

Rancang Bangun Aplikasi Deteksi Kemiripan Dokumen Teks Menggunakan Algoritma *Ratcliff/Obershelp*

Yudhy Lady Joane, Alicia Sinsuw, Agustinus Jacobus
Teknik Informatika Universitas Sam Ratulangi

120216101@student.unsrat.ac.id, alicia.sinsuw@unsrat.ac.id, a.jacobus@unsrat.ac.id

Abstrak – Kegiatan menjiplak tugas sering dilakukan mahasiswa yang merupakan tindakan plagiat, banyak tugas yang terkumpul dan waktu yang terbatas membuat dosen sulit untuk memeriksa tugas satu per satu. Oleh karena itu, diperlukan suatu aplikasi yang dapat mendeteksi kemiripan dokumen teks. Penelitian ini menggunakan algoritma *Ratcliff/Obershelp* yang dapat menilai seberapa mirip pola dari dua string yang berbeda. Pada pengujian algoritma dilakukan pengujian pada 12 dokumen dengan tingkat kesamaan dan posisi yang berbeda. Nilai yang didapat mendekati nilai prediksi dan nilai pada dokumen yang memiliki kesamaan sama tetapi posisi yang berbeda mendapatkan nilai yang berbeda..

Kata kunci : *Ratcliff/Obershelp*, *plagiat*, *Information Retrieval*.

I. PENDAHULUAN

Teknologi Informasi adalah seperangkat alat yang membantu manusia untuk bekerja dengan informasi dan melakukan tugas-tugas yang berhubungan dengan pemrosesan informasi[1]. Teknologi informasi merupakan alat yang sering digunakan dalam pengolahan dan pertukaran informasi.

Komputer dan perangkat lunak yang digunakan dalam pengolahan informasi sangat membantu masyarakat terlebih dalam bidang akademis terutama oleh mahasiswa. contohnya dalam penyelesaian dan pengumpulan tugas kuliah. Informasi yang ada diproses melalui perangkat lunak komputer dan menghasilkan *softcopy* yang tersimpan didalam komputer. Dari kemudahan yang ada terkadang malah disalah gunakan oleh mahasiswa sehingga memberikan dampak negatif. Mahasiswa sering menyalahgunakannya dengan menjiplak *softcopy* dokumen teks tugas mahasiswa lainnya dimana hal tersebut merupakan tindakan plagiat.

Menurut Permendiknas No 17 tahun 2010, pasal 1 ayat 1 plagiat itu sendiri merupakan perbuatan secara sengaja atau tidak sengaja dalam memperoleh nilai untuk suatu karya ilmiah, dengan mengutip sebagian atau seluruh karya dan/atau karya ilmiah orang lain, tanpa menyatakan sumber secara tepat dan memadai. Plagiat dapat dianggap tindakan pidana karena mencuri hak cipta orang lain dan pelaku plagiat disebut sebagai plagiator. Karena adanya alat bantu seperti komputer dan laptop sehingga memberikan kemudahan kepada mahasiswa untuk menjiplak. Plagiat sangat berdampak buruk bagi mahasiswa, antaranya mahasiswa menjadi malas dalam mengerjakan tugas kuliah dan hanya berharap kepada orang lain kemudian mahasiswa akan terbiasa melakukan penjiplakan sehingga dapat

mengurangi kreatifitas. Selain memiliki dampak buruk kepada mahasiswa, dosen atau para pengajar juga direpotkan dalam menganalisis satu per satu tugas yang dikumpulkan mahasiswa dengan membandingkan dua atau lebih dokumen secara manual. Cara tersebut kurang efektif dan efisien mengingat jumlah mahasiswa yang tidak sedikit sehingga memerlukan waktu yang lebih lama.

Dalam penelitian yang akan dilakukan, peneliti menggunakan algoritma *Ratcliff/Obershelp* dalam pencarian kesamaan string dari dokumen yang berbeda dan juga dalam penilaian tingkat kesamaan dokumen teks.

II. LANDASAN TEORI

A. Information Retrieval

Information Retrieval merupakan studi tentang sistem pengindeksan, pencarian, dan mengingat data, khususnya teks atau bentuk tidak terstruktur lainnya yang dapat memenuhi informasi dari dalam koleksi besar yang biasanya disimpan dalam komputer[2].

Information Retrieval merupakan suatu sistem yang menemukan informasi yang sesuai dengan kebutuhan *user* dari kumpulan informasi secara otomatis[3]. *Information Retrieval* sering digunakan untuk aplikasi pencarian contohnya seperti mesin pencarian internet.

