IMPLEMENTASI METODE JACCARD SIMILARITY TERHADAP TINGKAT KEMIRIPAN RUU DENGAN UU

(SKRIPSI)

Oleh

BONEMA TRI PRASETYO
1717051017

JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
2021

ABSTRACT

IMPLEMENTATION OF THE JACCARD SIMILARITY METHOD TOWARD THE LEVEL OF RELIABILITY OF THE BILL WITH THE LAW

BY

BONEMA TRI PRASETYO

Currently, with the passage of the Job Creation Act which adopts the Omnibus Law concept, many are getting rejection. The main objections were expressed by the workers related to various provisions, regarding wages and termination schemes. This will certainly become a big problem if this wave of rejection cannot be mediated and lasts for a long time. On the other hand, the occurrence of regulatory obesity in Indonesia causes overlapping between applicable laws. It is necessary to indicate the degree of similarity between the Draft Law and the Law in order to avoid overlapping regulations. Therefore, this research was conducted to find out how big the level of similarity between the Draft Law and the existing Law. In calculating the level of similarity in this study using the Jaccard Similarity method with the N-gram and hashing, then the accuracy of the method is measured using a confusion matrix table. According to legal experts (experts) that the similarity between the Draft Law and the Law is high if the level of similarity obtained is more than 25%.

Keywords : Draft Laws; Laws; Jaccard Similarity; N-grams; Hashing.

ABSTRAK

IMPLEMENTASI METODE *JACCARD SIMILARITY* TERHADAP TINGKAT KEMIRIPAN RUU DENGAN UU

OLEH

BONEMA TRI PRASETYO

Saat ini, dengan disahkan nya Undang-Undang Cipta Kerja yang mengadopsi konsep Omnibus Law, banyak mendapatkan penolakan. Penolakan utamanya dikemukakan oleh kalangan buruh terkait dengan berbagai ketentuan, soal upah dan skema pemutusan hubungan kerja. Hal ini tentunya akan menjadi menjadi masalah besar apabila gelombang penolakan ini tidak bisa dimediasi dan berlangsung dalam waktu yang lama. Disisi lain juga terjadinya obesitas regulasi di Indonesia menyebabkan saling tumpang tindih (*overlapping*) antar Undang-Undang yang berlaku. Perlunya mengindikasi seberapa besar tingkat kemiripan antara Rancangan Undang-Undang dengan Undang-Undang agar tidak terjadi tumpang tindih regulasi. Maka dari itu penelitian ini dilakukan untuk mengetahui seberapa besar tingkat kemiripan antara Rancangan Undang-Undang dengan Undang-Undang yang ada. Dalam menghitung tingkat kemiripan pada penelitian ini menggunakan metode *Jaccard Similarity* dengan *n-gram* dan *hashing* kemudian dilakukan pengukuran akurasi metode tersebut dengan tabel *confusion matrix*. Menurut ahli hukum (*expert*) bahwa kemiripan antara Rancangan Undang-Undang dengan Undang-Undang tinggi apabila tingkat kemiripan yang didapatkan lebih dari 25%.

Kata Kunci : Rancangan Undang-Undang; Undang-Undang; Jaccard Similarity; N-Gram; Hashing

IMPLEMENTASI METODE JACCARD SIMILARITY TERHADAP TINGKAT KEMIRIPAN RUU DENGAN UU

Oleh

BONEMA TRI PRASETYO

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh
Gelar SARJANA KOMPUTER

Pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam

JURUSAN ILMU KOMPUTER

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS LAMPUNG

2021

Judul Skripsi	: **IMPLEMENTASI METODE *JACCARD SIMILARITY* TERHADAP TINGKAT KEMIRIPAN RUU DENGAN UU**
Nama Mahasiswa	: Bonema Tri Prasetyo
Nomor Pokok Mahasiswa	: 1717051017
Program Studi	: S1 Ilmu Komputer
Fakultas	: Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

Aristoteles, M.Si.	**Rudy, S.H., LL.M., LL.D.**
NIP 19810521 200604 1 002	NIP 19810104 200312 1 001

2. Ketua Jurusan Ilmu Komputer

Didik Kurniawan, S.Si., M.T.
NIP 19800419 200501 1 004

MENGESAHKAN

1. Tim Penguji

Ketua : **Aristoteles, M.Si.** .....

Sekretaris : **Rudy, S.H., LL.M., LL.D.** .....

Penguji
Bukan Pembimbing : **Rico Andrian, S.Si., M.Kom.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Suripto Dwi Yuwono, S.Si., M.T.
NIP 19740705 200003 1 001

Tanggal Lulus Ujian Skripsi : **12 November 2021**

PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul "Implementasi Metode *Jaccard Similarity* Terhadap Tingkat Kemiripan RUU dengan UU" merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 12 November 2021
Yang menyatakan

Bonema Tri Prasetyo
NPM. 1717051017

RIWAYAT HIDUP

Penulis dilahirkan pada tanggal 12 Mei 1999 di Kalianda, Lampung Selatan. Sebagai anak ketiga dari empat bersaudara dengan Ayah bernama Bono dan Ibu bernama Endang Martini.

Penulis menyelesaikan pendidikan Taman Kanak-Kanak (TK) Bina Karya Kab. Lampung Selatan pada tahun 2005, menyelesaikan Sekolah Dasar (SD) di SDN 1 Way Urang Kec. Kalianda pada tahun 2011, kemudian menyelesaikan Sekolah Menengah Pertama (SMP) di MTsN Kalianda pada tahun 2014, dan menyelesaikan Sekolah Menengah Akhir (SMA) di SMA Kebangsaan Lampung Selatan pada tahun 2017.

Pada tahun 2017, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer FMIPA Universitas Lampung melalui jalur SNMPTN. Selama menjadi mahasiswa penulis menyelesaikan Kerja Praktik di Dinas Kependukan dan Pencatatan Sipil Kabupaten Lampung Selatan pada tahun 2020, dan penulis juga menyelesaikan Kuliah Kerja Nyata (KKN) Mandiri Putera Daerah di Desa Bumi Daya Kec. Palas Kab. Lampung Selatan pada tahun 2020

MOTTO

" Balas dendam terbaik adalah dengan memperbaiki dirimu."

“Ketika kau sedang mengalami kesusahan dan bertanya-tanya ke mana Allah, cukup ingat bahwa seorang guru selalu diam saat ujian berjalan.”

“Jangan menunggu. Takkan pernah ada waktu yang tepat.”

PERSEMBAHAN

Alhamdulillahi rabbil 'alamin skripsi ini telah selesai penulis kerjakan.

Skripsi yang disusun ini adalah suatu kebanggaan yang saya persembahkan kepada :

Kedua Orangtuaku yang Tercinta

Papa dan Mama yang senantiasa memberikan segala perhatian, kasih sayang, cinta, dukungan baik moril maupun materil serta doa terbaik bagi kesuksesan anak-anaknya yang tidak bisa diukur dan dibalas dengan apapun.

Mbak Indah Destiana, Mas Muhammad Iqbal, Ahmad Unsa El Farid, dan Keluarga Besar yang senantiasa memberikan dukungan, kasih sayang dan semangat, sehingga skripsi ini dapat selesai dikerjakan.

Kumbang Tech

Keluarga Besar Ilmu Komputer 2017

Serta Almamater Tercinta,
Universitas Lampung

SANWACANA

Alhamdulilahi rabbil ‘alamin puji syukur ke hadirat Allah SWT, yang telah melimpahkan rahmat-Nya kepada kita sehingga penulis dapat menyelesaikan skripsi di Jurusan Ilmu Komputer FMIPA Universitas Lampung dengan judul “Implementasi Metode *Jacccard Similarity* Terhadap Tingkat Kemiripan RUU dengan UU”.

