Pencarian Abstrak Tugas Akhir Mahasiswa Berdasarkan Tingkat Kemiripan Menggunakan Algoritma *Winnowing* dan *Jaccard Similarity* pada Universitas Budi Luhur

Wahyu Desena[1], Achmad Solichin [2*]
Program Studi Teknik Informatika, Fakultas Teknologi Informasi
Universitas Budi Luhur
Jl. Ciledug Raya, Petukangan Utara, Jakarta Selatan, 12260
achmad.solichin@budiluhur.ac.id*

Abstrak. Dokumen skripsi merupakan dokumen yang merepresentasikan penelitian yang dilakukan oleh mahasiswa jenjang strata satu. Untuk menghasilkan skripsi yang baik dibutuhkan studi literatur untuk mendukung penelitian tersebut. Pada Universitas Budi Luhur, sudah tersedia sistem pencarian literatur dalam bentuk dokumen skripsi. Namun demikian, pencarian data masih terbatas berdasarkan kesamaan judul dengan kata kunci yang diberikan. Hal tersebut mengakibatkan hasil pencarian tidak terlalu akurat. Oleh karena itu, pada penelitian ini diusulkan sistem yang mampu menyajikan hasil pencarian berdasarkan tingkat kemiripan dokumen. Metode yang digunakan adalah algoritma *Winnowing* berbasis *N-Gram* dengan perhitungan kemiripan metode *Jaccard Similarity*. Berdasarkan hasil pengujian, nilai k-gram dan w-gram mempengaruhi persentase kemiripan dokumen yang mana nilai terbaik adalah k-gram=3 dan w-gram=4. Prototipe sistem yang dihasilkan dapat menyajikan hasil pencarian beserta nilai kemiripan dokumen abstrak tugas akhir yang diinputkan dengan dokumen yang tersimpan di *repository*.

Kata Kunci: abstrak tugas akhir, pencarian dokumen, tingkat kemiripan, winnowing, jaccard similarity

1 Pendahuluan

Skripsi atau tugas akhir merupakan salah satu penilaian akhir bagi mahasiswa jenjang Strata-1 di suatu perguruan tinggi, termasuk di Universitas Budi Luhur (UBL). Dokumen skripsi merupakan dokumen yang merepresentasikan penelitian yang dilakukan oleh mahasiswa. Untuk menghasilkan skripsi yang baik dibutuhkan studi literatur untuk mendukung penelitian tersebut. Pada Universitas Budi Luhur, sudah tersedia sistem pencarian literatur dalam bentuk dokumen skripsi. Mahasiswa dapat melakukan pencarian dokumen skripsi dan karya ilmiah lainnya melalui sistem pencarian yang disediakan oleh Perpustakaan Universitas Budi Luhur. Namun demikian, pencarian data masih terbatas berdasarkan kesamaan judul dengan kata kunci yang diberikan. Hal tersebut mengakibatkan hasil pencarian tidak terlalu akurat karena seringkali kata kunci yang diberikan tidak selalu ditemukan pada judul. Oleh karena itu, diperlukan sebuah metode pencarian yang lebih akurat dan tidak hanya berdasarkan kesamaan kata pada judul.

Pencarian kesamaan diantara dua dokumen atau lebih membutuhkan suatu algoritma untuk menghitung similaritas dari teks yang terkandung di dalamnya. Secara umum kesamaan teks terbagi menjadi 4 pendekatan, yaitu berbasis *string*, korpus, *knowledge*, dan campuran (*hybrid*) [1]–[3]. Masing-masing pendekatan memiliki kelebihan dan kekurangan. Diantara keempat pendekatan tersebut, perhitungan kesamaan teks berdasarkan *string* merupakan metode yang lebih banyak digunakan, lebih cepat dan lebih mudah diimplementasikan [1], [4]. Akurasinya pun cukup baik. Perkembangan teknik perhitungan kesamaan teks merupakan bagian penting dari bidang ilmu pengolahan bahasa alami.

Saat ini juga telah banyak penelitian yang menerapkan berbagai metode kesamaan teks, terutama yang berbasis teks. Apriyanto dan Aribowo [5] mengusulkan sebuah aplikasi pengecekan similaritas judul skripsi menggunakan metode *Cosine Similarity*. Walaupun hasilnya sudah mampu menghasilkan tingkat kesamaan, namun penelitian baru menerapkan pengecekan similaritas pada judul skripsi saja. Metode *Cosine Similarity* juga digunakan pada beberapa penelitian lainnya untuk menghitung similaritas teks [6], [7]. Metode similaritas teks lainnya yang sering digunakan adalah *Jaccard Similarity*, seperti pada penelitian oleh [8], [9]. *Jaccard Similarity* merupakan algoritma yang membandingkan dua dokumen dengan menghitung kesamaan dari dokumen tersebut.

ISSN : 2655-139X (ONLINE)
ISSN : 0216-4221 (PRINT)

Selain algoritma similaritas, dalam menghitung tingkat kesamaan teks di antara dua dokumen, tahap pengolahan awal untuk mendapatkan fitur atau representasi sebuah teks juga memegang peranan yang penting. Beberapa metode yang sering digunakan adalah metode berbasis *N-Gram*, antara lain algoritma *Winnowing*, *Rabin-Karp* dan *Rolling Hash* [8], [10], [11]. N-gram merupakan sub-urutan dari sejumlah item dalam teks yang diberikan. Algoritma kemiripan *N-gram* membandingkan *n-gram* dari setiap karakter atau kata dalam dua *string*. Jarak dihitung dengan membagi jumlah *n-gram* yang serupa dengan jumlah maksimum *n-gram* [1]–[3].

Pada penelitian ini dikembangkan sebuah sistem menggunakan metode *N-gram* algoritma *Winnowing* dengan metode similaritas *Jaccard*. *N-gram* merupakan algoritma yang digunakan untuk mengambil potongan huruf sejumlah n dan mempunyai pengaruh yang tinggi terhadap hasil similaritas. Sistem yang dikembangkan mampu menampilkan hasil pencarian dokumen berdasarkan kemiripan dari dokumen yang dicari dengan dokumen yang terlah tersimpan di basis data. Pada penelitian ini secara khusus sistem dirancang menggunakan *dataset* yang berasal dari dokumen abstrak tugas akhir mahasiswa di Universitas Budi Luhur. Dengan adanya sistem ini diharapkan pencarian dokumen tugas akhir untuk keperluan studi literatur penelitian atau keperluan lain menjadi lebih akurat dan lebih tepat.

2 Metode Penelitian

2.1. Data Penelitian

Dataset yang digunakan dalam penelitian ini bersumber dari data dokumen abstrak tugas akhir mahasiswa Universitas Budi Luhur yang berbentuk dokumen *Portable Document Format* (PDF). Jumlah dokumen yang digunakan dalam penelitian ini adalah 200 dokumen yang berasal dari tugas akhir mahasiswa Fakultas Teknologi Informasi tahun 2015. Data tersebut kemudian diolah melalui tahap *preprocessing* sehingga data tersebut bisa digunakan sebagai *dataset*. Gambar 1 menyajikan contoh dokumen abstrak tugas akhir mahasiswa yang digunakan dalam penelitian ini.

Perpustakaan Universitas Budi Luhur

ABSTRAK

IMPLEMENTASI ALGORITMA *DEPTH-FIRST-SEARCH* SEBAGAI *MAZE-BASED LEVEL GENERATOR* PADA PROTOTIPE GAME 2D *TOP DOWN SHOOTER PROPOSTEROUS DEFIANCE* BERBASIS *MOBILE*

Oleh : Fadel Achmad Assegaf (1511500082)

Video Game pada umumnya merupakan jenis hiburan yang disukai oleh hampir semua kalangan orang saat ini, mulai dari kalangan anak-anak hingga kalangan dewasa. *Video game* juga sering digunakan untuk menghilangkan kejenuhan dalam menjalankan aktivitas sehari-hari dan juga untuk melatih diri seseorang untuk lebih berkonsentrasi dan fokus pada *video game* tersebut. Pembuatan aset-aset level pada *Video game* biasanya dibuat secara manual dan desain dan alur level akan selalu sama ketika level tersebut dimuat lagi. Maka dari itu, pembuatan *Video game* ini yang dikembangkan melalui Unity *Engine* bersifat 2D, dengan genre *top down shooter* dengan judul game *Proposterous Defiance* menggunakan Algoritma *Depth-First-Search* dalam pembuatan level secara otomatis. Level nantinya dapat terbuat dengan arah jalur yang acak dari *start* hingga menuju *finish*. Algoritma ini digunakan untuk pembuatan labirin *perfect* sehingga tidak terdapat jalan yang berulang, selain itu tidak ada sel yang terisolasi pada labirin. Algoritma DFS merupakan algoritma yang digunakan untuk membuat sebuah labirin *perfect*. Pembuatan labirin pada game *Proposterous Defiance* yang selalu *random* atau acak dengan algoritma *Depth-First-Search* dapat memberikan kesan *fresh* setiap kali level dimainkan, karena pembuatan level dalam *video game* ini berbentuk labirin *perfect* yang mana selalu memiliki alur yang berbeda setiap kali level dimainkan. Metode pengembangan menggunakan *Multimedia Development Life Cycle* (MDLC), yaitu suatu tahap yang di dalamnya terdapat pengolahan konsep, perancangan, pengumpulan bahan, pembuatan, testing, dan yang terakhir distribusi. Dengan MDLC ini, *video game* yang dibuat akan mendapatkan hasil akhir yang diharapkan, akurat, serta optimal. Berdasarkan penelitian yang telah dijalankan, dengan menggunakan metode MDLC dapat dsimpulkan penggunaan algoritma *Depth-First-Search* dalam pembuatan *maze-based level-generator* pada game 2D *Proposterous Defiance* telah terbukti dapat membuat alur jalan labirin yang selalu acak setiap kali level pada game dimuat.

Kata Kunci : *Depth-First-Search*, Labirin, MDLC, Unity 3D, *Video Game*

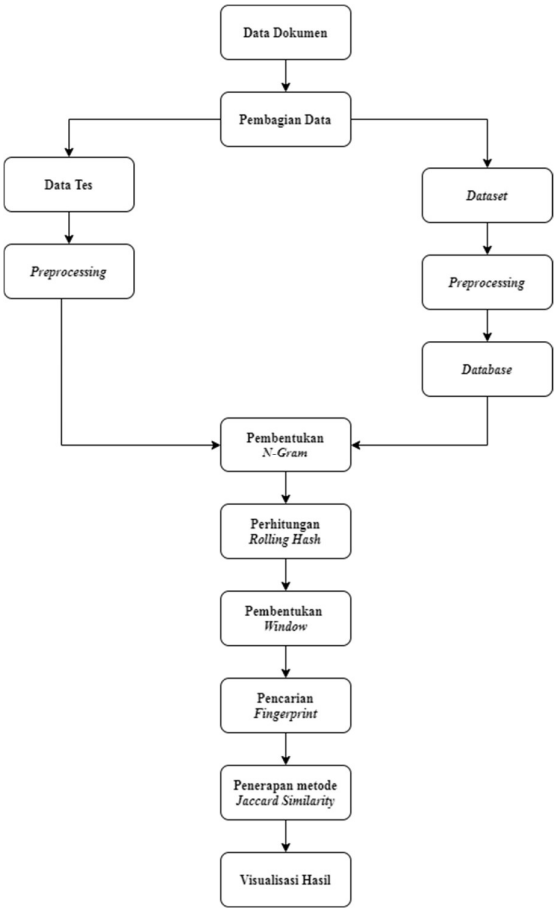
xv+70 halaman; 44 gambar; 2 tabel; 1 lampiran

Gambar 3. Contoh data abstrak tugas akhir mahasiswa

2.2. Penerapan Metode

Untuk membangun sistem pencarian dokumen berdasarkan similaritas menggunakan metode *Winnowing* dan *Jaccard similarity,* terdapat beberapa tahapan seperti tersaji pada Gambar 2. Dokumen abstrak tugas akhir yang telah terkumpul dibagi menjadi dua bagian yaitu sebagai *dataset* dan data uji. *Dataset* berfungsi sebagai data acuan dalam proses pencarian dokumen, sedangkan data uji berfungsi sebagai dokumen *query* dalam pencarian.

ISSN : 2655-139X (ONLINE)
ISSN : 0216-4221 (PRINT)



Gambar 4. Langkah Implementasi Metode

2.2.1. Data Dokumen

Pada tahap ini dilakukan proses pengumpulan data dokumen abstrak dari skripsi mahasiswa Budi Luhur. Dimana data dokumen ini ada yang dijadikan sebagai *dataset* dan ada juga yang menjadi data tes. *Dataset* ini berasal dari dokumen abstrak skripsi mahasiswa yang sudah dipublikasikan oleh perpustakaan Budi Luhur. Sedangkan data tes berasal dari dokumen abstrak mahasiswa yang akan menjalankan skripsi, sehingga mahasiswa bisa melakukan pengecekan similaritas di dokumen abstraknya.

2.2.2. Preprocessing

Pada tahapan *preprocessing* ini, dilakukan beberapa proses untuk mendapatkan *dataset* yang bersih, sehingga proses pengujian tingkat *similarity* lebih tepat dan akurat. Proses *preprocessing* yang digunakan pada penelitian ini diantaranya yaitu :

1. *Case Folding*. *Case Folding* merupakan proses penyetaraan kata yang mengandung huruf besar untuk diubah menjadi huruf kecil, misalnya Tujuan menjadi tujuan, Adapun menjadi adapun, dan seterusnya.
2. *Cleaning. Cleaning* merupakan proses untuk menghapus karakter selain a sampai z atau karakter selain huruf, karakter tersebut dihilangkan, seperti contoh 8266, / (garis miring), dan seterusnya sehingga hanya menyisakan karakter huruf saja.
3. Mengganti *Slang word*. *Slang word* merupakan proses mengganti kata tidak baku seperti “analisa” dan “rejeki”, kata yang tidak baku tersebut biasanya berupa singkatan atau bahasa yang kekinian, untuk itu supaya kata-kata dalam teks setara dengan EYD, maka kata tersebut digantikan dengan kata baku yang seharusnya yaitu menjadi “analisis” dan “rezeki”, pergantian kata-kata ini berdasarkan kamus yang terdapat dalam *library slang word*.

ISSN : 2655-139X (ONLINE)
ISSN : 0216-4221 (PRINT)

4. Menghapus *Stop word*. Dari penelitian sebelumnya, diperoleh kumpulan kata yang termasuk ke dalam *stoplist* yaitu kata umum yang dianggap tidak terlalu memiliki makna yang penting dan kemunculan kata ini sangat tinggi frekuensinya. Contoh kata yang dihilangkan antara lain kata "untuk", "dan", dan "bagi".
5. Menghapus Spasi. Pada proses ini dilakukan penghapusan spasi dari setiap kata, untuk melakukan proses selanjutnya, yaitu pada proses pencarian tingkat *similarity* sebuah dokumen.

2.2.3. N-Gram

N-gram adalah *substring* penggabungan karakter sejumlah *k* pada teks dokumen. Dalam menentukan hasil similaritas dokumen menggunakan metode *n-gram*, dokumen atau sekumpulan kata akan diproses dan akan dibentuk sebanyak *n-gram* atau memisahkan *string* sepanjang n yang akan dihitung pergeserannya secara terus menerus ke depan sejumlah nilai n sampai akhir dokumen. Sebagai contoh *n-gram* dari kalimat N = "Sortir berbasis Arduino", dengan nilai N=3 maka menjadi "sor ort rti tir irb rbe ber erb rba bas asi sis isa sar ard rdu dui uin ino".

2.2.4. Algoritme *Winnowing*

Winnowing adalah algoritma yang digunakan untuk melakukan proses pengecekan kesamaan kata (*document fingerprinting*) untuk mengidentifikasi tingkat *similarity.*

1. *Rolling Hash.* Pada tahapan ini dilakukan perhitungan *rolling hash* untuk mencari nilai *hash* dari setiap *string* yang sudah dipotong pada tahapan *n-gram*. Setiap *string* diubah menjadi ASCII lalu dihitung dengan persamaan (1) yang mana c1..ck merupakan nilai ASCII dari huruf pertama sebuah k-gram, b merupakan bilangan prima dan k adalah nilai *k-gram* yang digunakan. Setelah hasil dari hash pertama didapatkan selanjutnya digunakan rumus *rolling hash* seperti pada Persamaan (2) untuk menghitung nilai *hash* ke dua dan seterusnya tanpa menghitung ASCII yang sudah dihitung sebelumnya. Hasil dari perhitungan sebelumnya dikurangi hasil *hash* dari huruf pertama *gram* sebelumnya lalu dikali dengan hasil *hash* dari perhitungan ASCII terakhir dari gram sekarang

$$H\,(c1\ldots ck) = c1 * b^{(k-1)} + c2 * b^{(k-2)} + \ldots + ck * b^{(k-k)} \quad (1)$$

$$H\,(c2..ck+1) = \left(H(c1..cK) - c1 * b^{(k-1)}\right) * b + c^{(k+1)} \quad (2)$$

2. Pembentukan *Window.* Setelah nilai *hash* ditemukan semua maka langkah selanjutnya adalah pembentukan *window*. Nilai dari *rolling hash* akan dikelompokan berdasarkan nilai *w-gram*. Pada kasus ini nilai *window* yang dipergunakan adalah w=3.
3. Nilai *Fingerprint.* Proses selanjutnya setelah pembentukan *window* adalah mencari nilai *fingerprint.* Nilai *fingerprint* ditentukan berdasarkan nilai terkecil dari setiap *window* yang ada.

2.3. *Jaccard Similarity*

Pada tahapan ini untuk mencari nilai *similarity* pada sebuah dokumen dengan rumus *Jaccard similarity* dengan melihat *irisan* dan *union* dari *fingerprint* antar dua dokumen. Persamaan (3) merupakan persamaan *jaccard similarity* dari dokumen teks X dan Y. Irisan dari dokumen X dan Y dibagi dengan *union* dari dokumen X dan Y kemudian dikali 100% maka menghasilkan persentase *similarity* dari dokumen X dan Y.

$$Similarity(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} * 100\% \quad (3)$$

2.4. Rancangan Pengujian

Pengujian dilakukan untuk mengetahui tingkat similaritas sebuah dokumen abstrak dengan dokumen yang tersimpan di basis data (*repository*). Pada penelitian ini dilakukan pengujian terhadap nilai *k-gram* dan *w-gram* yang sama, dan pengujian terhadap berbagai variasi nilai *k-gram* dan *w-gram* yang berbeda. Tujuan pengujian untuk melihat pengaruh dari nilai *k-gram* dan *w-gram* terhadap hasil tingkat similaritas yang dihasilkan. Pengujian dilakukan dengan sejumlah dokumen abstrak tugas akhir yang dipilih secara acak. Pada langkah pengujian ini akan dijelaskan mengenai proses yang berjalan pada sistem yang dibangun, mulai dari *upload* dokumen, menjadikan dokumen sebagai *dataset* dan data tes sampai menampilkan hasil perhitungan tingkat *similarity.*

ISSN : 2655-139X (ONLINE)
ISSN : 0216-4221 (PRINT)

3 Hasil dan Pembahasan

3.1. Implementasi Metode

3.1.1. Data Dokumen

Data dokumen merupakan tahap untuk pengumpulan *dataset* dan proses pengujian data tes. Data yang digunakan berupa dokumen abstrak tugas akhir mahasiswa Universitas Budi Luhur dalam bentuk *PDF* yang kemudian dikonversi ke dalam bentuk teks. Data dokumen diperoleh dari perpustakaan Universitas Budi Luhur. Data dokumen yang sudah berbentuk teks akan dilakukan tahap *preprocessing* pembentukan *n-gram, rolling hash,* pembentukan *window* dan pencarian *fingerprint.* Sehingga diketahui tingkat *similarity* dari sebuah dokumen. Pada Tabel 1 merupakan contoh dokumen abstrak yang sebelumnya berupa file *PDF* kemudian dikonversi menjadi teks.

Tabel 1. Contoh data abstrak hasil ekstraksi file dokumen abstrak tugas akhir

ID	Isi abstrak
1	IMPLEMENTASI DATA MINING UNTUK MEMPREDIKSI PENERIMAAN PERMOHONAN CALON NASABAH ASURANSI KESEHATAN DENGAN MENGGUNAKAN ALGORITMA NAÏVE BAYES BERBASIS WEB PADA PT. ASURANSI SINARMAS Oleh : Lady Caesar Sevika Detale (1511503581) Kesehatan merupakan bagian yang amat penting bagi kehidupan manusia. PT. Asuransi Sinarmas adalah salah satu asuransi yang menyediakan produk asuransi kesehatan. Banyaknya masyarakat yang mendaftar dan mengajukan permohonan asuransi kesehatan, membuat pihak asuransi harus lebih memperhatikan mengenai penerimaan permohonan asuransi kesehatan. Hal tersebut dikarenakan pemberian jaminan kesehatan memiliki resiko yang cukup tinggi. Di sisi lain, banyaknya nasabah yang sering terlambat dan kurang peduli membayar premi, membuat pihak asuransi mengalami kerugian semakin besar. Maka dari itu, upaya untuk meminimalisir adanya permasalahan tersebut, pihak asuransi harus lebih memperhatikan lagi terhadap penerimaan permohonan calon nasabah.

3.1.2. Preprocessing

Setelah data dokumen didapat, maka dokumen akan memasuki tahap *preprocessing* pada dokumen yang sudah menjadi teks, proses yang dilakukan yaitu, *case folding,* menghapus karakter selain a-z, mengganti *slang word,* menghapus *stop word,* dan menghilangkan spasi antar karakter.

1. *Case Folding.* Pada Tabel 2 merupakan proses untuk merubah isi teks dokumen menjadi huruf kecil, melalui proses *case folding,* sehingga isi teks menjadi huruf kecil semua. Berikut contohnya.

Tabel 2. Contoh penerapan tahap case folding

Abstrak sebelum	**Abstrak setelah casefolding**
IMPLEMENTASI DATA MINING UNTUK MEMPREDIKSI PENERIMAAN PERMOHONAN CALON NASABAH ASURANSI KESEHATAN DENGAN MENGGUNAKAN ALGORITMA NAÏVE BAYES BERBASIS WEB PADA PT. ASURANSI SINARMAS Oleh : Lady Caesar Sevika Detale (1511503581) Kesehatan merupakan bagian yang amat penting bagi kehidupan manusia. PT. Asuransi Sinarmas adalah salah satu asuransi yang menyediakan produk asuransi kesehatan. Banyaknya masyarakat yang mendaftar dan mengajukan permohonan asuransi kesehatan, membuat pihak asuransi harus lebih memperhatikan mengenai penerimaan permohonan asuransi kesehatan. …	implementasi data mining untuk memprediksi penerimaan permohonan calon nasabah asuransi kesehatan dengan menggunakan algoritma naïve bayes berbasis web pada pt. asuransi sinarmas oleh : lady caesar sevika detale (1511503581) kesehatan merupakan bagian yang amat penting bagi kehidupan manusia. pt. asuransi sinarmas adalah salah satu asuransi yang menyediakan produk asuransi kesehatan. banyaknya masyarakat yang mendaftar dan mengajukan permohonan asuransi kesehatan, membuat pihak asuransi harus lebih memperhatikan mengenai penerimaan permohonan asuransi kesehatan. …

2. *Cleaning.* Pada Tabel 3 merupakan proses untuk menghapus karakter selain a-z. Sehingga isi teks yang bukan huruf a-z akan dihapus, seperti angka, simbol, dan seterusnya. Berikut contohnya.

Tabel 3. Contoh penerapan tahap cleaning

Abstrak sebelum	**Abstrak setelah proses**
implementasi data mining untuk memprediksi penerimaan permohonan calon nasabah asuransi kesehatan dengan menggunakan algoritma naïve bayes	implementasi data mining untuk memprediksi penerimaan permohonan calon nasabah asuransi kesehatan dengan menggunakan algoritma na ve

ISSN : 2655-139X (ONLINE)
ISSN : 0216-4221 (PRINT)

berbasis web pada pt. asuransi sinarmas oleh : lady caesar sevika detale (1511503581) kesehatan merupakan bagian yang amat penting bagi kehidupan manusia. pt. asuransi sinarmas adalah salah satu asuransi yang menyediakan produk asuransi kesehatan. banyaknya masyarakat yang mendaftar dan mengajukan permohonan asuransi kesehatan, membuat pihak asuransi harus lebih memperhatikan mengenai penerimaan permohonan asuransi kesehatan. …	bayes berbasis web pada pt asuransi sinarmas oleh lady caesar sevika detale kesehatan merupakan bagian yang amat penting bagi kehidupan manusia pt asuransi sinarmas adalah salah satu asuransi yang menyediakan produk asuransi kesehatan banyaknya masyarakat yang mendaftar dan mengajukan permohonan asuransi kesehatan membuat pihak asuransi harus lebih memperhatikan mengenai penerimaan permohonan asuransi kesehatan …

3. Mengganti slang word. Pada Tabel 4 merupakan proses untuk mengganti kata yang tidak baku menjadi kata baku, seperti kata analisa diubah menjadi analisis. Berikut contohnya.

Tabel 4. Contoh penerapan tahap slang word

Abstrak sebelum	Abstrak setelah proses
implementasi data mining untuk memprediksi penerimaan permohonan calon nasabah asuransi kesehatan dengan menggunakan algoritma na ve bayes berbasis web pada pt asuransi sinarmas oleh lady caesar sevika detale kesehatan merupakan bagian yang amat penting bagi kehidupan manusia pt asuransi sinarmas adalah salah satu asuransi yang menyediakan produk asuransi kesehatan banyaknya masyarakat yang mendaftar dan mengajukan permohonan asuransi kesehatan membuat pihak asuransi harus lebih memperhatikan mengenai penerimaan permohonan asuransi kesehatan …	implementasi data mining untuk memprediksi penerimaan permohonan calon nasabah asuransi kesehatan dengan menggunakan algoritma naive bayes berbasis web pada pt asuransi sinarmas oleh lady caesar sevika detale kesehatan merupakan bagian yang amat penting bagi kehidupan manusia pt asuransi sinarmas adalah salah satu asuransi yang menyediakan produk asuransi kesehatan banyaknya masyarakat yang mendaftar dan mengajukan permohonan asuransi kesehatan membuat pihak asuransi harus lebih memperhatikan mengenai penerimaan permohonan asuransi kesehatan …

4. Menghapus stop word. Pada Tabel 5 merupakan proses untuk menghapus kata yang diabaikan dalam tahap preprocessing, seperti kata dan, untuk, bagi, dan seterusnya. Berikut contohnya.

Tabel 5. Contoh penerapan tahap stop word

Abstrak sebelum	Abstrak setelah proses
implementasi data mining untuk memprediksi penerimaan permohonan calon nasabah asuransi kesehatan dengan menggunakan algoritma naive bayes berbasis web pada pt asuransi sinarmas oleh lady caesar sevika detale kesehatan merupakan bagian yang amat penting bagi kehidupan manusia pt asuransi sinarmas adalah salah satu asuransi yang menyediakan produk asuransi kesehatan banyaknya masyarakat yang mendaftar dan mengajukan permohonan asuransi kesehatan membuat pihak asuransi harus lebih memperhatikan mengenai penerimaan permohonan asuransi kesehatan …	implementasi data mining memprediksi penerimaan permohonan calon nasabah asuransi kesehatan algoritma naive bayes berbasis web asuransi sinarmas lady caesar sevika detale kesehatan asuransi sinarmas asuransi asuransi kesehatan banyaknya mendaftar mengajukan permohonan asuransi kesehatan asuransi memperhatikan penerimaan permohonan asuransi kesehatan …

5. Menghilangkan spasi antar karakter. Pada Tabel 6 merupakan proses untuk menghilangkan spasi antar karakter. Berikut contohnya.

Tabel 6. Contoh penerapan tahap stop word

Abstrak sebelum	Abstrak setelah proses
implementasi data mining memprediksi penerimaan permohonan calon nasabah asuransi kesehatan algoritma naive bayes berbasis web asuransi sinarmas lady caesar sevika detale kesehatan asuransi sinarmas asuransi asuransi kesehatan banyaknya mendaftar mengajukan	implementasidataminingmemprediksipenerimaan permohonancalonnasabahasuransikesehatanalgor itmanaivebayesberbasiswebasuransisinarmaslady caesarsevikadetalekesehatanasuransisinarmasasu ransiasuransikesehatanbanyaknyamendaftarmen gajukanpermohonanasuransikesehatanasuransime

ISSN : 2655-139X (ONLINE)
ISSN : 0216-4221 (PRINT)

permohonan asuransi kesehatan asuransi memperhatikan penerimaan permohonan asuransi kesehatan …	mperhatikanpenerimaanpermohonanasuransikesehatan …

3.1.3. Pembentukan N-Gram

Setelah dokumen melalui tahap *preprocessing,* kemudian masuk ke tahap pembentukan *n-gram* yaitu, proses memecahkan *string* teks yang dikelompokan berdasarkan nilai *n-gram* yang akan dihitung pergeserannya secara terus menerus ke depan sejumlah nilai n sampai akhir dokumen. Contoh tahapan pembentukan *n-gram* dengan nilai N=3. Berikut bisa dilihat di Tabel 7 berikut.

Tabel 7. Contoh penerapan tahap stop word

Cuplikan teks hasil preprocessing	N-gram dengan n=3
implementasidataminingmemprediksipenerimaan permohonancalonnasabahasuransikesehatanalgoritmanaivebayesberbasiswebasuransi…	imp lem ent asi dat ami nin gme mpr edi ksi pen eri maa npe rmo hon anc alo nna sab aha sur ans ike she ata nal gor itm ana ive bay esb erb asi swe bas ura nsi …

3.1.4. Algoritma *Winnowing*

Setelah pembentukan *n-gram* maka proses selanjutnya adalah perhitungan *rolling hash* untuk mencari nilai *hash* dari setiap *string* yang sudah dipotong pada tahap *n-gram*. Setiap *string* yang sudah dipotong diubah menjadi ASCII lalu dihitung menggunakan persamaan (1) dan (2). Tabel 8 menyajikan contoh perhitungan *rolling hash* dengan nilai k=3 dengan teks hasil n-gram "imp mpl ple lem eme men ent nta tas asi sip ipe pen eng nga gam ama man ana nan anf nfi fil ile".

Tabel 8. Contoh penerapan tahap stop word

Potongan teks	Desimal ASCII	Perhitungan	Hasil
imp	i=105 m=109 p=112	105*7^(3-1)+109*7^(3-2)+112*7^(3-3)	6020
mpl	m=109 p=112 l=108	109*7^(3-1)+112*7^(3-2)+108*7^(3-3)	6233
ple	p=112 l=108 e=101	112*7^(3-1)+108*7^(3-2)+101*7^(3-3)	6345
lem	l=108 e=101 m=109	108*7^(3-1)+101*7^(3-2)+109*7^(3-3)	6108
eme	e=101 m=109 e=101	101*7^(3-1)+109*7^(3-2)+101*7^(3-3)	5813
men	m=109 e=101 n=110	109*7^(3-1)+101*7^(3-2)+110*7^(3-3)	6158
ent	e=101 n=110 t=116	101*7^(3-1)+110*7^(3-2)+116*7^(3-3)	5835
nta	n=110 t=116 a=97	110*7^(3-1)+116*7^(3-2)+97*7^(3-3)	6299
tas	t=116 a=97 s=115	116*7^(3-1)+97*7^(3-2)+115*7^(3-3)	6478
asi	a=97 s=115 i=105	97*7^(3-1)+115*7^(3-2)+105*7^(3-3)	5663
sip	s=115 i=105 p=112	115*7^(3-1)+105*7^(3-2)+112*7^(3-3)	6482
ipe	i=105 p=112 e=101	105*7^(3-1)+112*7^(3-2)+101*7^(3-3)	6030
pen	p=112 e=101 n=110	112*7^(3-1)+101*7^(3-2)+110*7^(3-3)	6305
eng	e=101 n=110 g=103	101*7^(3-1)+110*7^(3-2)+103*7^(3-3)	5822
nga	n=110 g=103 a=97	110*7^(3-1)+103*7^(3-2)+97*7^(3-3)	6208
gam	g=103 a=97 m=109	103*7^(3-1)+97*7^(3-2)+109*7^(3-3)	5835
ama	a=97 m=109 a=97	97*7^(3-1)+109*7^(3-2)+97*7^(3-3)	5613
man	m=109 a=97 n=110	109*7^(3-1)+97*7^(3-2)+110*7^(3-3)	6130
ana	a=97 n=110 a=97	97*7^(3-1)+110*7^(3-2)+97*7^(3-3)	5620
nan	n=110 a=97 n=110	110*7^(3-1)+97*7^(3-2)+110*7^(3-3)	6179
anf	a=97 n=110 f=102	97*7^(3-1)+110*7^(3-2)+102*7^(3-3)	5625
nfi	n=110 f=102 i=105	110*7^(3-1)+102*7^(3-2)+105*7^(3-3)	6209
fil	f=102 i=105 l=108	102*7^(3-1)+105*7^(3-2)+108*7^(3-3)	5841
ile	i=105 l=108 e=101	105*7^(3-1)+108*7^(3-2)+101*7^(3-3)	6002

Setelah dilakukan perhitungan diperoleh hasil dari *rolling hash* "6020 6233 6345 6108 5813 6158 5835 6299 6478 5663 6482 6030 6305 5822 6208 5835 5613 6130 5620 6179 5625 6209 5841 6002". Tahap selanjutnya adalah

ISSN : 2655-139X (ONLINE)
ISSN : 0216-4221 (PRINT)

pembentukan *window* dari nilai *hash* yang diperoleh dengan mengelompokkan sebanyak nilai *w.* Berikut ini contoh hasil pembentukan *window* dengan w=4.

6020 | 6233 | 6345| 6108
5813 | 6158 | 5835| 6299
6478 | 5663 | 6482| 6030
6305 | 5822 | 6208| 5835
5613 | 6130 | 5620| 6179
5625 | 6209 | 5841| 6002

Setelah pembentukan *window* selesai, maka tahap selanjutnya adalah pencarian *fingerprint* dari setiap *window* yang ada. Nilai *fingerprint* ditentukan berdasarkan nilai terkecil dari setiap *window* yang ada. Hasil dari *fingerprint* dari contoh sebelumnya adalah "6020 | 5813 | 5663 | 5822 | 5613 | 5625"

3.1.5. *Jaccard Similarity*

Setelah fingerprint dari dokumen ditemukan langkah selanjutnya adalah perhitungan similaritas dari *fingerprint* yang ada menggunakan *Jaccard Similarity*. Tabel 9 menyajikan contoh fingerprint dari dua teks.

Tabel 9. Contoh teks dan nilai *fingerprint*

Teks	Contoh teks	Fingerprint
Teks 1	implementasipengamananfile	6020 | 5813 | 5663 | 5822 | 5613 | 5625
Teks 2	aplikasipengamananfile	5645 | 5663 | 5822 | 5613 | 5625

Perhitungan *jaccard similarity* menggunakan Persamaan (3) dengan rincian sebagai berikut:

$Similarity\ (X,Y) = \frac{|x \cap y|}{|x \cup y|} \times 100\%$

X = {6020,5813,**5663,5822,5613,5625**}
Y = {5645,**5663,5822,5613,5625**}
X∩Y = {5663,5822,5613,5625}
X∪Y = {6020,5813,5663,5822,5613,5625,5645}

$Similarity\ (X,Y) = \frac{|x \cap y|}{|x \cup y|} \times 100\%$

$Similarity\ (X,Y) = \frac{4}{7} \times 100\% = 0{,}571 \times 100\% =$ **57,1%**

Dengan demikian, diperoleh tingkat similaritas dari kedua dokumen adalah 57,1%

3.2 Pengujian

Pengujian merupakan salah satu hal yang perlu dilakukan dalam setiap pengembangan sistem untuk mengevaluasi, menganalisa dan mengetahui tingkat akurasi atau kesamaan hasil yang telah dicapai oleh sistem yang telah dirancang. Pengujian dilakukan dengan dua skenario yaitu pengujian dengan nilai *k-gram* dan *w-gram* yang sama dan pengujian dengan nilai *k-gram* dan *w-gram* yang berbeda.

3.2.1 Pengujian dengan nilai k-gram dan w-gram yang sama

Pengujian menggunakan 13 dokumen abstrak yang berbeda dari *dataset*. Pada pengujian ini digunakan nilai *k-gram* dan *w-gram* dari 2 sampai 5 dimana nilainya sama pada Abstrak_1513500346. Hasil pengujian dapat dilihat pada Tabel 10.

Tabel 10. Hasil pengujian similaritas dokumen

No	Data Uji	Hasil Uji Kemiripan			
		k=2 w=2	k=3 w=3	k=4 w=4	k=5 w=5
1	1511500025	46,96%	21,78%	5,84%	2,08%
2	1511500082	49,55%	24,41%	8,20%	3,31%
3	1511500132	44,95%	20,62%	8,66%	3,32%

ISSN : 2655-139X (ONLINE)
ISSN : 0216-4221 (PRINT)

4	1511500157	46,53%	25,15%	6,95%	4,88%
5	1511500199	50,51%	21,39%	7,11%	3,19%
6	1511500207	45,54%	23,93%	8,06%	5,26%
7	1511500215	44,66%	19,23%	7,65%	3,64%
8	1511500231	50,96%	21,43%	8,02%	1,96%
9	1511500249	51,40%	20,51%	8,33%	3,74%
10	1511500264	40,21%	23,57%	7,95%	3,14%
11	1511500272	47,66%	20,53%	7,48%	3,59%
12	1511500298	49,49%	24,24%	8,84%	3,64%
13	1511500314	43,33%	22,47%	6,55%	2,36%

3.2.2 Pengujian dengan nilai k-gram dan w-gram yang berbeda

Pengujian kedua menggunakan 13 dokumen abstrak yang sama dengan pengujian sebelumnya. Namun pada pengujian ini digunakan nilai *k-gram* = 3 dan *w-gram* = 4, *k-gram* = 4 dan *w-gram* = 3, *k-gram* = 2 dan *w-gram* = 5, *k-gram* = 5 dan *w-gram* = 2, dimana nilainya berbeda pada Abstrak_1513500346. Hasil pengujian dapat dilihat pada Tabel 11.

Tabel 11. Hasil pengujian nilai k-gram dan w-gram berbeda

No	Data Uji	Hasil Uji Kemiripan			
		k=3 w=4	k=4 w=3	k=2 w=5	k=5 w=2
1	1511500025	20,59%	7,58%	35,71%	3,17%
2	1511500082	22,15%	8,59%	36,54%	3,87%
3	1511500132	19,08%	7,14%	36,54%	3,93%
4	1511500157	21,37%	8,13%	44,44%	4,42%
5	1511500199	17,39%	7,39%	38%	3,96%
6	1511500207	23,62%	9,17%	40,43%	5,34%
7	1511500215	20,44%	6,58%	36,54%	3,97%
8	1511500231	21,64%	8,33%	35,29%	3,34%
9	1511500249	21,09%	8,39%	45,65%	4,20%
10	1511500264	26,50%	6,96%	41,86%	4,46%
11	1511500272	17,33%	6,88%	38,30%	3,66%
12	1511500298	23,26%	8,44%	44,44%	4,68%
13	1511500314	20,88%	6,92%	36,07%	4%

3.2.3. Pembahasan Hasil Penelitian

Hasil uji coba dari pencarian nilai *similarity* menggunakan metode *n-gram* dan *Jaccard Similarity* terhadap algoritma *winnowing* telah ditampilkan pada Tabel 10 menggunakan *k-gram* dan *w-gram* yang sama. Sementara itu, Tabel 11 menyajikan pengaruh penggunaan nilai *k-gram* dan *w-gram* berbeda terhadap hasil similaritas dokumen. Hasil pengujian terhadap 13 dokumen *dataset* memiliki derajat kesamaan yang berbeda-beda. Semakin kecil nilai *k-gram* dan *w-gram* maka derajat kesamaan atau *similarity* terhadap suatu dokumen mempunyai derajat kesamaan yang tinggi. Sebaliknya, semakin besar nilai *k-gram* dan *w-gram* maka derajat kesamaan suatu dokumen itu memiliki derajat yang rendah, jika nilai *k-gram* dan *w-gram* sama. Pada Tabel 10 membuktikan *k-gram* dan *w-gram* dengan nilai 2 menghasilkan *similarity* 51,40%, sedangkan *k-gram* dan *w-gram* dengan nilai 5 menghasilkan *similarity* yang rendah yaitu 5,26%.

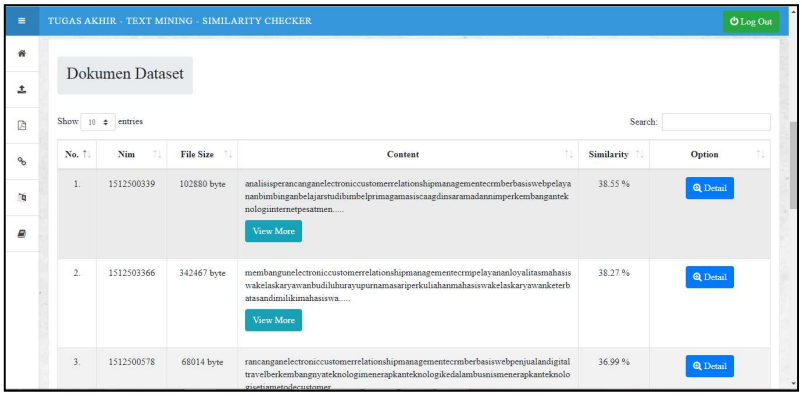
Penginputan nilai yang berbeda pada nilai *k-gram* dan *w-gram* dapat menghasilkan nilai *similarity* yang tinggi. Semakin kecil nilai *k-gram* dan *w-gram* maka akan semakin sering potongan suku kata akan dicocokan dan sering ditemukan nilai yang sama. Pada Tabel 11 dengan nilai *k-gram* = 2 dan *w-gram* = 5 menghasilkan nilai *similarity* sebesar 45,65%. Sementara dalam pengujian menggunakan data uji yang sama dengan dataset menghasilkan nilai

ISSN : 2655-139X (ONLINE)
ISSN : 0216-4221 (PRINT)

similaritas sebesar 100%. Dengan demikian, metode *Jaccard Similarity* memiliki kinerja yang baik untuk digunakan dalam pendeteksian nilai *similarity* terhadap suatu dokumen.

3.2.4 Prototipe Sistem

Gambar 3 menyajikan tampilan prototipe sistem. Pengguna dapat mengunggah dokumen abstrak dalam format PDF yang ingin dicari. Selanjutnya sistem akan memberikan tampilan hasil pencarian berupa dokumen abstrak hasil pencarian beserta persentase kesamaan (similaritas)-nya. Hasil pencarian diurutkan berdasarkan nilai similaritas yang paling tinggi hingga rendah. Untuk melihat isi dokumen hasil pencarian, pengguna dapat memilih tombol "Detail".



Gambar 5. Prototipe Sistem

4 Kesimpulan

Berdasarkan hasil evaluasi dan pengujian terhadap sistem pencarian dokumen abstrak tugas akhir mahasiswa Universitas Budi Luhur berbasis algoritma *Winnowing* dan *Jaccard Similarity*, dapat disimpulkan bahwa sistem mampu menyajikan hasil pencarian dokumen PDF yang menjadi masukan dengan baik. Penggunaan metode *Winnowing* dan *Jaccard Similarity* mampu menyajikan hasil similaritas dengan cukup akurat. Berdasarkan pengujian terhadap pengaruh nilai *k-gram* dan *w-gram* ternyata keduanya mempengaruhi nilai similaritas yang dihasilkan. Semakin besar nilai *k-gram* dan *w-gram* maka nilai similaritasnya semakin kecil. Oleh karena itu, perlu dikaji penggunaan nilai *k-gra*m dan *w-gram* yang tepat.

Referensi

[1] W. H. Gomaa and A. A. Fahmy, "A Survey of Text Similarity Approaches," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 13–18, 2013.
[2] M. Farouk, "Measuring Sentences Similarity: A Survey," *Indian J. Sci. Technol.*, vol. 12, no. 25, pp. 1–11, 2019.
[3] D. Chandrasekaran and V. Mago, "Evolution of Semantic Similarity—A Survey," *ACM Comput. Surv.*, vol. 54, no. 2, Feb. 2021.
[4] J. Wang and Y. Dong, "Measurement of Text Similarity : A Survey," *Information*, vol. 11, no. 421, pp. 1–17, 2020.
[5] I. Abdullah and E. Aribowo, "Rancang Bangun Aplikasi Pengecekan Kemiripan Judul Skripsi Dengan Metode Cosine Similarity (Studi Kasus : Program Studi Teknik Informatika UAD)," *J. Sarj. Tek. Inform.*, vol. 6, no. 2, pp. 43–52, 2018.
[6] D. A. R. Ariantini, A. S. M. Lumenta, and A. Jacobus, "Pengukuran Kemiripan Dokumen Teks Bahasa Indonesia Menggunakan Metode Cosine Similarity," *E-Journal Tek. Inform.*, vol. 9, no. 1, pp. 1–8, 2016.
[7] H. Sutikno and Saniati, "Implementasi Algoritma Cosine Similarity untuk Mendeteksi Kemiripan Topik

ISSN : 2655-139X (ONLINE)
ISSN : 0216-4221 (PRINT)

Judul," *JECSIT*, vol. 1, no. 1, pp. 51–61, 2021.
[8] Sunardi, A. Yudhana, and I. A. Mukaromah, "Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram dan Jaccard Similarity Terhadap Algoritma Winnowing," *TRANSMISI*, vol. 20, no. 3, pp. 105–110, 2018.
[9] K. Rinartha, "Simple Query Suggestion untuk Pencarian Artikel Menggunakan Jaccard Similarity," *J. Ilm. Rekayasa dan Manaj. Sist. Inf.*, vol. 3, no. 1, pp. 30–34, 2017.
[10] J. Priambodo, "Pendeteksian Plagiarisme Menggunakan Algoritma Rabin-Karp dengan Metode Rolling Hash," *J. Inform. Univ. PAMULANG*, vol. 3, no. 1, pp. 39–45, 2018.
[11] A. Filcha and M. Hayaty, "Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa," *JUITA*, vol. VII, no. 1, pp. 25–32, 2019.

ISSN : 2655-139X (ONLINE)
ISSN : 0216-4221 (PRINT)