# MENGUKUR TINGKAT KEMIRIPAN ANTARA RUU DENGAN UU MENGGUNAKAN ALGORITMA *RABIN KARP* DENGAN METODE *K-GRAM* DAN *HASHING*

**(Skripsi)**

**Oleh**

**FIKRI AULIA**
**NPM 1717051047**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**
**UNIVERSITAS LAMPUNG**
**BANDAR LAMPUNG**
**2021**

## ABSTRACT

## MEASURING THE SIMILARITY BETWEEN DRAFT LAWS AND LAWS USING THE RABIN KARP ALGORITHM WITH K-GRAM AND HASHING METHODS

**BY**

**FIKRI AULIA**

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 Rabin Karp algorithm with the k-gram and hashing method, 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; Rabin Karp; K-grams; Hashing.

# ABSTRAK

## MENGUKUR TINGKAT KEMIRIPAN ANTARA RUU DAN UU MENGGUNAKAN ALGORITMA *RABIN KARP* DENGAN METODE *K-GRAM* DAN *HASHING*

**OLEH**

**FIKRI AULIA**

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 algoritma *Rabin Karp* dengan metode *k-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; Rabin Karp; K-Gram; Hashing

# MENGUKUR TINGKAT KEMIRIPAN ANTARA RUU DENGAN UU MENGGUNAKAN ALGORITMA *RABIN KARP* DENGAN METODE *K-GRAM* DAN *HASHING*

**Oleh**

**FIKRI AULIA**

**Skripsi**

**Sebagai Salah Satu Syarat untuk Mencapai Gelar**
**SARJANA KOMPUTER**

**Pada**
**Jurusan Ilmu Komputer**
**Fakultas Matematika dan Ilmu Pengetahuan Alam**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**
**UNIVERSITAS LAMPUNG**
**BANDAR LAMPUNG**
**2021**

| | |
|---|---|
| Judul Skripsi | : **MENGUKUR TINGKAT KEMIRIPAN ANTARA RUU DENGAN UU MENGGUNAKAN ALGORITMA *RABIN KARP* DENGAN METODE *K-GRAM* DAN *HASHING*** |
| Nama Mahasiswa | : Fikri Aulia |
| Nomor Pokok Mahasiswa | : 1717051047 |
| 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 : **18 Oktober 2021**

## PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul "Mengukur Tingkat Kemiripan Antara RUU dan UU Menggunakan Algoritma *Rabin Karp* dengan Metode *K-gram* dan *Hashing*" 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, 18 Oktober 2021
Yang menyatakan

METERAI TEMPEL
10000
0230AAJX551892

Aulia
NPM. 1717051047

## RIWAYAT HIDUP

Penulis dilahirkan pada tanggal 20 September 1999 di Pariaman, Sumatera Barat. Sebagai anak pertama dari tiga bersaudara dengan Ayah bernama Muhammad Irsal dan Ibu bernama Afrida.

Penulis menyelesaikan pendidikan Taman Kanak-Kanak (TK) Perintis Kab. Kep. Mentawai pada tahun 2005, menyelesaikan Sekolah Dasar (SD) di SDN 27 Kp.Baru Kota Pariaman pada tahun 2011, kemudian menyelesaikan Sekolah Menengah Pertama (SMP) di SMPN 2 Pariaman pada tahun 2014, dan menyelesaikan Sekolah Menengah Akhir (SMA) di SMAN 1 Pariaman pada tahun 2017.

Pada tahun 2017, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer FMIPA Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa penulis menyelesaikan Kerja Praktik di PT. Perkebunan Nusantara 7 pada tahun 2020, dan penulis juga menyelesaikan Kuliah Kerja Nyata (KKN) Daring di Kelurahan Kp.Jawa II, Kota Pariaman, Sumatera Barat pada tahun 2020.

## MOTTO

**"Jika Tuah ada dibadan**
**Pasir digenggam menjadi Intan."**

**“Yakinlah, bahwa sesuatu yang sesuai dengan yang anda cita-citakan,**
**Sudah dipersiapkan sebaik mungkin oleh Allah SWT.”**

**“Dimana bumi dipijak, di situ langit dijunjung.”**
(Pepatah Minangkabau)

**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.

Raisa Irdayanti, Viona Irdayanti 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 “Mengukur Tingkat Kemiripan Antara RUU dan UU Menggunakan Algoritma *Rabin Karp* dengan Metode *K-gram* dan *Hashing*”.

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. Kedua saudara penulis Raisa dan Viona 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, 18 Oktober 2021
Yang menyatakan

Fikri Aulia
NPM. 1717051047

# DAFTAR ISI

**Halaman**

**DAFTAR ISI** ..... **xiii**

**DAFTAR GAMBAR** ..... **xv**

**DAFTAR TABEL** ..... **xvi**

**DAFTAR PSEUDOCODE** ..... **xvii**

**BAB I PENDAHULUAN** ..... **1**
1.1 Latar Belakang ..... 1
1.2 Rumusan Masalah ..... 4
1.3 Batasan Masalah ..... 4
1.4 Tujuan Penelitian ..... 5
1.5 Manfaat Penelitian ..... 5

**BAB II TINJAUAN PUSTAKA** ..... **6**
2.1 Omnibus Law ..... 6
2.2 Undang-Undang ..... 6
2.3 *Text Mining* ..... 7
2.4 *Preprocessing* ..... 7
2.5 *Naturan Language Processing* (NLP) ..... 8
2.6 *Natural Language Tool Kit* (NLTK) ..... 8
2.6.1 *Case Folding* ..... 8
2.6.2 *Tokenizing* ..... 8
2.6.3 *Filtering* ..... 9
2.6.4 *Stemming* ..... 9
2.7 Algoritma *Rabin Karp* ..... 9
2.8 *K-gram* ..... 10
2.9 *Hashing* ..... 10
2.9 *Similarity* ..... 10
2.10 *Confusion Matrix* ..... 11
2.10.1 *Precision* ..... 12
2.10.2 *Recall* ..... 12
2.10.3 *Accuracy* ..... 12

**BAB III METODOLOGI PENELITIAN ..... 13**
3.1 Waktu dan Tempat Penelitian ..... 13
3.2 Alat dan Data..... 13
3.2.1 Alat ..... 13
3.2.2 Data ..... 15
3.3 Tahap Penelitian..... 15
3.3.1 *Preprocesing* Dokumen ..... 17
3.3.2 Penghitungan Nilai *Hashing* ..... 20
3.3.3 Penghitungan Tingkat *Similarity* ..... 20
3.3.4 Pengujian Akurasi Metode ..... 21