Penulis mengucapkan terimakasih yang paling tulus dan sebesar-besarnya kepada semua pihak yang telah membantu serta mendukung dalam pelaksanaan penelitian dan penyusunan laporan skripsi ini, antara lain :

1. Kepada Papa dan Mama yang sangat penulis sayangi, yang senantiasa mengajarkan kebaikan, memberikan perhatian, kasih sayang, do’a terbaik, kepercayaan serta dukungan, atas setiap keputusan yang diambil oleh penulis hingga detik ini.
2. Ketiga saudara penulis Mbak Ana, Mas Iqbal, dan Unsa yang selalu memberikan perhatian, pengertian, dukungan, dan semangat dalam setiap langkah serta penyusunan skripsi ini.
3. Bapak Aristoteles, M.Si. selaku pembimbing utama dalam penelitian ini yang senantiasa memberikan arahan, bantuan, semangat, dan motivasi terbaik dalam menyelesaikan penelitian ini.
4. Bapak Rudy, S.H., LL.M., LL.D. selaku pembimbing kedua dalam penelitian ini yang senantiasa memberikan arahan, masukan, dan saran terbaik selama penyusunan laporan skripsi ini.
5. Bapak Rico Andrian, S.Si., M.Kom. selaku pembahas yang telah memberikan masukan, dan saran yang bermanfaat untuk perbaikan penelitian dan penulisan laporan skripsi ini.
6. Bapak Dr. Suripto Dwi Yuwono, S.Si., M.T. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

7. Bapak Didik Kurniawan, S.Si., M.T. selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
8. Ibu Astria Hijriani, S.Kom., M.Kom. selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
9. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc. Selaku Dosen Pembimbing Akademik yang selalu memberikan arahan dan semangat dalam menyelesaikan skripsi ini.
10. Seluruh Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu dan pelajaran terbaik selama penulis menempuh pendidikan di Jurusan Ilmu Komputer.
11. Seluruh Karyawan dan Staf Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan bantuan dalam banyak hal selama ini.
12. Keluarga Besar Kumbang Tech yang telah memberikan dukungan serta pengaruh terbesar kepada penulis dalam menjalankan pendidikan serta penyelesaian skripsi ini.
13. Seluruh teman-teman Ilmu Komputer 2017 yang telah memberikan kebersamaan, semangat, dan do'a selama penulis menempuh pendidikan di Jurusan Ilmu Komputer.

Penulis menyadari bahwa penyusunan skripsi ini masih jauh dari kata sempurna. Namun penulis sangat mengharapkan skripsi ini dapat bermanfaat bagi mahasiswa Ilmu Komputer pada khususnya dan para civitas akademik Universitas Lampung pada umumnya.

Bandar Lampung, 12 November 2021
Yang menyatakan

Bonema Tri Prasetyo
NPM. 1717051017

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

BAB I PENDAHULUAN

1.1 Latar Belakang

Presiden Joko Widodo dalam pidato pelantikannya di Sidang Paripurna MPR RI tanggal 20 Oktober 2019 menyampaikan hal-hal yang akan dilaksanakan nya selama periode 2019-2024. Salah satunya adalah menyederhanakan segala bentuk kendala regulasi. Presiden dalam pidato tersebut menyebutkan : "…Pemerintah akan mengajak DPR untuk menerbitkan dua undang-undang besar. Yang pertama, UU Cipta Lapangan Kerja. Yang Kedua, UU Pemberdayaan UMKM. Masing-masing UU tersebut akan menjadi Omnibus Law, yaitu satu UU yang sekaligus merevisi beberapa UU, bahkan puluhan UU. Puluhan UU yang menghambat penciptaan lapangan kerja langsung di revisi sekaligus. Puluhan UU yang menghambat pengembangan UMKM juga akan langsung direvisi." (Kementerian Luar Negeri, 2019:7)

Omnibus Law memiliki kesamaan konsep dengan omnibus bill. Definisi omnibus bill yakni "*a bill consisting of a number of related but separate parts that seek to amend and/or to enact one or several new Acts*." (House of Commons, Glossary of Parliamentary Procedure, 2011:38).(Sebuah UU yang terdiri dari sejumlah bagian terkait tetapi terpisah yang berupaya untuk mengubah dan/atau mencabut satu atau beberapa undang-undang yang ada dan/atau untuk membuat satu atau beberapa undang-undang baru)(Hantoro, 2019).

Penerapan Omnibus Law terhadap Tap MPR tentu akan berbeda dengan penerapan pada UU. Hal ini dapat ditelaah secara rasional, dikarenakan

perbedaan spektrum diantara dua hal tersebut. Melihat ke wilayah regional Asia Tenggara, Indonesia pada dasarnya tidak sendirian dalam pengalaman menerapkan konsep Omnibus Law. Filipina pernah merumuskan Omnibus Code of 1987 and Foreign Investment Act Of 1991(Michael, 2020).

Populasi peraturan perundang-undangan di Indonesia saat ini sudah mencapai pada angka 43.933 peraturan yang aktif dan berlaku di Indonesia, kemungkinan akan terus bertambah seiring waktu untuk mengatur setiap hal yang ada, dan yang dikhawatirkan adalah timbulnya peraturan yang terlalu banyak namun dengan kualitas yang buruk dan mengarah pada ketidakharmonisan antar peraturan yang ada (Rudy et al, 2021).

Saat ini, dengan disahkan nya UU Cipta Kerja yang mengadopsi konsep Omnibus Law, banyak mendapatkan penolakan. Narasi penolakan ini ternyata sudah lama terjadi. Penolakan utama nya dikemukakan kalangan buruh terkait dengan berbagai ketentuan soal upah dan skema pemutusan hubungan kerja. Hal ini tentunya akan menjadi kontraproduktif apabila gelombang penolakan ini tidak bisa dimediasi dan berlangsung dalam waktu yang lama(Arham et al., 2019).

Sebelumnya penelitian tentang tingkat kesamaan teks pernah dilakukan oleh beberapa orang dengan menggunakan metode yang berbeda-beda. Menurut (Nurdiana et al., 2016) dalam jurnalnya yang berjudul ” Perbandingan Metode *Cosine Similarit*y Dengan Metode *Jaccard Similarity* Pada Aplikasi Pencarian Terjemah Al-Qur’an Dalam Bahasa Indonesia” membandingkan tiga metode untuk mencari pencocokan kata yang lebih akurat dalam terjemahan Al-Qur’an. Metode *cosine*, *jaccard* dan *k-nearest neighbor* (K-NN) yang digunakan pada proses klasifikasi dokumen teks dengan hasil akhir dari percobaan 33 kali dengan *key* yang berbeda dan total 6326 dokumen di dapat metode cosine yang nilai kemiripannya tertinggi yaitu 41% dari metode *jaccard* 19% dan *k-nearest neighbor* (K-NN) 40%, karena metode *cosine similarity* mempunyai konsep normalisasi panjang vektor data dengan

membandingkan N-gram yang sejajar satu sama lain dari 2 pembanding. Berdeda dengan penelitian yang dilakukan oleh Nurdiana, penelitian yang dilakukan oleh (Sunardi et al., 2018) dalam penelitiannya yang berjudul " Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram Dan *Jaccard Similarity* Terhadap Algoritma Winnowing" mendapatkan hasil akhir dari pendeteksi plagiarisme menghasilkan persentase sampai 100% dengan menggunakan *Jaccard Similarity* dari dokumen atau sample dengan nilai n-gram 3 dan nilai w-gram 5. Jika dibandingkan dengan metode k-gram yang digunakan memperoleh kemiripan 83%, hasil ini cukup andal. Dengan demikian, metode Jaccard Similarity memiliki prospek untuk digunakan dalam deteksi plagiat.

