buku yang didigitalisasi ini melalui tahap pemindaian visual kemudian dikemas dalam format pdf (Portable Document Format). Hasil pemindaian yang berupa file pdf tidak bisa diubah ataupun diedit. Untuk mengubah atau mengedit dokumen tersebut harus di konversi terlebih dahulu ke bentuk lain seperti format

.doc. Dalam proses konversi ini sering didapati hasil konversi yang tidak beraturan. Diantaranya banyaknya kesalahan pengetikan kata (typo), spasi ganda, tanda baca yang tidak sesuai dan lain-lain. Untuk merapikan dokumen hasil konversi ini dilakukan dengan cara mengedit dokumen tersebut secara langsung.

Mengedit dokumen hasil konversi yang berantakan tentu memakan waktu dan tenaga yang tidak sedikit karena diperlukan ketelitian dan ketepatan dalam melakukan pengeditan. Ditambah lagi juga harus merapikan format penulisan yang tidak beraturan spasi maupun jarak antar baris tulisan. Selain itu untuk dokumen dalam jumlah yang banyak tentunya sangat memerlukan waktu yang lama. Untuk mempermudah proses mengedit diperlukan lah sebuah aplikasi yang dapat menyederhanakan serta memperbaiki teks hasil konversi yang tidak beraturan menjadi lebih rapi sehingga

lebih mudah untuk diedit. Proses perbaikan teks tersebut diperlukan sebuah pendekatan pencarian string khusus yaitu dengan pendekatan perkiraan (String Matching).

Beberapa penelitian sebelumnya tentang pendekatan pencarian string khusus pernah dilakukan, seperti penelitian dengan judul “Aplikasi Editor Pemeriksa Ejaan dan Rekomendasi Kata dalam Bahasa Indonesia Berbasis Android” [2]. Dalam penelitian ini digunakan algoritma Nazief dan Adriani. Algoritma Nazief dan Adriani adalah algoritma stemming yang dikembangkan oleh Bobby Nazief dan Mirna Adriani pada tahun 1996 sebagai hasil penelitian internal Universitas Indonesia. Algoritma ini dibuat berdasarkan aturan morfologi dan enkapsulasi afiks yang diizinkan dan dilarang. Sebelum melakukan pengecekan ejaan dengan algoritma Nazief dan Adriani, data yang di input user terlebih dahulu dicek menggunakan Apache Lucene. Apache Lucene akan melakukan crawling terhadap data yang sudah digunakan sebelumnya. Aplikasi yang dibuat pada penelitian ini berbasis android. Algoritma Nazief dan Ariani masih memiliki kekurangan yaitu pengecekan jika suatu kata memiliki dua suku kata, seperti ditandatangani.

Kemudian penelitian oleh dengan judul “Program Aplikasi Editor Kata Bahasa Indonesia Menggunakan Metode Approximate String Matching Dengan Algoritma Levenshtein Distance Berbasis Java” [3]. Metode yang digunakan pada penelitian ini adalah metode Approximate String Matching dengan menggunakan Algoritma Levenshtein Distance. Algoritma Levenshtein Distance bekerja dengan cara menghitung jumlah minimum transformasi suatu string menjadi string lain yang meliputi penggantian penghapusan dan penyisipan. Sistem editor kata Bahasa Indonesia untuk mengecek kebenaran ejaan kata memiliki akurasi yang cukup tinggi dan memiliki efisiensi waktu yang lebih cepat dibandingkan dengan pengecekan secara manual. Tingkat ketelitian sistem juga lebih tinggi dibandingkan dengan pengecekan secara manual terhadap teks yang memiliki jumlah kata yang relatif banyak. Sistem ini memiliki kelemahan semakin bervariasi kesalahan ejaan kata, maka, alternative membenaran kata yang muncul mungkin akan tidak sesuai dengan maksud kata yang

sebenarnya, karena sistem hanya mencari jarak yang terkecil (Priyana, 2013).

Selanjutnya penelitian tentang sebuah algoritma string matching Boyer Moore dengan judul “Implementasi Boyer Moore Pada Aplikasi Pencarian Rumus Matematika dan Fisika” [4]. Metode Boyer Moore pada penelitian ini digunakan untuk mencocokkan kata kunci yang dimasukkan dengan kata kunci yang ada di basis data. Kata kunci digunakan untuk mencari rumus yang berhubungan dengan soal. Metode string matching dengan algoritma boyer moore dapat diterapkan pada aplikasi pencarian rumus matematika dan fisika tingkat SMA yang disertai dengan pengujian algoritma sebanyak 30 kali dimana aplikasi dapat mencocokkan pola kata kunci pada soal serta menampilkan list atau kumpulan rumus matematika dan fisika tingkat SMA yang cocok dan sesuai dengan soal yang dikerjakan.

Berdasarkan pemaparan sebelumnya, pada tugas akhir ini penulis melakukan penelitian dengan judul “Penerapan Algoritma Boyer Moore dan Metode N-Gram pada Aplikasi Penyunting Naskah Teks Berbahasa Indonesia Berbasis Web”. Algoritma Boyer Moore akan menyimpan informasi pergeseran untuk melakukan pencarian string. Selain itu digunakan juga metode N-gram dalam Pengoreksian kesalahan kata. N-gram merupakan sebuah metode yang diaplikasikan sebagai pembangkitan kata atau karakter. Metode N-gram ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah n dari sebuah kata secara kontinuitas. Diharapkan aplikasi ini dapat membantu seorang editor dalam menemukan kesalahan kesalahan dalam penulisan naskah sehingga mempermudah dalam proses penyuntingan naskah.

