Rancang Bangun Aplikasi Kemiripan Dokumen Dengan Sumber – Sumber Internet

Virggi Eko Jacob[1)], Arie S. M. Lumenta[2)], Agustinus Jacobus[3)]
Teknik Elektro Universitas Sam Ratulangi, Jl.Kampus Bahu-Unsrat Manado, 95115
E-mail: virggijacob@gmail.com[1)], al@unsrat.ac.id[2)], a.jacobus@unsrat.ac.id[3)]

***Abstrak – E-learning* memberikan kemudahan bagi pengajar dan mahasiswa dalam kegiatan belajar menagajar, namun sering disalahgunakan oleh beberapa orang dalam menyelesaikan pekerjaan atau tugas-tugas yang diberikan. Penyalahgunaan sering dilakukan oleh mahasiswa, contohnya mahasiswa sering mengerjakan tugas kuliah dengan mengambil data dari sumber-sumber internet yang tidak jelas misalnya *wikipedia*, *wordpress* ataupun *blogspot*. Oleh karena itu perlu adanya aplikasi yang mampu mengukur tingkat kemiripan dokumen teks tugas mahasiswa yang dikumpulkan dengan data dari sumber-sumber internet yang tidak jelas yang dapat membantu mengidentifikasi setiap tugas mahasiswa dengan menguji dokumen tugas mahasiswa dengan artikel dari *wikipedia* menggunakan algoritama *Ratcliff/Obershelp*.**

Kata Kunci : *Information Retrieval, Ratcliff/Obershelp, Text Mining, Web Mining*.

Abstract _ E-learning makes it easy for teachers and students to learn learning activities, but is often misused by some people in completing work or assignments given. Abuse is often carried out by students, for example students often work on assignments by taking data from unclear internet sources such as wikipedia, wordpress or blogspot. Therefore it is necessary for an application that is able to measure the level of similarity of student task text documents collected with data from unclear internet sources that can help identify each student's task by testing student task documents with articles from wikipedia using the Ratcliff / Obershelp algorithm.

***Keywords : Information Retrieval, Ratcliff/Obershelp, Text Mining, Web Mining*.**

I. PENDAHULUAN

Perkembangan teknologi informasi saat ini berjalan sangat pesat, segala bidang relatif bisa dipandang sangat relevan berhubungan dengan komputer, dan tidak menutup mata betapa sangat berpengaruhnya komputer terhadap kemajuan zaman, baik segi kuantitas dan kualitas kepentingan manusia. Bagi institusi pendidikan pun peranan teknologi informasi sangatlah penting terutama untuk mencapai efisiensi dan efektifitas seluruh kegiatan yang terjadi antara pengajar, staff dan mahasiswa sehingga integrasi antara satu sistem dengan sistem lainnya dapat mencapai tujuan yaitu menghasilkan lulusan yang berkualitas.

Universitas Sam Ratulangi (UNSRAT) merupakan salah satu Universitas yang telah menerapkan sistem dalam hal administrasi maupun perkuliahan. Dalam hal perkuliahan UNSRAT telah menerapkan perkuliahan berbasis online yaitu *e-learning*. *E-learning* merupakan akronim dari *Electronic Learning*, adalah proses belajar mengajar jarak jauh yang menggunakan teknologi internet dan komputer sebagai media penyampaian informasi, digunakan untuk saling berbagi informasi antara dosen dan mahasiswa atau diskusi antara dosen dan mahasiswa baik materi kuliah maupun tugas kuliah. Hal ini tentu memberikan dampak yang positif dalam lingkungan akademis, tapi seiring dengan kemudahan yang diberikan terkadang disalahgunakan oleh beberapa orang dalam menyelesaikan pekerjaan atau tugas-tugas yang diberikan. Penyalahgunaan sering dilakukan oleh mahasiswa, contohnya dalam mengerjakan tugas kuliah, mahasiswa sering mengerjakan tugas kuliah dengan mengambil data dari sumber-sumber internet yang tidak jelas misalnya wikipedia, *wordpress* ataupun *blogspot*. Hal ini tentu sangatlah berdampak buruk bagi mahasiswa, selain itu dosen dan pengajar juga direpotkan dalam menganalisa satu per satu tugas mahasiswa, apalagi jika ingin membandingkan tugas dengan data dari sumber-sumber internet yang tidak jelas secara manual. Cara tersebut kurang efektif dan efesien mengingat jumlah mahasiswa yang tidak sedikit sehingga memerlukan waktu yang lebih lama untuk menganalisa tugas mahasiswa dengan data-data dari sumber internet yang tidak jelas.