Undang-Undang adalah Peraturan Perundang-undangan yang dibentuk oleh Dewan Perwakilan Rakyat dengan persetujuan bersama Presiden. Untuk menghasilkan Undang-undang yang berkualitas dibutuhkan persiapan yang kompeten, salah satu faktornya adalah originalitas. Kemampuan dalam menciptakan undang-undang yang original menjadi faktor penting. Teks plagiasi adalah bentuk kemiripan teks yang paling sering dijumpai. Deteksi plagiarisme dibedakan menjadi dua berdasarkan tugasnya yaitu secara intrinsik dan ekstrinsik. Plagiarisme adalah suatu tindakan mencuri dan publikasi dari penulis lain baik itu berupa bahasa, pikiran, gagasan, atau ekspresi dan merepresentasikan mereka sebagai salah satu karya asli sendiri (Stepchyshyn & Nelson, 2007). Penelitian ini menerapkan metode *jaccard similarity* untuk mendeteksi kemiripan teks pada dokumen UU dan RUU Republik Indonesia.

Oleh karena itu, melihat urgensi perlunya partisipasi masyarakat yang lebih aktif dalam proses perancangan UU bersama dengan pemerintah dan parlemen. Maka diperlukannya aplikasi yang dapat mengukur tingkat kemiripan antara RUU dengan UU. Dengan adanya aplikasi tersebut maka pembentukan regulasi yang menerapkan konsep Omnibus Law akan berjalan

dengan baik, tanpa mengesampingkan kepentingan pihak manapun baik dari elemen pemerintah maupun elemen non-pemerintah.

1.2 Rumusan Masalah

Rumusan masalah yang dibahas dalam penelitian ini yaitu bagaimana mengetahui tingkat kemiripan dari sebuah RUU terhadap UU yang telah disahkan menggunakan metode *jaccard similarity*.

1.3 Ruang Lingkup Penelitian

Ruang lingkup dalam penelitian ini adalah sebagai berikut.

1. Dataset yang digunakan dalam penelitian ini yaitu Undang-Undang Republik Indonesia tahun 2010 sampai 2020 berupa *file* berjenis pdf (*portable document format*).
2. Implementasi *text mining* yang digunakan menggunakan NLTK (*Natural Language Toolkit*) yang tersedia pada Python.
3. Pembobotan nilai pada kata yang dibandingkan menggunakan perhitungan nilai *hashing*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengukur tingkat akurasi kemiripan antara Rancangan Undang-Undang dan Undang-Undang dengan menggunakan metode *Jaccard Similarity*.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

1. Mengetahui tingkat akurasi kemiripan antara RUU dengan UU yang sudah *existing* menggunakan metode *jaccard similarity*.
2. Memudahkan pemerintah dalam melakukan pencarian tingkat kemiripan antara RUU dengan UU agar tidak terjadi obesitas regulasi.

BAB II TINJAUAN PUSTAKA

2.1 Undang-undang

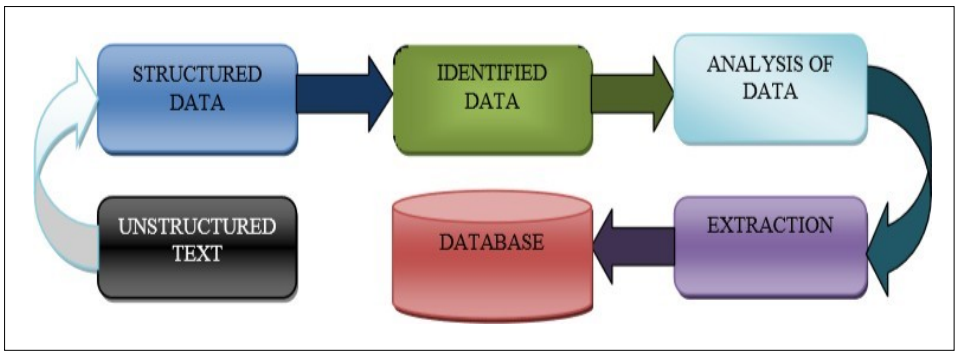
Undang-undang merupakan dasar hukum yang berlaku bagi negara hukum. Tahap pembentukan undang-undang dimulai dari tahap perencanaan, penyusunan, pembahasan, pengesahan atau penetapan, dan pengundangan. Mulai dari tahapan perencanaan dengan menyiapkan Rancangan Undang-Undang (RUU) yang harus disertai dengan naskah hasil penelitian/hasil kajian (naskah akademik), kemudian melalui tahap pembahasan di lembaga legislatif (DPR- RI) hingga tahapan pengundangan. Pembaharuan beberapa undang-undang merupakan contoh bagaimana pentingnya pembaharuan hukum dilakukan untuk mengikuti perkembangan yang ada di masyarakat (Fadli, 2018).

2.2 Omnibus Law

Omnibus law adalah undang-undang yang substansinya merevisi dan/atau mencabut banyak undang-undang. Konsep omnibus law menawarkan pembenahan permasalahan yang disebabkan karena peraturan yang terlalu banyak (*over regulasi*) dan tumpang tindih (*overlapping*) aturan Undang-Undang baru yang memuat beragam substansi aturan yang keberadaannya mengamandemen beberapa Undang-Undang dengan tujuan mengatasi terjadinya stagnasi pada pertumbuhan ekonomi (Rudy et al, 2021). Bila permasalahan tersebut diselesaikan dengan cara biasa, maka akan memakan waktu yang cukup lama dan biaya yang tidak sedikit. Belum lagi proses perancangan dan pembentukan peraturan perundang-undangan seringkali menimbulkan *deadlock* atau tidak sesuai kepentingan (Putra, 2020).

2.3 *Text Mining*

Salah satu variasi dari *data mining* yaitu *text mining*. *Text mining* bekerja dengan cara menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar (Kurniawan et al, 2012). *Text mining* bertujuan untuk memperoleh informasi yang berguna dari sekumpulan dokumen yang diklasifikasikan secara otomatis. Permasalahan yang biasa ditangani oleh *text mining* selain klasifikasi yaitu *information extraction, clustering, dan information retrieval* (Prakasa, 2016). Selain itu juga *text mining* mempunyai tujuan yaitu untuk mencari kata dalam sekumpulan dokumen dan melakukan analisa keterhubungan kata dalam dokumen tersebut (Praseptian, M.D., & Indriani, 2014). Proses *text mining* diilustrasikan pada Gambar 1.



Gambar 1. Proses *text mining* (Dang dan Ahmad, 2014)

2.4 Python

Python adalah bahasa pemrograman interpretatif multiguna. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks. Hal ini membuat Python sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai bahasa pemrograman lain. Bahasa ini muncul pertama kali pada tahun 1991, dirancang oleh seorang bernama Guido van Rossum. Sampai saat ini Python masih dikembangkan oleh Python Software Foundation. Bahasa Python mendukung hampir semua

sistem operasi, bahkan untuk sistem operasi Linux, hampir semua distronya sudah menyertakan Python di dalamnya (Anonymous, 2019).

2.5 *Natural Language Tool Kit* (NLTK)

Toolkit yang disediakan oleh bahasa pemrograman Python untuk mempermudah dalam menjalanan tugas-tugas pada *natural language processing* dengan menyediakan berbagai fungsi dan *wrapper*, serta corpora standar baik itu mentah atau pun *pre-processed* yang digunakan (Kholid Fuadi, 2013).