2. LANDASAN TEORI

2.1 *String Matching*

String matching merupakan suatu metode yang digunakan untuk mencocokkan lambang ataupun huruf (string) dalam suatu kasus deret [5]. Metode ini dipelajari untuk mengetahui bagaimana komputer bisa mendapatkan pola atau deretan pola yang telah diberikan. Dasarnya muncul string matching yaitu bagaimana mendapatkan sebuah simbol atau huruf dalam kumpulan teks dari sebuah kasus.

Dalam suatu kasus diberikan misal teks dengan x sebagai karakter dalam array $A[1...x]$ dan suatu pola lain yaitu dengan Y sebagai karakter dalam array $B[1...Y]$. bagaimana mendapatkan integer s yang disebut dengan valid shift? Terlebih dahulu valid shift ini memenuhi kriteria sebagai berikut dimana $0 \leq s < x - y$ dan $A[s + 1 \dots s + y] = B[1 \dots y]$. Jika B sebagai substring dari A , maka selanjutnya akan diketahui apakah B didalam A . Elemen-elemen B dan A dapat berupa karakter atau alfabet seperti $\{0,1\}$ atau $\{A,B,C,D,E,\dots,Z,a,b,c,d,\dots,z\}$.

Untuk kasus yang membutuhkan string matching dengan efisiensi tinggi dari aplikasi, dapat menggunakan algoritma Boyer Moore. Contoh penggunaan algoritma ini adalah untuk menyunting suatu teks, dan substitusi komentar kedalam teks. Algoritma ini memungkinkan string matching dengan ukuran alphabet moderat dan pola teks yang relatif panjang dalam kecepatan tinggi. Selama pengujian ada kemungkinan penempatan pola A dalam data teks B . jika karakter $B[i] = c$ tidak sebanding dengan pola $A[j]$ maka akan berjalan ketentuan tertentu :

1. Jika c tidak berisi dan dalam A boleh dimana saja, maka akan geser pola A melalui $T[i]$
2. Disaat kondisi lain, geser A sampai karakter c sehingga A dapat lurus dengan $B[i]$.

Teknik ini agar menghindari terjadinya tumpukan antara perbandingan pola pergeseran relatif terhadap data teks.

Dapat disimpulkan bahwa algoritma boyer moore relatif lebih cepat pada alfabet besar (panjang pola teks), hasilnya bukan untuk string biner atau membentuk pola pendek, untuk membuat string biner lebih diutamakan menggunakan algoritma knuth-morris-pratt. Sedangkan untuk pola-pola yang pendek dapat menggunakan algoritma naïve karena saat eksekusi berlangsung worst case akan berbentuk quadratic.

2.2 Algoritma Boyer Moore

Algoritma boyer moore adalah algoritma pencarian string yang paling efektif saat ini. Algoritma yang ditemukan oleh Bob Boyer dan J. Strother Moore ini telah menjadi standar untuk berbagai literatur pencarian string. Karakteristik utama dari algoritma boyer moore adalah algoritma ini melakukan pencocokan

string mulai dari kanan ke kiri. Dengan karakteristik tersebut, ketidakcocokan saat terjadi perbandingan string akan membuat pergerakan pattern melompati lebih jauh untuk menghindari perbandingan karakter string yang diperkirakan gagal [6].

Boyer moore merupakan salah satu algoritma pattern matching yang cukup terkenal (Eza Rahmanita, 2014). Algoritma ini menggunakan beberapa kasus pengecekan teks (input karakter yang akan dibaca) dengan pattern (pola yang akan disaring). Algoritma boyer moore adalah algoritma pencarian string yang mencari dengan cara membandingkan sebuah huruf dengan huruf yang ada di pattern yang dicari, dan menggeser pattern tersebut hingga posisinya sama dengan teks yang dicari dan membandingkan kata tersebut. Cara ini disebut character jump.

Character-jump technique melakukan suatu aksi ketika perbandingan antara dua karakter yang berbeda. Ada dua aksi tergantung pada teks s dan kata w yang dimiliki, jika p yaitu karakter pada s yang sedang diproses yang tidak cocok maka ada dua kemungkinan aksi. Mencari karakter yang sesuai dan cara penggeseran sebuah karakter perbandingan terakhir [6].

Karakteristik utama dari algoritma boyer moore adalah algoritma ini melakukan pencocokan string mulai dari kanan (belakang) dengan karakteristik tersebut, ketidakcocokan saat terjadi perbandingan string akan membuat pergerakan pattern melompat lebih jauh untuk menghindari karakter pada string yang diperkirakan gagal. Sehingga proses pencarian string akan lebih optimal.

Contoh cara kerja algoritma boyer moore ini adalah sebagai berikut:

Teks : mendaftar di unijoyo

Pattern : daftar

Tabel 1 Contoh Algoritma Boyer Moore

	m	e	n	d	a	f	t	a	r
1	d	a	f	t	a	r			
2		d	a	f	t	a	r		
3			d	a	f	t	a	r	
4				d	a	f	t	a	r

Dalam Tabel 1 dengan menggunakan algoritma boyer moore dalam pencarian string yang mencari dengan cara membandingkan sebuah huruf dengan huruf yang ada di pattern (m karakter = daftar) yang dicari, dan menggeser pattern sejauh m karakter tersebut hingga posisinya sama dengan teks = n karakter

yang dicari dan membandingkan kata tersebut. Dalam pergeserannya, m karakter akan mencocokkan huruf dimulai dari kanan ke kiri. Jika huruf di kanan tidak cocok dengan n karakter maka m karakter atau pattern akan menggeser sejauh m karakter untuk mencocokkan kembali huruf yang dimaksud, sehingga menjadi cocok.