**BAB IV HASIL PENELITIAN..... 24**
4.1 Analisis Kebutuhan Data ..... 24
4.2 *Text Mining*..... 24
4.3 Implementasi ..... 24
*4.3.1 Preprocessing*..... 25
4.4 Menghitung Nilai *Hashing* ..... 29
4.5 Pengukuran Tingkat Kemiripan ..... 35
4.6 Pengukuran Tingkat Akurasi Metode..... 40

**BAB V SIMPULAN DAN SARAN..... 43**
5.1 Simpulan..... 43
5.2 Saran ..... 43

**DAFTAR PUSTAKA ..... 44**

# DAFTAR GAMBAR

**Gambar** **Halaman**

Gambar 1. Alur penelitian ..... 16
Gambar 2. Hasil pengukuran tingkat kemiripan (*similarity*) ..... 37
Gambar 3. Pengukuran waktu berjalan program ..... 39

## DAFTAR TABEL

**Tabel** **Halaman**

Tabel 1. *Confusion Matrix* (Data, 2013). ..... 11
Tabel 2. Ilustrasi *Case Folding* ..... 17
Tabel 3. Ilustrasi *Tokenizing* ..... 17
Tabel 4. Ilustrasi *Filtering*..... 18
Tabel 5. Ilustrasi *Stemming* ..... 18
Tabel 6. Ilustrasi dokumen uji..... 19
Tabel 7. Ilustrasi dokumen sumber ..... 19
Tabel 8. Ilustrasi berjalannya hashing dengan *k-gram*=4 ..... 19
Tabel 9. Ilustrasi klasifikasi data..... 21
Tabel 10. Ilustrasi *Confusion Matrix* ..... 22
Tabel 11. Hasil *Case Folding* ..... 26
Tabel 12. Hasil *Tokenizing*..... 27
Tabel 13. Hasil *Filtering* ..... 27
Tabel 14. Hasil *Stemming* ..... 28
Tabel 15. *Replace*..... 29
Tabel 16. Menghitung nilai *hashing* ..... 30
Tabel 17. Pengukuran tingkat kemiripan (*similarity*) ..... 36
Tabel 18. Waktu program berjalan..... 38
Tabel 19. Klasifikasi data..... 40
Tabel 20. Hasil pengukuran metode yang digunakan ..... 41

# DAFTAR PSEUDOCODE

**PSEUDOCODE** **Halaman**

Pseudocode 1. Kode program konversi *file* ..... 25
Pseudocode 2. Kode program *preprocessing*..... 25
Pseudocode 3. Kode program penghitungan nilai *hashing*..... 30
Pseudocode 4. Kode program pengukuran tingkat kemiripan ..... 35

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Undang-Undang adalah peraturan Perundang-Undangan yang dibentuk oleh Dewan Perwakilan Rakyar (DPR) dengan persetujuan Presiden (Fadli, 2018). Undang-Undang yang dibentuk dan ditetapkan berisi peraturan yang mengatur banyak hal terkait kelangsungan hidup masyarakat dan mengatur berjalannya pemerintah.

Peraturan Perundang-undangan adalah semua peraturan tertulis yang memuat norma hukum yang mengikat secara umum dan dibentuk atau ditetapkan oleh lembaga negara atau pejabat yang berwenang melalui beberapa prosedur (Afif, 2018).

Rancangan Undang-Undang merupakan Undang-Undang yang diusulkan untuk disahkan dari lembaga legislatif kepada eksekutif. Rancangan Undang-Undang sendiri dapat berasal dari Presiden, DPR dan DPD. Setelah Rancangan Undang-Undang tersebut disahkan maka disebut sebagai Undang-Undang parlemen (Afif, 2018).

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 pemerintah telah mengesahkan Undang-Undang berbasis Omnibus Law yang menimbulkan banyak pertentangan dari kalangan buruh terkait ketenagakerjaan dan banyak lainnya. Omnibus law sendiri merupakan Undang-Undang yang dapat merevisi banyak Undang-Undang lainnya dengan tujuan untuk mengurangi obesitas regulasi, sehingga tidak ada lagi regulasi yang saling tumpang tindih dan tepat sasaran (Fitryantica, 2019).

Disahkannya Undang-Undang Cipta Kerja yang mengadopsi konsep Omnibus Law, Undang-Undang Cipta Kerja dinilai menjadi langkah awal yang pasti dalam penyederhanaan legislasi di Indonesia (Rudy et al, 2021). Banyak mendapatkan penolakan Penolakan utamanya 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 maka akan membuat kerugian dalam banyak hal terutama dalam organisasi atau perusahaan.

Di sisi lain, penolakan ini menimbulkan pertanyaan : Bagaimana sebenarnya keikutsertaan elemen non-pemerintah dalam hal ini mencakup pengusaha, akademisi, buruh, dan masyarakat pada umumnya, dalam perumusan rancangan Undang-Undang Cipta Kerja?. Berawal dari pertanyaan tersebut muncul urgensi untuk memberikan ruang yang cukup bagi elemen non-pemerintah untuk dapat menyampaikan pendapatnya terkait dengan proses perancangan Undang-Undang yang mengadopsi sistem Omnibus Law.

Melihat urgensi partisipasi masyarakat yang lebih aktif dalam proses perancangan Undang-Undang bersama dengan pemerintah dan parlemen, dengan dilakukannya penelitian ini untuk mengukur tingkat kemiripan (*similarity*) antara Rancangan Undang-Undang dengan Undang-Undang yang sudah ditetapkan sebelumnya, apakah rancangan tersebut dapat mewakili Undang-Undang yang telah ada atau tidak, dengan tujuan mengurangi obesitas regulasi yang ada di Indonesia.

Dalam penelitian yang dilakuan oleh (Mujahidin, 2013), menggunakan algoritma *rabin karp* untuk mendeteksi tingkat kemiripan dua dokumen yang di inputkan ke dalam sistem. *Output* dari penelitian tersebut yaitu sistem pemeriksaan kesamaan dua dokumen dengan hasil berupa persentase tingkat kemiripan, dalam penelitiannya untuk mendapatkan persentase kemiripan menggunakan *text mining* dengan beberapa proses didalamnya yaitu *case folding*, *tokenizing*, *filtering*, *stemming*, *tagging*, *dan* tahap *analyzing* setelah itu dilanjutkan dengan menghitung nilai *hashing* dan menghitung persentase *similarity*.

