Vol. 4, No. 2
November 2016
ISSN : 2338-1523
E-ISSN :2541-576X

PENGEMBANGAN APLIKASI PERBAIKAN KATA PADA DOKUMEN DENGAN MENERAPKAN METODE KNUTH MORRIS PRATT

Sandy Suwandana

Sekolah Tinggi Manajemen Informatika dan Komputer Gici
Komp. Batu Aji Centre Park Simpang Base Camp. Batam
Telp. (0778) 391333, Kode Pos 29439
e-mail: suwandanas@gmail.com

ABSTRAK

Hasil pengetikan laporan ataupun karangan yang berupa lembaran yang banyak pada file dokumen, tidak terlepas dari kesalahan-kesalahan dalam pengetikan kata. Dalam pemeriksaan untuk diperbaiki kesalahan hasil pengetikan kata tentu akan mengalami kesulitan. Hal ini akan memerlukan waktu yang tidak sedikit dan ketelitian yang lebih untuk memeriksa tiap-tiap kata yang ada. Untuk menemukan kata yang salah tersebut digunakan teknik string matching yang merupakan teknik pencarian dari sejumlah karakter yang disebut pattern dalam sejumlah besar teks. Dengan menerapkan algoritma Knuth Morris Pratt diharapkan hasil yang dicapai lebih cepat dan efisien dalam proses menemukan kata yang salah karena algoritma ini menggeser pattern dengan lebih cerdas yang meminimalkan jumlah perbandingan dari pattern terhadap teks.

Kata Kunci : Knuth Morris Pratt, KMP, String Matching, Perbaikan Kata

ABSTRACT

The results of typing a report or articles which a sheet that much on a document file, not apart from the mistakes in typing the word. In checking for repair errors in the results of typing words will certainly have difficulties. This will require substantial time and more accuracy to check every word there. To find the wrong word be used string matching techniques that are search techniques from amount of characters is called the pattern in a large amount of text. By applying the algorithm Knuth Morris Pratt expected results achieved more quickly and efficiently in the process of finding the wrong word because the algorithm is shifts the pattern more intelligently which minimizes the number of comparisons from pattern of the text.

Keywords: *Knuth Morris Pratt, KMP, String Matching, Word Repair*

PENDAHULUAN

Ketika mengetik pada sebuah komputer, misalnya dalam pembuatan laporan ataupun sebuah karangan tentunya tidak terlepas dari kesalahan-kesalahan pengetikan kata. Kesalahan-kesalahan yang terjadi sering diakibatkan karena faktor kebiasaan yang sering menyingkat kata ketika menulis menjadi kata yang tidak baku misalnya kata “yang” menjadi kata “yg”, kata “dengan” menjadi kata “dgn”. Adapun kesalahan-kesalahan lain yang terjadi adalah kata yang terbalik-balik susunan hurufnya misalnya pada kata “yang” terjadi kesalahan pengetikan menjadi “yagn”, kata “suatu” menjadi “suaut”, kata “sering” menjadi

ISSN : 2338-1523
E-ISSN :2541-576X

"serign", dan lain sebagainya. Kesalahan pada kata yang tidak baku misalnya kata "apotek" diketik menjadi "apotik", kata "aktivitas" diketik menjadi "aktifitas". Kesalahan lain yang berupa kekurangan huruf pada kata yang diketik, misalnya kata "komputer" menjadi "kompute" atau "komputr", kata "aplikasi" menjadi "aplikas" atau "apliksi" dan lain sebagainya. Kesalahan kekurangan huruf yang mana kata yang salah juga memiliki arti atau makna, misalnya kata "sedang" diketik menjadi "sedan", kata "siapa" diketik menjadi "siap" ataupun menjadi "sapa".

Hasil pengetikan laporan ataupun karangan yang berupa lembaran yang banyak, tentu menyulitkan untuk diperiksa dan diperbaiki dalam kesalahan pengetikan katanya. Hal ini akan memerlukan waktu yang tidak sedikit dan ketelitian yang lebih untuk memeriksa tiap-tiap kata yang ada pada hasil pengetikan laporan ataupun karangan.