2.3 N-Gram

N-gram adalah potongan N-karakter yang diambilkan dari suatu string. Untuk mendapatkan N-gram yang utuh ditempuh dengan menambahkan blank pada awal dan akhir string. Misalnya suatu string "TEXT" setelah ditambah awal dan akhir dengan "_" sebagai pengganti blank akan didapat N-gram sebagai berikut :

Unigram : T,E,X,T

Bigram : _T, TE, EX, XT, dan T_

Trigram : _TE, TEX, EXT, XT_, dan T__

Quadgram : _TEX, TEXT, EXT_, XT__ , dan T___

Dapat disimpulkan bahwa untuk string berukuran n akan dimiliki n unigram dan n+1 bigram, n+1 trigram, n+1 quadgram dan seterusnya. Penggunaan N-gram untuk matching kata memiliki keuntungan sehingga dapat diterapkan pada recovery pada input karakter ASCII yang terkena noise, interpretasi kode pos, information retrieval dan berbagai aplikasi dalam pemrosesan Bahasa alami [7].

Keuntungan N-gram dalam matching string adalah berdasarkan karakteristik N-gram sebagai bagian dari suatu string, sehingga kesalahan pada sebagian string hanya akan berakibat perbedaan pada sebagian N-gram. Sebagai contoh jika N-gram dari dua string dibandingkan, kemudian kita menghitung cacah N-gram yang sama dari dua string tersebut maka akan didapatkan nilai similaritas atau kemiripan dua string tersebut yang bersifat resistan terhadap kesalahan tekstual.

2.4 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble Chart, Bubble Diagram,

model proses, diagram alur kerja, atau model fungsi [8].

DFD ini adalah salah satu pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks daripada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisis maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

2.5 PHP

PHP atau kependekan dari Hypertext Preprocessor adalah salah satu bahasa pemrograman open source yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skripsi HTML (Firman, 2016). Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari. PHP merupakan bahasa scripting server – side, dimana pemrosesan datanya dilakukan pada sisi server. Sederhananya, serverlah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada client yang melakukan permintaan. Adapun pengertian lain PHP adalah akronim dari Hypertext Preprocessor, yaitu suatu bahasa pemrograman berbasis kode – kode (script) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke web browser menjadi kode HTML”.

2.6 MySQL

MySQL adalah sebuah database server, dapat juga berperan sebagai client sehingga sering disebut database client/server yang open source dengan kemampuan dapat berjalan baik di OS (Operating System) manapun, dengan platform Windows maupun [9].

Selain itu database ini memiliki beberapa kelebihan disbanding database lain, diantaranya adalah:

1. MySQL adalah sebuah software database yang open source, artinya program ini bersifat free atau bebas digunakan oleh

- siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
2. MySQL merupakan sebuah database client. Selain menjadi server yang melayani permintaan, MySQL juga dapat melayani query yang mengakses database pada server. Jadi MySQL dapat juga berperan sebagai client.
 3. MySQL mampu menerima query yang bertumpuk dalam satu permintaan atau disebut multi-threading.
 4. MySQL merupakan sebuah database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran gigabyte sekalipun.
 5. MySQL memiliki kecepatan dalam pembuatan tabel maupun peng-update-an tabel.

2.7 XAMPP

XAMPP merupakan tool yang menyediakan paket perangkat lunak kedalam satu buah paket. Dengan meng-install XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasinya secara otomatis untuk anda atau auto konfigurasi [10].

Banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola database dalam MySQL, salah satunya adalah phpMyAdmin. Dengan phpMyAdmin kita dapat membuat tabel, mengisi data dan lain-lain dengan mudah tanpa harus menghafal perintahnya.

2.8 Penyunting Naskah

Menurut KBBI (2007:1106) definisi penyuntingan adalah proses, cara, perbuatan menyunting atau sunting-menyunting. Sedangkan definisi menyunting adalah menyiapkan naskah siap cetak atau siap terbit dengan memperhatikan segi sistematika penyajian, isi, dan Bahasa (menyangkut ejaan, diksi, dan struktur kalimat), merencanakan dan mengarahkan penerbitan (surat kabar, majalah) dan menyusun atau merakit (film, pita rekaman) dengan cara memotong-motong dan memasang kembali [11].

Tujuan penyuntingan yang dilakukan oleh para penyunting adalah untuk menjadikan transkrip sebagai karya yang sempurna yang dapat dibaca dan dihayati dengan mudah oleh pembaca apabila diterbitkan kelak. Untuk

memastikan pengaliran atau penyebaran ide daripada penulis kepada pembaca dapat disampaikan dalam Bahasa yang dramatis, jelas, indah dan menarik.

3. METODE PENELITIAN

Metodologi Penelitian yang digunakan dalam penelitian ini meliputi studi literatur, analisis kebutuhan, perancangan sistem, pembuatan sistem, pengujian, dan analisis hasil pengujian.

Studi literatur dilakukan untuk penggalian informasi dengan mengkaji buku-buku yang berkaitan dengan sistem aplikasi yang akan dirancang.

Analisis kebutuhan dilakukan dengan tujuan untuk mengetahui dan mendata kebutuhan dalam pembangunan sistem. Analisis kebutuhan pada penelitian ini berupa perangkat lunak dan perangkat keras yang dibutuhkan.