Dalam bukunya (Feldman & Sanger, 2007) *Text Mining* yaitu salah satu teknik yang digunakan untuk melakukan klasifikasi data teks yang memiliki banyak variasi dengan mencari pola-pola menarik dari data teks tersebut. Sebelum dilakukannya *text mining* ada beberapa proses yang harus dilakukan yaitu *tokenizing*, *filtering*, *stemming*, *tagging*, dan *analyzing* (Pratama & Atmi, 2020).

Sedangkan pada penelitian yang dilakukan oleh (Siahaan et al., 2017) algoritma *rabin karp* digunakan untuk mencari pola dalam suatu teks serta kemiripan antara dua string, setiap kata yang akan dilakukan pencarian kemiripan memiliki panjang tertentu atau disebut *k-gram*, kemudian dari *k-gram* yang dimiliki dilakukan penghitungan nilai *hashing*, kemudian dilanjutkan dengan penghitungan tingkat similarity.

Dalam penelitiannya (Bahrul et al., 2019) menjelaskan bahwa *hashing* merupakan pengubahan nilai yang dimiliki sebuah *string* menjadi nilai yang mewakili nilai asli *string* tersebut, nilai hashing didapatkan melalui fungsi *hash* dengan memanfaatkan kode ASCII pada setiap *string*.

Kemudian penelitian yang dilakukan (Priambodo, 2018) pemeriksaan plagiarisme pada dokumen digital dengan cara memasukkan kata kunci tema

dokumen, dengan cara kerja lebih kurang seperti *search engine*, jika ada maka menampilkan dokumen yang memiliki plagiarisme yang cocok dengan kata kunci yang dicari. Pendeteksian ini menggunakan algoritma *rabin karp* dengan metode *rolling hash*.

Sedangkan penelitian yang dilakukan oleh (Rahmaddeni et al., 2018) yaitu sistem yang dapat mendeteksi plagiat pada abstrak proposal penelitian yang berbasis *website*, dengan memanfaatkan *text mining* sebagai praproses kemudian dilanjutkan dengan *parsing k-gram* dan *hashing*, setelah selesai penghitungan nilai *hashing* kemudian dihitung nilai *similarity*.

Dari penelitian yang telah dilakukan sebelumnya dan melihat urgensi partisipasi masyarakat yang lebih aktif dalam proses perancangan Undang-Undang bersama dengan pemerintah dan parlemen, maka penulis ingin melakukan penelitian dengan mengukur tingkat kemiripan antara Rancangan Undang-Undang dengan Undang-Undang menggunakan algoritma *rabin karp* dengan metode *k-gram* dan *hashing*.

## 1.2 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah bagaimana pengukuran tingkat kemiripan (*similarity*) antara Rancangan Undang-Undang dengan Undang-Undang yang telah ditetapkan sebelumnya dengan menggunakan algoritma *rabin karp* dengan metode *k-gram* dan *hashing.*

## 1.3 Batasan Masalah

Untuk melakukan penelitian ini, maka diperlukan batasan masalah sebagai berikut.

1. Jenis dokumen yang digunakan berjenis *file* pdf (*Portable Document Format*).

2. Pengujian menggunakan algoritma *rabin karp* dengan metode *k-gram* dan *hashing.*
3. Objek penelitian yaitu Undang-Undang Republik Indonesia tahun 2011-2020.

## 1.4 Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut:

1. Menerapkan algoritma *rabin karp* dengan metode *k-gram* dan *hashing*.
2. Menghitung tingkat kemiripan (*similarity*) antara Rancangan Undang-Undang dengan Undang-Undang yang telah ditetapkan**.**

## 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah mengetahui tingkat akurasi kemiripan (*similarity*) antara Rancangan Undang-Undang dengan Undang-Undang yang sudah *existing* menggunakan algoritma *rabin karp* dengan metode *k-gram* dan *hashing.*

.

# BAB II TINJAUAN PUSTAKA

## 2.1 Omnibus Law

Definisi daripada Omnibus Law dimulai dari kata Omnibus. Kata Omnibus berasal dari bahasa Latin, dimana artinya berkaitan dengan atau berurusan dengan berbagai objek dalam berbagai tujuan. Kata *Law* yang maka dapat didefinisikan sebagai hukum untuk semua (Busroh, 2017).

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).

Omnibus Law yang umumnya berkembang di negara bertradisikan *common law system* dan ini termasuk hal yang baru di Indonesia namun ada negara-negara lain telah menerapkan seperti Amerika Serikat, Kanada, Filipina, Australia dan masih banyak negara lain yang telah menerapkan omnibus law.

## 2.2 Undang-Undang

Undang-Undang merupakan hukum yang telah di sahkan oleh badan legislatf, sebelum terbentuknya Undang-Undang maka sebutannya adalah Rancangan Undang-Undang. Undang-Undang adalah ketaatan asas hukum yang diatur secara hirarkis. Norma hukum dari suatu negara manapun selalu berlapis-lapis dan berjenjang-jenjang. Norma yang di bawah berlaku bersumber dan berdasar

pada norma yang lebih tinggi, norma yang lebih tinggi berlaku bersumber dan berdasar pada norma yang lebih tinggi lagi sampai pada suatu norma yang tertinggi yang disebut Norma Dasar (Moonti, 2017).

## 2.3 *Text Mining*

*Text mining* merupakan proses ekstraksi pola (informasi dan pengetahuan yang berguna) dari banyaknya data yang tak terstruktur. Text mining adalah disiplin ilmu yang mencakup pengambilan informasi (*information retrieval*), analisa teks, ekstraksi informasi, kategorisasi, pengelompokan (*clustering*), visualisasi, penambangan data (*data mining*), dan pembelajaran mesin (*machine learning*) (Dahniawati et al., 2019).

Menurut (Dang & Ahmad, 2014) terdapat 5 langkah dasar pada *text mining*, yaitu mengumpulkan informasi data yang tidak terstruktur, kemudian mengkonversi informasi menjadi data terstruktur, lalu mengidentifikasi data yang sudah terstruktur, setelah itu menganalisis pola yang sudah teridentifikasi kemudian mengekstrak informasi dan disimpan ke dalam *database*.