Pengetikan kata yang mengalami kesalahan akan mempersulit dalam hal membaca hasil laporan ataupun karangan, terlebih lagi jika kesalahan yang dilakukan sangat banyak. Kata-kata yang salah dalam pengetikan tersebut tentu juga tidak sesuai dengan kata baku yang berlaku khususnya pada Bahasa Indonesia dan sudah seharusnya diperbaiki sesuai dengan Ejaan Yang Disempurnakan (EYD) di dalam Kamus Besar Bahasa Indonesia (KBBI).

Penelitian ini bertujuan untuk memecahkan permasalahan yang telah dirumuskan yaitu membangun suatu sistem yang mampu menyelesaikan perbaikan kata yang salah pada dokumen karena kesalahan kata yang berupa singkatan yang tidak baku dan berupa kata yang tidak baku.

Manfaat yang diharapkan dari penelitian ini adalah segala bentuk kesalahan ketik yang berupa singkatan yang tidak baku dan kata yang tidak baku pada sebuah penulisan dokumen dapat diselesaikan dengan cepat dan akurat.

LANDASAN TEORI

Pengembangan Sistem

Dalam membangun *software*, terdapat rangkaian langkah-langkah yang meliputi *communication*, *planning*, *modeling*, *constuction* dan *deployment* (Pressman, 2015).

1. *Communication*
   Sebelum pekerjaan teknis dapat dimulai, sangat penting untuk berkomunikasi dan berkolaborasi dengan stakeholder. Tujuannya adalah untuk memahami keinginan stakeholder dan mengumpulkan kebutuhan yang membantu mendefinisikan fitur software.
2. *Planning*
   Dalam membangun perangkat lunak (software) tentu terjadi aktivitas-aktivitas yang rumit sehingga dibutuhkan perencanaan yang dapat memandu aktivitas-aktivitas yang akan dilakukan. Perencanaan menggambarkan tugas-tugas atau kegiatan-kegiatan teknis yang akan dilakukan, resiko, sumber daya yang diperlukan dan jadwal kerja.
3. *Modeling*
   Pemodelan merupakan sketsa yang berguna untuk lebih memahami masalah dan bagaimana cara mengatasinya. Membuat model perlu dilakukan untuk lebih memahami kebutuhan perangkat lunak (software) dan desain yang dapat mencapai tujuan dari kebutuhan tersebut.
4. *Construction*
   Membangun apa yang sudah dirancang. Kegiatan ini menggabungkan peng-codingan (dalam bahasa pemrograman) dengan pengujian yang diperlukan untuk menemukan kesalahan pada kode.

ISSN : 2338-1523
E-ISSN :2541-576X

5. *Deployment*
   Kegiatan melakukan penyerahan perangkat lunak (baik yang sudah selesai ataupun hanya sebagian increment) ke stakeholder untuk mengevaluasi produk dan memberikan umpan balik berdasarkan evaluasi.

Model Pengembangan Sistem

Ada 2 (dua) pemodelan untuk menganalisis requirements (kebutuhan) yaitu analisis terstruktur dan pemodelan analisis yang disebut analisis berorientasi objek (object-oriented). Analisis terstruktur memandang data dan proses yang merubah data sebagai entitas yang terpisah. Objek data dimodelkan dengan cara mendefinisikan atribut dan hubungan mereka. Proses yang memanipulasi objek data dimodelkan dengan cara yang menunjukkan bagaimana mereka mengubah data sebagai aliran data objek melalui sistem. Sedangkan analisis berorientasi objek (object-oriented) berfokus pada definisi kelas dan cara di mana mereka berkolaborasi dengan satu sama lain untuk efek kebutuhan stakeholder.

String Matching

Teknik String Matching sudah sering digunakan untuk menemukan kecocokan kata ataupun proses untuk menemukan sebuah dokumen. Teknik ini membandingkan antara string yang dicari (pattern) dengan string lain yang biasa disebut string teksnya.

String dapat berupa kata, frase, atau kalimat. String Matching merupakan bagian penting dari sebuah proses pencarian string (String Searching) dalam sebuah dokumen. Hasil dari sebuah sebuah pencarian string dalam dokumen tergantung dari teknik dan cara pencocokan string yang digunakan.

String Matching merupakan suatu teknik yang digunakan untuk mencari atau menemukan string atau biasa disebut pattern dalam string lain yang jumlah karakternya sama dengan atau lebih besar dari string yang dicari.