Oleh karena itu perlu adanya aplikasi yang mampu mengukur tingkat kemiripan dokumen teks tugas mahasiswa yang dikumpulkan dengan data dari sumber-sumber internet yang tidak jelas.

A. *Information Retrieval*

Information Retrieval System atau sistem temu kembali informasi adalah suatu proses untuk mengidentifikasi, kemudian memanggil *(retrieve*) suatu dokumen dari suatu simpanan (*file*), sebagai jawaban atas pemintaan informasi. Pengertian lain menyatakan bahwa sistem temu kembali informasi adalah proses yang berhubungan dengan representasi, penyimpanan, pencarian dengan pemanggilan informasi yang relevan dengan kebutuhan informasi yang diinginkan pengguna.[1]

B. *Text Mining*

Text mining merupakan suatu proses yang melibatkan beberapa area teknologi. Namun secara umum proses-proses pada *text mining* mengadopsi proses *data mining*. Bahkan beberapa teknik dalam proses *text mining* juga menggunakan teknik-teknik data mining. Ada empat tahap proses pokok dalam text mining, yaitu pemrosesan awal terhadap teks (t*ext preprocessing*), transformasi teks *(text transformation*), pemilihan fitur (*feature selection*), dan penemuan pola (*pattern discovery*).

C. Text Preprocessing

Tahap ini melakukan analisis semantik (kebenaran arti) dan sintaktik (kebenaran susunan) terhadap teks. Tujuan dari pemrosesan awal adalah untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan lebih lanjut. Operasi yang dapat dilakukan pada tahap ini meliputi *part-of-speech* (*PoS*) *tagging*, menghasilkan parse tree untuk tiap-tiap kalimat, dan pembersihan teks. Selain itu pada tahapan ini biasanya juga dilakukan *case folding*, yaitu pengubahan karakter huruf menjadi huruf kecil. Contoh dari proses *part of speech* melakukan parsing terhadap seluruh kalimat dalam teks kemudian memberikan peran kepada setiap kata, misalnya : Ibu (subyek) pergi (predikat) ke (kata hub) pasar (keterangan). Hasil dari *part of speech tangging* dapat digunakan untuk *parse tree*, dimana masing-masing kalimat berdiri sebagai sebuah pohon mandiri. Untuk proses *parsing* sederhana tidak dibangun *parse tree* seperti cara sebelumnya. Pada proses *parsing* sederhana sistem akan memecah teks menjadi sekumpulan kata-kata, yang kemudian akan dibawa sebagai input tahap berikutnya pada proses *text mining*.[2]

1). Case folding

Tahap *case folding* adalah mengubah seluruh huruf dari "a" sampai dengan "z" dalam dokumen menjadi huruf kecil. Tidak semua dokumen konsisten dengan penggunaan huruf kapital. Maka dari itu *case folding* mengkonversi keseluruhan teks dalam dokumen menjadi huruf kecil.

2). *Cleansing*

Tahap cleansing digunakan untuk membersihkan dokumen dari simbol-simbol yang ada didalam dokumen.

3). Filtering

Tahap filtering adalah tahap mengambil kata-kata penting dari hasil *tokenizing* menggunakan algoritma *stopword removal* (membuang kata yang tidak memiliki makna). *Stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*.

4). *Penghapusan Spasi*

Pada proses penghapusan spasi setiap kata pada dokumen tugas dipisahkan, pada proses ini tahap yang dilakukan adalah memisahkan setiap kata yang dipisahkan oleh spasi, selanjutnya bagian dokumen yang memiliki karakter selain *alphabet* dan angka akan dipecah sesuai posisi karakter tersebut dan bagian yang hanya memiliki satu karakter non *alphabet* dan angka akan dibuang.