B. Text mining

Text mining merupakan salah satu bidang khusus dari data mining. *Text mining* lebih berfokus pada teknik dan metodologi dalam lingkup temu kembali sistem informasi. *Text mining* berfungsi untuk mendapatkan informasi dari sebuah dokumen teks. dalam penggalian informasi terdapat beberapa tahapan yang diantaranya adalah:

1. Text preprocessing

Text preprocessing merupakan tahap awal dari *Text mining* dimana data *text* akan dibersihkan sehingga *text* menjadi lebih terstruktur sebelum masuk ketahap berikutnya untuk diolah lebih lanjut. Sekumpulan karakter yang bersambungan (teks) harus dipecah-pecah menjadi lebih berarti. Hal tersebut dapat dilakukan dalam beberapa tingkatan yang berbeda. Suatu dokumen dapat di pecah menjadi suatu bab, paragraf, kalimat dan kata[4]. Tahapan *Text preprocessing* dalam penelitian ini meliputi:

a. Case Folding

Tahap *case folding* merupakan tahap dimana semua huruf dalam dokumen teks diubah menjadi huruf kecil dan hanya huruf a sampai z saja yang diterima.

b. *Filtering*

Pada tahap *filtering* dimana akan dilakukan penghapusan stopwords untuk mengeliminasi kata-kata yang tidak deskriptif dan yang sering muncul.

c. Penghapusan spasi

Tahap ini merupakan tahap dimana semua spasi yang terdapat pada dokumen teks dihilangkan. Karena algoritma *Ratcliff/Obershelp* merupakan algoritma untuk mendeteksi kemiripan teks dengan cara membandingkan string yang berbeda sehingga spasi yang terdapat pada dokumen teks harus dihilangkan sehingga dokumen teks menjadi sebuah string.

2. *Pattern Discovery/Text preprocessing*

Pattern Discovery/Text preprocessing merupakan tahap dari terakhir dari *Text mining* dimana dokumen teks yang telah dibersihkan dan sudah terstruktur akan diproses untuk mendapatkan pola dan informasi dari dokumen teks tersebut.

C. *Stop-word*

Stop-word adalah istilah untuk kata yang tidak memiliki keterkaitan makna yang kuat terhadap hubungan kata-kata lainnya pada suatu kalimat atau paragraf[5]. *Stop-word* merupakan kata yang umum dan sering digunakan.

Seperti yang telah dijelaskan diatas, *stop-word* merupakan kata yang sering muncul dalam suatu kalimat, dalam bahasa Indonesia kata-kata yang sering muncul seperti ‘ada’, ‘dan’, ‘untuk’, ‘di’, ‘adalah’ dan lain sebagainya.

Pengeleminasian *stop-word* mampu mengurangi kata dan karakter pada dokumen teks sehingga dapat mengurangi waktu eksekusi dalam pencarian kesamaan.

D. Algoritma *Ratcliff/Obershelp*

Algoritma *Ratcliff/Obershelp* menggunakan proses yang sama untuk memutuskan seberapa mirip dua pola satu dimensi. Karena string teks merupakan satu dimensi, algoritma ini mengembalikan nilai yang dapat di gunakan sebagai faktor kepercayaan atau persentase, menunjukkan kesamaan dua string[6].

Konsep pencocokan dari algoritma ini yaitu, pertama menemukan sub string terpanjang yang memiliki kesamaan dari string S1 dan S2 yang di sebut *anchor*. Nilai dari Km bertambah berdasarkan panjang dari *anchor*. Kemudian bagian yang tersisa dari string sebelah kiri dan kanan dari *anchor* harus diperiksa sebagai string-string yang baru (dengan kata lain mengulangi step 1). Proses tersebut di ulangi sampai semua karakter dari string S1 dan S2 di analisa.

Algoritma *Ratcliff/Obershelp* dinyatakan dengan rumus:

$$D_{ro} = \frac{2 * K_m}{|S_1| + |S_2|}$$

Persamaan 1. Algoritma *Ratcliff/Obershelp*

Km = Jumlah karakter yang sama

|S1| = Panjang dari String 1

|S2| = Panjang dari String 2

Mencari kesamaan kedua string MATHEMATICS dan MATEMATICA

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
S1	M	A	T	H	E	M	A	T	I	C	S
S2	M	A	T	E	M	A	T	I	C	A	

1. Panjang dari string S₁ :

$$|S_1|=11$$

Panjang dari string S₂ :

$$|S_2|=10$$

2. Substring yang terpanjang yang dimiliki oleh kedua string adalah EMATIC. Maka EMATIC merupakan sebuah *anchor*, dan Km=|EMATIC|=6