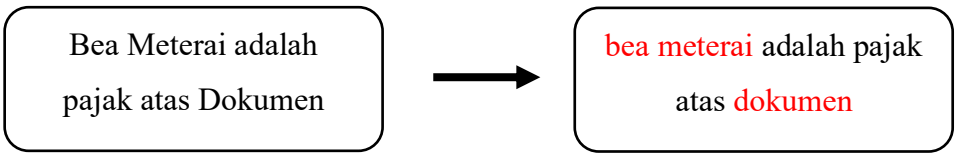
2.6 *Preprocessing*

Preprocessing yaitu proses pendahulu yang diterapkan terhadap data teks yang bertujuan untuk menghasilkan data numerik. Pada proses *preprocessing* merupakan tahap dimana deskripsi ditangani untuk dapat siap diproses memasuki tahap text mining (Nurhayati, 2011). Dalam penelitian ini terdapat 4 tahap *preprocessing* yaitu, *case folding, stopwords, tokenizing,* dan *stemming*.

Dalam jurnal (Aristoteles et al, 2012) menggunakan pengukuran bobot dalam dokumen berbahasa indonesia untuk melakukan peringkasan dokumen sehingga dokumen yang akan ringkas diproses sehingga memiliki bobot yang tidak terlalu besar namun tetap memiliki isi dan kesimpulan yang sama, dengan menghilangkan beberapa kalimat atau kata yang tidak penting.

2.6.1 *Case Folding*

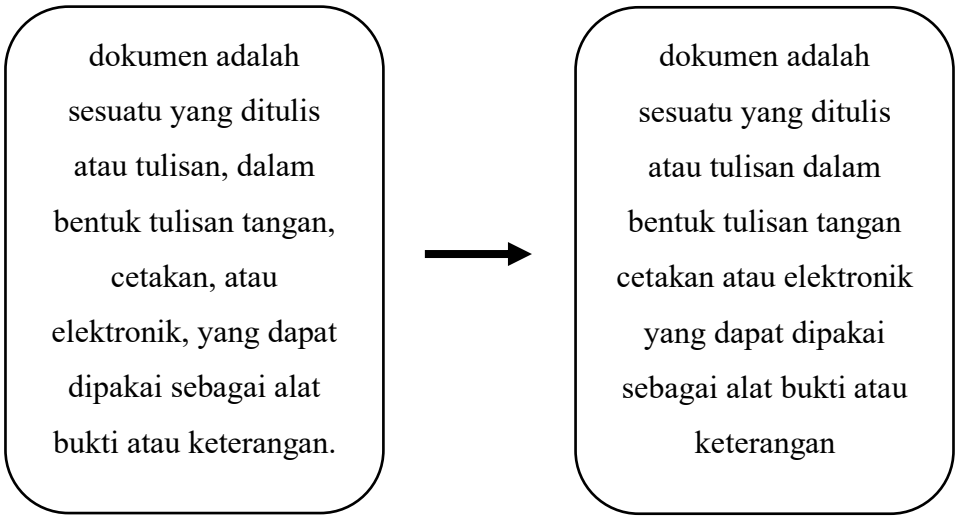
Menurut (Lutfi et al., 2018) *case folding* merupakan langkah konversi *font* dengan mengubah semua huruf menjadi huruf kecil. *Case folding* merupakan proses mengubah semua huruf di dalam dokumen menjadi huruf kecil serta menghilangkan karakter selain huruf (Wisnu & Hetami, 2015). Contoh *Case folding* dapat dilihat pada Gambar 2.



Gambar 2. Contoh *Case Folding*

2.6.2 *Stopword*

Menurut (Setiawan et al., 2013) pengertian *stopword* adalah sekumpulan kata yang tidak berhubungan (*irrelevant*) dengan subjek utama yang dimaksud meskipun kata-kata tersebut sering muncul di dalam data yang digunakan. Tujuan *stopword removal* ini adalah menghilangkan sebuah *array* kata kunci yang dianggap tidak penting sehingga hanya mengandung kata-kata yang bermakna atau sudah terbebas dari *stopword*. Contoh *stopword* dapat dilihat pada Gambar 3.

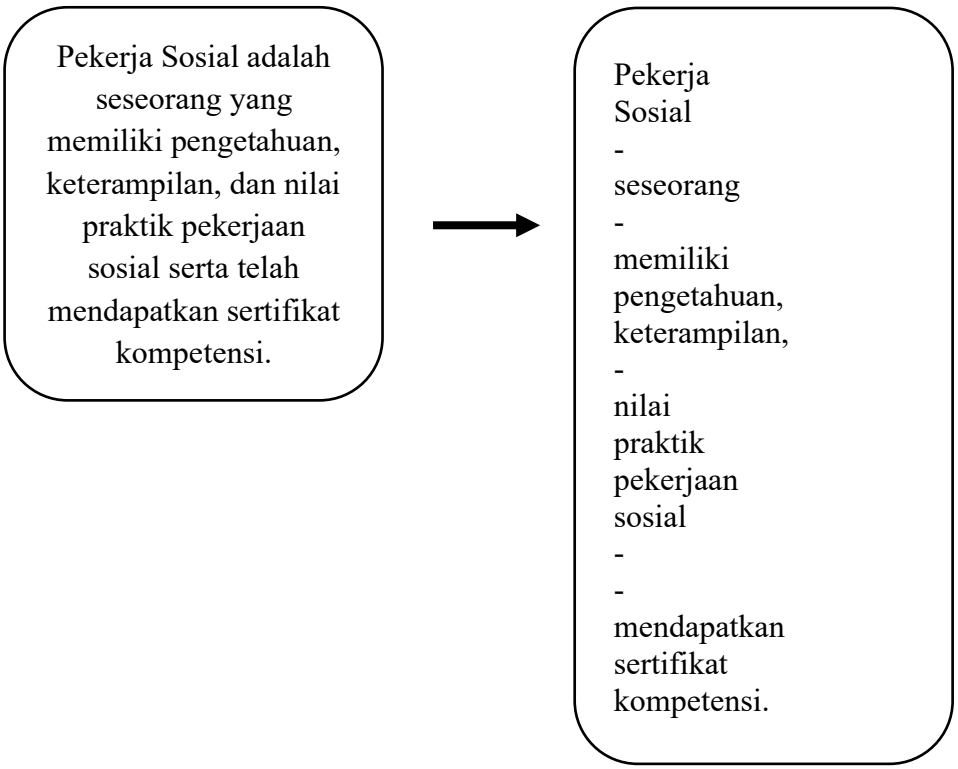


Gambar 3. Contoh *Stopword*

2.6.3 *Tokenizing*

Tokenizing adalah sebuah proses pemotongan kalimat menjadi setiap kata yang menyusunnya (Lutfi et al., 2018). Potongan-potongan kata ini disebut token atau term. *Tokenizing* memotong tiap kata dalam kalimat

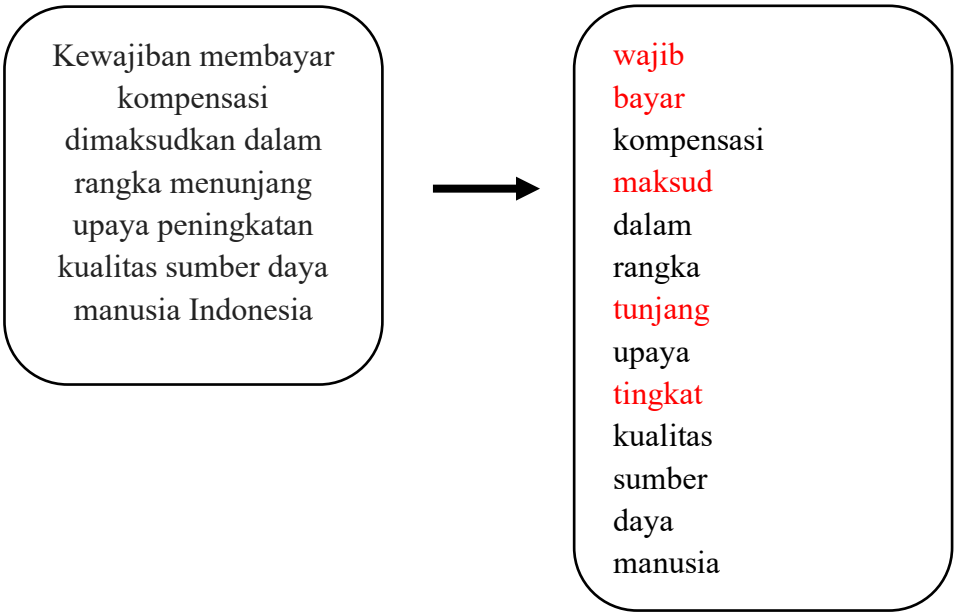
menggunakan spasi sebagai pembatas yang akan menghasilkan token (Wisnu & Hetami, 2015). Contoh *tokenizing* dapat dilihat pada Gambar 4.