D. Text Transformation

Transformasi teks atau pembentukan atribut mengacu pada proses untuk mendapatkan representasi dokumen yang diharapkan. Pendekatan representasi dokumen yang lazim digunakan adalah model "*bag of words*" dan model ruang vektor (*vector space model*). Transformasi teks sekaligus juga melakukan pengubahan kata-kata ke bentuk dasarnya dan pengurangan dimensi kata di dalam dokumen. Tindakan ini diwujudkan dengan menerapkan stemming dan menghapus *stopwords*. *Stopwords* merupakan kata umum yang sering muncul dan dianggap tidak mempunyai makna. *Stopword* umumnya dimanfaatkan dalam task *information retrival*, termasuk oleh *Google*. Contoh *stopwords* untuk bahsa Inggris diantaranya "*of*","*the*". Sedangkan untuk bahasa Indonesia diantaranya "yang", "di", "ke", "dari", "juga".

E. Feature Selection

Pemilihan fitur (kata) merupakan tahap lanjut dari pengurangan dimensi pada proses transformasi teks. Walaupun tahap sebelumnya sudah melakukan penghapusan kata-kata yang tidak deskriptif (*stopwords*), namun tidak semua kata-kata di dalam dokumen memiliki arti penting. Oleh karena itu, untuk mengurangi dimensi, pemilihan hanya dilakukan terhadap kata-kata yang relevan yang benar-benar merepresentasikan isi dari suatu dokumen.

F. Pattern Discovery

Pattern discovery merupakan tahap penting untuk menemukan pola atau pengetahuan (*knowledge*) dari keseluruhan teks. Tindakan yang lazim dilakukan pada tahap ini adalah operasi *text mining*, dan biasanya menggunakan teknik-teknik *data mining*. Dalam penemuan pola ini, proses text mining dikombinasikan dengan proses-proses *data mining*.

G. Web Mining

Web mining bertujuan untuk menemukan informasi atau pengetahuan yang bermanfaat dari struktur *web hyperlinks*, halaman *web*, dan data penggunaan *web*. Berdasarkan jenis data primer yang digunakan dalam proses penggalian informasi, *web mining* dapat dikategorikan menjadi 3 jenis, yaitu: *web structure mining*, *web content mining*, dan *web usage mining*.[3]

1). Web Structure Mining

Web structure mining bertujuan untuk menemukan pengetahuan yang bermanfaat dari *hyperlinks*, di mana *hyperlinks* tersebut menggambarkan mengenai struktur *Web*. *Hyperlink* merupakan sebuah tautan yang terdapat pada suatu halaman *web* dan merujuk ke bagian lain pada halaman yang sama atau ke halaman lain. Pemanfaatan yang paling populer dari *web structure mining* adalah untuk menentukan tingkat otoritas suatu halaman *web*. Mesin pencari *Google* menggunakan informasi tersebut untuk menentukan urutan hasil pencariannya. Sebuah algoritma *web structure mining*, *Page Rank*, ditemukan oleh dua pendiri *Google*: *Larry Page* dan *Sergey Brin*. *Web structure mining* dapat juga diaplikasikan untuk mengkluster atau mengklasifikasikan halaman *web*.

2). *Web Content Mining*

Web content mining bertujuan untuk mengekstrak informasi atau pengetahuan yang bermanfaat dari isi halaman *web*. Terdapat dua kategori dalam *web content mining*: ekstraksi data terstruktur dan *text mining*. Ide mengenai ekstraksi data terstruktur berasal dari hasil pengamatan bahwa kebanyakan situs *web* menampilkan informasi penting yang berasal dari basisdata mereka menggunakan suatu template tertentu. Kita dapat mengidentifikasi template tersebut dengan mencari pola-pola yang berulang dalam halaman *web*. Selain data terstruktur, halaman *web* juga mengandung banyak sekali teks yang tidak terstruktur yang ditulis dalam bahasa natural. Penggalian informasi dari teks seperti ini merupakan *domain* dari *text mining*. Salah satu hal yang penting untuk dilakukan

dalam *text mining* adalah mengekstrak pendapat atau sentimen orang-orang dalam tinjauan produk, forum, jejaring sosial, dan *blog*.

3). *Web Using Mining*

Web usage mining bertujuan untuk menangkap dan memodelkan pola perilaku dan profil dari pengunjung *web*. Pola-pola tersebut dapat digunakan untuk meningkatkan pemahaman mengenai perilaku dari segmen-segmen pengunjung *web* yang berbeda, untuk memaksimalkan tata letak dan struktur dari situs *web*, dan untuk memberikan informasi yang sesuai dengan profil pengunjung. Berbeda dengan dua jenis *web mining* sebelumnya, sumber data primer dari web *usage mining* adalah log akses *web server*, bukan halaman *web*.