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
S1	M	A	T	H	E	M	A	T	I	C	S
S2	M	A	T	E	M	A	T	I	C	A	

3. Di sebelah kiri dari *anchor* tersisa kumpulan huruf MATH dan MAT. Sub string yang terpanjang dan memiliki kesamaan dari kumpulan huruf tersebut adalah MAT. Maka, Km= 6+|MAT|=9

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
S1	M	A	T	H	E	M	A	T	I	C	S
S2	M	A	T	E	M	A	T	I	C	A	

4. MAT merupakan substring yang berada di awal dari kedua string S₁ dan S₂, disebelah kiri substring tersebut sudah tidak terdapat huruf lagi. Pada sebelah kanan dari MAT, kita memiliki H di dalam string S₁ dan pada string S₂ sudah tidak memiliki karakter lagi. Maka Km tetap sama dan kita lanjutkan pada karakter sebelah kanan dari *anchor*.

5. Disebelah kanan dari *anchor* terdapat karakter S dan A. Karena kedua karakter tersebut berbeda maka mereka tidak cocok. Maka nilai dari Km tetap 9. Jadi kita memiliki semua data yang kita perlukan untuk menghitung nilai dari *Ratcliff/Obershelp*.

Penilaian *Ratcliff/Obershelp* untuk string MATHEMATICS dan MATEMATICA adalah:

$$D_{ro} = \frac{2 * 9}{10 + 11} = \frac{18}{21} = 0.857$$

Jadi, dari kedua string MATHEMATICS dan MATEMATICA memiliki nilai kesamaan 0.857 yang bisa dikatakan sama.

E. Plagiarisme

Plagiarisme berasal dari kata Latin, “plagiarius”, yang berarti “pencuri”. Plagiarisme merupakan tindakan mengambil tulisan atau hasil karya orang lain seperti tulisan atau hasil karyanya sendiri tanpa persetujuan dari pemilik tulisan atau hasil karya tersebut. Plagiarisme adalah tindakan penyalahgunaan, pencurian/perampasan, penerbitan, pernyataan, atau menyatakan sebagai milik sendiri

sebuah pikiran, ide, tulisan, atau ciptaan yang sebenarnya milik orang lain[7].

Sedangkan menurut Kamus Besar Bahasa Indonesia (KBBI) Plagiarisme adalah tindakan penyalahgunaan, pencurian/perampasan, penerbitan, pernyataan, atau menyatakan sebagai milik sendiri sebuah pikiran, ide, tulisan, atau ciptaan yang sebenarnya milik orang lain.

F. DFD (*Data flow diagram*)

Data flow diagram (DFD) disebut juga dengan Diagram Arus Data (DAD). DFD adalah: suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data, dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut[8].

Data flow diagram (DFD) adalah sebuah alat yang menggambarkan aliran data sampai sebuah sistem selesai, dan kerja atau proses dilakukan dalam sistem tersebut[9]. Terdapat beberapa ahli yang pernah mendefinisikan simbol-simbol DFD, diantaranya adalah Gane/Serson dan Yourdon/De Marco yang mendefinisikan simbol DFD pada tabel 1 tabel diagram alir data.

Tabel 1 diagram alir data

Gane/Sarson	Yourdon/De Marco	Keterangan
		Entitas eksternal, dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi diluar sistem
		Orang, unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi.
		Aliran data dengan arah khusus dari sumber ke tujuan
		Penyimpanan data atau tempat data direfer oleh proses.

DFD memiliki beberapa tingkatan level yaitu:

1. Diagram Konteks
Diagram konteks merupakan perwakilan dari seluruh proses yang terdapat pada sebuah sistem. Diagram ini sangat sederhana untuk dibuat karena dalam diagram konteks hanya gambaran besar dari suatu sistem dan sama sekali tidak memuat penyimpanan.
2. Diagram Nol (diagram level-1)
Diagram ini merupakan pemecahan dari diagram konteks.
3. Diagram Rinci
Diagram ini merupakan diagram yang digunakan untuk menguraikan atau pemecahan proses yang ada pada diagram nol.

III. Metodologi Penelitian

A. Analisis Data

Dalam analisis data dilakukan penganalisaan data dengan melakukan *text preprocessing* dan *processing*. Dimana dalam *text preprocessing* akan dilakukan pembersihan data teks dan pada tahap *processing*

dimana algoritma *Ratcliff/Obershelp* akan diimplementasikan.