Gambar 4. Contoh *Tokenizing*

2.6.4 *Stemming*

Stemming adalah proses pengubahan kata berimbuhan menjadi bentuk kata dasar (Lutfi et al., 2018). *Stemming* digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut yang sesuai dengan struktur morfologi Bahasa Indonesia yang baik dan benar. Contoh *stemming* dapat dilihat pada Gambar 5.



Gambar 5. Contoh *Stemming*

2.7 N-Gram

N-gram adalah potongan n karakter dalam suatu *string* tertentu atau potongan n kata dalam suatu kalimat tertentu (Lisangan, 2015). N-gram dapat dibedakan berdasarkan berapa jumlah huruf yang dipergunakan dalam pemisahan huruf untuk setiap kata antara lain dengan nilai n = 1, 2, 3 atau 4. n = 1 biasanya disebut dengan Uni-gram, n = 2 disebut dengan Bi-gram, n = 3 disebut dengan Tri-gram dan n = 4 disebut Quad-gram.

Blank ditambahkan pada awal dan akhir suatu string untuk mengetahui batas awal dan akhir suatu *string*(Zaman et al., 2015). Misal dalam kata “SAMA” akan didapatkan N-gram sebagai berikut:

Uni-gram : S, A, M, A

Bi-gram : _S, SA, AM, MA, A_

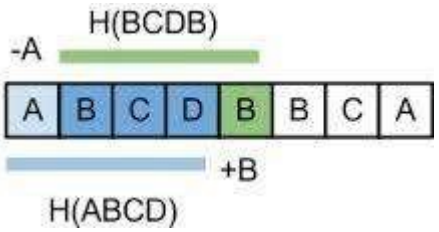
Tri-gram : _SA, SAM, AMA, MA_, A__

Quad-gram : _SAM, SAMA, AMA, MA__, A___

Keuntungan N-Gram dalam pencocokan *string* karena karakteristik N-Gram sebagai bagian dari suatu string, sehingga kesalahan pada sebagian string hanya akan berakibat perbedaan pada sebagian N-Gram. Sebagai contoh jika N-Gram dari dua string dibandingkan, kemudian menghitung cacah N-Gram yang sama dari dua *string* tersebut maka akan didapatkan nilai kesamaan dua *string* tersebut yang bersifat resistan terhadap kesalahan tekstual (Padró, M. & Padró, 2004).

2.8 *Rolling Hash*

Rolling hash merupakan salah satu metode *hashing* yang memberikan kemampuan untuk menghitung nilai *hash* tanpa mengulangi seluruh *string*. Nilai *hash* merupakan nilai numerik yang dibentuk dari kode ASCII. Berikut ini merupakan ilustrasi contoh menemukan pola "bcdb" pada *string* "abcdbbca" yang ditunjukkan pada Gambar 6 (Wibowo & Hastuti, 2016).



Gambar 6. Ilustrasi pencarian pola pada *string* menggunakan metode window rolling hash(Wibowo & Hastuti, 2016)

Berikut adalah formula (1) untuk menghitung hashing.

$$H = (c_1 * 10^{(k-1)} + c_2 * 10^{(k-2)} + \cdots + c_n * 10^{(k-k)}) \bmod n \quad \ldots(1)$$

Keterangan :

H = *Hashing* untuk setiap *k-gram*

c = Nilai ASCII karakter

k = Nilai $k - gram$/jumlah karakter yang dihitung pada gram

n = basis bilangan prima

2.9 *Jaccard Similarity*

Jaccard Similarity atau *Jaccard Coefficient* yang dikembangkan oleh Paul Jaccard pada tahun 1901 merupakan algoritma yang fungsinya untuk membandingkan dua sampel yaitu dokumen yang satu dengan yang lainnya berdasarkan kata yang dimilikinya. *Jaccard similarity* biasanya digunakan untuk membandingkan dokumen dan menghitung nilai kemiripan (*similarity*) dari dua buah objek atau dokumen (Sunardi et al., 2018). Namun *Jaccard similarity* memiliki kelemahan dimana perhitungannya tidak memperhatikan *term frequency*(berapa kali suatu term terdapat di dalam suatu dokumen). *Jaccard similarity* dapat dirumuskan sebagai berikut:

$$Jaccard\ (X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \ldots(2)$$

Dimana :

X = *String* dalam dokumen 1

Y = *String* dalam dokumen 2

Rumus tersebut digunakan untuk mencari persamaan dan perbedaan pada dua sampel. Sebagai contoh jika diketahui:

X = jurusan ilmu komputer

Y = ilmu komputer fmipa unila

$X \cap Y$ = {ilmu, komputer}

$X \cup Y$ = {jurusan, ilmu, komputer, fmipa, unila}

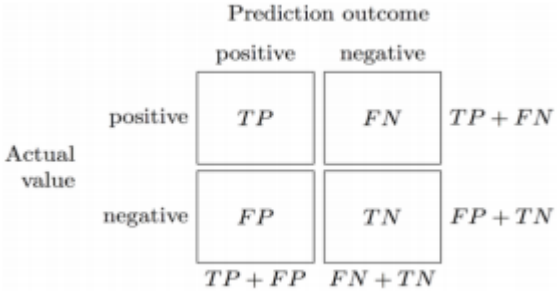
maka akan menghasilkan nilai:

$$Jaccard\ (X,Y) = \frac{|ilmu\ komputer|}{|jurusan\ ilmu\ komputer\ fmipa\ unila|}$$

$$Jaccard\ (X,Y) = \frac{|2|}{|5|} = 0{,}4 \text{ x } 100\% = 40\%$$

2.10 *Confusion Matrix*

Confusion Matrix adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Ada empat istilah yang merupakan representasi hasil proses klasifikasi pada *confusion matrix* yaitu True Positif, True Negatif, False Positif, dan False Negatif (Anggreany, 2020). Contoh *confusion matrix* dapat dilihat pada Gambar 7.



Gambar 7. *Confusion Matrix*(Filcha & Hayaty, 2019)

True Positive (TP) merupakan jumlah prediksi data benar dengan label benar.

True Negative (TN) merupakan jumlah prediksi data benar dengan label salah.

False Positive (FP) merupakan jumlah prediksi data salah dengan label benar.

False Negative (FN) merupakan jumlah prediksi data salah dengan label salah.

2.10.1 *Precission*

Precission adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem(Data, 2013).

$$Precission = \frac{TP}{TP+FP}\ x\ 100\% \ldots(3)$$

2.10.2 *Recall*

Recall adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi(Data, 2013).

$$Recall = \frac{TP}{TP+FN}\ x\ 100\% \ldots(4)$$

2.10.3 *Accuracy*

Accuracy adalah tingkat kedekatan antara nilai prediksi dengan nilai aktual(Data, 2013).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}\ x\ 100\% \quad \ldots (5)$$

BAB III METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian dilaksanakan pada semester ganjil tahun akademik 2020/2021 di Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung yang berada di Jl. Soemantri Brojonegoro No. 1 Gedong Meneng, Bandar Lampung. Adapun alur penelitian secara garis besar dibagi menjadi 3 tahap, yaitu pengumpulan data, implementasi *text mining,* dan analisis hasil implementasi.