Arumugam, et al (2011) mengatakan algoritma String Matching digunakan untuk menemukan kejadian dari pola P panjang M dalam teks T. Banyak teknik dan algoritma telah dirancang untuk memecahkan masalah ini. Algoritma ini memiliki banyak aplikasi termasuk pengambilan Informasi, query database, dan analisis urutan DNA dan protein. Pada dasarnya algoritma String Matching ini menggunakan pola untuk memindai teks. Ukuran teks diambil dari file sama dengan panjang pola. Pendekatan ini menyejajarkan ujung kiri pola dan teks. Kemudian memeriksa apakah pola terjadi dalam teks dan menggeser pola ke kanan. Ini mengulangi prosedur yang sama lagi dan lagi sampai ujung kanan dari pola pergi ke ujung kanan teks.

Algoritma Knuth Morris Pratt

Algoritma ini digagas pada tahun 1974 oleh Donald Knuth dan Vaughan Pratt, dan secara terpisah oleh James H.Morris. Ketiganya menerbitkan bersama-sama pada tahun 1977.

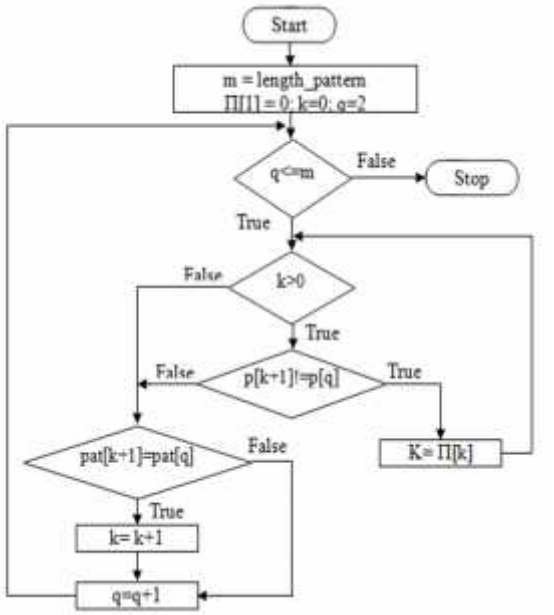
MShukla, et al (2014) juga mengatakan bahwa algoritma Knuth Morris Pratt (KMP) ini merupakan algoritma pencocokan pengurutan ke depan yang linear dengan memindai string dari kiri ke kanan yang menghitung pencocokan dalam dua bagian yaitu yang pertama dengan menghitung tabel KMP dan selanjutnya melakukan pencarian dengan urutan yang linear. Suthar, et al (2015) mengatakan, algoritma ini mengurangi waktu untuk mencari string dibandingkan dengan algoritma Brute Force. Algoritma ini menjamin bahwa string pencarian tidak akan membutuhkan lebih dari perbandingan karakter N. Algoritma

ISSN : 2338-1523
E-ISSN :2541-576X

Knuth Morris Pratt (KMP) memiliki 2 komponen yaitu (Jain et al, 2012):

1. Prefix Function (Π)

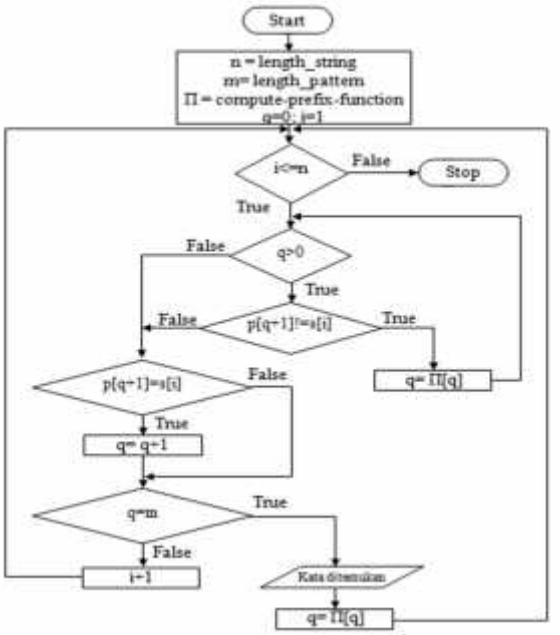
Fungsi ini merangkum pengetahuan tentang bagaimana pencocokan pola terhadap pergeseran pattern. Informasi ini dapat digunakan untuk menghindari pergeseran yang percuma dari pattern 'p'. Dengan kata lain berarti menghindari backtracking pada string 'S'. Tahapan Prefix function:



Gambar 1. Tahapan *Prefix Function*

2. KMP Matcher

Dengan pattern 'p', string 'S' dan prefix function 'Π' sebagai input, menemukan terjadinya 'p' pada 'S' dan mengembalikan jumlah pergeseran dari 'p' setelah ditemukan. Tahapan algoritma KMP-Matcher:



Gambar 2. Tahapan *KMP Matcher*

Kelebihan dan Kekurangan Algoritma KMP

Menurut Shukla, et al (2014), kelebihan dari algoritma Knuth Morris Pratt ini adalah:

1. KMP merupakan algoritma yang sederhana, cepat dan mudah untuk diimplementasikan.
2. Algoritma KMP sangat efektif dalam banyak deteksi sub DNA yang berurutan.
3. Penggunaan tabel KMP merupakan fitur yang bermanfaat besar pada algoritma ini. Tabel ini mengurangi perbandingan yang tidak perlu dan backtracking.

Sedangkan kelemahan atau kekurangan dari algoritma ini adalah sebagai berikut:

1. Saat melakukan percobaan diamati bahwa seberapa panjang kecocokan ditemukan, bahkan algoritma tidak ada hubungannya jika kecocokan tidak ditemukan.
2. Penggunaan tabel kecocokan parsial, dapat meningkatkan kompleksitas ruang untuk pattern yang relatif lebih panjang. Kompleksitas waktu dari kasus terburuk adalah O(m+n). Untuk menampung tabel

ISSN : 2338-1523
E-ISSN :2541-576X

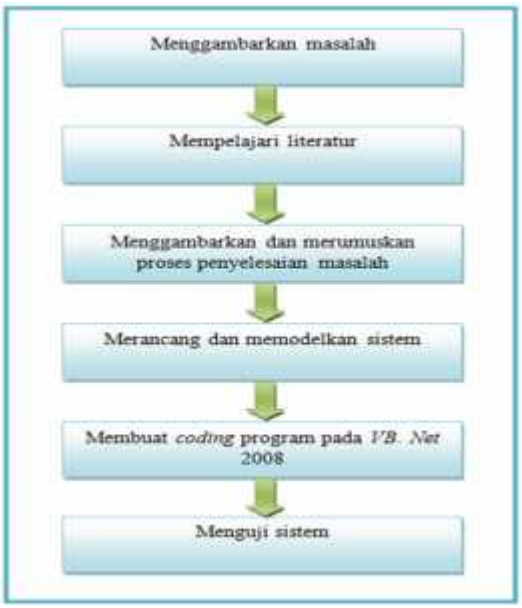
kecocokan parsial algoritma membutuhkan O(m) lebih banyak ruang.

METODOLOGI PENELITIAN

Metodologi penilitian ini memberikan gambaran langkah-langkah yang mencakup dari awal penelitian sampai dengan akhir penelitian. Langkah-langkah ini perlu disusun sebelum melakukan penelitian agar nantinya penelitian dapat terlaksana dengan terstruktur sehingga mendapatkan proses yang maksimal. Metodologi penelitian ini memuat tentang kerangka kerja penelitian yang berupa tahapan-tahapan yang harus dilalui selama melakukan penelitian.

Kerangka Kerja

Kerangka kerja dalam penelitian ini menggambarkan langkah-langkah yang harus dilewati. Kerangka kerja disusun agar penelitian yang dilakukan terlaksana dengan terstruktur dan jelas.



Gambar 3. Kerangka Kerja

ANALISIS DAN PERANCANGAN

Analis Penerapan Algoritma Knuth Morris Pratt

Algoritma ini digunakan untuk membantu dalam menemukan kata yang salah. Cara kerja aplikasi yang dirancang ini akan mampu menemukan kata-kata yang salah seperti singkatan kata yang tidak baku misalnya kata “dgn” dan penggunaan kata yang tidak baku misalnya kata “kwitansi” yang terdapat pada sebuah dokumen dan secara otomatis memperbaikinya sesuai dengan kata yang benar yang terdapat pada database.