1. *Text preprocessing*

Tahap ini melakukan analisis semantik (kebenaran arti) dan sintaktik (kebenaran susunan) terhadap teks. Tujuan dari tahapan ini adalah untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan lebih lanjut. Berikut adalah tahapan *text preprocessing* yang akan digunakan adalah:

- a. Data Input
Berikut adalah contoh paragraf yang akan dibersihkan melalui tahapan *text preprocessing* sebelum kemudian diolah lebih lanjut.

Olahraga, secara teratur bisa membuat tubuh menjadi SEHAT dan BUGAR.

- b. *Case Folding*
Case folding merupakan tahapan dimana semua huruf yang ada didalam dokumen diubah menjadi huruf kecil dan hanya huruf ‘a’ sampai dengan ‘z’ yang dapat diterima. Karakter selain huruf dihilangkan dan dianggap delimiter (pembatas).

Input:
Olahraga, secara teratur bisa membuat tubuh menjadi SEHAT dan BUGAR ,

Output:
olahraga, secara teratur bisa membuat tubuh menjadi sehat dan bugar.

- c. *Cleansing*
Pada tahap ini dokumen akan dibersihkan dari tanda baca dan simbol-simbol yang ada di dalam dokumen.

Input:
olahraga, secara teratur bisa membuat tubuh menjadi sehat dan bugar.

Output:
olahraga secara teratur bisa membuat tubuh menjadi sehat dan bugar

- d. *Filtering*
Pada tahap *filtering* akan menggunakan algoritma stoplist dimana kata-kata yang kurang penting dan tidak memiliki pengaruh akan dibuang. Penghapusan stopword dilakukan dengan membandingkan kata-kata yang ada didalam dokumen dengan stopword yang sebelumnya telah disimpan. Jika ada kata yang sama, kata tersebut akan dihilangkan. Kata-kata yang dimaksudkan sebagai stopword adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-words. Contoh stopword adalah “yang”, “dan”, “di”, “dari” dan lain – lain. Berikut adalah

contoh proses dari *filtering* dengan algoritma stoplist.

Input:
olahraga secara teratur bisa membuat tubuh menjadi sehat dan bugar
Output:
olahraga teratur membuat tubuh sehat bugar

- e. Penghapusan Spasi
Setelah kalimat dibersihkan maka akan dilakukan penghapusan spasi agar menjadi satu string. Penghapusan spasi dilakukan karena algoritma yang digunakan merupakan pendeteksian string. Berikut adalah hasil dari kalimat yang telah menjadi string.

Input:
olahraga teratur membuat tubuh sehat bugar
Output:
olahragateraturmembuattubuhsehatbugar

2. *Text Processing*

Pada proses ini adalah dimana algoritma *Ratcliff/Obershelp* akan di implementasikan. Cara kerja algoritma ini adalah untuk menemukan *subsequence* pencocokan terpanjang berdekatan dari dua data string. Proses yang sama kemudian diterapkan secara rekursif untuk potongan urutan ke kiri dan ke kanan *subsequence* yang cocok sampai tidak ditemukan lagi *subsequence* yang sama.

a. *Sequence Matching*

Tahap ini mencocokkan dua buah string untuk mendapatkan *subsequence* yang dimiliki oleh String 1 dan String 2. Dibawah ini adalah teks dokumen yang telah melewati tahap preprocessing.

- Menghitung Banyak Karakter
Pencarian nilai |S1| dan |S2| dimana |S1| adalah banyak karakter yang dimiliki String 1 sedangkan |S2| adalah banyak karakter yang dimiliki oleh String 2.

String 1
berolahragamembuattubuhsehatdanbugar
|S1|= 36
String 2
olahragasecarateraturmembuattubuhsehatdanbugar
|S2|= 46

- Pencarian *subsequence*
Pencarian *subsequence* yang sama-sama dimiliki oleh string1 dan string2, *subsequence* terpanjang dinamakan *anchor*.

String 1
Berolahraga|membuattubuhsehatdanbugar|
String 2
Olahragasecarateratur|membuattubuhsehatdanbugar|

Anchor :
membuattubuhsehatdanbugar
Km= 25
Setelah *Anchor* telah ditemukan, selanjutnya mencari *subsequence* lainnya yang terdapat pada string1 dan string2

String1
ber|olahraga||membuattubuhsehatdanbugar|
String 2
|olahraga|secarateratur|membuattubuhsehatdanbugar|
Substring baru : olahraga
Menggabung substring-substring yang telah ditemukan dari String 1 dan String 2, maka total karakter dari substring-substring yang di tambahkan adalah seperti berikut

$$Km = 25 + |olahraga| = 25 + 8 = 33$$

b. Menghitung Kesamaan

Tahap ini dilakukan setelah semua substring yang sama telah di temukan dari proses sequence matching dan dimana persamaan dari algoritma *Ratcliff/Obershelp* akan digunakan.