## 2.4 *Preprocessing*

Dalam jurnal (Bahrul et al., 2019) *preprocessing* adalah kumpulan langkah untuk menyetarakan sebuah dokumen berisi teks dalam metode penelitiannya dengan tujuan untuk mengurangi atau menghilangkan kata atau bahkan kalimat yang tidak memiliki makna atau tidak penting, menghilangkan spasi dan karakter selain abjad dan angka.

Dalam jurnal (Aristoteles et at., 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.

## 2.5 *Naturan Language Processing* (NLP)

Pemrosesan bahasa alami adalah rangkaian teknik komputasi secara teoritis untuk menganalisis dan mempresentasikan teks yang terjadi secara alami pada satu atau lebih tingkat analisis linguistik untuk tujuan mencapai pemrosesan bahasa mirip manusia dalam menyelesaikan beberapa *task* atau aplikasi (Liddy, 2001). Beberapa istilah yang sering digunakan dalam NLP seperti token, *sentence*, *tokenization*, *filtering*, *case folding*, *stemming*, *stopword* dan *corpus*.

## 2.6 *Natural Language Tool Kit* (NLTK)

*Toolkit* yang disediakan oleh bahasa pemrograman *python* untuk mempermudah dalam menjalankan tugas-tugas pada *natural language processing* dengan menyediakan berbagai fungsi dan *wrapper*, serta *corpora* standar baik itu mentah atau pun *pre-processed* yang digunakan (Fuadi, 2013). Beberapa *pre-processed* yang digunakan dari *Natural Language Tool Kit* digunakan dalam penelitian ini adalah sebagai berikut.

### 2.6.1 *Case Folding*

*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 (Brata & Hetami, 2015).

### 2.6.2 *Tokenizing*

*Tokenizing* adalah sebuah proses pemotongan kalimat menjadi setiap kata yang menyusunnya. Potongan-potongan kata ini disebut token atau *term*. *Tokenizing* memotong tiap kata dalam kalimat menggunakan spasi sebagai *delimiter* yang akan menghasilkan token (Brata & Hetami, 2015).

### 2.6.3 *Filtering*

Filtering adalah menyaring kata yang didapat dari proses *tokenizing* yang dianggap tidak penting atau tidak memiliki makna dalam proses *text mining* yang disebut *stoplist* (Brata & Hetami, 2015)

### 2.6.4 *Stemming*

Mengembalikan kata-kata yang diperoleh dari hasil *filtering* ke bentuk dasarnya, menghilangkan imbuhan awal (*prefix*) dan imbuhan akhir (*sufix*) sehingga didapat kata dasar. S*temming* memerlukan masukan berupa kata yang terdapat dalam suatu dokumen, dengan menghasilkan keluaran berupa *root word* (Brata & Hetami, 2015).

## 2.7 Algoritma *Rabin Karp*

Algoritma *rabin karp* merupakan algoritma yang digunakan untuk *string matching* dengan memberikan *fingerprint* pada *string* tersebut, mencocokkan antara *string* yang dicari (n) pada sekumpulan *string* yang lebih panjang dari *string* yang dicari (m) dengan memanfaatkan nilai *hashing* untuk setiap *string* yang ada karena tanpa nilai *hash* akan membutuhkan waktu yang lama untuk mendapatkan *string* yang akan dicari  Algoritma *Rabin Karp* ditemukan oleh Michael O. Rabin dan Richard M. Karp. Algoritma ini menggunakan metode *hash* dalam mencari suatu kata. Teori ini jarang digunakan untuk mencari kata tunggal, namun cukup penting dan sangat efektif bila digunakan untuk pencarian jamak (Atmopawiro, 2006).

Algoritma ini dapat digunakan dalam pengembangan *website* ataupun aplikasi *desktop* yang bertujuan untuk mencari persamaan kata atau kalimat yang ada pada sebuah dokumen berisikan teks dengan cara kerjanya lebih kurang seperti *search engine*.

## 2.8 *K-gram*

Metode ini menghasilkan rangkaian substring sejumlah *k-gram*, dimana k adalah parameter yang dipilih oleh user. *K-gram* mengambil *substring* karakter huruf sejumlah k dari sebuah kata yang secara kontinuitas dari teks sumber hingga akhir dokumen. Tahapan ini disebut dengan *tokenizing k-gram* atau pemotongan karakter dengan nilai k yang diasumsikan 3, 4 dan 5 karakter. Nilai *k-gram* bisa kita tetapkan sesuai kebutuhan (Ramadhani, 2015). *K-gram* digunakan untuk pemenggalan karakter sejumlah k dan akan bergeser terus ke kanan sebanyak 1(satu) karakter.

## 2.9 *Hashing*

Dalam penelitian (Putri & Utama, 2017) *Similarity Using* Rabin-Karp *Algorithm hashing* adalah nilai terpenting dalam algoritima *rabin karp*. Nilai hashing berdasarkan jumlah huruf yang dibandingkan diperoleh dengan cara mengalikan nilai ASCII dengan bilangan yang telah ditentukan yang bilangan pokoknya adalah bilangan prima, jika ada dua *string* yang sama maka nilai *hashing* dari *string* tersebut juga harus sama. Basis bilangan prima digunakan dengan tujuan untuk menghindari nilai *hashing* yang sama pada *k-gram* yang berbeda (Mujahidin, 2013). Berikut adalah formula (1) untuk menghitung hashing.

$$H = (\mathrm{c}_{\mathbf{1}} * 10^{(k-1)} + \mathrm{c}_{\mathbf{2}} * 10^{(k-2)} + \cdots + \mathrm{c}_{\mathbf{n}} * 10^{(k-k)}) \text{ mod n} \quad \ldots(1)$$

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 *Similarity*

Inti dari *k-gram* dibagi menjadi dua tahap, yaitu membagi kata menjadi *k-gram* dan mengelompokkan hasil *terms* dari *k-gram* yang sama. Kemudian untuk

menghitung *similarity* dari kumpulan kata tersebut maka digunakan rumus pengukuran nilai similaritas untuk pasangan kata yang didapatkan (Rahmaddeni et al., 2018).