Tabel 1. Pattern yang Dicari

No.	*Pattern*
1	AKTIFITAS
2	SDG

Teks = PEMILIK APOTEK SDG TIDAK BERAKTIFITAS
Pattern = AKTIFITAS
Penyelesaian:
Terlebih dahulu membuat tabel prefix pada pattern.

Indeks	1	2	3	4	5	6	7	8	9
P	A	K	T	I	F	I	T	A	S
Π									

Langkah pertama, M = jumlah_karakter dan Π[1] = 0.

Indeks	1	2	3	4	5	6	7	8	9
P	A	K	T	I	F	I	T	A	S
Π	0								

Selanjutnya bandingkan P[i+1] dengan P[j]. Jika terjadi kecocokan maka i=i+1.
Π[j] = i; j=j+1, di mana j≤M
i=0 dan j=2
P[1] ≠ P[2] maka Π[2]= 0

Indeks	1	2	3	4	5	6	7	8	9
P	A	K	T	I	F	I	T	A	S
Π	0	0							

ISSN : 2338-1523
E-ISSN :2541-576X

i=0 dan j=3
P[1] ≠ P[3] maka Π[3]= 0

Indeks	1	2	3	4	5	6	7	8	9
P	A	K	T	I	F	I	T	A	S
Π	0	0	0						

i=0 dan j=4
P[1] ≠ P[4] maka Π[4]= 0

Indeks	1	2	3	4	5	6	7	8	9
P	A	K	T	I	F	I	T	A	S
Π	0	0	0	0					

i=0 dan j=5
P[1] ≠ P[5] maka Π[5]= 0

Indeks	1	2	3	4	5	6	7	8	9
P	A	K	T	I	F	I	T	A	S
Π	0	0	0	0	0				

i=0 dan j=6
P[1] ≠ P[6] maka Π[6]= 0

Indeks	1	2	3	4	5	6	7	8	9
P	A	K	T	I	F	I	T	A	S
Π	0	0	0	0	0	0			

i=0 dan j=7
P[1] ≠ P[7] maka Π[7]= 0

Indeks	1	2	3	4	5	6	7	8	9
P	A	K	T	I	F	I	T	A	S
Π	0	0	0	0	0	0	0		

i=0 dan j=8
P[1] = P[8] maka i=i+1 dan Π[8]= 1

Indeks	1	2	3	4	5	6	7	8	9
P	A	K	T	I	F	I	T	A	S
Π	0	0	0	0	0	0	0	1	

i=1 dan j=9
P[2] ≠ P[9] maka i= Π[i] → i= Π[1] → i= 0 kemudian bandingkan kembali, P[1] ≠ P[9] maka Π[9]= 0

Indeks	1	2	3	4	5	6	7	8	9
P	A	K	T	I	F	I	T	A	S
Π	0	0	0	0	0	0	0	1	0

Setelah nilai Π tabel prefix pada pattern diketahui maka tahapan selanjutnya adalah mencocokkan pattern dengan teks. Tahap ini merupakan tahap kedua dari algoritma Knuth Morris Pratt (KMP) yaitu tahap KMP Matcher.

Langkah 1:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern	A	K	T	I	F	I	T	A	S														
Π	0	0	0	0	0	0	0	1	0														

Pada langkah 1, Pattern[1] tidak sama dengan Teks[1] kemudian dilakukan pergeseran pada pattern. Karena nilai dari Π[1] = 0, maka pergeseran dilakukan sebanyak satu karakter ke kanan.

Langkah 2:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern		A	K	T	I	F	I	T	A	S													
Π		0	0	0	0	0	0	0	1	0													

Pada langkah 2 juga tidak terjadi kecocokan. Pattern[2] tidak sama dengan teks[2], kemudian dilakukan pergeseran pattern kembali sebanyak satu karakter ke kanan.