Penilaian *Ratcliff/Obershelp* untuk String 1 dan String 2 adalah:

$$D_{ro} = \frac{2 \times 33}{36 + 46} = \frac{66}{82} = 0.804$$

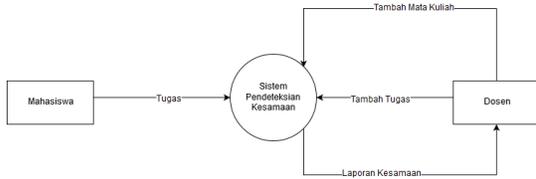
B. Pemodelan Proses Sistem

Pemodelan proses adalah sekumpulan tahap, tugas dan aktivitas yang dibutuhkan untuk secara efisien mentransformasikan kebutuhan pemakai ke suatu solusi perangkat lunak yang efektif. Pemodelan proses perangkat lunak bertujuan untuk merepresentasikan aktivitas yang terjadi selama pembuatan perangkat lunak dan perubahan-perubahannya. Pemodelan proses pada penelitian ini menggunakan tools *Data flow diagram* (DFD). *Data flow diagram* (DFD) merupakan suatu cara atau metode untuk membuat rancangan sebuah sistem yang mana berorientasi pada alur data yang bergerak pada sebuah sistem nantinya.

1. DFD Diagram Konteks

DFD diagram konteks menggambarkan satu lingkaran besar yang mewakili aliran data dari keseluruhan sistem. Pada gambar 1 DFD diagram konteks dapat dilihat bahwa sistem memiliki 2 aktor yaitu dosen dan mahasiswa.

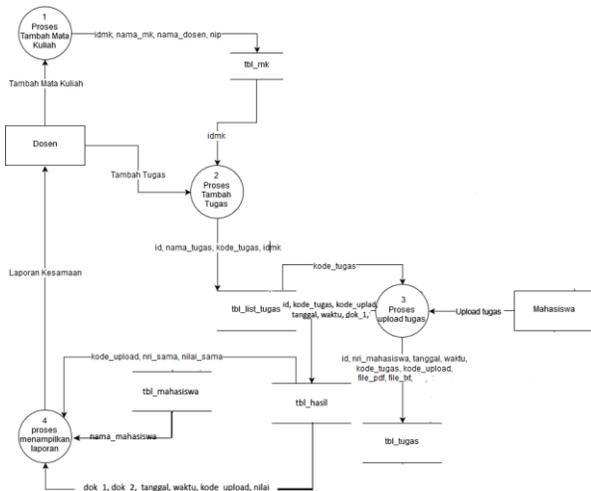
Dosen dapat membuat mata kuliah baru dan tugas baru agar dapat di akses oleh mahasiswa kemudian sistem akan menampilkan hasil nilai kesamaan dari dokumen tugas mahasiswa. Dan untuk aktor mahasiswa dapat mengunggah tugas.



Gambar 1 Diagram Konteks

2. DFD Level 0

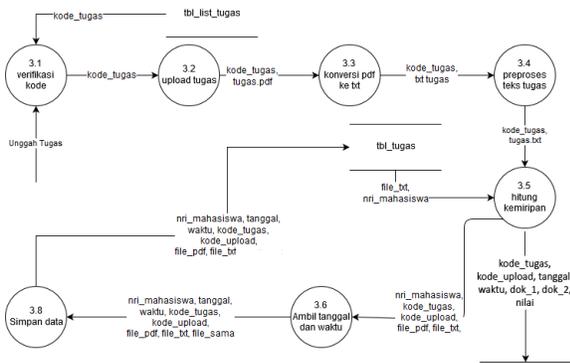
DFD Level 0 merupakan lanjutan dari DFD diagram konteks di mana pada Level 0 proses aliran data lebih terperinci. Pada gambar 2 DFD Level 0 memiliki beberapa proses yang ada pada sistem yaitu, proses tambah mata kuliah, proses tambah tugas, proses upload tugas dan proses menampilkan laporan. Selain proses-proses tersebut terdapat juga table basis data yaitu tbl_mk, tbl_list_tugas, tbl_nilai dan tbl_tugas.