H. Personal Hypertext Preprocessor (PHP)

PHP adalah program aplikasi yang bersifat *server side*, yang artinya hanya dapat berjalan pada sisi *server* saja dan tidak dapat berfungsi tanpa adanya sebuah *server* di dalamnya. *PHP* juga bukan sebuah bahasa pemrograman yang lengkap.

Maksudnya program ini tidak menyertakan sebuah compiler tersendiri yang membuat program hasilnya menjadi program .exe yang dapat dijalankan sendiri. Program ini akan selalu membutuhkan sebuah server pendukung yang disebut *Web Server* dan program *PHP* itu sendiri untuk menjalankan semua *script* program.

PHP merupakan sebuah bahasa pemrograman yang berlisensi *open source*. *Script* ini dapat bercampur dengan *Script Tag HTML* sehingga karena kemampuannya tersebut, ia disebut sebagai bahasa yang embeded pada *Tag HTML*.[4]

I. MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standar *SQL* (*Structured Query Language*). *MySQL* juga dapat berperan sebagai *client/server*, yang *open source* dengan kemampuan dapat berjalan baik di OS (*Operating System*) manapun. Selain itu *database* ini memiliki kelebihan dibanding *database* lain, diantaranya adalah :

1. *MySQL* sebagai *Database Management System* (*DBS*)
2. *MySQL* sebagai *Relation Database Management System (RDBMS)*
3. *MySQL* adalah sebuah software database yang bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
4. *MySQL* merupakan database server, jadi dengan menggunakan database ini, dapat dihubungkan ke media internet sehingga dapat diakses dari jauh.
5. Selain menjadi *server* yang melayani permintaan, *MySQL* juga dapat melakukan *query* yang mengakses database pada *server*.
6. Mampu menerima *query* yang bertumpuk dalam satu permintaan atau yang disebut *Multi-Threading*.
7. Mampu menyimpan data yang berkapasitas besar hingga berukuran *gigabyte* sekalipun.
8. Memiliki kecepatan dalam pembuatan tabel maupun *update* tabel.
9. Menggunakan bahasa permintaan standar yang bernama *SQL* (*Structure Query Language*) yaitu sebuah bahasa permintaan yang distandarkan pada beberapa *database server* seperti *oracle*.

Dengan beberapa kelebihan yang dimiliki, *MySQL* menjadi sebuah program *database* yang sangat terkenal digunakan. Pada umumnya *MySQL* digunakan sebagai *database* yang diakses melalui *web*.[4]

J. JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman *JavaScript*, Standar ECMA-262 Edisi ke-3 - Desember 1999. *JSON* merupakan *format* teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga *C* termasuk *C*, *C++*, *C#*, *Java*, *JavaScript*, *Perl*, *Python* dll. Oleh karena sifat-sifat tersebut, menjadikan *JSON* ideal sebagai bahasa pertukaran-data.

JSON terbuat dari dua struktur:

1). Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.

2). Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini.[5]

K. Algoritma Ratcliff/Obershelp

L.

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 bagaimana sama 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* di nyatakan dengan rumus (1):

$$D_{ro} = \frac{2*K_m}{|S_1|+|S_2|}. \quad (1)$$

Km = Jumlah karakter yang sama
|S1| = Panjang dari String 1
|S2| = Panjang dari String 2

Mencari kesamaan kedua string PAGISARAPANNASIGRENG dan MAKANNASIGORENGPAGI, kita dapat menghitung nilai kemiripan dengan menggunakan (1).

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[16]	[17]	[18]	[19]	[20]	[21]	[22]
S1	P	A	G	I	S	A	R	A	P	A	N	N	A	S	I	G	O	R	E	N	G
S2	M	A	K	A	N	N	A	S	I	G	O	R	E	N	G	P	A	G	I		

1. Panjang dari string S1 :
|S1|=22
Panjang dari string S2 :
|S2|=20

2. Substring yang terpanjang yang dimiliki oleh kedua string adalah ANNASIGORENG. Maka ANNASIGORENG merupakan sebuah anchor, dan Km= |ANNASIGORENG|=12

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[16]	[17]	[18]	[19]	[20]	[21]	[22]
S1	P	A	G	I	S	A	R	A	P	A	N	N	A	S	I	G	O	R	E	N	G
S2	M	A	K	A	N	N	A	S	I	G	O	R	E	N	G	P	A	G	I		

3. Di sebalah kiri dan kanan dari anchor tersisa kumpulan huruf PAGISARAP dan MAKPAGI. Substring yang terpanjang dan memiliki kesamaan dari kumpulan huruf tersebut adalah PAGI. Maka, Km= 12+|PAGI|=16

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[16]	[17]	[18]	[19]	[20]	[21]	[22]
S1	P	A	G	I	S	A	R	A	P	A	N	N	A	S	I	G	O	R	E	N	G
S2	M	A	K	A	N	N	A	S	I	G	O	R	E	N	G	P	A	G	I		

4. PAGI merupakan substring yang berada di awal dari string S1 dan akhir dari string S2, disebelah kiri substring tersebut sudah tidak terdapat huruf lagi. Pada sebelah kanan dari PAGI, kita memiliki SARAP di dalam *string* S1 dan pada string S2 memiliki MAK pada *substring* S2. Sudah tidak ada kesamaan *string* lagi dalam S1 dan S2. Maka Km tetap sama dan kita lanjutkan pada karakter sebelah kanan dari *anchor*.
5. Disebelah kanan dari *anchor* tidak terdapat lagi karakter. Karena sudah tidak ada karkater, maka nilai dari Km tetap 16. Jadi kita memiliki semua data yang kita perlukan untuk menghitung nilai dari *Ratcliff/Obershelp*.
   Penilaian *Ratcliff/Obershelp* untuk *string* PAGISARAPANNASIGORENG dan MAKANNASIGORENGPAGI adalah:

$$D_{ro} = \frac{2 \times 16}{22 + 20} = \frac{32}{42} = 0.761$$

Jadi, dari kedua string PAGISARAPANNASIGORENG dan MAKANNASIGORENGPAGI memiliki nilai kesamaan 0.761 yang bisa dikatakan sama.

II. METODOLOGI PENELITIAN

A. *Obyek Penelitian*

Obyek penelitian ini dilakukan dengan menggunakan berkas tugas dari mahasiswa yang di uji dengan *wikipedia* menggunakan algoritma *Ratcliff/Obershelp*.

B. *Pengumpulan Data*

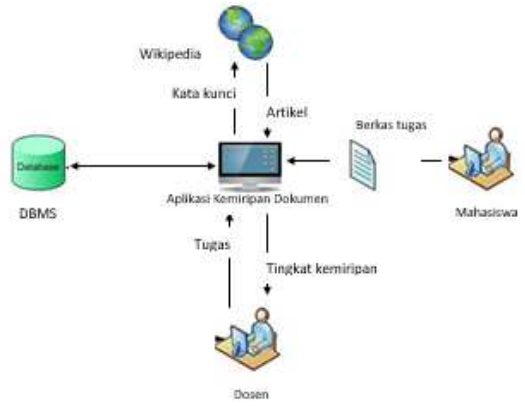
Dalam penelitian ini saya hanya menggunakan metode pengambilan data secara primer yaitu studi literatur yang merupakan pengumpulan data dengan cara mencari referensi-referensi terkait yang dibutuhkan untuk penelitian. Referensi tersebut dapat berupa buku-buku, jurnal-jurnal, tulisan penelitian dan juga artikel-artikel dari internet yang memiliki kaitan dengan kasus yang sedang dilakukan dalam penelitian.

C. *Desain Arsitektur Sistem*

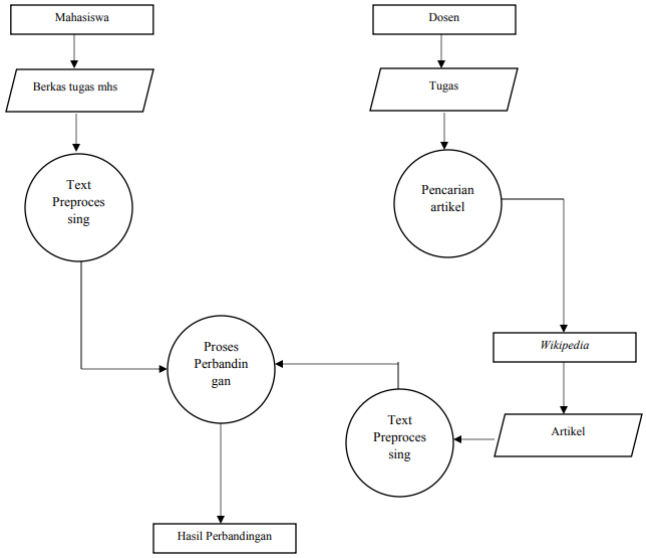
Pada tahap desain arsitektur sistem ini akan menjelaskan apa saja yang digunakan dan proses yang terjadi pada aplikasi. Dimana dosen mengakses sistem yang kemudian memberikan tugas dari matakuliah yang di ampuh, beserta kata kunci yang akan digunakan untuk mencari artikel dari *wikipedia*, mahasiswa kemudian mengakses sistem dan mengerjakan tugas dari matakuliah yang dipilih dengan menunggah berkas tugas ke dalam aplikasi kemudian sistem akan langsung melakukan pengujian dengan artikel dari *wikipedia*, hasil dari pengujian akan langsung tersimpan dalam *database*.

D. *Diagram Prinsip Kerja*

Pada diagram prinsip kerja ini, akan menjelaskan secara umum prinsip kerja dari sistem yang dibangun, sistem dibangun dalam sebuah aplikasi *web,* diamana terdapat dua actor yang dapat mengakses sistem ini yaitu mahasiswa dan dosen. Mahasiswa hanya dapat mengakses aplikasi dan melihat setiap tugas dari matakuliah yang dipilih dan diminta untuk mengunggah file tugas dalam bentuk ekstensi *pdf** ke dalam aplikasi, kemudian aplikasi akan melakukan konversi berkas tugas mahasiswa *pdf** ke *txt**, setalah itu file *txt** tugas mahasiswa akan melewati tahap *preprocessing*, kemudian file **txt* tugas mahasiswa akan diuji dengan artikel dari *wikipedia* sesuai dengan kata kunci yang diberikan Dosen, sebelum masuk dalam tahap pengujian artikel terlebih dahulu di konversi dari file **html* ke **txt* kemudian artikel *wikipedia* dalam bentuk **txt* akan melewati juga tahap *preprocessing*, setelah itu kedua file **txt* tugas mahasiswa dan **txt* artikel *wikipedia* akan melewati tahap pengujian menggunakan algoritma yang diterapkan yaitu *Ratcliff/Obershelp,* setelah selesai melakukan pengujian maka hasil kemiripan dari berkas tugas mahasiswa dengan artikel dari *wikipedia* akan langsung tersimpan dalam database. Nilai kemiripan dari hasil pengujian hanya bisa dilihat oleh dosen.



Gambar 1 Desain Arsitektur Sistem



Gambar 2 Diagram Prinsip Kerja Sistem

III. HASIL DAN PEMBAHASAN

A. *Pengujian Aplikasi*

1. *Pengujian Model*

Pada tahap ini perangkat lunak yang di bangun akan dilakukan pengujian sebagai salah satu cara untuk mengetahui apakah perangkat lunak sesuai dengan hasil yang di harapkan. Terdapat 2 pengujian yang dilakukan yaitu :

a. Pengujian Waktu Eksekusi

Pengujian 5 dokumen tugas yang di unggah dengan 1 konten di internet sesuai kata kunci yang dicari. Waktu yang dibutuhkan saat mengunggah tugas adalah 12.1 detik. Untuk hasil keseluruhan pengujian waktu eksekusi dapat di lihat pada tabel 1 pengujian waktu eksekusi.

TABEL I PENGUJIAN WAKTU EKSEKUSI

No	Dokumen	Total Dokumen	Banyak Karakter	Waktu Eksekusi (detik)
1	Tugas 1	1	871	11,5
2	Tugas 2	1	2.405	20,0
3	Tugas 3	1	4.325	1.05,5
4	Tugas 4	1	7.633	1.28,6
5	Tugas 5	1	7.633	1.32,3

TABEL 2 PENGUJIAN ALGORITMA

No	Dokumen	*Wikipedia*	Prediksi	Nilai	Dalam Desimal
1	Dok 1	Artikel	0%	14%	0.14
2	Dok 2	Artikel	25%	35%	0.35
3	Dok 3	Artikel	50%	67%	0.67
4	Dok 4	Artikel	75%	72%	0.72
5	Dok 5	Artikel	100%	54%	0.84

b. Pengujian Algoritma

Pengujian algoritma merupakan pengujian yang dilakukan untuk mengetahui tingkat akurasi dari algoritma yang digunakan yaitu *Ratcliff/Obershelp*. Dalam pengujian ini terdapat dokumen mahasiswa yang akan bandingkan dengan artikel dari *Wikipedia* sesuai kata kunci dari tugas yang diberikan. dokumen yang akan dibandingkan sengaja dibuat dengan tingkat kesamaan yaitu 0%, 25%, 50%, 75% dan 100%. Hasil dari pengujian algoritma dapat di lihat pada tabel 2.

IV. PENUTUP

A. *Kesimpulan*

Aplikasi pendeteksi kemiripan telah berhasil di buat dan mampu mengukur tingkat kemiripan dokumen teks tugas mahasiswa dengan sumber-sumber internet (*wikipedia*), dengan menggunakan algoritma pengujian *Ratcliff/Obershelp*. Pengujian berdasarkan 5 dokumen mahasiswa (tabel 1 dan 2) yang di uji kemiripannya dengan artikel dari wikipedia, dalam pengujian berkas tugas dengan artikel dimiliki hasil yaitu ada yang mendekati prediksi, ada yang di bawah prediksi, dan ada yang melewati prediksi. Jadi bisa dikatakan bahwa aplikasi berfungsi sebagaimana yang diharapkan. Waktu eksekusi yang diperlukan untuk mendeteksi kemiripan 80% - 100% yaitu 1 menit 32 detik. Banyaknya karakter dalam dokumen tugas mahasiswa mempengaruhi waktu eksekusi dari proses pengujian dengan artikel *wikipedia*.

B. *Saran*

Perlu adanya penelitian lanjutan untuk menentukan algoritma atau metode yang terbaik dalam pengujian kemiripan dokumen tugas mahasiswa dan *wikipedia*. Pengujian juga bisa dikembangkan untuk mendeteksi gambar, dan di uji dengan sumber internet lain (*blogspot* atau *worldpress*).

KUTIPAN

[1] K. D. Putung, A. Lumenta, and A. Jacobus, "Penerapan Sistem Temu Kembali Informasi Pada Kumpulan Dokumen Skripsi," *E-journal Tek. Inform.*, 2016.

[2] F. Z. Tala, "A Study of Stemming E□ects on Information Retrival in Bahasa Indonesia," *Master Log. Proj. Inst. Logic, Lang. Comput. Univ. van Amsterdam Netherlands*, 2003.

[3] "Web data mining: exploring hyperlinks, contents, and usage data," *Choice Rev. Online*, 2013.

[4] S. Pebrianto, "Nugroho, Bunafit . Latihan Membuat Aplikasi Web PHP dan MySQL Dengan Dreamweaver MX (6, 7, 2004) dan 8, Gava Media, Yogyakarta,2008," *Nugroho, Bunafit*, 2008.

[5] J. Friesen and J. Friesen, "Introducing JSON," in *Java XML and JSON*, 2019.

[6] A. Menggunakan, / Ratcliff, Y. Obershelp, J. Lady, A. Sinsuw, and A. Jacobus, "Rancang Bangun Aplikasi Deteksi Kemiripan Dokumen Teks," *J. Tek. Inform.*, 2017.

SEKILAS TENTANG PENULIS

Saya bernama Virggi Eko Jacob. Lahir pada tanggal 20 Maret 1994 di Manado.

Saya mulai menempuh pendidikan di SD Negeri 30 Manado (1999-2006). Kemudian melanjutkan ke SMP Negeri 2 Manado (2006-2009). Setelah itu saya menempuh pendidikan di SMA Negeri 4 Manado (2009-2012). 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. Penulis membuat skripsi demi memenuhi syarat sarjana (S1) dengan penelitian berjudul Rancang Bangun Aplikasi Kemiripan Dokumen Dengan Sumber – Sumber Internet yang dibimbing oleh Ir. Arie S.. M. Lumenta, ST., MT, dan Agustinus Jacobus, ST., MCs sehingga pada tanggal 3 Mei 2019 resmi lulus di Teknik Elektro Program Studi Teknik Informatika Universitas Sam Ratulangi Manado dengan menyandang gelar Sarjana Komputer (S.Kom) dan mendapatkan hasil predikat sangat memuaskan.