Langkah 3:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern			A	K	T	I	F	I	T	A	S												
Π			0	0	0	0	0	0	1	0													

Langkah 4:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern				A	K	T	I	F	I	T	A	S											
Π				0	0	0	0	0	0	1	0												

Langkah 5:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern					A	K	T	I	F	I	T	A	S										
Π					0	0	0	0	0	0	1	0											

Langkah 6:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern						A	K	T	I	F	I	T	A	S									
Π						0	0	0	0	0	0	1	0										

ISSN : 2338-1523
E-ISSN :2541-576X

Pada langkah 6 terjadi kecocokan yaitu pada indeks ke-6 antara pattern “A” dengan teks “A”, maka tidak dilakukan pergeseran pattern. Kemudian dilakukan perbandingan kembali ke indeks selanjutnya

Langkah 7:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern						A	K	T	I	F	I	T	A	S									
Π						0	0	0	0	0	0	1	0										

Pada langkah 7 tidak terjadi kecocokan. Karena pada langkah sebelumnya sudah terjadi kecocokan karakter, maka kemudian cek nilai Π dari indeks sebelumnya. Karena nilai Π dari Pattern “A” adalah 0, maka untuk perbandingan selanjutnya adalah karakter teks dari index yang sama yaitu indeks ke 7 dengan karakter pertama dari pattern. Jadi pattern bergeser sebanyak 1 karakter ke kanan.

Langkah 8:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern							A	K	T	I	F	I	T	A	S									
Π							0	0	0	0	0	0	0	1	0								

Pada langkah 8 tidak terjadi kecocokan, kemudian dilakukan pergeseran pattern kembali sebanyak satu karakter ke kanan.

Langkah 9:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern								A	K	T	I	F	I	T	A	S							
Π								0	0	0	0	0	0	1	0								

Langkah 10:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern									A	K	T	I	F	I	T	A	S						
Π									0	0	0	0	0	0	1	0							

Langkah 11:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern										A	K	T	I	F	I	T	A	S					
Π										0	0	0	0	0	0	1	0						

Langkah 12:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern											A	K	T	I	F	I	T	A	S				
Π											0	0	0	0	0	0	1	0					

Langkah 13:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern												A	K	T	I	F	I	T	A	S			
Π												0	0	0	0	0	0	0	1	0			

Langkah 14:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern													A	K	T	I	F	I	T	A	S		
Π													0	0	0	0	0	0	1	0			

Langkah 15:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern														A	K	T	I	F	I	T	A	S	
Π														0	0	0	0	0	0	1	0		

Langkah 16:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern															A	K	T	I	F	I	T	A	S
Π															0	0	0	0	0	0	1	0	

Pada langkah 16 terjadi kecocokan yaitu pada indeks ke-15, maka tidak dilakukan pergeseran pada pattern. Kemudian dilakukan perbandingan kembali ke indeks selanjutnya.

Langkah 17:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern															A	K	T	I	F	I	T	A	S
Π															0	0	0	0	0	0	1	0	

Pada langkah 17 terjadi kecocokan kembali, maka tidak dilakukan pergeseran pattern. Kemudian dilakukan perbandingan kembali ke indeks selanjutnya.

Langkah 18:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern															A	K	T	I	F	I	T	A	S
Π															0	0	0	0	0	0	1	0	

Langkah 19:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern															A	K	T	I	F	I	T	A	S
Π															0	0	0	0	0	0	1	0	

Langkah 20:

ISSN : 2338-1523
E-ISSN :2541-576X

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern															A	K	T	I	F	I	T	A	S
Π															0	0	0	0	0	0	0	1	0

Langkah 21:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern															A	K	T	I	F	I	T	A	S
Π															0	0	0	0	0	0	0	1	0

Langkah 22:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern															A	K	T	I	F	I	T	A	S
Π															0	0	0	0	0	0	0	1	0

Langkah 23:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern															A	K	T	I	F	I	T	A	S
Π															0	0	0	0	0	0	0	1	0

Langkah 24:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern															A	K	T	I	F	I	T	A	S
Π															0	0	0	0	0	0	0	1	0

Pada langkah 24 terjadi kecocokan karakter kembali dan semua karakter pada pattern telah cocok. Jadi pada langkah 32 telah ditemukan kata yang dicari yaitu kata yang salah pada teks dan selanjutnya kata ini akan ditampilkan ke daftar kata yang salah oleh sistem yang nantinya akan diganti dengan kata yang benar yaitu "AKTIVITAS".