Gambar 2 DFD Level 0

3. DFD Level 1 Upload Tugas

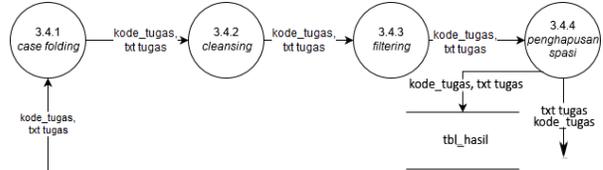
DFD Level 1 Proses Upload Tugas menjelaskan alir data yang terjadi didalam proses upload tugas yang dilakukan oleh mahasiswa sebagai salah satu aktor.



Gambar 3 Diagram Level 1 Upload Tugas

4. DFD Level 2 Preprocessing Teks Tugas

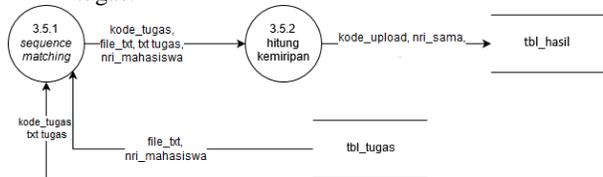
DFD level 2 preprocessing teks tugas menjelaskan aliran sistem yang terjadi pada proses preprocessing, dimana preprocessing merupakan tahap untuk membersihkan data teks sebelum masuk ke proses perhitungan kesamaan.



Gambar 4 DFD Level 2 Preprocessing Tugas Teks

5. DFD Level 2 Hitung Kemiripan

DFD level 2 hitung kemiripan menjelaskan aliran sistem yang terjadi saat penghitungan kemiripan tugas.



Gambar 5 DFD Level 2 Hitung Kemiripan

IV. HASIL DAN PEMBAHASAN

A. Interface Sistem

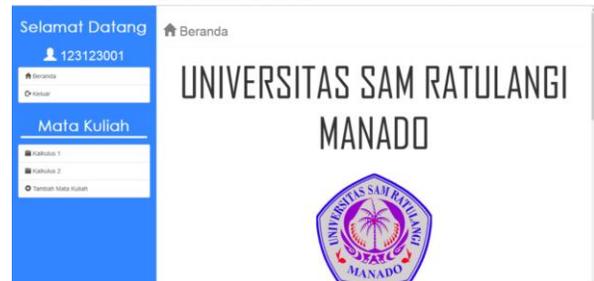
1. Halaman Login

halaman login dibawah ini adalah tampilan dari halaman awal aplikasi. Dari halaman tersebut dapat dilihat bahwa aplikasi memiliki dua user, yaitu user dosen dan juga user mahasiswa.



Gambar 6 Tampilan Halaman Login

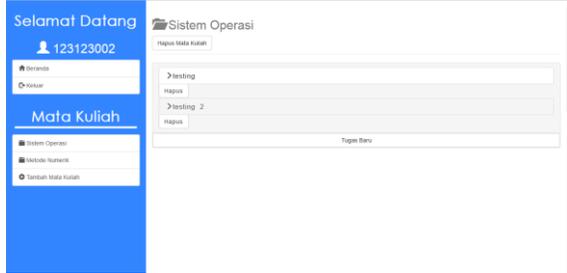
2. Halaman Utama Dosen



Gambar 7 tampilan utama dosen

Gambar 7 Tampilan Halaman Utama dosen merupakan halaman awal dari *user* dosen setelah melakukan login. Pada halaman terdapat menu beranda, keluar, list seluruh mata kuliah yang di ampuh dan juga tambah mata kuliah baru.

3. Halaman List Tugas



Gambar 8 List Tugas

Gambar 8 ini menampilkan format isian data prediksi mahasiswa di semester dimana mahasiswa sedang menjalankan atau baru selesai perkuliahan. Isian yang ada berupa NIM, Jurusan atau Program Studi dan IP Semester serta SKS yang disesuaikan dengan semester mahasiswa.

4. Halaman Tambah Tugas Baru

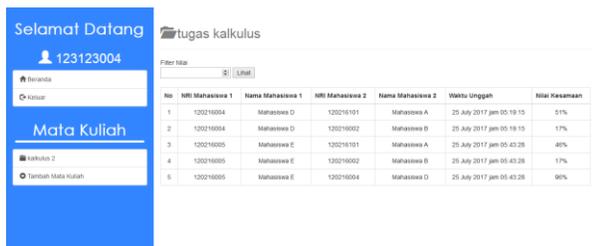
gambar 9 adalah tampilan untuk menambah tugas baru. Pada halaman tersebut terdapat form untuk mengisi id tugas dan nama tugas yang harus di isi untuk menambahkan tugas baru.