3.2 Alat dan Data

Adapun alat dan bahan yang digunakan dalam penelitian ini antara lain:

1. Perangkat Keras (*Hardware*)

   Perangkat keras yang digunakan dalam penelitian ini adalah satu unit laptop dengan spesifikasi:

   a. *Processor* : AMD A12-9700P CPU 2.5GHz
   b. *Installed RAM* : 8.00 GB
   c. *System type* : 64-bit Operating System
   d. *Operating System* : Windows 10 Home Single Language

2. Perangkat Lunak (*Software*)

   Perangkat lunak yang digunakan dalam penelitian ini adalah:

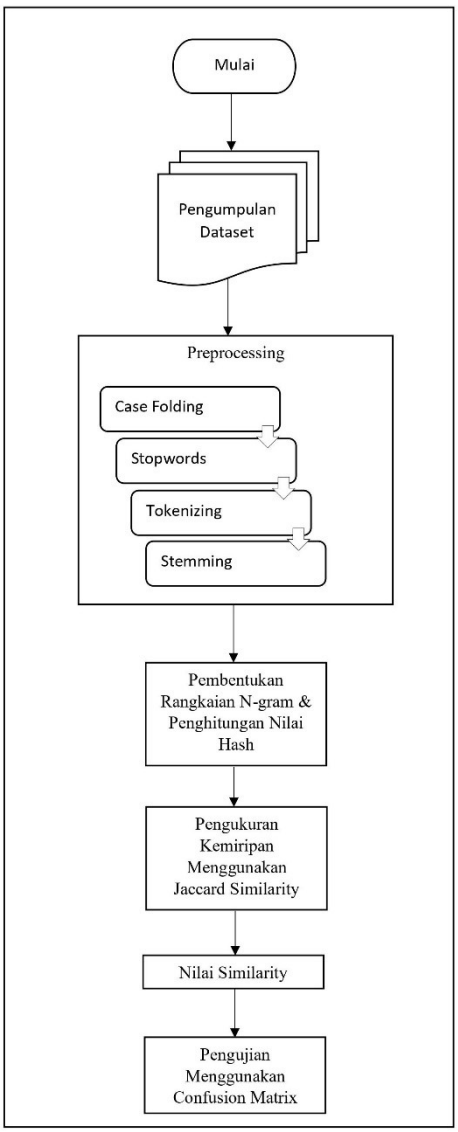
   a. Jupyter Notebook
   b. Anaconda Navigator
   c. Visual Studio Code
   d. Web Browser Google Chrome

3. Data

   Data yang digunakan dalam penelitian ini ialah berupa data rancangan undang-undang dan undang-undang Republik Indonesia yang telah disahkan pemerintah berformat *file pdf*. Data tersebut bersumber dari Fakultas Hukum Universitas Lampung dan Jaringan Dokumen dan Informasi Hukum Nasional. Dokumen undang-undang yang digunakan untuk dataset merupakan undang-undang dari tahun 2010 sampai 2020. Lalu undang-undang tersebut diklasifikasikan menjadi beberapa kategori untuk memudahkan proses mencari kesamaan antar dokumen.

3.3 Tahapan Penelitian

Tahapan penelitian yang dilakukan melalui beberapa tahapan yaitu pengumpulan dataset, *preprocessing*, pembentukan rangkaian n-gram, perhitungan nilai kemiripan dengan metode *jaccard similarity*, memunculkan nilai *similarity*. Alur penelitian dapat dilihat pada Gambar 8.



Gambar 8. Alur penelitian implementasi *text mining* untuk *similarity* undang-undang

3.3.1 Pengumpulan Data

Tahap awal dari penelitian ini adalah dengan mengumpulkan sumber data yang akan dipergunakan untuk dibandingkan yaitu Rancangan Undang-Undang dan Undang-Undang Republik Indonesia yang telah disahkan. Pengumpulan data dilaksanakan pada semester ganjil tahun akademik 2020/2021. Undang-undang yang dikumpulkan sebelumnya telah diunduh pada *website* https://peraturan.go.id/ dan juga dari Fakultas Hukum

Universitas Lampung untuk dimasukkan kedalam *database*. Undang-undang yang dipergunakan sebagai dataset hanya undang-undang yang berformat pdf(*portable document format*). Jika sudah, dokumen undang-undang akan diubah menjadi file berformat *txt* untuk mempermudah pencarian kata. Konversi format dokumen dari *pdf* ke *txt* menggunakan *library* dalam pemrograman bahasa Python yaitu dengan *pdfminer*. Kemudian dokumen hasil konversi akan melalui tahap *preprocessing* untuk memudahkan pengolahan pencarian kemiripan teks antar dokumen.

3.3.2 *Preprocessing*

Tahap *preprocessing* teks dalam penelitian ini melalui 4 tahapan yaitu, *case folding*, *stopwords*, *tokenizing*, dan *stemming*.

1. *Case Folding*

   Pada tahap ini, semua huruf yang ada pada setiap dokumen diubah menjadi huruf kecil. Berikut ilustrasi dari *case folding* pada Tabel 1.

   Tabel 1. Ilustrasi *Case Folding*

Data Input	Hasil Case Folding
Berdasarkan sejarah, Indonesia mewarisi sistem hukum civil law.	berdasarkan sejarah, indonesia mewarisi sistem hukum civil law.

2. *Stopwords*

   Tahap ini bertujuan untuk menghilangkan kata yang tidak memiliki arti atau makna pada dokumen. Berikut ilustrasi dari *Stopword* pada Tabel 2.

Tabel 2. Ilustrasi *Stopword*

Data Hasil Case Folding	Hasil Stopword
berdasarkan sejarah, indonesia mewarisi sistem hukum civil law.	berdasarkan sejarah indonesia mewarisi sistem hukum civil law

3. *Tokenizing*

   Ini merupakan sebuah langka digunakan untuk pemotongan *string* input berdasarkan setiap kata yang menyusunnya, paragraf dan kalimat dipotong berdasarkan kata penyusunnya. Pada tahap ini juga dibuat daftar kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Berikut ilustrasi *tokenizing* pada Tabel 3.

Tabel 3. Ilustrasi *Tokenizing*

Data Hasil Stopword	Hasil Tokenizing
berdasarkan sejarah indonesia mewarisi sistem hukum civil law	- sejarah indonesia mewarisi sistem hukum civil law

4. *Stemming*

   Pada tahap *stemming*, kata yang terdapat pada dokumen diubah menjadi kata dasar dengan membuang imbuhan awal dan imbuhan akhir. Berikut ilustrasi *stemming* pada Tabel 4.

Tabel 4. Ilustrasi *Stemming*

Data Hasil Stopword	Hasil Tokenizing
-	-
sejarah	sejarah
indonesia	indonesia
mewarisi	waris
sistem	sistem
hukum	hukum
civil	civil
law	law

3.3.3 Pembentukan Rangkaian N-Gram

Hasil dari *preprosessing* akan menghasilkan data teks murni yang akan digunakan pada proses n-gram. N-gram adalah *substring* penggabungan karakter sejumlah *n* pada teks dokumen. Berikut ini ilustrasi pembentukan rangkaian n-gram menggunakan *string* "hukum" pada Tabel 5.