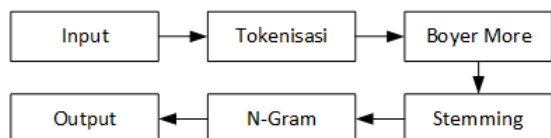
Setelah dilakukan analisis kebutuhan, kemudian dilakukan perancangan sistem. Perancangan sistem meliputi pembuatan *website* sebagai antarmuka, pembuatan *database*, dan pembuatan program aplikasi pencarian dan pengoreksian kata dalam sebuah teks Bahasa Indonesia.

Tahap pengujian merupakan tahap yang bertujuan untuk menguji sistem secara keseluruhan setelah tahap integrasi. Semua elemen sistem diuji untuk melihat unjuk kerja dari sistem, baik itu unit *input*, unit pemroses, kemudian unit *output* yaitu antarmuka *website*.

4. PERANCANGAN

4.1 Perancangan Sistem

Sistem pada aplikasi ini dibangun menggunakan Bahasa pemrograman PHP pada sistem aplikasi berbasis web. Sistem ini bekerja dengan memasukkan data berupa teks naskah. Tahap selanjutnya adalah sistem akan melakukan proses tokenisasi terhadap naskah yang dimasukkan. Tokenisasi adalah proses untuk membagi teks yang dapat berupa kalimat, paragraf atau dokumen, menjadi token-token. Data hasil tokenisasi akan dicocokkan dengan data yang ada di database. Jika data yang dimasukkan tidak ditemukan dalam database maka akan diganti dengan data yang paling mendekati. Diagram blok pada perancangan sistem dapat dilihat pada Gambar 1.

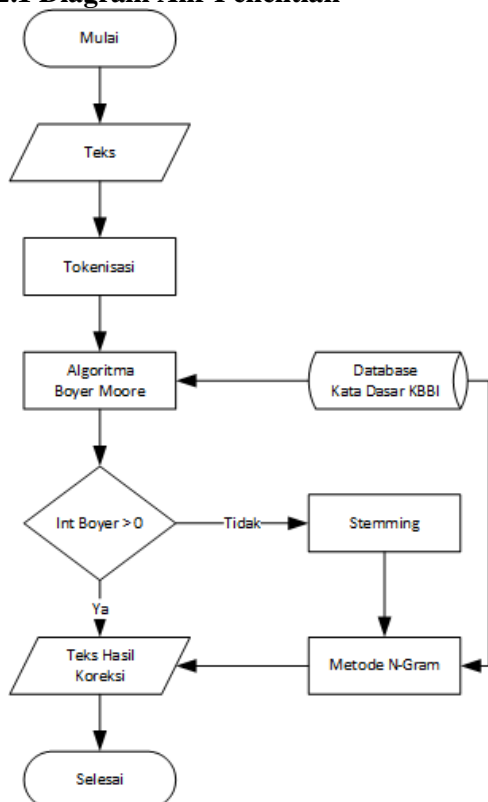


Gambar 1 Diagram Blok Perancangan Sistem

4.2 Perancangan Aplikasi

Dalam penelitian ini, perancangan aplikasi berisikan Diagram Alir Penelitian, *Data Flow Diagram* dan Perancangan *Database*.

4.2.1 Diagram Alir Penelitian



Gambar 2 Diagram Alir Aplikasi Penyunting Naskah

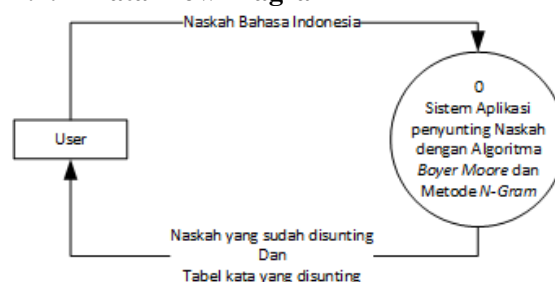
Berdasarkan Gambar 2, dapat dijelaskan bahwa proses penyuntingan naskah dimulai dengan pengguna memasukkan naskah teks Bahasa Indonesia ke dalam aplikasi. Kemudian pada proses tokenisasi sistem akan memecah teks dari input berdasarkan pembatas tertentu yang terdapat dalam *string* tersebut. Tokenisasi yang dirancang pada aplikasi ini memecah teks dengan *whitespace* atau spasi sebagai pembagi, lalu mengubahnya menjadi huruf kecil agar seragam, serta membuang seluruh tanda baca yang terdapat dalam teks.

Selanjutnya pada tahap Boyer Moore sistem akan mulai mencocokkan kesamaan antara teks masukan yang sudah dipecah

dengan teks kata dasar yang ada di *database*. Jika belum ada kecocokan maka sistem melakukan proses *Stemming*, mengubah kata yang berimbuhan menjadi bentuk kata dasar.

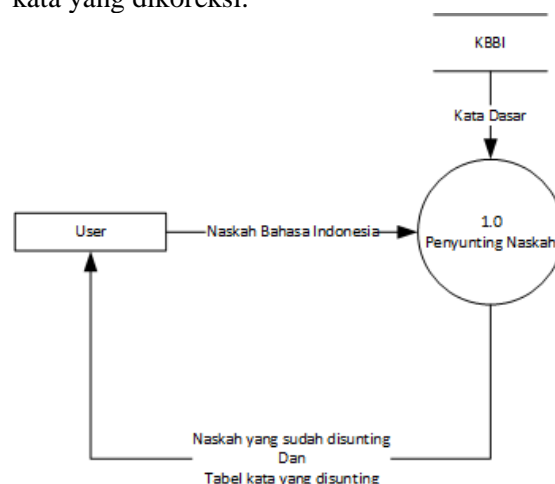
Metode *N-Gram* adalah pemotongan karakter yang diambil dari suatu *string*. Metode ini digunakan untuk mencari kata yang paling mirip dengan kata masukan yang tidak ditemukan di *database*. Tahap terakhir adalah menampilkan hasil teks yang sudah di koreksi dan juga daftar tabel yang berisi kata-kata yang ditemukan kesalahan penulisan dan diubah ke data baku yang sesuai dengan KBBI.