Gambar 9 tambah tugas baru

5. Halaman Laporan Tugas

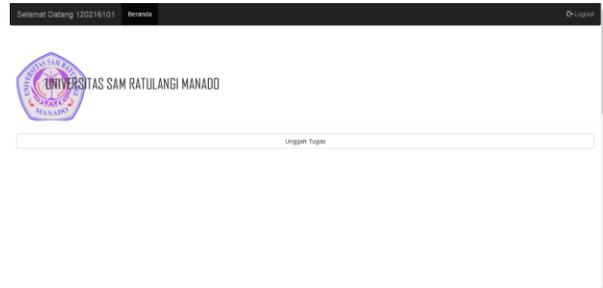
gambar 10 adalah tampilan saat list tugas di pilih. Halaman tersebut menampilkan seluruh mahasiswa yang telah mengumpulkan tugas beserta laporan jika tugas itu sama dengan mahasiswa lain. Dalam laporan tersebut terdapat detail NRI mahasiswa yang sama beserta nilai kemiripan tugas juga terdapat fungsi filter untuk menyaring data yang tampil sesuai dengan nilai yang kita dimasukkan.



Gambar 10 laporan tugas

6. Halaman Awal *User* Mahasiswa

Gambar 11 adalah tampilan awal dari halaman awal mahasiswa setelah melakukan login dalam halaman ada button tambah tugas untuk masuk ke dalam tugas mata kuliah.



Gambar halaman utama *user* mahasiswa

B. Pengujian Model

Pada tahap pengujian perangkat lunak ini akan dilakukan pengujian dari perangkat lunak yang dibangun sebagai evaluasi untuk mengetahui apakah perangkat lunak sesuai dengan hasil yang diharapkan. Terdapat 2 pengujian yang dilakukan yaitu :

1. Pengujian Waktu Eksekusi

pengujian 6 dokumen merupakan hasil dari pengujian 6 dokumen tugas mahasiswa. Waktu yang dibutuhkan saat mengunggah tugas ke 6 adalah 111 detik 1410 *microsecond* atau 1 menit 51 detik 1410 *microsecond*.

Untuk hasil keseluruhan pengujian waktu eksekusi yang telah dilakukan dapat dilihat pada tabel 1 pengujian waktu eksekusi.

Tabel 2 pengujian waktu eksekusi

No	Dokumen	Total Dokumen Tersimpan	Banyak Karakter	Waktu Eksekusi (<i>microsecond</i>)
1	Tugas 1	1	1.739	0.0017
2	Tugas 2	2	1.739	12.5663
3	Tugas 3	3	1.739	41.5436
4	Tugas 4	4	1.739	43.7672
5	Tugas 5	5	1.739	55.9577
6	Tugas 6	6	1.739	111.1410

2. Pengujian Algoritma

Pengujian algoritma merupakan pengujian yang dilakukan untuk mengetahui tingkat akurasi dari algoritma yang digunakan yaitu *Ratcliff/Obershelp*. Dalam pengujian ini terdapat 1 dokumen yang akan dibandingkan dengan 11 dokumen lainnya yang memiliki tingkat kesamaan yang berbeda sehingga total dokumen yang akan di unggah adalah 12 dokumen. 11 dokumen yang akan dibandingkan sengaja dibuat dengan tingkat kesamaan yaitu 0%, 25%, 50%, 75% dan 100%. Selain tingkat kesamaan yang dibuat berbeda, masing-masing dokumen dengan tingkat kesamaan yang sama diletakan pada baris yang berbeda.

Hasil dari pengujian algoritma adalah seperti pada tabel

Tabel 3 Hasil Pengujian Algoritma

No	Dokumen A	Dokumen B	Posisi	Prediksi	Nilai
1	Dokumen 2	Dokumen 1	-	0%	17%
2	Dokumen 3	Dokumen 1	Atas	25%	34%
3	Dokumen 4	Dokumen 1	Tengah	25%	33%
4	Dokumen 5	Dokumen 1	Bawah	25%	26%
5	Dokumen 6	Dokumen 1	Atas	50%	57%
6	Dokumen 7	Dokumen 1	Tengah	50%	53%
7	Dokumen 8	Dokumen 1	Bawah	50%	51%
8	Dokumen 9	Dokumen 1	Atas	75%	78%
9	Dokumen 10	Dokumen 1	Tengah	75%	79%
10	Dokumen 11	Dokumen 1	Bawah	75%	76%
11	Dokumen 12	Dokumen 1	-	100%	100%

Dari tabel dapat dilihat bahwa nilai yang didapat mendekati nilai yang di prediksi dan pada posisi yang berbeda dengan kesamaan yang sama nilai yang di dapatkan berbeda, kesamaan kalimat dengan posisi akhir dokumen memiliki nilai lebih dekat dengan nilai prediksi dibandingkan dengan kalimat yang berada pada awal dokumen.