Tabel 5. contoh proses pembentukan rangkaian n-gram

Atribut	Nilai Array
Rangkaian string dengan n gram 3 terhadap string "hukum"	[0] => huk [1] => uku [2] => kum

3.3.4 Penghitungan Nilai *Hash/Rolling Hash*

Proses untuk mengubah rangkaian *string* menjadi rangkaian nilai *hash* menggunakan *rolling hash*. *Rolling hash* memungkinkan untuk menghitung nilai *hash* tanpa melakukan *rehashing* kembali dari iterasi pertama.

$$\mathbf{H}(c_1 \dots c_2) = c_1 * b^{k-1} + c_2 * b^{k-2} * \dots + c_{k-1} * b + c_k$$

Untuk menghitung nilai *hash* pertama dilakukan perhitungan dengan menggunakan rumus diatas. $H(c_1 \dots c_n)$ merupakan nilai *hash* pertama, dimana *b* adalah nilai konstan bilangan prima, *k* adalah nilai *kgram* dan

c_n merupakan nilai ascii dari karakter. Berikut ini merupakan contoh perhitungan nilai *hash* menggunakan rangkaian n-gram yang ditujukan pada Tabel 6.

Tabel 6. contoh perhitungan *subtring* dengan menggunakan *hash*

Perhitungan *hash* iterasi pertama	[0] => huk h = 104, u = 117 , k = 107, b = 7, k = 3 H = $\boldsymbol{c_1 * b^{k-1} + c_2 * b^{k-2} * ... + c_{k-1} * b + c_k}$ H = (104*7^2) + (117*7^1) + (107*7^0) H = 5096 + 819 + 107 H = 6022
Perhitungan *hash* iterasi kedua	[1] => uku u = 117, k = 107 , u = 117, b = 7, k = 3 H = $\boldsymbol{c_1 * b^{k-1} + c_2 * b^{k-2} * ... + c_{k-1} * b + c_k}$ H = (117*7^2) + (107*7^1) + (117*7^0) H = 5733 + 749 + 117 H = 6599
Perhitungan *hash* iterasi ketiga dan seterusnya	[2] => kum k = 107, u = 117 , m = 109, b = 7, k = 3 H = $\boldsymbol{c_1 * b^{k-1} + c_2 * b^{k-2} * ... + c_{k-1} * b + c_k}$ H = (107*7^2) + (117*7^1) + (109*7^0) H = 5243 + 819 + 109 H = 6171

Setiap *string* akan mendapatkan nilai *hash* yang berbeda-beda bedasarkan nilai dari n-gram yang dipakai. Setelah mendapatkan nilai *hash* sampai dengan *n-substring*, selanjutnya akan dilakukan perhitungan dengan metode *jaccard similarity*.

3.3.5 Pengukuran Kemiripan Menggunakan *Jaccard Similarity*

Nilai dari proses *hashing* tiap dokumen akan dicocokan menggunakan *jaccard coefficient* untuk mengukur persentase kemiripan teks. Output dari proses ini adalah nilai persentase kemiripan antara 2 dokumen.

Penghitungan *jaccard coefficient* dilakukan berdasarkan rumus *hashing* berikut:$J(A,B) = |A \cap B||A \cup B|$, dimana J(A,B) adalah nilai kemiripan antara dataset A dan B, $A \cap B$ adalah irisan/data yang sama dari A dan B, dan $A \cup B$ adalah *union*/gabungan data dari A dan B. Dari hasil tersebut dikalikan 100 untuk menghasilkan nilai persentase. Berikut rumus dari *jaccard coefficient*.

$$\textbf{\textit{Jaccard (A,B)}} = \frac{|A \cap B|}{|A \cup B|} \; \boldsymbol{x} \; \mathbf{100}\%$$

Bila terdapat potongan *string*

A = {civil law}

B = {common law}

Kemudian tentukan nilai pembobotan menggunakan n-gram dan perhitungan *hashing* dengan nilai n = 5, maka

A = {3118, 3320, 3583, 3273, 3337, 3304}

B = {3139, 3429, 3381, 3390, 3433, 3387, 3304}

$A \cap B$ = {3304}

$A \cup B$ = {3118, 3320, 3583, 3273, 3337, 3304, 3139, 3429, 3381, 3390, 3433, 3387}

$$\textbf{\textit{Jaccard (A,B)}} = \frac{|1|}{|12|} \; \boldsymbol{x} \; \mathbf{100}\%$$

$$\textbf{\textit{Jaccard (A,B)}} = \mathbf{0,083} \; \boldsymbol{x} \; \mathbf{100}\%$$

***Jaccard (A,B)* = 8,3%**

Jadi tingkat kesamaan antara *string* A dan *string* B setelah melalui proses perhitungan *jaccard similarity* adalah 8,3%.

3.3.6 Pengujian Akurasi dengan *Confusion Matrix*

Nilai kemiripan yang didapatkan dengan menggunakan metode *jaccard similarity* akan melalui tahapan pengujian akurasi menggunakan *confusion Matrix*. Pengujian ditujukan untuk mengukur keakuratan nilai yang didapat dari sistem maupun dengan pengecekan manual oleh seorang *expert*. Contoh pengujian dengan *confusion matrix* dapat dilihat pada Tabel 7.

Tabel 7. Ilustrasi Klasifikasi dengan *Confusion Matrix*

No	Rancangan Undang-Undang	Undang-Undang	Jaccard Similarity		Expert		Klasifikasi
			Tinggi	Rendah	Tinggi	Rendah	
1	RUU No 1 Tahun 2021.pdf	UU No 1 Tahun 2019.pdf	✓		✓		TP
2	RUU No 1 Tahun 2021.pdf	UU No 1 Tahun 2020.pdf	✓		✓		TP
3	RUU No 1 Tahun 2021.pdf	UU No 1 Tahun 2018.pdf	✓		✓		TP
4	RUU No 1 Tahun 2021.pdf	UU No 1 Tahun 2017.pdf	✓		✓		TP
5	RUU No 1 Tahun 2021.pdf	UU No 1 Tahun 2016.pdf		✓		✓	TN
6	RUU No 1 Tahun 2021.pdf	UU No 1 Tahun 2015.pdf		✓		✓	TN
7	RUU No 1 Tahun 2021.pdf	UU No 1 Tahun 2014.pdf		✓		✓	TN
8	RUU No 1 Tahun 2021.pdf	UU No 1 Tahun 2013.pdf	✓			✓	FN

Penentuan tinggi atau rendah hasil dari *similarity* didasarkan pada persentase yang didapatkan. Apabila nilai *similarity* lebih dari atau sama dengan 25% maka dikategorikan tinggi. Sebaliknya, jika nilai *similarity* dibawah 25% maka dikategorikan rendah. Dari hasil penujian pada Tabel 7 klasifikasi data uji maka diketahui tabel *confusion matrix* pada Tabel 8.

Tabel 8. Ilustrasi *Confusion Matrix*

		Prediksi (Jaccard Similarity)	
		Tinggi	Rendah
Actual (Expert)	Tinggi	TP (4)	FN (1)
	Rendah	FP (0)	TN (3)

Tabel 8 merupakan hasil klasifikasi dari Tabel 7 yang menyatakan tinggi atau rendah dua data yang diuji.