Penghitungan tingkat kemiripan (*similarity*)

$$s = \frac{2*ks}{d1+d2} * 100\% \quad \ldots(2)$$

Keterangan :

s = tingkat kemiripan (*similarity*)

ks = jumlah *hashing* yang sama antara dokumen uji dengan dokumen yang lain

d1 = jumlah *hashing* yang ada pada dokumen pertama

d2 = jumlah *hashing* yang ada pada dokumen kedua

## 2.10 *Confusion Matrix*

*Confusion matrix* digunakan untuk memeriksa tingkat akurasi algoritma dan metode yang digunakan oleh peneliti. *Confusion matrix* merupakan sebuah tabel yang menyatakan jumlah data uji yang benar dan jumlah data uji yang salah yang kemudian diklasifikasikan (Filcha & Hayaty, 2019).

| | | Nilai Sebenarnya | |
|---|---|---|---|
| | | *TRUE* | *FALSE* |
| Nilai Prediksi | *TRUE* | TP<br>(*True Positive*)<br>*Correct Result* | FP<br>(*False Positive*)<br>Unexpected Result |
| | *FALSE* | FN<br>(*False Negative*)<br>*Missing Result* | TN<br>(*True Negative*)<br>*Correct absense of result* |

Tabel 1. *Confusion Matrix* (Data, 2013).

Keterangan Tabel 1.

1. *True Positive* (TP) apabila prediksi atau program pengukuran tingkat kemiripan bernilai tinggi dan menurut *expert* juga bernilai tinggi.
2. *False Negative* (FN) apabila prediksi atau program pengukuran tingkat kemiripan bernilai tinggi sedangkan menurut *expert* bernilai rendah.
3. *True Negative* (TN) apabila prediksi atau program pengukuran tingkat kemiripan bernilai rendah dan menurut *expert* juga bernilai rendah.
4. *False Positive* (FP) apabila prediksi atau program pengukuran tingkat kemiripan bernilai rendah sedangkan menurut *expert* bernilai tinggi.

### 2.10.1 *Precision*

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

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

### 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 METODOLOGI 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 beralamat di Jl. Soemantri Brojonegoro No. 1 Gedong Meneng, Bandar Lampung.

## 3.2 Alat dan Data

Alat dan data yang digunakan dalam penelitian ini untuk menunjang dan mendukung pelaksanaan penelitian antara lain:

### 3.2.1 Alat

Alat yang digunakan dalam pengolahan data untuk menemukan tingkat kemiripan antara Rancangan Undang-Undang dengan Undang-Undang yaitu sebagai berikut:

a. Perangkat Keras (*Hardware*)

   Perangkat keras yang digunakan dalam penelitian ini adalah sebuah laptop dengan spesifikasi sebagai berikut:

   1) *Processor*: Intel®Core™ i7-6700HQ CPU.
   2) *Installed* RAM: DDR4 16GB 2400Mhz.
   3) *Harddisk*: 1 TB 5400 RPM.
   4) *Solid State Drive* 2.5” 240GB.

b. Perangkat Lunak (*Software*)

1. *Lucid chart*

   Perangkat lunak berbasis *website* yang digunakan untuk membuat representasi visual dari urutan task seperti *flowchart*, *organization chart*, *process maps*, *ER* Diagram, *Network* diagram, dan sebagainya. Dan ini juga dapat bekerja dengan berkolaborasi dengan orang lain dalam pembuatan suatu diagram.

2. *Python*

   Python merupakan bahasa pemrograman tingkat tinggi yang bersifat interpretatif dan multiplatform (dapat bekerja diberbagai platform seperti MS Windows, Linux, Macintosh dan lainnya). Python dikembangkan oleh Guido Van Rossum pada tahun 1990 di Amsterdam yang merupakan 16 kelanjutan dari bahasa pemrograman ABC. Python dapat digunakan secara bebas oleh siapapun (Baktikominfo, 2019).

3. *Jupyter Notebook*

   Jupyter membutuhkan Python untuk diinstal karena didasarkan pada bahasa Python. Salah satu alat yang akan mengotomatisasi instalasi Jupyter dari GUI yaitu *software* Anaconda.

4. *Natural Language Tool Kit* (NLTK)

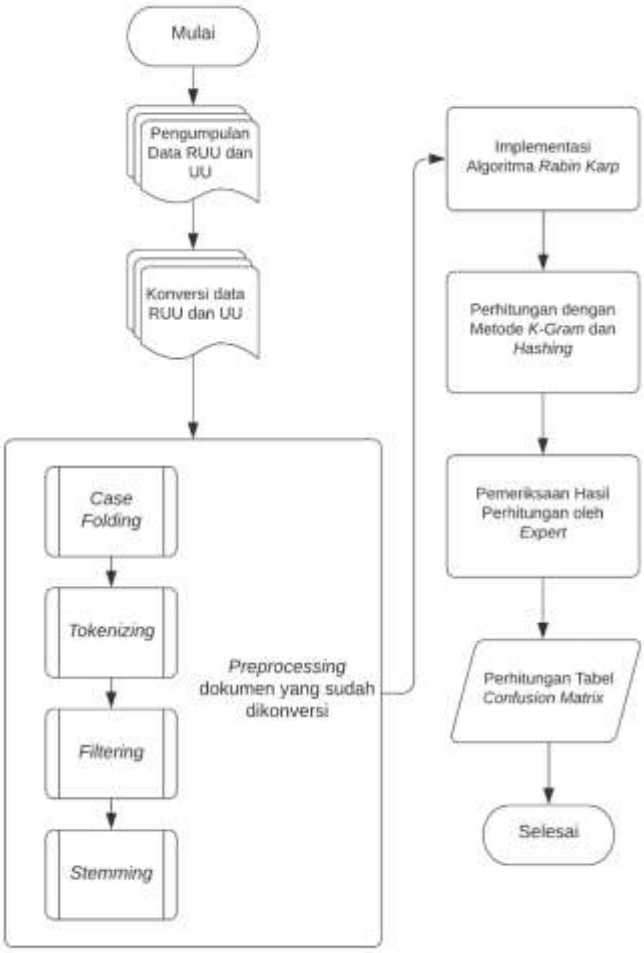
   NLTK merupakan platform terkemuka yang bersifat *open source* dan gratis untuk membangun program Python. NLTK disebut sebagai "perpustakaan yang luar biasa untuk bermain dengan bahasa alami" karena menyediakan antarmuka yang mudah digunakan dan serangkaian pustaka pemrosesan teks untuk klasifikasi, tokenisasi, *stemming*, *tagging*, parsing, penalaran semantik, dan pustaka pemrosesan bahasa alami lainnya. NLTK tersedia untuk Windows, Mac OS X, dan Linux (Syafrial Fachri Pane, 2020) (Fuadi, 2013).