4.2.2 Data Flow Diagram



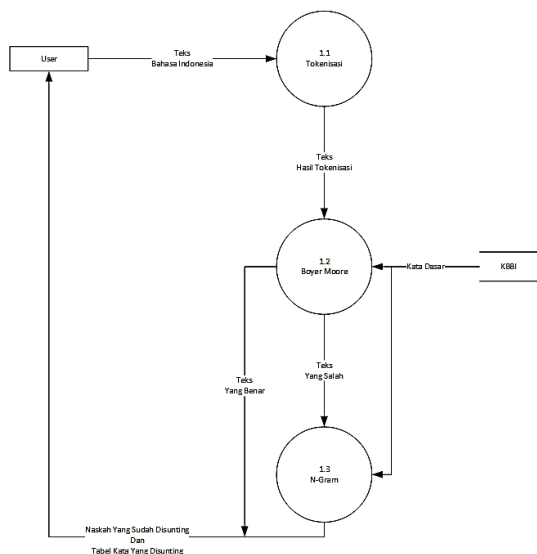
Gambar 3 DFD Level 0

Berdasarkan Gambar 3 aliran data bersumber dari masukan oleh pengguna kedalam sistem, yaitu teks naskah dalam Bahasa Indonesia. Untuk data pembanding menggunakan *database* dari sistem berupa kata dasar yang terdapat dalam KBBI. Kemudian diproses dan menghasilkan keluaran berupa naskah yang sudah di koreksi serta daftar kata-kata yang dikoreksi.



Gambar 4 DFD Level 1

Dari gambar 4 proses menyunting naskah dengan cara membandingkan teks masukan dengan kata dasar yang ada di *database* untuk kemudian diubah ke kata yang benar dan ditampilkan ke *User*.



Gambar 5 DFD Level 2

Gambar 5 menggambarkan proses tokenisasi memecah kalimat menjadi kata, menghapus semua karakter selain huruf dan mengubah semua menjadi huruf kecil. Proses Boyer Moore mencari kata yang tidak sesuai dengan kata dasar yang ada di *database*. Proses *N-Gram*, mengganti kata dengan kata dari *database* yang paling mendekati kata yang tidak ditemukan.

4.2.3 Perancangan Database

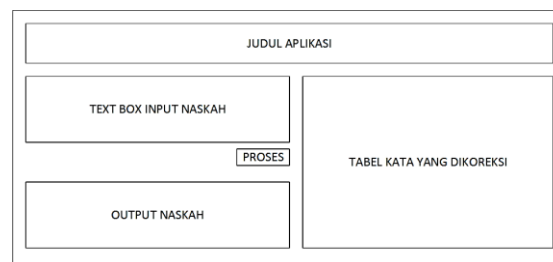
Dalam membangun sistem ini menggunakan satu tabel dalam *database* bernama *db_kbbi*. Tabel ini bernama *tb_katadasar*, tabel ini digunakan untuk menyimpan data kata dasar dari Kamus Besar Bahasa Indonesia untuk digunakan sebagai data pembandingan. Untuk isi dari tabel dapat dilihat pada Tabel 2.

Tabel 2 Tabel *tb_katadasar*

No	Nama Field	Type dan Panjang Field	Keterangan
1	<i>id_katadasar</i>	Integer(10)	Primary key
2	<i>katadasar</i>	Varchar(70)	

4.3 Perancangan Antar Muka

Perancangan antarmuka dari sistem dalam penelitian ini dibuat menggunakan Bahasa pemrograman *PHP* yang ditampilkan pada *web browser*. Sistem akan dibangun pada satu menu tampilan utama.



Gambar 6 Rancangan Antar Muka Aplikasi

Berdasarkan Gambar 6 tersebut, untuk melakukan proses penyuntingan naskah pertama user memasukkan teks naskah dalam Bahasa Indonesia ke Text Box Input Naskah. Setelah Text Box diisi dengan teks naskah, user mengklik tombol proses untuk memulai proses penyuntingan naskah. Setelah proses penyuntingan yang dilakukan oleh sistem selesai, maka hasil dari naskah yang telah disunting akan ditampilkan pada kotak output naskah dan kata-kata yang disunting akan ditampilkan pada tabel kata yang dikoreksi.

5. IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1 Implementasi Algoritma Boyer Moore dan Metode N-Gram Pada Perangkat Lunak

5.1.1. Tokenisasi

Proses tokenisasi ini seluruh isi teks dalam naskah teks Bahasa Indonesia dijadikan huruf kecil dengan *strtolower*. Kemudian akan menghapus karakter selain huruf yang ada pada naskah teks sesuai dengan yang sudah terdeklarasi dalam kode program. Dalam proses ini akan dihapus juga *spasi* yang lebih dari satu yang mana sebelumnya setiap kata yang akan dijadikan *array* dengan *explode(" ", trim(\$tanpaAngka))* kemudian akan disatukan kembali dan setiap kata akan diberikan satu buah *spasi* dengan *implode(" ", \$arrTemp)* sehingga menjadikan hasil tokenisasi benar-benar hanya mengandung sebuah *spasi* di antara setiap kata. Berikut ini adalah contoh dari naskah teks uji ketika melalui proses tokenisasi.