Kemudian dilanjutkan dengan pattern berikutnya yaitu "SDG". Karena melakukan pencarian dengan pattern baru maka tahap algoritma KMP digunakan kembali dari tahap awal. Sama seperti sebelumnya, terlebih dahulu mencari nilai Π pada pattern.

Indeks	1	2	3
P	S	D	G
Π			

M=jumlah_karakter
Π[1] = 0

Indeks	1	2	3
P	S	D	G
Π	0		

Selanjutnya bandingkan P[i+1] dengan P[j]. Jika terjadi kecocokan maka i=i+1.
Π[j] = i; j=j+1, di mana j≤M
i=0 dan j=2
P[1] ≠ P[2] maka Π[2]= 0

Indeks	1	2	3
P	S	D	G
Π	0	0	

i=0 dan j=3
P[1] ≠ P[3] maka Π[3]= 0

Indeks	1	2	3
P	S	D	G
Π	0	0	0

Setelah nilai Π tabel prefix pada pattern diketahui maka tahapan selanjutnya adalah KMP Matcher yaitu mencocokkan pattern dengan teks.

Langkah 1:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern	S	D	G																				
Π	0	0	0																				

Langkah 2:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern		S	D	G																			
Π		0	0	0																			

Langkah 3:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern			S	D	G																		
Π			0	0	0																		

Langkah 4:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern				S	D	G																	
Π				0	0	0																	

Langkah 5:

ISSN : 2338-1523
E-ISSN :2541-576X

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern					S	D	G																
Π					0	0	0																

Langkah 6:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern						S	D	G															
Π						0	0	0															

Langkah 7:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern							S	D	G														
Π							0	0	0														

Langkah 8:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern								S	D	G													
Π								0	0	0													

Pada langkah 8 terjadi kecocokan yaitu pada indeks ke-8, maka tidak dilakukan pergeseran pada pattern. Kemudian dilakukan perbandingan kembali ke indeks selanjutnya.

Langkah 9:

Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern								S	D	G													
Π								0	0	0													

Pada langkah 9 terjadi kecocokan kembali, maka tidak dilakukan pergeseran pattern. Kemudian dilakukan perbandingan kembali ke indeks selanjutnya.

Langkah 10:

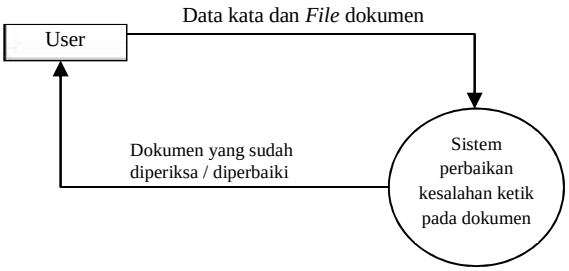
Indeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Teks	M	E	R	E	K	A		S	D	G		B	E	R	A	K	T	I	F	I	T	A	S
Pattern								S	D	G													
Π								0	0	0													

Pada langkah 10 terjadi kecocokan kembali dan semua karakter pada pattern telah cocok. Jadi pada langkah 10 telah ditemukan kata yang dicari yaitu “SDG” yang merupakan kata yang salah pada teks dan selanjutnya kata ini akan ditampilkan ke daftar kata yang salah oleh sistem.

Setelah itu proses KMP Matcher untuk pencarian pattern “SDG” dapat dihentikan karena sudah ditemukan. Hal ini dikarenakan sistem hanya cukup menemukan satu dan apabila ada kata “SDG” lagi pada dokumen, maka kata yang ditemukan ini sudah cukup mewakili kata “SDG” yang lainnya untuk kemudian diperbaiki dengan kata yang benar.

Perancangan Model Sistem

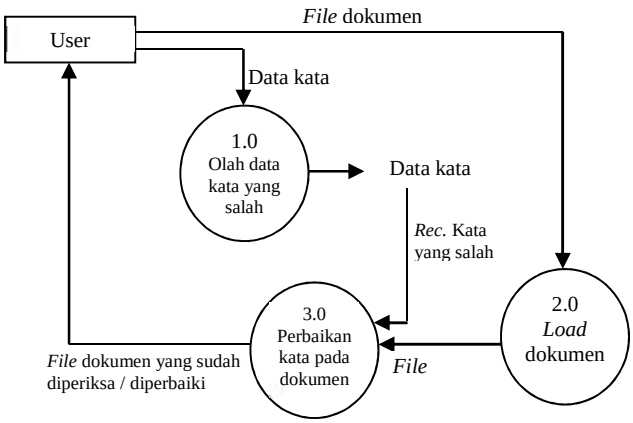
1. Data Flow Diagram
   a. Level Konteks



Gambar 4. DFD Level Konteks Sistem yang Dirancang

b. Level 0

ISSN : 2338-1523
E-ISSN :2541-576X



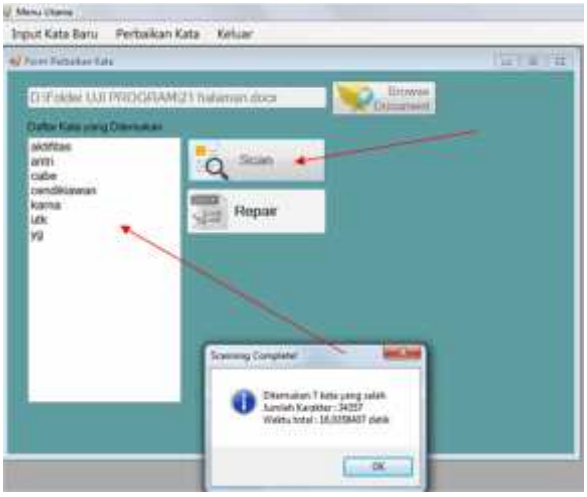
Gambar 5. DFD Level 0 Sistem yang Dirancang

IMPLEMENTASI

Implementasi sistem dilakukan setelah tahap analisis dan perancangan dilakukan.

Gambar 6. Memilih File Dokumen

Gambar 7. Setelah Memilih File Dokumen



Gambar 8. Mencari Kata yang Salah

Gambar 9. Memperbaiki Kata yang Salah

KESIMPULAN

1. Sistem yang dibangun dapat menemukan kata yang salah yang berupa kata yang tidak baku dan berupa singkatan dan juga dapat memberikan perbaikan terhadap kata-kata yang salah.
2. Penerapan algoritma *Knuth Morris Pratt* dapat menyelesaikan masalah yang terjadi untuk menemukan kesalahan penulisan kata yang berupa kata yang tidak baku dan berupa singkatan untuk kemudian diperbaiki oleh sistem.

DAFTAR PUSTAKA

ISSN : 2338-1523
E-ISSN :2541-576X

Arumugam. et al, 2011 "Text Analyzer", International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Vol. 1(1).

Buulolo Efori, 2013, "Implementasi Algoritma String Matching Dalam Pencarian Surat Dan Ayat Dalam Bible Berbasis Android", Pelita Informatika Budi Darma, Vol. 3.

Dewanti Cherly, 2016, "Analisa Jitu Soal-Soal UN 2016 Semua Jurusan SMK", Pustaka Ilmu Semesta.

Góngora. et al, 2012, "State of the Art for String Analysis and Pattern Search Using CPU and GPU Based Programming", Journal of Information Security (JIS), Vol 3: 314-318

Hussain. et al, 2013, "Improved Approach for Exact Pattern Matching (Bidirectional Exact Pattern Matching), IJCSI International Journal of Computer Science Issues, Vol. 10(1).

Jain. et al, 2012, "Comparative Study on Text Pattern Matching for Heterogeneous System", International Journal of Computer Science & Engineering Technology (IJCSET), Vol. 3(11)

Hutahaean Jeperson, 2015, "Konsep Sistem Informasi", Yogyakarta, Deepublish.

Prasad. et al, 2010, "Single Pattern Search Implementations in a Cluster Computing Environment", 4th IEEE International Conference on Digital Ecosystems and Technologies.

Pressman, 2015, "Software Engineering : A Practitioners Approach, 8th Edition", Raghu Srinivasan.

Pressman, 2001, "Software Engineering A : Practitioners Approach, 5th Edition", Thomas Casson.

Shukla. et al, 2014, "An Analysis on three Influential DNA Sequencing Algorithms", International Journal of Application or Innovation in Engineering & Management (IJAIEM), Vol. 1(3).

Suthar. et al, 2015, "A Survey Paper on String Matching", International Journal for Scientific Research & Development (IJSRD), Vol. 3(5)