### 3.2.2 Data

Data yang digunakan dalam penelitian ini yaitu berupa Rancangan Undang-Undang dan Undang-Undang yang telah disahkan oleh pemerintah dengan format .pdf data ini bersumber dari Fakultas Hukum Universitas Lampung dan Jaringan Dokumen dan Informasi Hukum Nasional, kemudian dilakukan pemrosesan dengan menggunakan alat serta *software* dengan mengikuti algoritma *rabin karp* dengan metode *k-gram* dan *hashing* untuk mengukur tingkat kemiripannya.

## 3.3 Tahap Penelitian

Penelitian dilakukan dengan beberapa tahapan yaitu studi literatur, pengumpulan data Undang-Undang, pengujian antara Rancangan Undang-Undang dan Undang-Undang yang sudah ada, menggunakan algoritma *rabin* k*arp* dengan metode *k-gram* dan *hashing*, pengujian tingkat kemiripan (*similarity*) antara Rancangan Undang-Undang dan Undang-Undang secara manual, dan pembuatan laporan. Alur penelitian dapat dilihat pada gambar 1.




Gambar 1. Alur penelitian

Penelitian ini dilakukan dengan tahap awal pengumpulan data berupa Undang-Undang yang telah disahkan oleh pemerintah. Kemudian dilanjutkan dengan tahap *preprocessing* untuk setiap Undang-Undang yang bertujuan untuk menghilangkan *noise*, menyeragamkan bentuk kata, dan mengurangi *volume* kata, kemudian dilakukan pemeriksaan tingkat kemiripan menggunakan algoritma *rabin karp* dengan metode *k-gram* dan *hashing*. Tingkat kemiripan (*similarity*) yang didapatkan antara Rancangan Undang-Undang dengan Undang-Undang kemudian diperiksa secara manual untuk melihat tingkat keakuratan dari metode yang digunakan.

### 3.3.1 *Preprocesing* Dokumen

Berikut adalah urutan langkah *preprocessing* yang dilakukan pada penelitian ini

1. ***Case Folding***

   Ini merupakan sebuah langkah digunakan untuk mengubah setiap huruf yang ada pada data menjadi huruf kecil (Filcha & Hayaty, 2019). Berikut ilustrasi *case folding* dapat lihat pada Tabel 1 .

Tabel 2. Ilustrasi *Case Folding*

| Data *Input* | Hasil *Case Folding* |
|---|---|
| Dalam memulai strategi pemasaran, langkah yang harus dilakukan yaitu identifikasi pasar. | dalam memulai strategi pemasaran, langkah yang harus dilakukan yaitu identifikasi pasar. |

2. ***Tokenizing***

   Ini merupakan sebuah langka digunakan untuk pemisahan kata berdasarkan susunan kata. Hasil dari pemisahan kata disebut token (Filcha & Hayaty, 2019). Berikut ilustrasi *tokenizing* dapat lihat pada Tabel 2.

Tabel 3. Ilustrasi *Tokenizing*

| Data *Case Folding* | Hasil Tokenizing |
|---|---|
| dalam memulai strategi pemasaran, langkah yang harus dilakukan yaitu identifikasi pasar. | dalam<br>memulai<br>strategi<br>pemasaran,<br>langkah<br>yang<br>harus<br>dilakukan<br>yaitu<br>identifikasi<br>pasar. |

### 3. *Filtering*

Ini merupakan sebuah langkah digunakan untuk menghilangkan kata yang dianggap tidak penting, karakter atau tanda baca pada data (Puspaningrum et al., 2019). Berikut ilustrasi *filtering* dapat lihat pada Tabel 3.

Tabel 4. Ilustrasi *Filtering*

| Data *Tokenizing* | Hasil *Filtering* |
|---|---|
| dalam | |
| memulai | memulai |
| strategi | strategi |
| pemasaran, | pemasaran |
| langkah | langkah |
| yang | |
| harus | |
| dilakukan | dilakukan |
| yaitu | |
| identifikasi | identifikasi |
| pasar. | pasar |

### 4. *Stemming*

Ini merupakan sebuah langkah digunakan untuk pencarian kata dasar dari setiap kata yang telah di tokenisasi dengan membuang imbuhan diawal dan diakhir kata (Puspaningrum et al., 2019). Berikut ilustrasi *stemming* dapat lihat pada Tabel 4.

Tabel 5. Ilustrasi *Stemming*

| Data *Filtering* | Hasil *Stemming* |
|---|---|
| memulai | mulai |
| strategi | strategi |
| pemasaran, | pasar |
| langkah | langkah |
| dilakukan | lakui |
| identifikasi | identifikasi |
| pasar | pasar |

Dalam algoritma *rabin karp* untuk menentukan tingkat kemiripan data input melalui beberapa tahapan dengan metode *k-gram* dan *hashing*. Setelah melalui *preprocessing* data input dilakukan pemenggalan kata sehingga menjadi potongan-potongan karakter dan kemudian potongan karakter dibandingkan dengan dokumen yang sudah ada sebelumnya. Setiap pemenggalan kata selalu bergeser satu karakter sesudahnya. Berikut ini ilustrasi *k-gram* dalam algoritma *rabin karp* dengan panjang karakter = 4.

Tabel 6. Ilustrasi dokumen uji

| |
|---|
| ‘plag’ ‘lagi’ ‘agia’ ‘iari’ ‘aris’ ‘rism’ ‘isme’ ‘smej’ ‘meji’ ‘ejip’ ‘jipl’ ‘ipla’ ‘plak’ ‘lakm’ ‘akme’ ‘kmel’ ‘mela’ ‘elan’ ‘lang’ ‘angg’ ‘ngga’ ‘ggar’ ‘garh’ ‘arha’ ‘rhak’ ‘hake’ ‘akci’ ‘kcip’ ‘cipt’ ‘ipta’ |

Tabel 7. Ilustrasi dokumen sumber

| |
|---|
| ‘plag’ ‘lagi’ ‘agia’ ‘giar’ ‘iari’ ‘aris’ ‘rism’ ‘isme’ ‘smeg’ ‘megi’ ‘egia’ ‘giat’ ‘iatj’ ‘atji’ ‘tjip’ ‘jipl’ ‘ipla’ ‘plak’ ‘lakk’ ‘akka’ ‘kkar’ ‘kary’ ‘arya’ ‘ryal’ ‘yala’ ‘alan’ ‘lang’ ‘angg’ ‘ngga’ ‘ggar’ ‘garh’ ‘arha’ ‘rhak’ ‘hake’ ‘akei’ ‘kcip’ ‘cipt’ ‘ipta’ |

Tabel 8. Ilustrasi berjalannya hashing dengan *k-gram*=4

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| p | l | a | g | i | a | r | i | s | m | e |
| p | l | a | g | i | a | r | i | s | m | e |
| p | l | a | g | i | a | r | i | s | m | e |
| p | l | a | g | i | a | r | i | s | m | e |
| p | l | a | g | i | a | r | i | s | m | e |
| p | l | a | g | i | a | r | i | s | m | e |
| p | l | a | g | i | a | r | i | s | m | e |
| p | l | a | g | i | a | r | i | s | m | e |

### 3.3.2 Penghitungan Nilai *Hashing*

Penghitungan nilai *hashing* terhadap dokumen.