Berikut ini adalah contoh dari naskah teks uji ketika melalui proses tokenisasi.

Sprinter tercepat dunia kelas junior Lalu Muhamad Zohri, turut prihatin dengan musibah gempa bumi yang terjadi Pulau Lombok, yang berkekuatan 6,4 skala richter pada 29 Juli 2018. Keprihatinan itu dia wujudkan dengan memberi bantuan, bagi korban gempa bumi.

sprinter tercepat dunia kelas junior lalu muhamad zohri turut prihatin dengan musibah gempa bumi yang terjadi pulau lombok yang berkekuatan skala richter pada juli keprihatinan itu dia wujudkan dengan memberi bantuan bagi korban gempa bumi

5.1.2. Algoritma Boyer Moore

Pada proses algoritma *boyer moore*, string hasil tokenisasi dibandingkan dengan string yang ada di *database*. Proses pencocokan dilakukan mulai dari karakter string yang paling akhir sehingga lompatan terjadi lebih jauh dan membuat proses pencocokan lebih cepat. String terus digeser hingga ditemukan kecocokan diantara keduanya.

String dicocokkan per huruf nya. Jika tidak ditemukan maka akan terus di geser hingga ditemukan kecocokan antar *string* masukan dan *database* yang dibandingkan. Proses akan terus berulang dari kata pertama hingga kata terakhir.

Berikut ini adalah contoh teks uji setelah melewati proses *boyer moore*.

sprinter tercepat dunia kelas junior lalu muhamad zohri turut prihatin dengan musibah gempa bumi yang terjadi pulau lombok yang berkekuatan skala richter pada juli keprihatinan itu dia wujudkan dengan memberi bantuan bagi korban gempa bumi

sprinter tercepat **dunia kelas junior lalu** muhamad zohri **turut prihatin dengan musibah gempa bumi yang terjadi pulau lombok yang** berkekuatan **skala richter pada juli** keprihatinan itu dia wujudkan dengan memberi **bantuan bagi korban gempa bumi**

5.1.3. Stemming

Proses *stemming* merupakan proses yang akan menjadikan setiap kata menjadi kata dasar. Pada proses *stemming* ini merupakan proses penghapusan imbuhan yang terdapat pada setiap kata. Baik itu berupa awalan, akhiran maupun gabungan keduanya.

Berikut adalah contoh teks yang sudah melalui proses *stemming*.

sprinter tercepat **dunia kelas junior lalu** muhamad zohri **turut prihatin dengan musibah gempa bumi yang terjadi pulau lombok yang** berkekuatan **skala richter**

pada juli keprihatinan itu dia wujudkan dengan memberi bantuan bagi korban gempa bumi

sprinter **cepat** dunia kelas junior lalu muhamad zohri turut prihatin dengan musibah gempa bumi yang **jadi** pulau lombok yang **kuat** skala richter pada juli **prihatin** itu dia **wujud** dengan **beri** bantu bagi korban gempa bumi

5.1.4. Metode N-Gram

Kata yang di masukkan setelah melalui *stemming* akan di pecah kembali dengan menggunakan *N-Gram* sehingga menjadi dua karakter untuk setiap *string*-nya atau bisa disebut juga dengan *Bigram*.

Kata hasil *N-Gram* akan dicari tingkat kemiripannya dengan kata dasar yang ada di *database* yang juga sudah melalui proses *N-Gram*. Dari semua proses pengecekan tingkat kemiripan, jika tidak ditemukan yang cocok atau nilai tingkat kemiripannya 1, maka akan diambil kata yang tingkat kemiripannya sama dengan atau lebih dari 70% sebagai kata pengganti. Jika semua tingkat kemiripan dibawah 70% maka kata masukkan tersebut akan dianggap sebagai kata asing (bukan Bahasa Indonesia) karena kata tersebut tidak ada di *database* kata dasar KBBI.

5.2 Sistem Kerja Aplikasi

Dimulai dengan memasukkan teks kedalam kotak teks. Kemudian *user* mengklik tombol proses. Proses pertama kali yang diproses oleh sistem aplikasi yaitu proses tokenisasi. Data hasil tokenisasi kemudian mengalami proses algoritma *Boyer Moore* yaitu proses pencocokan awal setiap kata masukan dengan data kata dasar yang ada di *database* KBBI.

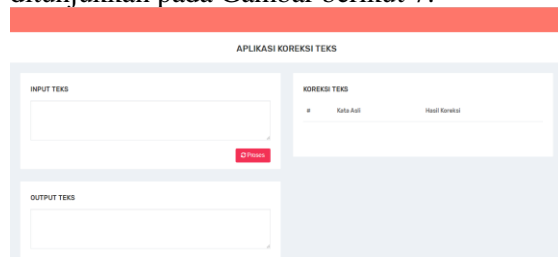
Selanjutnya untuk kata yang tidak ditemukan kecocokan dengan seluruh kata yang ada di *database*, setiap kata tersebut akan mengalami proses pemecahan kata dasar yaitu proses penghapusan imbuhan yang terdapat pada kata tersebut. Proses pemecahan kata yang berimbuhan ini disebut sebagai proses *Stemming*.