V. PENUTUP

A. Kesimpulan

Berdasarkan dari pembahasan-pembahasan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan dari Skripsi “Rancang Bangun Aplikasi Pendeteksian Kesamaan Dokumen Teks Dengan Algoritma *Ratcliff/Obershelp*” adalah sebagai berikut :

1. Sistem dapat melakukan proses pengidentifikasian kesamaan dokumen tugas mahasiswa menggunakan algoritma *Ratcliff/Obershelp*.
2. Banyak tugas yang terkumpul dan banyaknya karakter didalam dokumen tugas pada suatu tugas mempengaruhi waktu eksekusi dari proses pengidentifikasian kesamaan dokumen.
3. Posisi kalimat yang sama mempengaruhi nilai kesamaan dari pendeteksian.

B. Saran

Saran yang ingin disampaikan untuk pengembangan lebih lanjut penelitian ini adalah sebagai berikut :

1. Pada pendeteksian kemiripan dokumen teks ini dapat menambahkan proses stemming dalam preprocessing teks agar jumlah data teks berkurang dan dapat meminimum waktu pemrosesan data teks.
2. Pada pendeteksian kemiripan dokumen teks ini dapat di analisa menggunakan algoritma yang lain yang mungkin jauh lebih akurat agar dapat mendapatkan hasil yang lebih akurat pula.

DAFTAR REFERENSI

- [1] Haag, S dan Keen P. 1996. *Information Technology, Tomorrow's Advantage Today*. McGraw-Hill.
- [2] K. Putung, A. Lumenta, A Jacob, 2016. “Penerapan Sistem Temu Kembali Informasi pada Kumpulan Dokumen Skripsi.” E-journal Teknik Informatika, Volume 8, No 1.

- [3] F. Amin, Purwatingtyas, 2015. "Rancang Banun *Information Retrieval System (IRS)* Bahasa Jawa Ngoko pada Palintangan Penjabar Semangad dengan Metode *Vector Space Model (VSM)*". Jurnal Teknologi Informasi DINAMIK Volume 20, No 1.
- [4] A. Indranandita, B. Susanto, A. Rachmat, 2008. “Sistem Klasifikasi dan Pencarian Jurnal Dengan Menggunakan Metode Naïve Bayes dan Vector Space Model”. 10 Jurnal Informatika, Volume 4, No. 2.
- [5] A. Firdaus, Ernawati, A. Vatesria, 2014. “Aplikasi Pendeteksi Kemiripan Pada Dokumen Teks Menggunakan Algoritma Nazief & Adriani dan Metode Cosine Similarity.” Jurnal Teknologi Informasi 2014 Vol. 10
- [6] Ilya Ilyankou, 2014. “*Comparison of Jaro-Winkler and Ratcliff/Obershelp algorithms in spell check.*” *IB Extended Essay Computer Science*.
- [7] Eko Nugroho, 2011. “Perancangan Sistem Deteksi Plagiarisme Dokumen Teks Dengan Menggunakan Algoritma *Rabin-Karp*.” Tugas Akhir. Malang: Universitas Brawijaya Malang.
- [8] R. Afyeni, 2014. “Perancangan Data Flow Diagram untuk Sistem Informasi Sekolah (Studi Kasus pada SMA Pembangunan Laboratorium UNP)”. Jurnal TEKNOIF, Volume 2, No. 1
- [9] Indrajani, 2011. “Perancangan Basis Data dalam All in 1, PT. Elex Media Komputindo”. Jakarta.

SEKILAS TENTANG PENULIS



Saya bernama Yudhy Lady Joane dan merupakan anak kedua dari pasangan Daniel N dan Youla K, lahir di Palu pada tanggal 9 Agustus 1993. Asal daerah Manado.

Saya menyelesaikan pendidikan sekolah dasar di SD Katolik Kristus Raja Jayapura (1998 - 2005). Kemudian melanjutkan studi tingkat pertama di SMPN 1 Jayapura (2005 - 2008) dan selanjutnya saya menempuh pendidikan tingkat atas di SMAN 1 Manado (2008- 2011).

Setelah itu, di tahun 2012 saya melanjutkan pendidikan ke salah satu perguruan tinggi yang berada di Manado yaitu Universitas Sam Ratulangi Manado, dengan mengambil Program Studi S-1 Teknik Informatika di Jurusan Elektro Fakultas Teknik.