*Hashing* = $((c_1 * 10^{k\text{-gram-}1}) + \ldots + (c_k * 10^{k\text{-gram-}k\text{-gram}}))$ mod n … (1)

*Hashing* = 'plag'

c = bilangan ascii dari karakter

$k - gram$ = 4

n = 10007

*Hashing* = $((112*10^3) + (108*10^2) + (97*10^1) + (103*10^0))$ mod 10007

*Hashing* = (112000) + (10800) + (970) + (103) mod 10007

*Hashing* = (123873) mod 10007

*Hashing* = 96

Didapatkan nilai *hashing* terhadap penggalan karakter '*plag*' yaitu 96

### 3.3.3 Penghitungan Tingkat *Similarity*

Penghitungan tingkat kemiripan (*similarity*)

$$s = \frac{2*ks}{d1+d2} * 100\% \quad \ldots(2)$$

Keterangan :

s = tingkat kemiripan (*similarity*)

ks = jumlah *hashing* yang sama antara dokumen uji dengan dokumen yang lain

d1 = jumlah *hashing* yang ada pada dokumen pertama

d2 = jumlah *hashing* yang ada pada dokumen kedua

Menurut (Alamsyah, 2017) Beberapa tingkatan kemiripan (*similarity*) yang menjadi ketentuan sebagai berikut

1. Apabila dokumen uji memiliki tingkat *similarity* = 0% maka dokumen uji berbeda dengan dokumen kedua atau dokumen yang telah ada sebelumnya.

2. Apabila dokumen uji memiliki tingkat *similarity* < 15% maka dokumen uji memiliki sedikit kesamaan rendah dengan dokumen kedua atau dokumen yang telah ada sebelumnya.
3. Apabila dokumen uji memiliki tingkat *similarity* 15% s/d 50% maka dokumen uji memiliki tingkat kesamaan sedang dengan dokumen kedua atau dokumen yang telah ada sebelumnya.
4. Apabila dokumen uji memiliki tingkat *similarity* > 50% maka dokumen uji memiliki tingkat kesamaan tinggi dengan dokumen kedua atau dokumen yang telah ada sebelumnya.
5. Apabila dokumen uji memiliki tingkat *similarity* 100% maka dokumen uji memiliki tingkat kesamaan sama dengan dokumen kedua atau dokumen yang telah ada sebelumnya.

### 3.3.4 Pengujian Akurasi Metode

Pengujian akurasi pengukuran tingkat kemiripan antara Rancangan Undang-Undang dengan Undang-Undang menggunakan algoritma *rabin karp* dengan metode *k-gram* dan *hashing* apakah dua dokumen *input* memiliki tingkat kemiripan yang tinggi atau rendah menggunakan tabel *Confusion matrix*. Dapat dilihat pada Tabel 7 merupakan ilustrasi klasifikasi data uji.

Tabel 9. Ilustrasi klasifikasi data

| No | Data *Input* | Undang-Undang | *Rabin Karp* | | *Expert* | | Klasifikasi |
|---|---|---|---|---|---|---|---|
| | | | Tinggi | Rendah | Tinggi | Rendah | |
| 1 | RUU No 1 Tahun 2021.pdf | UU No 1 Tahun 2019.pdf | ✓ | | ✓ | | TP |
| 2 | | UU No 1 Tahun 2020.pdf | ✓ | | ✓ | | TP |
| 3 | | UU No 1 Tahun 2018.pdf | ✓ | | ✓ | | TP |
| 4 | | UU No 1 Tahun 2017.pdf | | ✓ | ✓ | | FN |
| 5 | | UU No 1 Tahun 2016.pdf | | ✓ | | ✓ | TN |

| No | Data *Input* | Undang-Undang | *Rabin Karp* | | *Expert* | | Klasifikasi |
|---|---|---|---|---|---|---|---|
| | | | Tinggi | Rendah | Tinggi | Rendah | |
| 6 | | UU No 1 Tahun 2015.pdf | | ✓ | | ✓ | TN |
| 7 | | UU No 1 Tahun 2014.pdf | | ✓ | | ✓ | TN |
| 8 | | UU No 1 Tahun 2013.pdf | ✓ | | | ✓ | FP |

Tabel 9 merupakan tabel klasifikasi hasil pengujian antara Rancangan Undang-Undang dengan 20 Undang-Undang yang ada, didapatkan 6 hasil uji yang sesuai antara program *rabin karp* dengan ahli hukum (*expert*).

Tabel 10. Ilustrasi *Confusion Matrix*

| | | Nilai Sebenarnya (*expert*) | |
|---|---|---|---|
| | | Tinggi | Rendah |
| Nilai Prediksi (*Rabin Karp*) | Tinggi | 3 | 1 |
| | Rendah | 1 | 3 |

Tabel 10 merupakan hasil klasifikasi dari Tabel 9 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*.

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

$$Precision = \frac{3}{3 + 1}\ x\ 100\%$$

$$Precision = 75\%$$

**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*.

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

$$Recall = \frac{3}{3 + 1}\ x\ 100\%$$

$$Recall = 75\%$$

**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*.

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

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

$$Accuracy = 75\%$$

# BAB V SIMPULAN DAN SARAN

## 5.1 Simpulan

Simpulan yang diperoleh berdasarkan penelitian yang telah dilakukan adalah telah dilakukannya penelitian pengukuran tingkat kemiripan (*similarity*) antara Rancangan Undang-Undang dengan Undang-Undang yang telah ditetapkan sebelumnya dengan menggunakan algoritma *rabin karp* dengan metode *k-gram* dan *hashing*.

## 5.2 Saran

Saran yang diberikan berdasarkan penelitian yang telah dilakukan adalah sebagai berikut.

1. Data input lebih diperbaiki karena didalam *file* undang-undang terdapat gambar yang membuat praproses menjadi lambat dan bahkan menimbulkan *error* yang mengakibatkan *file* tidak dapat dieksekusi.
2. Pada penelitian selanjutnya dapat dikembangkan menggunakan algoritma atau metode lainnya yang berbeda untuk mendapatkan tingkat kemiripan (*similarity*) yang lebih optimal.
3. Pada penelitian selanjutnya dapat dikembangkan juga dikarenakan pada penelitian ini, data tidak akan memiliki kemiripan 100% apabila data tidak sama. Karena telah dicoba sebelumnya dengan kalimat yang memiliki arti yang sama namun dengan susunan kata yang berbeda memiliki tingkat kemiripan yang berbeda.
4. Pada penelitian selanjutnya dapat dikembangkan dengan memberikan tampilan *user interface* yang menarik untuk *file input* dan hasil persentase tingkat kemiripan (*similarity*).

# DAFTAR PUSTAKA


Afif, Z. (2018). *Pembentukan Peraturan Perundang-Undangan Berdasarkan Pancasila Dan Undang-Undang Dasar Negara Kesatuan Republik Indonesia*. *Open Journal System (OJS)*, *7*(September).

Alamsyah, N. (2017). *Deteksi Plagiarisme Tingkat Kemiripan Judul Skripsi Dengan Algoritma Winnowing*. *Technologia: Jurnal Ilmiah*, *8*(4), 205. https://doi.org/10.31602/tji.v8i4.1119

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.

Atmopawiro, A. (2006). *Pengkajian dan Analisis Tiga Algoritma Efisien Rabin-Karp, Knuth-Morris-Pratt, dan Boyer-Moore dalam Pencarian Pola dalam Suatu Teks*. In *STEI ITB*.

Bahrul Khoir, A., Qodim, H., Busro, B., & Rialdy Atmadja, A. (2019). *Implementation of rabin-karp algorithm to determine the similarity of synoptic gospels*. *Journal of Physics: Conference Series*, *1175*(1). https://doi.org/10.1088/1742-6596/1175/1/012120

Baktikominfo. (2019). *Bahasa Pemrograman Python : Pengertian, Sejarah, Kelebihan Dan Kekurangannya*. https://www.baktikominfo.id/en/informasi/pengetahuan/bahasa_pemrograman_python_pengertian_sejarah_kelebihan_dan_kekurangannya-954

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

Busroh, F. F. (2017). *Konseptualisasi Omnibus Law Dalam Menyelesaikan Permasalahan Regulasi Pertanahan*. *Arena Hukum*, *10*(2), 227–250. https://doi.org/10.21776/ub.arenahukum.2017.01002.4

Dahniawati, D., Indriati, & Sutrisno. (2019). *Deteksi Plagiarisme pada Artikel Berita Berbahasa Indonesia menggunakan BM25*. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *3*(5), 4508–4515.

Dang, S., & Ahmad, P. H. (2014). *Text Mining : Techniques and its Application. IJETI International Journal of Engineering & Technology Innovations*, *1*(March), 22–25.

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

Elizabeth D. Liddy. (2001). *Natural language processing*. *Artificial Intelligence*, *19*(2), 131–136. https://doi.org/10.1016/0004-3702(82)90032-7

Fadli, M. (2018). *Pembentukan Undang-Undang Yang Mengikuti Perkembangan Masyarakat*. *Jurnal Legislasi Indonesia*, *15*(1), 49–58.

Feldman, R., & Sanger, J. (2007). *The Text Mining Handbook Advanced Approaches to Analyzing Unstructured Data*. *Basin Research*, *34*(2), 239–240. https://doi.org/10.1046/j.1365-2117.2001.00145.x

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

Fitryantica, A. (2019). *Harmonisasi Peraturan Perundang-Undangan Indonesia melalui Konsep Omnibus Law*. *Gema Keadilan*, *6*(3), 300–316.

Fuadi, K. (2013). *Pengenalan NLP ( Natural Language Processing ) dengan Python Contents*. 1–13.

Moonti. (2017). *Ilmu Perundang-Undangan*. *4*(1), 29.

Mujahidin, Z. (2013). *Implementasi Metode Rabin Karp Untuk Mendeteksi Tingkat Kesamaan Dua Dokumen*. *Jurnal Tugas Akhir*.

Pratama, E. E., & Atmi, R. L. (2020). *A Text Mining Implementation Based on Twitter Data*. *Journal of Computers for Society*, *1*(1), 91–100.

Priambodo, J. (2018). *Pendeteksian Plagiarisme Menggunakan Algoritma Rabin-Karp dengan Metode Rolling Hash*. *Jurnal Informatika Universitas Pamulang*, *3*(1), 39. https://doi.org/10.32493/informatika.v3i1.1518

Puspaningrum, E. Y., Nugroho, B., Setiawan, A., & Hariyanti, N. (2019). *Detection of Text Similarity for Indication Plagiarism Using Winnowing Algorithm Based K-gram and Jaccard Coefficient*. *Journal of Physics: Conference Series*, *1569*(2). https://doi.org/10.1088/1742-6596/1569/2/022044

Putri, R. E., Putera, A., & Siahaan, U. (2017). *Examination of Document Similarity Using Rabin-Karp Algorithm. International Journal of Recent Trends in Engineering and Research*, *3*(8), 196–201. https://doi.org/10.23883/ijrter.2017.3404.4sndk

Rahmaddeni, D. S. dan A. (2018). *SATIN – Sains dan Teknologi Informasi Sistem Pendeteksi Tingkat Kesamaan Teks pada Pengusulan Proposal. SATIN - Sains Dan Teknologi Informasi*, *4*(2), 84–91.

Ramadhani, S. (2015). *Sistem Pencegahan Plagiarism Tugas Akhir Menggunakan Algoritma Rabin-Karp ( Studi Kasus : Sekolah Tinggi Teknik Payakumbuh )*. *Jurnal Teknologi Informasi & Komunikasi Digital Zone*, *6*(1), 44–52.

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

Siahaan, A. P. U., Rahim, R., Mesran, M., & Siregar, D. (2017). *K-Gram As A Determinant Of Plagiarism Level in Rabin-Karp Algorithm. International Journal of Scientific & Technology Research*, *06*(07), 350–353. https://doi.org/10.31219/osf.io/yxjnp

Syafrial Fachri Pane, Y. A. S. (2020). *Big Data: Classification Behavior Menggunakan Python*. Kreatif Industri Nusantara.