Setelah melalui proses *stemming* maka didapatlah kata dasar dari kata masukan yang berimbuhan. Kata tersebut kemudian dicocokkan kembali dengan *database* kata dasar. Jika ada kata yang masih belum

menemukan kecocokan dengan kata yang ada di *database*, maka kemudian proses akan dilanjutkan dengan proses perhitungan tingkat kemiripan kata dengan kata dasar yang ada di *database* menggunakan metode *N-Gram*.

5.3 Tampilan Aplikasi

Dalam penelitian ini sistem dibuat menggunakan Bahasa pemrograman PHP yang berbasis *web*. Aplikasi hanya memiliki satu tampilan utama. Tampilan utama aplikasi ditunjukkan pada Gambar berikut 7.



Gambar 7 Tampilan Aplikasi

5.4 Pengujian Aplikasi

Pengujian dilakukan dengan mengkonversi sebuah teks dari format pdf menjadi doc kemudian teks hasil konversi tersebut dimasukkan ke aplikasi untuk di dirapikan penulisannya.

1. Pengujian Data 1



Gambar 8 Hasil Scan Data Uji 1

Gambar 8 adalah teks hasil pemindaian visual yang sudah dalam bentuk pdf. Teks tersebut merupakan isi dari sebuah buku yang berjudul "Nur Muhammad Menyingkap Asal Mula Kejadian Makhluk".

Setelah melalui proses pengecekan melalui aplikasi, hasil teks menjadi satu halaman. Dari hasil aplikasi didapat data-data sebagai berikut.

Jumlah kata sebelum proses : 439 Kata
 Jumlah kata setelah proses : 471 Kata
 Waktu proses : 72 Detik
 Jumlah kata terdeteksi salah : 47 Kata

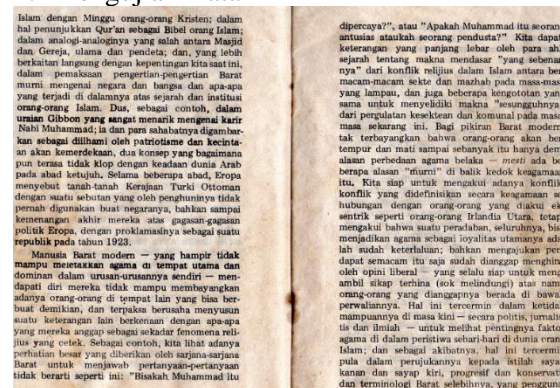
Jumlah kata asing : 34 Kata
 Jumlah kata yang dikoreksi : 13 Kata

Tabel 3 Kata Hasil Koreksi Pengujian Data 1

Kata Asli	Hasil Penyuntingan
sesunquhnya	sesungguhnya
smpat	empat
rmakan	makan
kecuall	kecuali
mustaqim	mustakim
rnakhluk	makhluk
baligh	balig
rmangkok	rangkok
hujjan	hujan
bintang	bintan
bintang	bintan
ramadhan	ramadan
zabarjut	zabarjad

Tabel 3 merupakan table yang berisi kata-kata yang salah penulisannya dan kata-kata hasil koreksi aplikasi. Berdasarkan hasil pengujian dari 439 kata yang diproses, 34 kata diantaranya dideteksi sebagai kata asing dan 13 kata yang salah pengetikannya.

2. Pengujian Data 2



Gambar 9 Hasil Scan Data Uji 2

Gambar 9 adalah teks hasil pemindaian visual yang sudah dalam bentuk pdf. Teks tersebut merupakan isi dari sebuah buku yang berjudul "Kebangkitan Islam di Mata Seorang Sarjana Barat". Selanjutnya file pdf tersebut dikonversi ke bentuk doc.

Setelah melalui proses pengecekan melalui aplikasi, hasil teks menjadi satu halaman. Dari hasil aplikasi didapat data-data sebagai berikut.

Jumlah kata sebelum proses : 442 Kata
 Jumlah kata setelah proses : 454 Kata
 Waktu proses : 72 Detik
 Jumlah kata terdeteksi salah : 32 Kata
 Jumlah kata asing : 25 Kata

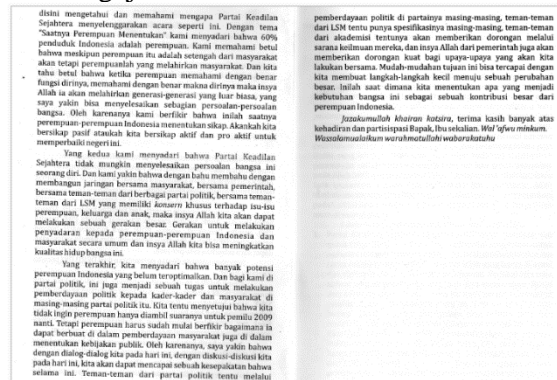
Jumlah kata yang dikoreksi : 7 Kata

Tabel 4 Kata Hasil Koreksi Pengujian Data 2

Kata Asli	Hasil Penyuntingan
Sangat	sangat
Meletakkan	meletakkan
Rnenyusun	menyusun
Mreka	reka
Relijius	religius
Loyalitas	loyalitas
Epini	opini

Tabel 4 merupakan hasil dari proses pengoreksian kata yang terdapat kesalahan pengejaannya. Berdasarkan hasil pengujian dari 442 kata yang diproses, dengan algoritma *boyer moore* aplikasi dapat mendeteksi sebanyak 32 kata yang tidak sesuai dengan *database* KBBI. Dari 32 kata, 25 kata diantaranya dideteksi sebagai kata asing dan 7 kata yang salah penyetikannya.

