Jurnal Coding, Sistem Komputer Untan
Volume 04, No.1 (2016), hal. 33-44 ISSN : 2338-493x

APLIKASI PENDETEKSI PLAGIAT TERHADAP KARYA TULIS BERBASIS WEB MENGGUNAKAN *NATURAL LANGUAGE PROCESSING* DAN ALGORITMA *KNUTH-MORRIS-PRATT*

[1]Rio Alamanda, [2]Cucu Suhery, [3]Yulrio Brianorman
[1][2][3]Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak
Telp./Fax.: (0561) 577963
e-mail:
[1]rioalamanda@gmail.com, [2]csuhery@gmail.com,
[3]rionorman@gmail.com

Abstrak

Dalam bidang akademik, salah satu syarat kelulusan untuk mendapat gelar akademik adalah membuat karya tulis, seperti: skripsi, thesis dan disertasi. Akan tetapi tidak jarang, karya tulis yang dihasilkan tersebut adalah hasil plagiat. Plagiat adalah penjiplakan atau pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seolah-olah karangan dan pendapat sendiri. Untuk pengecekan manual dengan jumlah tersebut membutuhkan waktu lama, oleh karena itu dibutuhkan aplikasi yang dapat membantu proses pengecekan agar lebih efisien. Tujuan dari penelitian ini adalah membuat aplikasi berbasis web yang dapat mendeteksi plagiat terhadap karya tulis menggunakan Natural Language Processing dan Algoritma Knuth–Morris-Pratt. Sistem pendeteksian kemiripan jurnal dimulai dari algoritma perangkuman, yaitu TF/IDF. Kemudian proses dilanjutkan pada proses Tokenizing, Filtering dan Stemming. Selanjutnya dilanjutkan dengan Algoritma Knuth-Morris-Pratt dan proses menghitung persentase kemiripan dengan Cosine Similarity. Dari hasil pengujian yang telah dilakukan, rata-rata persentase kemiripan yang dihasilkan dari pendeteksian tanpa menggunakan proses TF-IDF sebesar 45,98%, nilai tersebut lebih rendah dibandingkan dengan pendeteksian menggunakan proses TF-IDF, Tokenizing, Filtering dan Stemming yang menghasilkan persentase kemiripan 41,09%. Sedangkan tanpa menggunakan proses Stemming yaitu 40,58% serta tanpa menggunakan proses TF-IDF dan Stemming sebesar 40,54%.

Kata kunci: Karya tulis, *Knuth-Morris-Pratt*, *Natural Language Processing*, Plagiat

1. PENDAHULUAN

Karya tulis dapat diartikan sebagai sebuah hasil olah pikir manusia yang dituangkan ke dalam tulisan. Karya tulis dibagi menjadi 2 jenis, yaitu karya tulis fiksi dan karya tulis non fiksi atau ilmiah. Karya tulis fiksi adalah karya tulis yang merupakan rekayasa, khayalan, atau hasil imajinasi manusia.

Dalam bidang akademik, salah satu syarat kelulusan untuk mendapat gelar akademik adalah membuat karya tulis, seperti: skripsi, thesis dan disertasi. Akan tetapi, tidak sedikit mahasiswa yang karya tulisnya adalah hasil plagiat (pencontekan).

Plagiat adalah penjiplakan atau pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seolah-olah karangan dan pendapat sendiri.

Jumlah mahasiswa S1 di Indonesia kurang lebih 3.630.375 orang [1], maka ada sekitar 3.630.375 pula hasil karya ilmiah yang dibuat oleh mahasiswa. Dari jumlah tersebut, maka kemungkinan besar ada beberapa hasil plagiat. Untuk pengecekan manual dengan jumlah tersebut membutuhkan waktu lama, oleh karena itu dibutuhkan aplikasi yang dapat membantu proses pengecekan agar lebih efisien.

Pada penelitian sebelumnya yang dilakukan oleh salah satu mahasiswa dari Program Studi Teknik Informatika Institut Teknnologi Bandung dengan judul penelitian "Penerapan *Pattern Matching* untuk Deteksi Plagiarisme" [2], pendeteksian plagiarisme ditujukan untuk tugas kuliah dengan penerapan Algoritma *Knuth-Morris-Pratt* (KMP) dan Algoritma *Boyer-Moore*. Proses pendeteksian dilakukan dengan membandingkan 2 buah tugas kuliah. Kesimpulannya adalah pendeteksian masih kurang efektif disebabkan oleh pencocokan string yang diimplementasikan pada kedua algoritma tersebut merupakan pencocokan *exact matching*. Artinya, *pattern* yang ada pada teks harus sama persis dengan pattern yang sedang dibandingkan. Sehingga bisa saja terjadi kesalahan perhitungan jumlah kemunculan untuk contoh uji kasus yang lebih rumit atau untuk kasus-kasus unik. Kemudian, input yang dapat dimasukkan oleh user masih terbatas yaitu untuk *file* dengan format txt saja.

Algoritma yang akan digunakan dalam aplikasi ini adalah Algoritma *Knuth-Morris-Pratt*. Berdasarkan permasalahan dari penelitian sebelumnya yang mengatakan bahwa pendeteksian masih kurang efektif karena pencocokan string dalam Algoritma *Knuth-Morris-Pratt* merupakan pencocokan *exact* matching, dapat diatasi dengan penerapan *Natural Language Processin* (NLP) dalam proses pendeteksian. Dengan penerapan NLP tersebut, dapat menjadikan isi dari setiap teks menjadi kata tanpa imbuhan atau kata serta menerapkan proses peringkasan isi teks dari sebuah karya tulis. Diharapkan dua proses tersebut dapat menambah tingkat keefektifan hasil kemiripan dan waktu yang dibutuhkan untuk melakukan proses pendeteksian. Kemudian input yang dapat dimasukkan oleh *user* tidak hanya file dengan format txt saja, tetapi juga format doc dan pdf. Tetapi, aplikasi yang dirancang tidak dapat membandingkan dokumen yang diproteksi dan hasil *scanning*.

Algoritma *Knuth-Morris-Pratt* (KMP) adalah salah satu algoritma pencarian *string* yang dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersamaVaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977.

Aplikasi ini dirancang berbasis *web*, agar lebih fleksibel dan dapat diakses kapan saja, dimana saja dan oleh siapa saja.

2. TINJAUAN PUSTAKA

2.1 Plagiarisme

Plagiat dapat dianggap sebagai tindak pidana karena mencuri hak cipta orang lain. Di dunia pendidikan, pelaku plagiarisme dapat mendapat hukuman berat seperti dikeluarkan dari sekolah/universitas. Pelaku plagiat disebut sebagai plagiator. Berikut adalah penjelasan mengenai definisi dan jenis plagiat.

Dalam buku Bahasa Indonesia: Sebuah Pengantar Penulisan Ilmiah [3], menggolongkan hal-hal berikut sebagai tindakan plagiarisme :

1. Mengakui tulisan orang lain sebagai tulisan sendiri,
2. Mengakui gagasan orang lain sebagai pemikiran sendiri
3. Mengakui temuan orang lain sebagai kepunyaan sendiri
4. Mengakui karya kelompok sebagai kepunyaan atau hasil sendiri,
5. Menyajikan tulisan yang sama dalam kesempatan yang berbeda tanpa menyebutkan asal-usulnya
6. Meringkas dan memparafrasekan (mengutip tak langsung) tanpa menyebutkan sumbernya, dan
7. Meringkas dan memparafrasekan dengan menyebut sumbernya, tetapi rangkaian kalimat dan pilihan katanya masih terlalu sama dengan sumbernya.

2.2 *Natural Language Processing* (NLP)

Natural Language Processing (NLP) adalah salah satu bidang ilmu komputer yang merupakan cabang dari kecerdasan buatan, dan bahasa (linguistik) yang berkaitan dengan interaksi antara komputer dan bahasa alami manusia, seperti bahasa

Indonesia atau bahasa Inggris. Tujuan utama dari studi NLP adalah membuat mesin yang mampu mengerti dan memahami makna bahasa manusia lalu memberikan respon yang sesuai.

2.3 Algoritma *Knuth-Morris-Pratt*

Algoritma Knuth-Morris-Pratt adalah salah satu algoritma pencarian string, dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977. Jika melihat algoritma Brute Force lebih mendalam, dapat diketahui bahwa dengan mengingat beberapa perbandingan yang dilakukan sebelumnya dapat meningkatkan besar pergeseran yang dilakukan. Hal ini akan menghemat perbandingan, yang selanjutnya akan meningkatkan kecepatan pencarian.

2.4 *Cosine Similarity*

Metode *Cosine Similarity* adalah metode untuk menghitung similaritas antara dua dokumen. Penentuan kesesuaian dokumen dengan *query* dipandang sebagai pengukuran (*similarity measure*) antara *vector document* (D) dengan *vector query* (Q). Semakin sama suatu vector dokumen dengan *vector query* maka dokumen dapat dipandang semakin sesuai dengan *query*. Rumus yang digunakan untuk menghitung cosine similarity adalah sebagai berikut:

$$cosSim(X, d_j) = \frac{\sum_{i=1}^{m} xi.\, d_{ji}}{\sqrt{\left(\sum_{i=1}^{m} x_i\right)^2}.\sqrt{\left(\sum_{i=1}^{m} xd_{ji}\right)^2}} \quad (1)$$

dimana X adalah dokumen uji, d_j adalah dokumen uji, x_i dan d_{ji} adalah nilai bobot yang diberikan pada setiap *term* pada dokumen. Kedekatan *query* dan dokumen diindikasikan dengan sudut yang dibentuk. Nilai *cosinus* yang cenderung besar mengindikasikan bahwa dokumen cenderung sesuai *query*. Dalam proses membandingkan dokumen yang sesuai dengan dokumen yang telah ada atau dokumen lainnya, maka digunakan perhitungan dengan rumus pada persamaan (1) untuk mengetahui angka similaritas dari dokumen tersebut.

Untuk menentukan jenis plagiarisme antara dokumen yang diuji ada 5 jenis penilaian persentase *similarity* [4] :

1. Hasil uji 0% berarti kedua dokumen tersebut benar-benar berbeda baik dari segi isi dan kalimat secara keseluruhan
2. Hasil uji kurang dari 15% berarti kedua dokumen tersebut hanya mempunyai sedikit kesamaan
3. Hasil uji 15% sampai 50% berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang
4. Hasil uji lebih dari 50% berarti dapat dikatakan bahwa dokumen tersebut mendekati plagiarism
5. Hasil uji 100% menandakan bahwa dokumen tersebut adalah plagiat karena dari awal sampai akhir mempunyai isi yg sama persis.

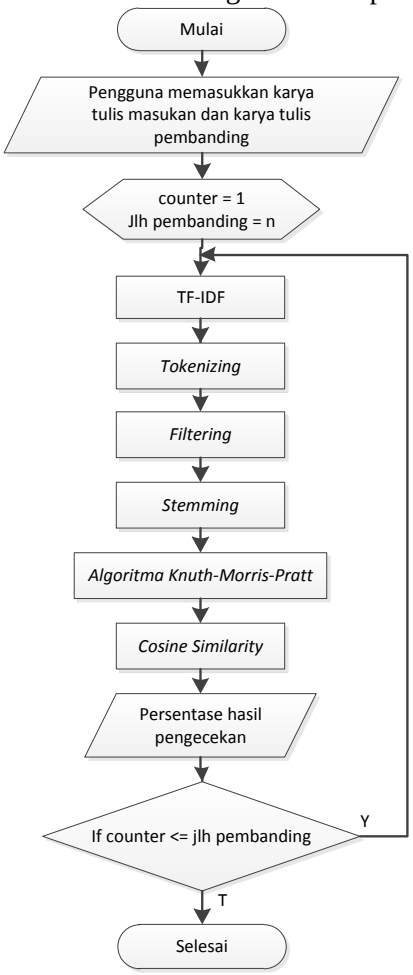
3. METODOLOGI PENELITIAN

Proses penelitian dimulai dengan melakukan studi literatur, yaitu mencari dan mempelajari literatur yang berhubungan dengan pembuatan tugas akhir, yaitu melalui *e-book*, artikel, tutorial serta jurnal yang berhubungan dengan tugas akhir. Kemudian melakukan analisis kebutuhan perangkat lunak. Selanjutnya dilakukan perancangan sistem yaitu merancang aplikasi secara keseluruhan dengan menggabungkan seluruh algoritma yang akan digunakan dan disesuaikan dengan kebutuhan. Perancangan sistem dimulai dari perancangan algoritma perangkuman dengan menggunakan cabang-cabang ilmu dari *Natural Language Processing* (NLP), yaitu TF/IDF, *tokenizing*, *filtering* dan *Stemming*. Setelah itu dilanjutkan dengan perancangan algoritma Knuth-Morris-Pratt. Setelah kedua algoritma tersebut berhasil, maka dilakukan penggabungan algoritma, sehingga akan menghasilkan sistem utuh yang sesuai dengan kebutuhan. Setelah melakukan perancangan sistem, dilanjutkan dengan melakukan pengujian untuk mengetahui kinerja sistem. Setelah dilakukan pengujian dilakukan analisa untuk mendapatkan kesimpulan akhir dari proses penelitian dan melakukan implementasi.

4. PERANCANGAN SISTEM

4.1 Perancangan Aplikasi

Dalam penelitian ini, perancangan aplikasi berisikan Diagram Alir Aplikasi.



Gambar 1. Diagram Alir Aplikasi Pendeteksian Plagiat

Berdasarkan Gambar 1, dapat dijelaskan bahwa proses pendeteksian plagiat terhadap karya tulis adalah sebagai berikut:

1. Mulai
   Pertama pengguna memasukkan karya tulis masukkan dan karya tulis pembanding kedalam aplikasi yang nantinya akan tersedia form untuk memasukkan kedua karya tulis tersebut. Kemudian pengguna menekan tombol deteksi untuk memulai melakukan proses pendeteksian.
2. TF-IDF
   Tahap pertama yang akan dilalui adalah TF-IDF. Pada proses ini isi teks dari karya tulis yang awalnya memiliki banyak sekali kalimat akan di rangkum menjadi beberapa kalimat saja untuk memaksimalkan waktu pendeteksian. Peringkasan dilakukan dengan cara menghitung setiap bobot kalimat dalam karya tulis. Untuk mendapatkan bobot kalimat dilakukan dengan cara menghitung bobot setiap kata dalam seluruh karya tulis. Selanjutnya bobot tiap kata dijumlahkan hingga menjadi bobot kalimat. Kalimat-kalimat dengan bobot tertinggi akan ditampilkan menjadi hasil ringkasan. Jumlah kalimat yang ditampilkan pada hasil ringkasan tergantung dengan persentase peringkasan yang ditentukan oleh pengguna. Sebagai nilai *default*, yaitu 50% yang artinya adalah setengah dari seluruh isi dari setiap karya tulis.
3. *Tokenizing*
   Pada tahap ini sistem akan memecah teks dari inputan berdasarkan pembatas tertentu yang terdapat dalam *string* tersebut. *Tokenizing* 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.
4. *Filtering*
   Pada tahap ini sistem akan mengambil kata-kata penting dari hasil tahap sebelumnya yaitu pada tahap *tokenizing*. Pada sistem yang dibuat saat ini akan menggunakan algoritma *stoplist* (membuang kata yang kurang penting). *Stoplist*/*stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*.
5. *Stemming*
   Tahap *Stemming* adalah tahap mencari root kata dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke kata dasar.
6. Algoritma *Knuth-Morris-Pratt*
   Pada tahap ini sistem akan mulai mencocokkan kesamaan antara dua karya tulis yang telah dimasukkan kedalam sistem untuk mengetahui seberapa besar kemiripannya. Perhitungan penggeseran pada algoritma ini adalah sebagai berikut, bila terjadi

ketidakcocokan pada saat pattern sejajar dengan *teks[i..i + n – 1]*, dapat dianggap bahwa ketidakcocokan pertama terjadi di antara *teks[I + j] dan pattern [j]*,dengan *0 < j < n*.berarti, *teks [i..i + j – 1] = pattern [0..j – 1] dan a = teks [I + j]*tidak sama dengan *b = pattern[j]*. Ketika proses menggeser, sangat beralasan bila ada sebuah awalan *u* dari *pattern* akan sama dengan sebagian akhiran *u* dari sebagian teks. Sehingga proses bisa menggeser *pattern* agar awalan *u* tersebut sejajar dengan akhiran dari *u*. Dengan kata lain, pencocokan *string* akan berjalan secara efesien bila mempunyai tabel yang menentukan berapa panjang pergeseran seandainya terdeteksi ketidakcocokan di karakter ke-*j* dari *pattern*. Tabel itu harus memuat *next[j]* yang merupakan posisi karakter *pattern[j]* setelah digeser, sehingga proses bisa menggeser *pattern* sebesar *j –next[j] relative* terhadap teks.

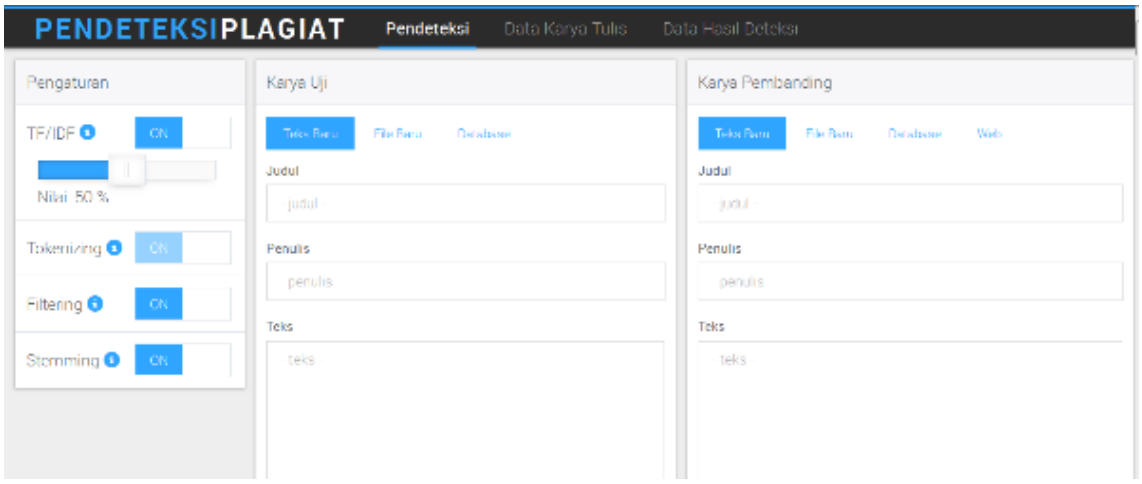
7. *Cosine Similarity*
   Metode *Cosine Similarity* adalah metode untuk menghitung similaritas antara dua dokumen. Penentuan kesesuaian dokumen dengan *query* dipandang sebagai pengukuran (*similarity measure*) antara *vector dokumen* (D) dengan *vector query* (Q). Semakin sama suatu *vector dokumen* dengan *vector query* maka dokumen dapat dipandang semakin sesuai dengan *query*.
8. Persentase hasil kemiripan
   Tahap terakhir adalah menampilkan nilai persentase kemiripan antara kedua karya ilmiah yang telah dimasukkan kedalam sistem.

5. HASIL DAN PEMBAHASAN

Implementasian sistem pendeteksi plagiat pada karya tulis ini dibangun menggunakan bahasa pemrograman PHP yang hasilnya berupa aplikasi berbasis *web*. Untuk penyimpanan data hasil pengujian dan beberapa data yang membantu dalam proses *filtering* dan proses *Stemming* menggunakan MySQL. Pengujian sistem akan dilakukan dengan mendeteksi kemiripan terhadap 10 karya tulis. Pengujian dilakukan dalam 4 tahap yaitu pertama, pendeteksian melalui proses TF-IDF, *Tokenizing*, *Filtering*, *Stemming*, Algoritma *Knuth-Morris-Pratt* dan *Cosine Similarity*. Kedua, pendeteksian tanpa melalui proses TF-IDF. Ketiga, pendeteksian tanpa melalui proses *Stemming* dan yang keempat, melakukan proses pendeteksian tanpa melalui proses TF-IDF dan *Stemming*.

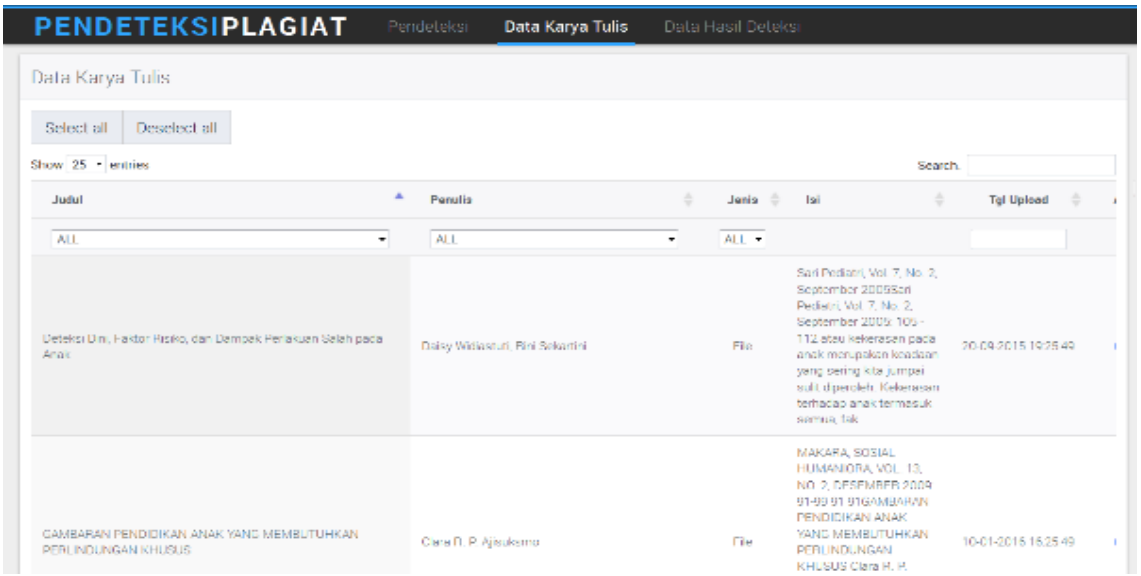
5.1 Tampilan Aplikasi

Dalam penelitian ini sistem dibuat dengan menggunakan bahasa pemrograman PHP yang berbasis web. Pada tampilan aplikasi terdiri dari 3 menu yaitu menu Pendeteksi, Data karya tulis dan Data hasil deteksi. Tampilan menu Pendeteksi pada aplikasi ditunjukkan pada Gambar 2 berikut.



Gambar 2. Tampilan Menu Pendeteksi Pada Aplikasi

Tampak pada Gambar 2, bagian kiri dari tampilan merupakan navigasi untuk pengaturan proses TF-IDF, *tokenizing*, *filtering* dan *Stemming*. Selanjutnya tampilan menu Data karya tulis dapat dilihat pada gambar berikut ini.



Gambar 3. Tampilan Menu Data Karya Tulis Pada Aplikasi

Pada menu data karya tulis memuat semua data karya tulis yang pernah di masukkan kedalam sistem. *User* dapat pula mengubah dan menghapus karya tulis yang telah disimpan dalam data karya tulis tersebut. Kemudian tampilan menu selanjutnya adalah Data Hasil Deteksi yang dapat dilihat pada gambar berikut.

Gambar 4. Tampilan Menu Data Hasil Deteksi Pada Aplikasi

Tampilan menu data hasil deteksi memuat pencatatan data hasil dari pendeteksian yang pernah dilakukan. User dapat melihat detail dari pendeteksian dan melakukan pengecekkan ulang untuk mendapatkan hasil persentase baru apabila data karya tulis pernah di ubah sebelumnya pada menu data karya tulis. Pencatatan hasil deteksi ini dapat pula dihapus oleh *user*.

5.2 Pengujian Aplikasi

Pengujian perangkat keras bertujuan untuk mengetahui apakah perangkat keras yang sebelumnya didesain dan dibuat dapat berjalan dengan lancar atau tidak, sehingga

dapat dianalisa kesalahan-kesalahan dalam proses pembuatan perangkat keras. Pengujian dilakukan dengan mengecek kemiripan dari 5 dokumen uji yang telah disiapkan sebelumnya. Kelima dokumen tersebut akan dicek nilai persentase kemiripannya antara dokumen karya tulis yang satu dengan dokumen karya tulis yang lainnya. Selain nilai persentasenya, akan dihitung pula lama waktu proses dari setiap pengecekkan kemiripan yang dilakukan oleh sistem. Dokumen 1 akan di cek kemiripannya dengan dokumen 1, dokumen 2, dokumen 3, dokumen 4 dan dokumen 5 kemudian dokumen 2 di cek kemiripannya dengan dokumen 1 dan seterusnya sampai kelima dokumen tersebut masing-masing dicek kemiripannya. Berikut ini adalah tabel dari hasil pengecekkan antar dokumen.

Tabel 1. Persentase Kemiripan antar Dokumen

Dok. Pembanding / Dok. Uji	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	% (menit:detik)									
1	100 (1:50)	38,51 (1:27)	28,16 (1:45)	41,08 (1:51)	27,8 (2:23)	31,33 (2:10)	16,56 (2:11)	48,83 (2:18)	52,2 (2:28)	36,71 (2:30)
2	38,51 (1:41)	100 (0:17)	46,71 (1:10)	56,88 (1:01	47,46 (0:57)	34,16 (1:11)	19,74 (1:18)	62,19 (1:19)	73,11 (1:2)	39,33 (0:59)
3	28,16 (2:53)	46,71 (1:8)	100 (2:25)	44,38 (2:12)	43,46 (1:49)	48,1 (2:30)	21,06 (2:39)	55,57 (2:45)	55,78 (2:58)	34,65 (2:3)
4	41,08 (2:58)	56,88 (2:33)	44,38 (2:35)	100 (2:45)	36,96 (2:39)	26,71 (2:45)	14,45 (2:43)	75,19 (2:55)	75,66 (2:58)	29,01 (2:55)
5	27,8 (2:45)	47,46 (2:33)	43,46 (2:49)	36,96 (2:51)	100 (1:8)	35,71 (2:21)	23,04 (2:22)	41,61 (2:21)	47,2 (2:49)	47,57 (2:54)
6	31,33 (2:33)	34,16 (2:35)	48,1 (2:22)	26,71 (2:45)	35,71 (2:49)	100 (2:1)	15,07 (2:50)	41,98 (2:56)	34,52 (2:59)	18,65 (2:54)
7	16,56 (2:53)	19,74 (2:45)	21,06 (2:55)	14,45 (2:42)	23,04 (2:50)	15,07 (2:21)	100 (2:59)	30,12 (2:53)	19,02 (2:51)	15,53 (2:59)
8	48,83 (2:59)	62,19 (2:58)	55,57 (2:45)	75,19 (2:51)	41,61 (2:49)	41,98 (2:55)	30,12 (2:52)	100 (2:12)	77,87 (3:1)	39,18 (3:1)
9	52,2 (2:48)	73,11 (2:44)	55,78 (2:35)	75,66 (2:56)	47,2 (2:39)	34,52 (2:41)	19,02(3:3)	77,87 (2:59)	100 (3:5)	50,24 (3:9)
10	36,71 (2:55)	39,33 (2:35)	34,65 (2:58)	29,01 (2:58)	47,57 (2:57)	18,65 (2:48)	15,53 (3:1)	39,18 (3:0)	50,24 (3:11)	100 (2:45)

Nilai persentase kemiripan yang paling kecil adalah nilai persentase saat mengecek kemiripan Dokumen 4 dengan Dokumen 7 yang menghasilkan persentase kemiripan sebesar 14,45 %. Dengan hasil persentase sebesar itu mengartikan bahwa kedua dokumen karya tulis yang di cek merupakan karya tulis yang memiliki sedikit kesamaan, ada beberapa isi dari kedua dokumen karya tulis yang sama menurut sistem pada saat melakukan pengecekkan. Nilai persentse kemiripan yang paling besar adalah nilai persentase saat melakukan pengecekkan kemiripan Dokumen 8 dan Dokumen 9 yang menghasilkan persentase kemiripan sebesar 77,87 % yang mana mengartikan bahwa kedua dokumen karya tulis tersebut mendekati plagiarisme, ada banyak isi yang sama dari kedua dokumen karya tulis saat melakukan pengecekkan. Nilai persentase kemiripan yang melebihi angka 50 % yang mana mengartikan bahwa jika nilai persentase sebesar itu menunjukkan dokumen karya tulis merujuk pada tindakan plagiat atau penjiplakan karya tulis. Rata-

rata hasil persentase dari setiap proses adalah 41,09 %. Lama waktu yang dibutuhkan oleh sistem dalam setiap proses tersebut bervariasi. Semakin banyak jumlah halaman dari dokumen karya tulis yang dicek maka semakin lama pula waktu yang dibutuhkan oleh sistem.

Disamping hasil persentase itu semua tidaklah mengartikan bahwa dokumen tersebut benar-benar plagiat tanpa dicek secara manual. Sistem ini digunakan seperti pada maanfaatnya yaitu sebagai bahan pertimbangan dalam menentukan plagiarisme serta membantu meningkatkan efisiensi waktu dalam pendeteksian kemiripan karya tulis.

Dilakukan pula pengujian sistem dengan melakukan pendeteksian tanpa melalui proses TF-IDF, melakukan pendeteksian tanpa proses *Stemming* dan melakukan pendeteksian tanpa proses TF-IDF dan proses *Stemming*. Berikut tabel hasil dari pengujian sistem yang dilakukan saat melakukan pendeteksian tanpa melalui proses TF-IDF.

Tabel 2. Persentase Kemiripan antar Dokumen Tanpa Proses TF-IDF

Dok. Pembanding / Dok. Uji	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	% (menit:detik)									
1	100 (2:21)	39,55 (1:53)	27,83 (3:26)	42,88 (3:8)	25,92 (3:2)	32,06 (3:5)	16,11 (3:0)	50,56 (3:15)	55,93 (3:20)	38,4 (2:59)
2	39,55 (1:58)	100 (0:20)	46,8 (2:21)	57,09 (2:29)	45,11 (2:13)	33,32 (2:56)	19,3 (2:49)	64,93 (3:0)	74,17 (2:57)	37,18 (2:59)
3	27,83 (3:49)	46,8 (1:29)	100 (1:20)	43,79 (2:12)	43,36 (2:33)	45,51 (2:41)	22,48 (2:42)	55,47 (3:1)	55,81 (3:1)	34,54 (3:10)
4	42,88 (3:1)	57,09 (1:45)	43,79 (1:47)	100 (2:0)	35,26 (2:20)	27,17 (2:59)	15,13 (2:51)	76,33 (2:53)	78,88 (2:55)	28,16 (3:19)
5	25,92 (3:8)	45,11 (1:49)	43,36 (1:51)	35,26 (2:2)	100 (2:1)	34,35 (2:55)	21,49 (2:39)	41,85 (2:58)	43,16 (3:3)	44,1 (3:12)
6	32,06 (3:11)	33,32 (2:0)	45,51 (2:19)	27,17 (2:13)	34,35 (2:32)	100 (3:1)	15,17 (2:49)	43,91 (3:1)	33,82 (3:11)	18,7 (3:13)
7	16,11 (3:25)	19,3 (2:1)	22,48 (2:21)	15,13 (2:25)	21,49 (2:42)	15,17 (3:6)	100 (2:40)	29,51 (3:12)	18,46 (3:17)	15,26 (3:21)
8	50,56 (3:39)	64,93 (2:11)	55,47 (2:24)	76,33 (2:29)	41,85 (2:35)	43,91 (3:11)	29,51 (2:59)	100 (2:48)	79,8 (3:21)	43,34 (2:51)
9	55,93 (3:33)	74,17 (2:25)	55,81 (2:39)	78,88 (2:34)	43,16 (2:45)	33,82 (3:17)	18,46 (2:53)	79,8 (3:0)	100 (2:48)	47,11 (3:1)
10	38,4 (3:32)	37,18 (2:34)	34,54 (2:38)	28,16 (2:38)	44,1 (2:59)	18,7 (3:29)	15,26 (3:0)	43,34 (3:12)	47,11 (3:11)	100 (2:51)

Nilai persentase kemiripan yang paling kecil adalah nilai persentase saat mengecek kemiripan Dokumen 4 dengan Dokumen 7 yang menghasilkan persentase kemiripan sebesar 15,13%. Hasil persentase kemiripan terkecil naik 0.68% dibanding dengan proses pendeteksian menggunakan TF-IDF. Nilai persentse kemiripan yang paling besar adalah nilai persentase saat melakukan pengecekkan kemiripan Dokumen 8 dan Dokumen 9 yang menghasilkan persentase kemiripan sebesar 79,8%. Hasil persentase kemiripan tertinggi berbeda dengan proses pendeteksian menggunakan TF-IDF dan persentasenya naik 1.93%. Rata-rata hasil persentase dari setiap proses adalah 45,98%. Rata-rata hasil persentase dari setiap proses naik 4,89% dibanding dengan proses pendeteksian menggunakan TF-IDF. Hasil persentase

kemiripan tanpa menggunakan proses TF-IDF berubah nilainya dibandingkan dengan proses menggunakan TF-IDF. Hasil persentase kemiripan tersebut berubah dikarenakan ketika tanpa proses TF-IDF yaitu menghapus beberapa kata-kata atau kalimat dan menyisakan kata-kata atau kalimat yang penting saja memiliki banyak kesamaan antar karya tulis mengakibatkan nilai persentase kemiripan menjadi naik begitu pula sebaliknya ketika tanpa proses TF-IDF memiliki banyak kata-kata atau kalimat yang berbeda antar karya tulis menjadikan nilai persentase kemiripan menurun. Beberapa waktu proses pendeteksian tanpa menggunakan TF-IDF berubah menjadi bertambah lebih lama karena jumlah kata menjadi lebih banyak dalam proses pendeteksian.

Berikut tabel hasil dari pengujian sistem yang dilakukan saat melakukan pendeteksian tanpa melalui proses *Stemming*.

Tabel 3. Persentase Kemiripan antar Dokumen Tanpa Proses *Stemming*

Dok. Pembanding / Dok. Uji	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	% (menit:detik)									
1	100 (1:42)	29,45 (1:25)	28,25 (2:55)	38,41 (2:1)	23,98 (2:40)	20,5 (2:45)	10,59 (2:50)	41,39 (3:0)	40,39 (3:1)	30,59 (2:34)
2	29,45 (1:21)	100 (0:17)	40,94 (1:8)	48,45 (2:10)	37,92 (2:32)	24,78 (2:44)	13,72 (2:52)	51,94 (3:1)	55,76 (3:12)	35,6 (2:40)
3	28,25 (2:46)	40,94 (1:2)	100 (1:12)	36,57 (2:9)	41,38 (2:41)	42,82 (2:32)	22,28 (2:59)	49,15 (2:58)	50,31 (3:2)	31,37 (2:49)
4	38,41 (2:53)	56,88 (1:31)	36,57 (1:34)	100 (2:18)	30,37 (2:49)	26,71 (2:55)	14,45 (2:57)	74,9 (3:3)	63,17 (3:12)	29,01 (2:53)
5	23,98 (2:16)	37,92 (1:30)	41,38 (1:40)	30,37 (2:23)	100 (2:0)	38,3 (2:53)	16,89 (3:1)	34,32 (3:11)	42,28 (3:11)	45,65 (2:50)
6	20,5 (2:45)	24,78 (1:39)	42,82 (1:30)	19,98 (2:31)	38,3 (2:32)	100 (2:1)	6,3 (3:9)	37,21 (2:59)	26,05 (3:4)	14,41 (2:49)
7	10,59 (2:59)	13,72 (1:41)	22,28 (1:29)	8,52 (2:21)	16,89 (2:47)	6,3 (2:34)	100 (2:20)	11,11 (3:0)	16,25 (3:19)	12,38 (3:1)
8	41,39 (2:56)	51,94 (1:39)	49,15 (1:33)	74,9 (2:19)	34,32 (2:38)	37,21 (3:1)	11,11 (2:59)	100 (2:43)	66,75 (3:21)	35,81 (3:12)
9	40,39 (3:1)	55,76 (1:27)	50,31 (1:47)	63,17 (2:31)	42,28 (2:57)	26,05 (3:5)	16,25 (3:4)	66,75 (3:12)	100 (3:9)	44,31 (3:9)
10	30,59 (3:3)	35,6 (1:46)	31,37 (1:50)	25,17 (2:30)	45,65 (2:59)	14,41 (3:11)	12,38 (3:11)	35,81 (3:20)	44,31 (3:23)	100 (3:0)

Nilai persentase kemiripan yang paling kecil adalah nilai persentase saat mengecek kemiripan Dokumen 6 dengan Dokumen 7 yang menghasilkan persentase kemiripan sebesar 6,3%. Hasil persentase kemiripan terkecil menurun 8,83% dibanding dengan proses pendeteksian menggunakan *Stemming*. Nilai persentse kemiripan yang paling besar adalah nilai persentase saat melakukan pengecekkan kemiripan Dokumen 4 dan Dokumen 8 yang menghasilkan persentase kemiripan sebesar 74,9 %. Hasil persentase kemiripan tertinggi menurun 4,9% dibanding dengan proses pendeteksian menggunakan *Stemming*. Rata-rata hasil persentase dari setiap proses adalah 40,58%. Rata-rata hasil persentase dari setiap proses menurun 5,4% dibanding dengan proses pendeteksian menggunakan *Stemming*. Hasil persentase

kemiripan tanpa menggunakan proses *Stemming* berubah nilainya dibandingkan dengan proses menggunakan *Stemming*. Hasil persentase kemiripan tersebut berubah dikarenakan ketika tanpa proses *Stemming* yaitu mengubah setiap kata menjadi kata dasar memiliki banyak kesamaan antar karya tulis mengakibatkan nilai persentase kemiripan menjadi naik begitu pula sebaliknya ketika tanpa proses *Stemming* memiliki banyak kata-kata atau kalimat yang berbeda antar karya tulis menjadikan nilai persentase kemiripan menurun. Sistem tidak melakukan perubahan setiap kata menjadi kata dasar menjadikan proses pendeteksian tidak memakan waktu yang lama.

Berikut tabel hasil dari pengujian sistem yang dilakukan saat melakukan pendeteksian tanpa melalui proses TF-IDF dan proses *Stemming*.

Tabel 4. Persentase Kemiripan antar Dokumen Tanpa Proses TF-IDF dan Proses *Stemming*

Dok. Rembanding / Dok. Uji	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	% (menit:detik)									
1	100 (2:35)	31,24 (1:52)	27,89 (3:1)	39,76 (3:22)	22,9 (3:1)	24,12 (3:9)	11,46 (3:19)	45,79 (3:11)	41,42 (3:19)	32,55 (3:23)
2	31,24 (2:0)	100 (0:14)	40,27 (3:2)	48,36 (3:12)	35,9 (3:12)	25,11 (3:1)	13,55 (3:11)	52,9 (3:1)	56,82 (3:12)	33,28 (3:22)
3	27,89 (4:15)	40,27 (1:49)	100 (0:42)	34,76 (3:12)	40,4 (3:8)	41,9 (3:8)	25,16 (3:19)	46,12 (3:37)	49,32 (3:37)	31,11 (3:31)
4	39,76 (3:29)	48,36 (1:46)	34,76 (3:23)	100 (1:1)	27,8 (3:2)	21,29 (3:12)	9,44 (3:21)	74,85 (3:28)	66,99 (3:29)	24,01 (3:30)
5	22,9 (3:30)	35,9 (1:42)	40,4 (3:12)	27,8 (3:24)	100 (1:43)	38,19 (3:13)	16,59 (3:30)	31,27 (3:23)	39,55 (3:22)	42,69 (3:29)
6	24,12 (3:33)	25,11 (1:45)	41,9 (3:42)	21,29 (3:15)	38,19 (3:2)	100 (1:56)	10,36 (3:12)	36,47 (3:28)	26,2 (3:20)	16,98 (3:41)
7	11,46 (3:45)	13,55)1:53)	25,16 (3:12)	9,44 (3:35)	16,59 (3:12)	10,36 (3:7)	100 (1:36)	13,61 (3:33)	16,37 (3:31)	12,19 (3:39)
8	45,79 (3:49)	52,9 (1:42)	48,12 (3:21)	74,85 (3:12)	31,27 (3:18)	36,47 (3:30)	13,61 (3:10)	100 (1:9)	68,95 (3:30)	37,54 (3:41)
9	43,56 (3:57)	56,82 (1:53)	49,32 (3:12)	66,99 (3:45)	39,55 (3:23)	26,2 (3:29)	16,37 (3:31)	68,95 (3:18)	100 (1:12)	41,42 (3:42)
10	32,55 (4:36)	33,28 (1:54)	31,11 (3:25)	24,01 (3:48)	42,69 (3:32)	16,98 (3:39)	12,19 (3:29)	37,54 (3:28)	41,42 (3:49)	100 (3:49)

Nilai persentase kemiripan yang paling kecil adalah nilai persentase saat mengecek kemiripan Dokumen 4 dengan Dokumen 7 yang menghasilkan persentase kemiripan sebesar 9,44%. Hasil persentase kemiripan terkecil menurun 3,14% dibanding dengan proses pendeteksian menggunakan TF-IDF dan *Stemming*. Nilai persentse kemiripan yang paling besar adalah nilai persentase saat melakukan pengecekkan kemiripan Dokumen 4 dan Dokumen 8 yang menghasilkan persentase kemiripan sebesar 74,85%. Hasil persentase kemiripan tertinggi menurun 0,05% dibanding dengan proses pendeteksian menggunakan TF-IDF dan *Stemming*. Rata-rata hasil persentase dari setiap proses adalah 40,54%. Rata-rata hasil persentase dari setiap proses menurun 0,55% dibanding dengan proses pendeteksian menggunakan TF-IDF dan *Stemming*. Hasil persentase kemiripan tanpa menggunakan

TF-IDF dan *Stemming* berubah nilainya dibandingkan dengan proses menggunakan TF-IDF dan *Stemming*. Hasil persentase kemiripan tersebut berubah dikarenakan ketika tanpa proses TF-IDF yaitu menghapus beberapa kata-kata atau kalimat dan menyisakan kata-kata atau kalimat yang penting saja dan tanpa proses *Stemming* yaitu mengubah setiap kata menjadi kata dasar memiliki banyak kesamaan antar karya tulis mengakibatkan nilai persentase kemiripan menjadi naik begitu pula sebaliknya ketika tanpa proses TF-IDF dan tanpa proses *Stemming* memiliki banyak kata-kata atau kalimat yang berbeda antar karya tulis menjadikan nilai persentase kemiripan menurun. Waktu proses pendeteksian menjadi sedikit lebih lama dibandingkan dengan proses menggunakan TF-IDF dan *Stemming* dikarenakan tanpa melalui TD-IDF menjadikan isi karya tulis banyak untuk di cek kemiripannya.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan yang didapat setelah proses pengerjaan tugas akhir ini adalah sebagai berikut:

1. Penerapan TF-IDF menjadikan isi dari karya tulis terangkum dengan memunculkan kalimat yang mengandung kata-kata yang sering muncul dalam isi karya tulis tersebut. Kata-kata yang sering muncul dianggap merupakan kata penting yang dibahas dalam karya tulis, sehingga kalimat yang terdapat kata-kata dengan kemunculan yang banyak akan dianggap sebagai ide pokok dari karya itu.
2. Pengujian pendeteksian menggunakan TF-IDF menghasilkan rata-rata persentase kemiripan 41,09 % dan waktu proses 2 menit 27 detik, sedangkan tanpa TF-IDF menghasilkan persentase kemiripan 45,98% dan waktu proses 3 menit 23 detik. Hal ini membuktikan bahwa proses TF-IDF dapat mempersingkat waktu proses pendeteksian kemiripan dibandingkan dengan tanpa melalui proses TF-IDF. Selain itu dapat mempengaruhi persentase kemiripan, karena, pada hasil rangkuman terdapat banyak kesamaan atau sebaliknya membuang kata-kata yang sama.
3. Penerapan *Tokenizing* untuk menghilangkan karakter selain huruf isi dari karya tulis dilakukan dengan cara mengubah setiap huruf menjadi huruf kecil, kemudian menghapus *delimiter* dan spasi yang lebih dari satu. Hasilnya adalah susunan kata dengan pemisah sebuah spasi saja.
4. Penerapan *Filtering* untuk menghapus kata-kata penghubung antar kata dan kata yang tidak memiliki arti penting pada isi dari karya tulis dilakukan dengan cara membandingkan setiap kata dalam karya tulis dengan data *stopword* yang ada di database. Ketika kata karya tulis dan data *stopword* sama, maka kata tersebut akan dihapus.
5. Penerapan *Stemming* untuk menjadikan setiap kata dari isi karya tulis menjadi kata dasar dilakukan dengan cara membandingkan setiap kata dengan data *dictionary* pada *database* yang mencatat kata-kata berimbuhan. Ketika kata dari isi karya tulis dibandingkan dengan data *dictionary* terindikasi kata yang memiliki imbuhan, maka kata tersebut akan diubah menjadi kata dasar dengan menghapus imbuhan yang dimilikinya. Namun, proses perubahan tersebut juga akan merubah kata ketika ada suatu nama orang atau nama perusahaan yang ketika di cocokkan dengan database kata dasar termasuk kata yang berimbuhan, maka sistem akan mengubah nama tersebut menjadi kata dasar sesuai dengan proses *Stemming*. Nama “Petra” akan dianggap kata yang memiliki awalan pe-, sehingga hasil setelah proses *Stemming* adalah “tra”.
6. Pengujian pendeteksian menggunakan *Stemming* menghasilkan rata-rata persentase kemiripan 41,09 % dan waktu proses 2 menit 27 detik, sedangkan tanpa Stemming meng-hasilkan persentase kemiripan 40,58% dan waktu proses 2 menit 23 detik. Hal ini membuktikan bahwa pendeteksian kemiripan melalui proses Stemming menjadikan waktu proses lebih lama

dibandingkan dengan ketika tidak melalui proses Stemming.

7. Algoritma *Knuth-Morris-Pratt* untuk mendeteksi kemiripan terhadap karya tulis dilakukan dengan cara mencari setiap kata yang sama pada kedua karya tulis yang dideteksi. Hasil pencarian kemiripan tersebut kemudian dihitung persentase kemiripannya menggunakan metode *Cosine Similarity*.
8. Pengujian pendeteksian dokumen berjumlah 8 halaman dan 9 halaman menghasilkan persentase 28,16% dan waktu proses 1 menit 45 detik, sedangkan pendeteksian dokumen berjumlah 8 halaman dan 13 menghasilkan persentase 36,71% dan waktu proses 2 menit 30 detik. Hal ini membuktikan bahwa jumlah halaman dapat mempengaruhi waktu proses dan persentase kemiripan, karena algoritma pencocokannya adalah dengan mencocokkan kata satu-persatu pada masing-masing karya tulis.
9. Rata-rata persentase kemiripan yang dihasilkan dari pendeteksian tanpa menggunakan proses TF-IDF sebesar 45,98%, nilai tersebut lebih rendah dibandingkan dengan pendeteksian menggunakan proses TF-IDF, *Tokenizing*, *Filtering* dan *Stemming* yang menghasilkan persentase 41,09%, tanpa menggunakan proses *Stemming* yaitu 40,58% serta tanpa menggunakan proses TF-IDF dan *Stemming* sebesar 40,54%

6.2 Saran

Saran untuk perbaikan dan pengembangan dari tugas akhir ini adalah:

1. Proses *Stemming* sebaiknya memiliki basis data kata dasar bahasa indonesia secara lengkap, sehingga ketika ditemukan awalan seperti “pe-“ pada nama “petra”, kata tersebut tidak langung dipotong menjadi “tra”, akan tetapi harus dicocokkan terlebih dahulu dengan basis data kata dasar yang ada, apabila kata tersebut ditemukan dalam basis data kata dasar, maka akan dipotong, akan tetapi apabila tidak ditemukan, maka kata tersebut dianggap sebagai kata yang tidak memiliki kata dasar.
2. Aplikasi dapat ditambahkan fitur untuk pendeteksian ke internet, sehingga pendeteksian tidak hanya dilakukan pada basis data yang tersedia, akan tetapi juga dapat melakukan pendeteksian keseluruh *website* yang ada di dunia.
3. Sebagai pengembangan kedepan dapat pula aplikasi ini dibuat kedalam bentuk sistem aplikasi berbasis android, sehingga dapat memudahkan pengecekkan dengan *smartphone* android dimanapun dan kapanpun.

DAFTAR PUSTAKA

[1] Pangkalan Data Pendidikan Tinggi, Direktorat Jendral Pendidikan Tinggi. (2015). *Grafik Jumlah Mahasiswa Aktif Berdasarkan Jenjang Pendidikan*. http://forlap.dikti.go.id/mahasiswa/homegraphjenjang, diakses tanggal 20 Oktober 2014.

[2] Putri, Rifki Afiana (2013). *Penerapan Pattern Matching untuk Deteksi Plagiarisme*. Bandung: Institut Teknologi Bandung.

[3] Utorodewo, Felicia. (2007). *Bahasa Indonesia: Sebuah Pengantar Penulisan Ilmiah*. Depok: Lembaga Penerbit FEUI.

[4] Mutiara, Benny. (2008). *Anti Plagiarism Application with Algorithm Karp-Rabin at Thessis in Gunadarma University*. Depok: Universitas Gunadarma

[5] Ekaputri, Gahayu Handari dan Yulie Anneria Sinaga (2011). Aplikasi Algoritma Pencarian *String Knuth-Morris-Pratt* dalam Permainan *Word Search.* Bandung: Institut Teknologi Bandung.