1. Precision

Precision adalah ketepatan program pengukuran tingkat kemiripan (*similarity*) melakukan pengambilan data yang diminta atau dibutuhkan yang diukur dari tabel *confusion matrix* (Data, 2013), berikut ilustrasi perhitungan nilai *precision*.

$$Precision = \frac{TP}{TP + FP}\ x\ 100\%$$

$$Precision = \frac{4}{4 + 0}\ x\ 100\%$$

$$Precision = 100\%$$

2. Recall

Recall adalah tingkat keberhasilan program pengukuran tingkat kemiripan (*similarity*) dalam menemukan data yang diminta atau

dibutuhkan oleh pengguna diukur dari tabel *confusion matrix* (Data, 2013), berikut ilustrasi perhitungan nilai *recall*.

$$Recall = \frac{TP}{TP + FN}\ x\ 100\%$$

$$Recall = \frac{4}{4 + 1}\ x\ 100\%$$

$$Recall = 80\%$$

3. Accuracy

Accuracy adalah tingkat kedekatan antara nilai prediksi oleh program pengukuran tingkat kemiripan (*similarity*) dengan nilai sebenarnya (*expert*) diukur dari tabel *confusion matrix* (Data, 2013), berikut ilustrasi perhitungan nilai *accuracy*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}\ x\ 100\%$$

$$Accuracy = \frac{4 + 3}{4 + 3 + 0 + 1}\ x\ 100\%$$

$$Accuracy = 87.5\%$$

Berdasarkan perhitungan dari Tabel 8 dengan menggunakan rumus *confusion matrix* didapatkan nilai *precision* sebesar 100%, *recall* sebesar 80%, dan *accuracy* sebesar 87,5%. Nilai tersebut menjadi acuan dalam keabsahan *similarity* yang didapatkan dengan metode *jaccard similarity*.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari penelitian didapatkan kesimpulan yaitu telah dilakukan pengukuran tingkat kemiripan antara Rancangan Undang-Undang dengan Undang-Undang yang berformat *pdf(portable document format)* dan berhasil mencari nilai *similarity* antar dokumen menggunakan metode *jaccard similarity*.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, didapatkan saran sebagai berikut.

1. Data yang diinputkan harus disempurnakan karena ada beberapa dokumen yang berformat *.pdf* berisi *file image*. Hal tersebut yang mempersulit pengolahan data pada tahap *preprocessing*.
2. Perlu dikembangkannya pencarian nilai *similarity* UU kedalam sebuah *website* agar memudahkan pengguna lain dalam mencari tingkat kemiripan RUU dengan UU.
3. Pada penelitian selanjutnya dapat dikembangkan dengan berbagai metode agar mendapatkan nilai *similarity* yang lebih akurat.

DAFTAR PUSTAKA

Anggreany, M. S. (2020). *Confusion Matrix*. Https://Socs.Binus.Ac.Id/2020/11/01/Confusion-Matrix/.

Anonymous. (2019). *Pendahuluan Python*. https://belajarpython.com/tutorial/apa-itu-python

Arham, S., Saleh, A., & Kunci, K. (2019). Omnibus Law Dalam Perspektif Hukum Indonesia. *Uit.E-Journal.Id*.

Aristoteles, Yeni Herdiyeni, Ahmad Ridha, J. A. (2012). Text Feature Weighting For Summarization Of Document Bahasa Indonesia Using Genetic Algorithm. *International Journal of Computer Science Issues*, *9*(3), 1–6.

Data, M. (2013). *Perbedaan: precision, recall & accuracy*. https://dataq.wordpress.com/2013/06/16/perbedaan-precision-recall-accuracy/

Fadli, M. (2018). Pembentukan Undang-Undang Yang Mengikuti Perkembangan Masyarakat. *Jurnal Legislasi Indonesia*, *15*(1), 49–58. http://e-jurnal.peraturan.go.id/index.php/jli/article/view/12/pdf

Filcha, A., & Hayaty, M. (2019). Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa. *JUITA : Jurnal Informatika*, *7*(1), 25. https://doi.org/10.30595/juita.v7i1.4063

Hantoro, N. M. (2019). Parliamentary Review OMNIBUS LAW. *Parliamentari Review*, *1*(1), 1–9.

Kurniawan, B., Effendi, S., & Sitompul, O. (2012). Klasifikasi Konten Berita dengan Metode Text Mining. *Kurniawan, B., Effendi, S., & Sitompul, O. . (2012). Klasifikasi Konten Berita Dengan Metode Text Mining. 1 No.1, 14–19.*, *1 No.1*, 14–19.

Lisangan, E. A. (2015). Implementasi n-Gram Technique dalam Deteksi Plagiarisme pada Tugas Mahasiswa. *Lisangan, E.A., 2015, Implementasi n-Gram Technique Dalam Deteksi Plagiarisme Pada Tugas Mahasiswa. Jurnal Tematika, Vol. 1 No. 2, ISSN: 2303-387824- 30*, *1*.

Lutfi, A. A., Permanasari, A. E., dan Fauziati, S. (2018). Sentiment analysis in the sales review by indonesian marketplace by utilizing support vector machine. *JISEBI*, *4(1)*, 57–64.

Michael, T. (2020). Bentuk Pemerintahan Perspektif Omnibus Law. *Jurnal Ius Constituendum*, *5*(1), 159. https://doi.org/10.26623/jic.v5i1.2222

Nurdiana, O., Jumadi, J., & Nursantika, D. (2016). Perbandingan Metode Cosine Similarity Dengan Metode Jaccard Similarity Pada Aplikasi Pencarian Terjemah Al-Qur'an Dalam Bahasa Indonesia. *Jurnal Online Informatika*. https://doi.org/10.15575/join.v1i1.12

Nurhayati, S. (2011). Implementasi Text Mining Untuk Klasifikasi Kesenian Tradisional Dengan Metode NBC (Naive Bayes Classifier). *Nurhayati, S. (2011). Implementasi Text Mining Untuk Klasifikasi Kesenian Tradisional Dengan Metode NBC (Naive Bayes Classifier).*

Padró, M. & Padró, L. (2004). Comparing methods for language identification. *Padró, M., & Padró, L. (2004). Comparing Methods for Language Identification. Procesamiento Del Lenguaje Natural, 33.*, *33*.

Prakasa, S. . (2016). Text Mining. *Text Mining. Sekolah Tinggi Teknologi Telematika Telkom Purwokerto*.

Praseptian, M.D., & Indriani, A. (2014). *Implementasi Text Mining dalam Klasifikasi Buku dengan Metode Naïve Bayes Classifier Studi Kasus pada*

Perpustakaan STMIK PPKIA Tarakanita Rahmawati. 243–247.

Putra, A. (2020). Penerapan Omnibus Law Dalam Upaya Reformasi Regulasi. *Jurnal Legislasi Indonesia*.

Rudy. Aristoteles. Kurniawan, R. C. (2021). *Model Omnilaw: Solusi Pemecahan Masalah Penyederhanaan Legislasi Dalam Rangka Pembangunan Hukum*. Pusaka Media.

Setiawan, A., Kurniawan, E., & Handiwidjojo, W. (2013). Implementasi Stop Word Removal Untuk Pembangunan Aplikasi Alkitab Berbasis Windows 8. *Jurnal EKSIS*, *6(2)*, 1–11.

Stepchyshyn, V., & Nelson, R. S. (2007). Library plagiarism policies. *Stepchyshyn, V., & Nelson, R. S. (Eds.). (2007). Library Plagiarism Policies (Vol. 37). Assoc of Cllge & Rsrch Libr.*, *37*.

Sunardi, S., Yudhana, A., & Mukaromah, I. A. (2018). Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram Dan Jaccard Similarity Terhadap Algoritma Winnowing. *Transmisi*, *20*(3), 105. https://doi.org/10.14710/transmisi.20.3.105-110

Wibowo, R. K., & Hastuti, K. (2016). Penerapan Algoritma Winnowing Untuk Mendeteksi Kemiripan Teks pada Tugas Akhir Mahasiswa. *Techno.Com*, *15*(4), 303–311.

Wisnu, D., & Hetami, A. (2015). Perancangan Information Retrieval (IR) Untuk Pencarian Ide Pokok Teks Artikel Berbahasa Inggris Dengan Pembobotan Vector Space Model. *Jurnal Ilmiah Teknologi Dan Informasi ASIA*, *9(1)*, 53–59.

Zaman, B., Hariyanti, E., & Purwanti, E. (2015). Sistem Deteksi Bahasa pada Dokumen menggunakan N-Gram. *Multinetics*, *1*(2), 21. https://doi.org/10.32722/vol1.no2.2015.pp21-26