3. Pengujian Data 3



Gambar 10 Hasil Scan Data Uji 3

Gambar 10 adalah teks hasil pemindaian visual yang sudah dalam bentuk pdf. Teks tersebut merupakan isi dari sebuah buku yang berjudul "Political Empowerment of Woman". Selanjutnya *file* pdf tersebut dikonversi ke bentuk doc.

Setelah melalui proses pengecekan melalui aplikasi, hasil teks menjadi satu halaman. Dari hasil aplikasi didapat data-data sebagai berikut.

Jumlah kata sebelum proses : 377 Kata
 Jumlah kata setelah proses : 395 Kata
 Waktu proses : 66 Detik
 Jumlah kata terdeteksi salah : 20 Kata
 Jumlah kata asing : 14 Kata
 Jumlah kata yang dikoreksi : 6 Kata

Tabel 5 Kata Hasil Koreksi Pengujian Data 3

Kata Asli	Hasil Penyuntingan
berfikir	berpikir
konsern	konser
inenyetujui	menyetujui
berfikir	berpikir
kesepakatan	kesepakatan
partisipasi	partisipasi

Tabel 5 merupakan daftar-daftar kata-kata yang terdeteksi terdapat kesalahan ejaan. Berdasarkan hasil pengujian dari 377 kata yang diproses, dengan algoritma *boyer moore*, aplikasi dapat mendeteksi dapat mendeteksi sebanyak 20 kata yang tidak sesuai dengan KBBI. Dari 20 kata tersebut terdapat 14 kata asing dan 6 kata yang salah penyetikannya. 5 dari 6 kata dikoreksi dengan benar menggunakan metode N-Gram.

6. PENUTUP

6.1 Kesimpulan

Dari hasil uji coba aplikasi berjalan dengan baik dengan kesimpulan sebagai berikut:

1. Penerapan Algoritma Boyer Moore pada aplikasi untuk mencari kata yang salah dalam sebuah teks mendapatkan keberhasilan 100%. Kata-kata pada teks di bandingkan dengan kata-kata dalam database dan dengan Algoritma Boyer Moore kata yang tidak sesuai dengan database berhasil dikenali dan dianggap sebagai kata yang salah.
2. Penerapan Metode N-Gram pada aplikasi pengoreksi teks, dari hasil pengujian mendapatkan rata-rata tingkat keberhasilan mencapai 89,68%. Masih ada beberapa kata yang diperbaiki tidak sesuai dengan kata yang seharusnya karena perhitungan dengan Metode N-Gram mendapatkan hasil yang lebih tinggi dengan kata lain pada database.

6.2 Saran

Saran untuk perbaikan dan pengembangan dari tugas akhir ini adalah :

1. Aplikasi dapat ditambahkan fitur untuk menambah database kata dasar sehingga jika ada kata baru atau kata khusus yang tidak terdapat dalam database KBBI dapat langsung ditambahkan pada saat pengoreksian ulang.

2. Sebagai pengembangan kedepan dapat pula aplikasi ini dibuat kedalam bentuk sistem aplikasi berbasis android, sehingga dapat memudahkan pengoreksiandengan menggunakan smartphone android dimanapun dan kapanpun.

<http://faizalabbassulaiman.blogspot.com/2015/10/xampp.html>.

- [11] Abasawatawalla, "Pengertian Dan Penjelasan Penyuntingan Naskah," 2012. [Online]. Available: <http://abasawatawalla01.blogspot.com/2012/12/penyuntingan-naskah.html>.

7. DAFTAR PUSTAKA

- [1] A. Ibrahim, "Pengertian Literatur dan Jenis-jenisnya," 2015. [Online]. Available: <https://pengertiandefinisi.com/pengertian-literatur-dan-jenis-jenisnya/>.
- [2] F. R. Dwi, "Aplikasi Editor Pemerikas Ejaan dan Rekomendasi Kata Dalam Bahasa Indonesia Berbasis Android," *Petra Christian University*, p. 6, 2011.
- [3] D. R. Priyana, "PROGRAM APLIKASI EDITOR KATA BAHASA INDONESIA MENGGUNAKAN METODE APPROXIMATE STRING MATCHING DENGAN ALGORITMA LEVENSHTAIN DISTANCE BERBASIS JAVA," *Program Studi Teknik Informatika Universitas Brawijaya*, p. 10, 2013.
- [4] H. Agung, "Implementasi Booyer-Moore Pada Aplikasi Pencarian Rumus Matematika Dan Fisika," *Program Studi Teknik Informatika Universitas Bunda Mulia*, p. 12, 2016.
- [5] Ritma, "Materi String Matching," 5 10 2016. [Online]. Available: <https://dailyritma.wordpress.com/2016/10/05/materi-string-matching/>.
- [6] S. M. Eza Rahmanita, "Pencarian String Menggunakan Algoritma Boyer Moore Pada Dokumen," *Jurnal Ilmian NERO Vol. 1 No. 1*, p. 12, 2014.
- [7] A. Hamzah, "Deteksi Bahasa Untuk Dokumen Teks Berbahasa Indonesia," *Seminar Nasional Informatika 2010 (semnasIF 2010)*, p. 9, 2010.
- [8] Yurindra, *Software Engineering*, Yogyakarta: Deepublish, 2017.
- [9] S. M. Teuku Radillah, *Visual Basic 6.0 Return*, Yogyakarta: Deepublish, 2018.
- [10] F. A. Sulaiman, "XAMPP," 2015. [Online]. Available: