IMPLEMENTASI *TEXT MINING* TERHADAP UNDANG-UNDANG BERBASIS *OMNIBUS LAW* MENGGUNAKAN *WORD2VEC* DAN *SOFT COSINE SIMILARITY MEASURE*

(SKRIPSI)

Oleh

Adinda Rizky Febiyanto
1717051042

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022

IMPLEMENTASI *TEXT MINING* TERHADAP UNDANG-UNDANG BERBASIS *OMNIBUS LAW* MENGGUNAKAN *WORD2VEC* DAN *SOFT COSINE SIMILARITY MEASURE*

Oleh

ADINDA RIZKY FEBIYANTO

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

2022

ABSTRAK

IMPLEMENTASI *TEXT MINING* TERHADAP UNDANG-UNDANG BERBASIS *OMNIBUS LAW* MENGGUNAKAN *WORD2VEC* DAN *SOFT COSINE SIMILARITY MEASURE*

Oleh

ADINDA RIZKY FEBIYANTO

Perkembangan sebuah negara tidak bisa terlepas dari perkembangan legislasi yang terjadi didalamnya. Salah satu contoh perkembangan legislasi yang fleksibel adalah penerapan *Omnibus law* yang dimulai oleh Presiden Joko Widodo untuk penyederhanaan legislasi. Namun, penerapan *Omnibus law* dengan pendekatan manual akan memakan banyak waktu dan biaya. Oleh karena itu untuk menyelaraskan penerapan *Omnibus law* di Indonesia yang mewarisi sistem hukum *Civil Law,* diperlukan pendekatan teknologi terbaru agar proses penyelarasan dapat dilakukan dengan efektif dan efisien untuk setiap jenis peraturan perundang-undangan yang ada di Indonesia.

Salah satu pendekatan yang dapat digunakan sebagai solusi penyederhanaan legislasi adalah menggunakan *Text mining* dan *Natural Language Processing (NLP)*. Penelitian sebelumnya dalam bidang hukum terfokus untuk dapat menemukan similaritas antar dokumen yang dapat dimanfaatkan untuk menemukan relevansi antar dokumen Undang-Undang. Beberapa metode yang pernah digunakan dalam penelitian sebelumnya antara lain pendekatan *network-based* dan *embedding-based* serta metode pengukuran similaritas *cosine similarity*.

Berbeda dengan penelitian sebelumnya yang mayoritas menggunakan pengukuran similaritas menggunakan cosine similarity, penelitian ini dilakukan dengan menggunakan kombinasi *embedding-based Word2Vec* dan pengukuran similaritas *soft cosine similarity measure*. Peneltian dilakukan melalui dua skenario pengukuran. Hasil pengukuran skenario pertama mendapatkan nilai *Precision* sebesar 67%, *Recall* sebesar 80%, *Accuracy* sebesar 80% dan *F-β measure* sebesar 73%. Hasil pengukuran similaritas skenario kedua mendapatkan nilai *Precision* sebesar 67%, *Recall* sebesar 67%, *Accuracy* sebesar 73%, dan *F-β measure* sebesar 67%.

Kata Kunci : *Text mining*, *Natural Language Processing*, Pengukuran Similaritas, *Word2Vec*, *Soft Cosine Similarity Measure*

ABSTRACT

IMPLEMENTATION OF TEXT MINING ON OMNIBUS LAW-BASED LAW USING WORD2VEC AND SOFT COSINE SIMILARITY MEASURE

By

ADINDA RIZKY FEBIYANTO

The development of a country cannot be separated from the development of legislation that occurs in it. One example of the development of flexible legislation is the implementation of the Omnibus law initiated by President Joko Widodo to simplify legislation. However, implementing the Omnibus law with a manual approach will take a lot of time and money. Therefore, to harmonize the application of Omnibus law in Indonesia, which inherits the Civil Law legal system, the latest technological approach is needed so that the alignment process can be carried out effectively and efficiently for each type of legislation in Indonesia.

One approach that can be used as a solution to simplifying legislation is to use Text mining and Natural Language Processing (NLP). Previous research in the field of law has focused on finding similarities between documents that can be used to find relevance between legal documents. Several methods that have been used in previous research include network-based and embedding-based approaches and cosine similarity measurement methods.

In contrast to previous studies which mostly used similarity measurements using cosine similarity, this research was conducted using a combination of embedding-based Word2Vec and soft cosine similarity measures. The research was conducted through two measurement scenarios. The results of the first scenario measurement get a Precision value of 67%, Recall of 80%, Accuracy of 80%, and F-β measure of 73%. The results of the second scenario similarity measurement get a Precision value of 67%, Recall of 67%, Accuracy of 73%, and F-β measure of 67%.

Keywords: Text mining, Natural Language Processing, Similarity Measurement, Word2Vec, Soft Cosine Similarity Measure

Judul Skripsi : **IMPLEMENTASI *TEXT MINING* TERHADAP UNDANG-UNDANG BERBASIS *OMNIBUS LAW* MENGGUNAKAN *WORD2VEC* DAN *SOFT COSINE SIMILARITY MEASURE***

Nama Mahasiswa : *Adinda Rizky Febiyanto*

Nomor Pokok Mahasiswa : 1717051042

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

Aristoteles, S.Si., 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, S.Si., M.Si.** .....

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

Penguji Utama : **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 : **04 Februari 2022**

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Adinda Rizky Febiyanto

NPM : 1717051042

Dengan ini menyatakan bahwa skripsi saya dengan judul " IMPLEMENTASI TEXT MINING TERHDAP UNDANG-UNDANG BERBASIS OMNIBUS LAW MENGGUNAKAN SOFT COSINE SIMILARITY MEASURE" adalah benar hasil karya sendiri dan bukan orang lain. Seluruh tulisan yang tertuang didalam skripsi ini telah mengikuti kaidah penulisan karya tulis ilmiah Universitas Lampung. Jika di kemudian hari terbukti skripsi saya adalah hasil penjiplakan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Bandar Lampung, 23 Februari 2022

Penulis

Adinda Rizky Febiyanto

NPM. 1717051042

RIWAYAT HIDUP

Penulis dilahirkan di Bandar Lampung pada tanggal 30 September 1999 sebagai anak pertama dari dua bersaudara dari pasangan Bapak Agus Setiyanto dan Ibu May Saroh. Penulis menyelesaikan pendidikan Sekolah Dasar (SD) di SD 01 Yapindo pada tahun 2005. Kemudian melanjutkan ke jenjang pendidikan menengah pertama di Sekolah Menengah Pertama (SMP) Yapindo pada tahun 2011. Kemudian melanjutkan pendidikan menengah atas di Sekolah Menengah Atas (SMA) di SMAN 01 Gading Rejo yang diselesaikan pada tahun 2017.

Penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung pada tahun 2017 melalui jalur Seleksi Bersama Masuk Perguruan Tinggi (SBMPTN). Selama menjadi mahasiswa, penulis mengikuti beberapa kegiatan antara lain.

1. Anggota Muda Jurusan Ilmu Komputer Himpunan Mahasiswa Jurusan Ilmu Komputer periode 2017/2018.
2. Anggota Bidang Eksternal Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom) pada periode 2017/2018.
3. Organizing Committee Program Global Village by AIESEC Unila tahun 2018.
4. Asisten Dosen Jurusan Ilmu Komputer untuk mata kuliah Logika semester genap tahun ajaran 2018/2019.
5. Asisten Dosen Jurusan Ilmu Komputer untuk mata kuliah Struktur Data semester ganjil tahun ajaran 2019/2020

6. Asisten Dosen Jurusan Ilmu Komputer untuk mata kuliah Sistem Informasi Geografis semester genap tahun ajaran 2019/2020.
7. Asisten Dosen Jurusan Ilmu Komputer untuk mata kuliah Rekayasa Perangkat Lunak pada tahun 2018,2019, dan 2020.
8. Ketua Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom) periode 2019/2020.
9. Praktik Kerja Lapangan (PKL) sebagai magang Sistem Informasi di PT. Kereta Api Indonesia Divisi Regional IV Tanjungkarang.
10. Latihan Kader I Himpunan Mahasiswa Islam (HMI) Cabang Bandar Lampung Komisariat Teknik Universitas Lampung tahun 2019.
11. KKN periode ke-II tahun ajaran 2020/2021 di desa Bratasena, Kabupaten Tulang Bawang.
12. Pimpinan Sidang Konferensi Cabang Himpunan Mahasiswa Islam (HMI) Cabang Bandar Lampung tahun 2020.
13. Latihan Kader II Tingkat Nasional Himpunan Mahasiswa Islam (HMI) Cabang Bukittinggi, Provinsi Sumatera Barat.
14. Wakil Ketua Dewan Perwakilan Mahasiswa (DPM-U KBM Unila) periode 2021/2022.
15. Finalis 20 besar Lomba Essay Nasional Asosiasi Fintech Syariah Indonesia tahun 2021 dengan judul essay “ Tata Kelola Model Text Mining Berbasis Prinsip Syariah dalam Pemberantasan Pinjaman Online Ilegal di Indonesia”.

MOTTO

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya.” – QS Al Baqarah 286

“Cogito Ergo Sum”- Descartes

“Yakinkan dengan Iman, Usahakan dengan Ilmu, Sampaikan dengan Amal”
“ Yakin Usaha Sampai !”

Ku Persembahkan untuk
Kedua Orang Tua Tercinta

SANWACANA

Puji syukur penulis senantiasa terucap atas kehadirat Allah SWT, karena telah memberikan rahmat dan hidayah-Nya kepada saya sehingga saya dapat menyelesaikan skripsi dengan judul **“ Implementasi Text Mining Terhadap Undang-Undang Berbasis Omnibus Law Menggunakan Word2Vec dan Soft Cosine Similarity Measure”**. Shalawat serta salam penulis sanjungkan kepada junjungan besar Nabi Muhammad SAW yang senantiasa kita bersama nantikan syafaat nya di yaumil akhir.

Penyelesaian skripsi ini tidak terlepas dari adanya bimbingan, kritik, bantuan, semangat dan motivasi yang tiada henti selama penelitian, penulisan serta dalam penyelesaian studi. Saya ucapkan terimakasih kepada :

1. Bapak Aristoteles, S.Si.,M.Si. sebagai pembimbing utama yang telah membimbing, memberikan kritik dan saran, sehingga skripsi ini dapat selesai dengan baik.
2. Bapak Rudy, S.H., LL.M., LL.D. sebagai pembimbing kedua yang telah membimbing dan memberikan ilmu yang bermanfaat.
3. Bapak Rico Andrian, S.Si., M.Kom. sebagai pembahas yang telah memberikan masukan yang bermanfaat untuk penyelesaian skripsi ini.
4. Bapak Dr. Suripto Dwi Yuwomo, M.T. selaku Dekan FMIPA Universitas Lampung.
5. Bapak Didik Kurniawan, S.Si., M.T. selaku Ketua Jurusan Ilmu Komputer Universitas Lampung.
6. Ibu Astria Hijriani, S.Kom., M.Kom. selaku Sekretaris Jurusan Ilmu Komputer Universitas Lampung.
7. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc. selaku Pembimbing Akademik saya di Jurusan Ilmu Komputer Universitas Lampung.

8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu selama menempuh perkuliahan di Jurusan Ilmu Komputer.
9. Ibu Ade Nora Maela, Bang Zainuddin, yang telah membantu urusan administrasi.
10. Bapak, Ibu, Adik, serta keluarga yang senantiasa mendoakan, menyemangati, serta mendukung pendidikan dan segala pilihan yang saya pilih.
11. Rekan seperjuangan Similarity Check, Irham Setyaki, Bonema Tri Prasetyo, dan Fikri Aulia.
12. Rekan-Rekan Kumbang Tech, Wildan Mutaqin, Patar Siahaan, dan Fauzi Riski sebagai teman seperjuangan yang mendukung kegiatan akademik dan non-akademik.
13. Rekan MCL yang menemani selama kegiatan perkuliahan.
14. Yolanda Maresta, untuk setiap diskusi, canda tawa, dan motivasi yang telah diberikan.
15. Rekan dan Senior Himpunan Mahasiswa Islam (HMI) Cabang Bandar Lampung Komisariat Teknik Universitas Lampung.
16. Rekan dan Senior Himpunan Mahasiswa Jurusan Ilmu Komputer FMIPA Jurusan Ilmu Komputer.
17. Rekan-rekan Ilmu Komputer Angkatan 2017 yang membersamai selama masa studi di Jurusan Ilmu Komputer Universitas Lampung.
18. Serta semua pihak yang tidak dapat penulis sebutkan satu-persatu, penulis mengucapan terimakasih atas doa, dukungan, dan semangat sehingga penulis dapat menyelesaikan skripsi ini.

Akhir kata, penulis menyadari bahwa masih terdapat banyak kekurangan di dalam penyusunan skripsi ini dan jauh dari kesempurnaan, besar harapan penulis, skripsi ini dapat bermanfaat bagi pembaca.

Bandar Lampung, Februari 2022
Penulis,

Adinda Rizky Febiyanto

DAFTAR ISI

Halaman

DAFTAR ISI ..... **i**

DAFTAR GAMBAR ..... **v**

DAFTAR TABEL ..... **vi**

DAFTAR *PSEUDOCODE* ..... **viii**

BAB 1 PENDAHULUAN **1**

1.1 Latar Belakang 1

1.2 Rumusan Masalah 4

1.3 Batasan Masalah 5

1.4 Tujuan Penelitian 5

1.5 Manfaat Penelitian 5

BAB 2 TINJAUAN PUSTAKA **6**

2.1 Studi Literatur 6

2.1.1 *Similarity Analysis of Legal Judgments* 9

2.1.2 *Soft Similarity dan Soft Cosine Measure : Similarity of Features in Vector Space Model* 9

2.1.3 *Measuring Similarity Among Legal Court Case Documents* 10

2.1.4 *Document Similarity Detection Using Indonesian Language Word2Vec Model* 10

2.1.5 *Document Similarity of Czech Court Decisions* 11

2.1.6 *Consideration About Learning Word2Vec* 11

2.2 Landasan Teori 12

2.2.1 *Text mining* 12
2.2.2 *Preprocessing* 13
2.2.2.1 *Extraction* 13
2.2.2.2 *Case Folding* 13
2.2.2.3 *Filtering* 14
2.2.2.4 *Stopwords Elimination* 14
2.2.3 *Exploratory Data Analysis (*EDA*)* 14
2.2.4 *Term Frequency Inverse Document Frequency* (*TF-IDF*) dan *Pivoted Normalization* 15
2.2.5 *Word2Vec* 17
2.2.5.1 Model *CBOW* 19
2.2.5.2 Model *Skip-gram* 24
2.2.6 *Soft Cosine Similarity Measure* 33
2.2.7 *Confusion Matrix* 35
BAB 3 METODOLOGI PENELITIAN 36
3.1 Objek Penelitian 36
3.2 Waktu dan Tempat Penelitian 37
3.3 Data dan Perangkat Penelitian 37
3.3.1 Data 37
3.3.2 Perangkat Penelitian 37
3.4 Desain Penelitian 38
3.4.1 Pengumpulan dan Pengelompokkan Data 40
3.4.1.1 Pengumpulan dan Pengelompokkan Data Latih 40
3.4.1.2 Pengumpulan dan Pengelompokkan Data Uji 41
3.4.2 *Preprocessing* Data 43
3.4.2.1 *Extraction* 43

3.4.2.2 *Case Folding* 43

3.4.2.3 Filtering 44

3.4.2.4 Stopwords Elimination 44

3.4.3 Exploratory Data Analysis (EDA) 45

3.4.4 Pembentukan *Corpus* Data Latih,Data Uji Kueri & Data Uji Pembanding 46

3.4.5 Pembentukan Matriks *TF-IDF* Untuk Data Latih, Data Uji Kueri & Data Uji Pembanding 46

3.4.6 Pembentukan Model *Word2Vec* 48

3.4.7 Pembentukan Matriks *Similarity* 49

3.4.8 Pengukuran Similaritas 50

3.4.9 Validasi Hasil 51

3.4.10 Penarikan Kesimpulan 52

BAB 4 HASIL DAN PEMBAHASAN 55

4.1 *Preprocessing* 55

4.1.1 Hasil *Extraction* 55

4.1.2 Hasil *Case folding* 58

4.1.3 Hasil *Filtering* 58

4.1.4 Hasil *Stopwords Elimination* 59

4.2 Hasil *Exploratory Data Analysis* (EDA) 61

4.3 Hasil Pembentukan *Corpus* Data Latih, Data Uji Kueri & Data Uji Pembanding 67

4.4 Hasil Pembentukan Matriks *TF-IDF* Untuk Data Latih Dan Data Uji 71

4.5 Hasil Pembentukan Model *Word2Vec* 72

4.6 Hasil Pembentukan Matriks *Similarity* 75

4.7 Hasil Pengukuran Similaritas 76

4.8 Pembahasan 78

4.8.1 Pembahasan Skenario Pengukuran Pertama 79

4.8.2 Pembahasan Skenario Pengukuran Kedua 82

BAB 5 SIMPULAN DAN SARAN 87

5.1 Simpulan 87

5.2 Saran 88

DAFTAR PUSTAKA 89

LAMPIRAN 92

DAFTAR GAMBAR

Gambar Halaman

Gambar 2.1 Ilustrasi Arsitektur *Word2Vec (CBOW & Skip-gram)* 18
Gambar 2.2 Ilustrasi Model CBOW..... 23
Gambar 2.3 Ilustrasi Model *Skip-gram* 31
Gambar 2.4 Ilustrasi Model *Skip-gram Negative Sampling*..... 32
Gambar 3.1 Diagram Alir Penelitian 39
Gambar 4.1 *Document-term Matrix* Data Latih..... 62
Gambar 4.2 Visualisasi *Word*cloud Data Latih Untuk Dokumen 2020-11 65
Gambar 4.3 Hasil Pembentukan Matrik *TF-IDF* Untuk Data Latih 71
Gambar 4.4 Hasil Pembentukan Matriks *TF-IDF* Untuk Data Uji 71
Gambar 4.5 Visualisasi Kata Relevan dan Tidak Relevan dengan Kata Input "Investasi" 74

DAFTAR TABEL

Tabel Halaman

Tabel 1.1 Permasalahan Regulasi (Sadiawati, 2015) 3
Tabel 2.1 Daftar Penelitian Terkait 7
Tabel 3.1 Jumlah Data Latih 40
Tabel 3.2 Data Uji Kueri 42
Tabel 3.3 Data Uji Indeks Pembanding 42
Tabel 3.4 Ilustrasi Tahap *Case folding* 43
Tabel 3.5 Ilustrasi *Filtering* 44
Tabel 3.6 Ilustrasi *Stopwords Elimination* 45
Tabel 3.7 Konfigurasi Pembentukan Matriks *TF-IDF* 47
Tabel 3.8 Konfigurasi Awal Pembentukan Model *Word2Vec* 48
Tabel 3.9 Ilustrasi Matriks *Similarity* 50
Tabel 3.10 Ilustrasi *Confusion Matrix* Validasi Hasil 51
Tabel 4.1 Hasil *Case folding* 58
Tabel 4.2 Hasil *Filtering* Angka, Pungtuasi, dan *ASCII* non-cetak 59
Tabel 4.3 Hasil *Filtering* Karakter Fungsional 59
Tabel 4.4 Hasil *Stopwords Elimination* 60
Tabel 4.5 Daftar Stopwords Tambahan 63
Tabel 4.6 Daftar Dokumen Kosong 66
Tabel 4.7 Jumlah Data Latih Setelah Tahap EDA 66
Tabel 4.8 Hasil Pembentukan *Corpus* Data Latih Untuk Dokumen 1946-19 69
Tabel 4.9 Hasil Pembentukan *Corpus* Data Uji Untuk Dokumen 210615_RUU_PelayananPublik 69
Tabel 4.10 Daftar Kata Yang Tidak Dapat Tervektorisasi 70
Tabel 4.11 Konfigurasi Akhir Pembentukan Model *Word2Vec* 72
Tabel 4.12 Daftar Sepuluh Kata Relevan dan Non-Relevan dengan Kata Investasi 74

Tabel 4.13 Hasil Pengukuran Similaritas Skenario Pertama 77
Tabel 4.14 Hasil Pengukuran Similaritas Skenario Kedua 78
Tabel 4.15 Validasi Pengukuran Similaritas Skenario Pertama..... 80
Tabel 4.16 *Confusion Matrix* Skenario Pengukuran Pertama 81
Tabel 4.17 Evaluasi Skenario Pertama Pengukuran Similaritas 81
Tabel 4.18 Validasi Pengukuran Similaritas Skenario Kedua 83
Tabel 4.19 *Confusion Matrix* Skenario Pengukuran Kedua..... 84
Tabel 4.20 Evaluasi Skenario Kedua Pengukuran Similaritas..... 85

DAFTAR *PSEUDOCODE*

Pseudocode Halaman

Pseudocode 4.1 Kode Program Ekstraksi Data Latih 56
Pseudocode 4.2 Kode Program Ekstraksi Data Uji 56
Pseudocode 4.3 Kode Program *Case folding* dan *Filtering*..... 57
Pseudocode 4.4 Kode Program *Filtering*..... 57
Pseudocode 4.5 Kode Program *Stopwords Elimination* 60
Pseudocode 4.6 Kode Program Pembentukan *Document-Term Matrix* Data Latih 61
Pseudocode 4.7 Kode Program Pembentukan Daftar *Stopwords* Tambahan 63
Pseudocode 4.8 Kode Program Memperbarui Data Latih Dengan *Stopwords* Tambahan 64
Pseudocode 4.9 Kode Program Memperbarui Data Uji dengan *Stopwords* Tambahan 64
Pseudocode 4.10 Kode Program Visualisasi *Word*cloud Data Latih..... 65
Pseudocode 4.11 Kode Program Pembentukan *Corpus* Data Latih 68
Pseudocode 4.12 Kode Program Pembentukan *Corpus* Data Uji..... 68
Pseudocode 4.13 Kode Program Pembentukan Model *Word2Vec*..... 73
Pseudocode 4.14 Kode Program Pembentukan Matriks *Similarity*..... 75
Pseudocode 4.15 Kode Program Pengukuran Similaritas..... 76

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perkembangan sebuah negara tidak lepas dari perkembangan sistem hukum yang ada didalamnya. Perkembangan sistem hukum tersebut tentu tidak lepas dari banyak permasalahan yang dihadapinya. *Omnibus law* menjadi isu yang populer bagi masyarakat Indonesia belakangan ini. Penerapan *omnibus law* ditujukan untuk melakukan penyederhanaan legislasi yang dilakukan oleh Presiden Joko Widodo untuk menyederhanakan investasi dan birokrasi. Penerapan *omnibus law* di Indonesia menghadapi banyak permasalahan seperti jumlah peraturan yang terlalu banyak (over regulasi) dan tidak harmonisnya regulasi yang ada. Tentu hal ini akan membutuhkan waktu dan biaya yang besar apabila dilakukan melalui pendekatan secara tradisional (Rudy dkk., 2021).

Secara perspektif hukum *Omnibus law* memiliki kesamaan konsep dengan *Omnibus Bill*. Black (Black & Frey, 1934) mendefinisikan *Omnibus Bill* sebagai berikut : *"A legislative bill,including in one act various separate and distinct matters, and particularly one joining a number of different subjects in one measure in such a way as to compel the executive authority to accept provisions which he does not approve or else defeat the whole enactment."*(Sebuah RUU dalam satu bentuk yang mengatur bermacam-macam hal yang terpisah dan berbeda, dan seringkali menggabungkan sejumlah subjek yang berbeda dalam satu cara , sedemikian rupa sehingga dapat memaksa eksekutif untuk menerima ketentuan yang tidak disetujui atau juga membatalkan seluruh pengundangan).

Berdasarkan definisi tersebut dapat diterjemahkan menjadi “Sebuah Undang-Undang 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”.

Pengunaan sistem *Omnibus law* yang pernah diterapkan di Indonesia tentu akan berbeda dengan saat ini (Hantoro et al., 2020) yang akan diterapkan pada Undang-Undang yang memiki spektrum hukum yang lebih luas. Indonesia dalam regional Asia Tenggara tidak sendiri dalam pengalaman menggunakan konsep *Omnibus law*. Filipina pernah menggunakan *Omnibus Code of 1987 and Foreign Investment Act Of 1991*.

Penyederhaan legislasi tahap awal dilakukan oleh Presiden Joko Widodo melalui penerbitan dua Undang-Undang yang mengadopsi pendekatan *omnibus law*. Urgensi penyederhaan legislasi ini didukung dengan data yang disadur dari (Setiadi, 2020) yang menyatakan bahwa Undang-Undang yang saat ini berada di Indonesia berjumlah sebanyak 1687 Undang-Undang. Melihat banyak nya Undang-Undang yang ada tersebut sudah tentu kompleksitas dalam pembentukan regulasi atau legislasi di masa mendatang akan semakin sulit. Hal ini disebut dengan *overregulated* (Setiadi, 2020) yang menyebabkan kualitas dan disharmoni legislasi memburuk.

Selain menyebabkan buruknya kualitas dan peluang disharmoni legislasi yang semakin besar, *overregulated* akan menyebabkan pertumbuhan ekonomi Indonesia bergerak tidak cepat sehingga kontraproduktif (Malau, 2014) dengan langkah yang sudah diambil oleh pemerintah Indonesia sebagai salah satu aktor aktif dalam pengembangan hukum persaingan di ASEAN melalui Masyarakat Ekonomi Asean (MEA). Hal lain yang dapat dijadikan contoh bahwa *overregulated* menyebabkan perekonomian Indonesia bergerak dengan lambat adalah Indonesia berada dalam posisi ke-6 negara ramah investasi dalam regional ASEAN (Hartono & Hardiwinoto, 2018)

Penyederhanaan legislasi yang dicitakan tentu harus bersamaan dengan harmonisasi legislasi agar tidak terjadi *overregulated* yang lebih parah atau bahkan kekosongan hukum. Untuk melakukan harmonisasi, secara umum permasalahan yang ditemukan terbagi menjadi empat hal (Sadiawati, 2015). Permasalahan tersebut akan disajikan dalam Tabel 1.1.

Tabel 1.1 Permasalahan Regulasi (Sadiawati, 2015)

Jenis Permasalahan	**Pengertian**
Konflik Regulasi	Konflik regulasi merupakan sebuah kondisi apabila terdapat pasal atau ketentuan yang nyata-nyata bertentangan dengan peraturan lainnya.
Inkonsistensi	Inkonsisten yang berarti ketika terdapat peraturan atau pengaturan yang tidak konsisten dalam satu peraturan perundang-undangan beserta turunannya.
Multitafsir Regulasi	Dikelompokkan menjadi multitafsir apabila ditemukan ketidakjelasan pada objek dan subjek yang diatur sehingga menimbulkan ketidakjelasan rumusan bahasa dan sistematika yang tidak jelas.
Tidak Operasional	Dinyatakan tidak operasional apabila sebuah regulasi tidak memiliki daya guna, namun masih tetap berlaku.

Tabel 1.1 menjelaskan tentang permasalahan regulasi yang terjadi di Indonesia.

Berdasarkan permasalahan-permasalahan yang telah disebutkan. Dapat diambil kesimpulan bahwa dibutuhkan pengembangan teknologi dalam proses penyederhanaan dan harmonisasi legislasi di Indonesia.

Salah satu pengembangan implementasi teknologi dalam proses penyederhanaan dan harmonisasi legislasi adalah *text mining* (Zhong et al., 2020). *Text mining* dapat diterjemahkan menjadi sistem yang mengelola data teks untuk mendeteksi pola kebahasaan untuk mendapatkan informasi (Sebastiani, 2002). Penerapan *text mining* dapat dimanfaatkan untuk *text summarization*, *document retrieval*, dan *assesing document similarity* (Witten, 2004).

Bentuk penerapan *text mining* untuk dokumen berbahasa Indonesia salah satu nya dapat dilihat dalam penelitian dengan judul "*Text Feature Weighting For Summarization Of Document Bahasa Indonesia Using Genetic Algorithm"* (Aristoteles et al., 2012). Dalam penelitian tersebut dijabarkan penggunaan *text mining* untuk merangkum sebuah dokumen berdasarkan pembobotan menggunakan *Genetic Algorithm.* Namun, perangkuman dokumen tidak cocok untuk diterapkan untuk membantu harmonisasi Undang-Undang, dikarenakan proses perangkuman ditakutkan akan menghilangkan bagian penting dalam Undang-Undang tersebut.

Untuk itu diperlukan bentuk lain dari penerapan *text mining* yang dapat digunakan untuk mencari hubungan semantik antar dokumen tanpa menghilangkan esensi dari sebuah dokumen tersebut. Hubungan semantik tersebut akan digunakan untuk mencari tingkat similaritas antar dokumen Undang-Undang. Dalam melakukan penilaian similaritas terlebih dahulu yang akan dilakukan adalah proses pembobotan kata dengan menggunakan *Word2Vec* yang dikembangkan oleh (Mikolov, Chen, et al., 2013). Proses pembobotan ini dilakukan untuk merubah data Undang-Undang menjadi kumpulan vektor. Setelah itu, dilakukan penghitungan menggunakan metode *Soft Cosine Similarity Measure* (Sidorov et al., 2014) sehingga akan didapatkan tingkat similaritas diantara Undang-Undang. Tingkat similaritas ini akan memberikan informasi bagi penyusun Undang-Undang untuk menentukan status sebuah Undang-Undang, dan menjadi bahan pertimbangan dalam pembentukan Undang-Undang baru.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, didapatkan permasalahan pokok yang perlu di teliti dan dijadikan dasar dalam pengerjaan penelitian ini yakni, Bagaimana mengimplementasikan *Text mining* untuk mencari tingkat similaritas antar Undang-Undang menggunakan kombinasi metode pembobotan kata *Word2Vec* dan metode pengukuran similaritas *Soft Cosine Similarity Measure*?

1.3 Batasan Masalah

Untuk melakukan penelitian ini secara fokus, maka diperlukan batasan masalah terhadap implementasi *text mining* terhadap data Undang-Undang dan Rancangan Undang-Undang sebagai berikut.

1. Pembobotan kata dilakukan dengan menggunakan *Word2Vec*.
2. Pengukuran similaritas antar dokumen Undang-Undang atau Rancangan Undang-Undang dilakukan menggunakan pengukuran *soft cosine similarity measure*.
3. Data yang digunakan terdiri atas Undang-Undang (UU) dan Rancangan Undang-Undang (RUU). Data Undang-Undang yang digunakan adalah Undang-Undang dalam rentang tahun 1945-2020. Data Rancangan Undang-Undang yang digunakan adalah Rancangan Undang-Undang Tentang Pelayanan Publik dan Pertambangan yang diberikan oleh mitra penelitian yang berasal dari Fakultas Hukum Universitas Lampung.
4. Jenis data yang digunakan adalah dokumen teks di dalam format *Portable Document File (.pdf)*.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengimplementasikan dan menguji akurasi metode *Word2Vec* dan *soft cosine similarity measure* untuk pengukuran similaritas terhadap dokumen Undang-Undang atau Rancangan Undang-Undang.

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah :

1. Terbentuknya *corpus* Undang-Undang yang berisikan Undang-Undang dalam rentang tahun 1945-2020.
2. Terbentuknya model *Word2Vec* berdasarkan data Undang-Undang dalam rentang tahun 1945-2020.
3. Hasil pengukuran similaritas menggunakan *Word2Vec* dan soft cosine *similarity* menjadi sumber informasi tingkat relevansi antar dokumen Undang-Undang atau Rancangan Undang-Undang.

BAB 2 TINJAUAN PUSTAKA

2.1 Studi Literatur

Sistem hukum di Indonesia yang sudah lama mewarisi sistem hukum *civil law* tentu akan menjadi tantangan tersendiri di dalam pembentukaan Undang-Undang yang menerapkan konsep *omnibus law* (Hantoro et al., 2020). Disarikan dari *Parliamentary Review* Tentang *Omnibus law* (Hantoro et al., 2020), pada dasarnya Indonesia sudah lama tidak menganut sistem undang-undang payung dan semua undang-undang memiliki kedudukan yang setingkat. Permasalahan yang akan menjadi tantangan penerapan Undang-Undang yang mengadopsi *Omnibus law* adalah terkait dengan penerapan teknik dan asas peraturan perundang-undangan serta harmonisasi antar peraturan perundang-undangan.

Berdasarkan permasalahan yang sudah dirangkum di dalam latar belakang, terdapat permasalahan utama yakni bagaimana mengimplementasikan *text mining* yang bertujuan untuk mencari tingkat similaritas antar Undang-Undang sehingga harmonisasi Undang-Undang dapat dilakukan dengan baik. Pada studi literatur akan dijabarkan penelitian terkait yang berguna sebagai landasan di dalam melakukan penelitian. Tabel 2.1 akan menyajikan daftar penelitian yang terkait dengan implementasi *text mining* untuk mencari similaritas antar dokumen Undang-Undang atau Rancangan Undang-Undang.

Tabel 2.1 Daftar Penelitian Terkait

Penelitian	Data	Metode	Hasil
Similarity Analysis of Legal Judgements (Kumar et al., 2011)	**Data Latih :** 2430 data *Judgement from Supreme Court India* **Data Uji :** 20 *pair of judgements from Supreme Court India*.	**Metode pembobotan kata :** *TF-IDF* **Metode pengukuran similaritas :** *All-term cosine similarity*, *Legal-term cosine similarity*, *Co-citation similarity*, dan *Bibliographic coupling similarity*	Metode pengukuran *all-term cosine similarity* dan *co-citation similarity* tidak efisien, *sedangkan* metode pengukuran *legal-term cosine similarity* dan *bibliographic coupling similarity* efektif untuk menemukan *similar judgements*.
Soft Similarity and Soft Cosine Measure : Similarity of Features in Vector Space Model (Sidorov et al., 2014)	*Entrance Exams Task* (Peñas et al., 2013) yang terdiri dari : 12 dokumen uji, 60 pertanyaan (5 pertanyaan untuk setiap dokumen 240 pilihan jawaban (4 untuk setiap pertanyaan)	**Metode transformasi data :** *Integrated Syntactic Graph* **Metode pengukuran similaritas :** *Cosine similarity* dan *Soft cosine measure*	Pengukuran *soft cosine measure* mungkin untuk di kombinasikan dengan algoritma *word embedding* lainnya untuk mendapatkan variasi hasil yang mungkin lebih baik. Metode pengukuran *soft cosine measure* memberikan hasil yang lebih baik dibandingkan *cosine similarity* dan hasil terbaik yang didapatkan dalam (Peñas et al., 2013) yakni sebesar 0.45
Measuring Similarity Among Legal Court Case Documents (Mandal et al., 2017)	**Data Latih :** *33545 court case documents of India Supreme Court*(1950-2016) **Data Uji :** *50 pair of case documents based on* (Kumar et al., 2013)	**Metode *Word embedding* :** *TF-IDF Vectors*, *Latent Dirichlet Allocation (LDA)*, *Word2Vec* ***Metode Document embedding* :** *Doc2Vec* **Metode Pengukuran similaritas :** *Cosine Similarity* **Representasi dokumen :** *Whole Document*, *Summ-document*, *Paragraph-Avg*,dan *RFC-Avg of document*	Didapatkan korelasi representasi dokumen + metode dengan penilaian ahli. *Whole document* + *Doc2Vec* = 0.69 *Summary* + *LDA* = 0.61 *Paragpraph Avg* + *Word2Vec* = 0.59 *RFC-Avg* + *LDA* = 0.51. Metode *Doc2Vec* menunjukkan hasil paling baik dibandingkan dengan metode lainnya dengan korelasi positif sebesar 0..69.

Tabel 2.1 (lanjutan)

Penelitian	Data	Metode	Hasil
Document Similarity Detection Using Indonesian Language Word2Vec Model (Ramadhanti & Mariyah, 2019)	**Data Latih :** 353238 koleksi artikel dari *wikipedia* bahasa Indonesia **Data Uji :** 10 pasang jurnal artikel saintifik	**Metode *word embedding*:** *Word2Vec*,dan *TF-IDF* **Metode pengukuran similaritas :** *Cosine Similarity* **Representasi dokumen :** Keseluruhan dokumen dan Representasi paragraf dokumen	Hasil pengukuran similaritas menggunakan *Word2Vec + cosine similarity* lebih baik apabila dibandingkan dengan *TF-IDF + cosine similarity*, untuk seluruh pasangan artikel saintifik yang diujikan.
Document Similarity of Czech supreme court decissions (Novotná, 2020)	**Data Latih :** *Czech Court Decission Corpus* 1.0 (111977 dokumen) **Data Uji :** *Supreme Court decision, 26 Cdo 1471/2013*	**Metode *document embedding* :** *Doc2Vec* **Metode pengukuran similaritas :** *Cosine similarity*	Didapatkan 10 dokumen paling relevan dengan *data testing*, dimana delapan diantara nya diputuskan oleh hakim yang sama. Performa *Doc2Vec* kurang baik apabila mendapatkan input berupa kata teks yang terlalu panjang.
Considerations about Learning Word2Vec (Di Gennaro et al., 2021)	*Google set* (Mikolov, Chen, et al., 2013) 19544 data analogi, dua grup utama perbandingan (*semantic* dengan 5 kategori dan 8869 data analogi), dan (*syntactic* dengan 9 kategori dan 19675 data analogi)	**Metode *word embedding :*** *Word2Vec*. Melakukan eksperimen *hyper-parameter* dalam *Word2Vec* yakni *learning-rate, negative-sampling, vector-size, window size* dan hubungannya dengan jumlah *epochs* dalam sebuah proses latih.	Model *Word2Vec* bekerja dengan cara meminimalkan hasil dari *loss function* dalam setiap *epochs*. Performa model *Word2Vec* ditentukan dengan pemilihan *hyper-parameter* yang tepat dan besaran vektor yang tepat.

Tabel 2.1 menyajikan daftar penelitian yang berkaitan dengan pengukuran similaritas antar dokumen Undang-Undang dan Rancangan Undang-Undang.

Penelitian terkait ini akan menjadi acuan dan perbandingan dengan penelitian ini. Beberapa penelitian yang digunakan sebagai acuan dalam penelitian ini akan dijabarkan lebih lanjut.

2.1.1 *Similarity Analysis of Legal Judgments*

Penelitian ini dilakukan oleh (Kumar et al., 2011). Pada penelitian ini menggunakan data latih berupa 2430 dokumen *judgment* yang berasal dari *Supreme Court India*, dan data uji berupa 20 pasangan *judgment* yang berasal dari *Supreme Court India*. Penelitian ini dilakukan dengan menggunakan metode pembobotan kata *term frequency-inverse document frequency* (TF-IDF) dan pengukuran similaritas menggunakan *all-term cosine similarity*, *legal-term cosine similarity*, *co-citation similarity*, dan *bibliographic coupling similarity*.

Hasil yang didapatkan dalam penelitian ini adalah metode pengukuran *all-term cosine similarity* dan *co-citation similarity* tidak efisien untuk melakukan pengukuran similaritas antar dokumen *judgment*. Sedangkan metode pengukuran *legal-term cosine similarity* dan *bibliographic coupling similarity* efisien untuk digunakan dalam pengukuran similaritas antar dokumen *judgment*.

2.1.2 *Soft Similarity dan Soft Cosine Measure : Similarity of Features in Vector Space Model*

Penelitian ini dilakukan oleh (Sidorov et al., 2014). Pada penelitian ini menggunakan data berupa *entrance exam task* yang di usulkan oleh (Peñas et al., 2013) yang terdiri atas 12 dokumen uji dengan 60 pertanyaan (5 pertanyaan pada setiap dokumen uji), dimana setiap pertanyaan memiliki 4 pilihan jawaban.

Penelitian ini dilakukan dengan cara menggunakan *integrated syntactic graph* untuk melakukan transformasi data, dan menggunakan metode pengukuran similaritas *cosine similarity* dan *soft cosine similarity*.

Hasil yang didapatkan dalam penelitian ini adalah *soft cosine measure* sangat mungkin untuk dikombinasikan dengan metode pembobotan kata lainnya yang berbasis *vector space model*. *Soft cosine measure* berhasil meraih performa tertinggi dengan hasil sebesar 0.45.

2.1.3 *Measuring Similarity Among Legal Court Case Documents*

Penelitian ini dilakukan oleh (Mandal et al., 2017). Pada penelitian ini menggunakan data latih berupa 33545 dokumen *court case* yang berasal dari *Indian Supreme Court* dalam rentang tahun 1950-2016, dan data latih berupa 50 pasang dokumen *court case* yang di sadur dari (Kumar et al., 2013). Penelitian ini menggunakan metode pembobotan kata berupa *term frequency-inverse document frequency (TF-IDF), latent-dirichlet allocation (LDA), Word2Vec*, dan pembobotan dokumen menggunakan *Doc2Vec*. Untuk melakukan pengukuran similaritas, penelitian ini menggunakan pengukuran *cosine similarity*. Representasi dokumen yang digunakan dala penelitian ini berupa *whole document* (keseluruhan dokumen), *summary of document* (rangkuman dokumen), *paragraph average* (rerata bobot dalam setiap paragraf) , dan *Reason for citation (RFC) Average of document*.

Penelitian ini mendapatkan hasil yakni, *Whole document* + *Doc2Vec* = 0.69, *Summary* + *LDA* = 0.61, *Paragpraph Avg* + *Word2Vec* = 0.59, *RFC-Avg* + *LDA* = 0.51. Dimana kombinasi representasi keseluruhan dokumen menggunakan *Doc2Vec* mendapatkan hasil korelasi positif dengan penilaian ahli sebesar 0.69.

2.1.4 *Document Similarity Detection Using Indonesian Language Word2Vec Model*

Penelitian ini dilakukan oleh (Ramadhanti & Mariyah, 2019). Pada penelitian ini digunakan data latih berupa 353238 artikel dalam wikipedia berbahasa Indonesia, dan data uji berupa sepuluh pasang artikel saintifik. Penelitian ini menggunakan metode pembobotan kata *Word2Vec* dan *term frequency-inverse document frequency (TF-IDF)*, serta pengukuran similaritas menggunakan metode pengukuran *cosine similarity*. Penelitian ini dilakukan dengan cara membandingkan hasil pengukuran similaritas untuk representasi keseluruhan dokumen dan rerata bobot paragraf dalam dokumen. Hasil yang didapatkan adalah penggunaan *Word2Vec* jauh baik dibandingkan dengan penggunaan *TF-IDF* untuk memberikan pembobotan setiap kata.

2.1.5 *Document Similarity of Czech Court Decisions*

Penelitian ini dilakukan oleh (Novotná, 2020). Pada penelitian ini data terbagi menjadi data latih dan data uji. Data latih yang digunakan berupa 111977 dokumen yang berasal dari *Czech Court Decision Corpus* 1.0. Data uji yang digunakan berupa 26 pasang dokumen *Czech Court Decision*. Dalam melakukan pembobotan dokumen, penelitian ini menggunakan metode *Doc2Vec*, dan pengukuran similaritas antar dokumen menggunakan metode *cosine similarity*.

Hasil yang didapatkan dalam penelitian ini adalah didapatkan 10 pasang dokumen dengan nilai similaritas tertinggi, dimana delapan diantaranya dibuat oleh hakim yang sama, selain itu disimpulkan bahwa performa *Doc2Vec* tidak terlalu baik apabila menggunakan dokumen input yang terlalu panjang.

2.1.6 *Consideration About Learning Word2Vec*

Penelitian ini dilakukan oleh (Di Gennaro et al., 2021). Pada penelitian ini data yang digunakan adalah *Google set* oleh (Mikolov, Chen, et al., 2013). Dimana, data yang digunakan terdiri atas 19544 data analogi uji semantik dan uji sintaksis. Metode pembobotan kata yang digunakan dalam penelitian ini adalah *Word2Vec*. Penelitian ini dilakukan untuk menemukan hubungan antara proses *learning* yang dilakukan *Word2Vec* dalam berbagai macam nilai *hyper-parameter* dengan *epochs* atau banyak nya perulangan yang dilakukan dalam pembentukan *Word2Vec*.

Jenis *hyper-parameter* yang di teliti dalam penelitian ini antara lain, *negative sampling*, *vector size embedding*, dan *window size*. Dalam setiap pengujian untuk jenis *hyper-parameter* yang telah disebutkan, disimpulkan bahwa *Word2Vec* bekerja dengan cara meminimalisir nilai *loss function* pada setiap *epochs* yang dilakukan. Tercapainya nilai *loss function* yang minimal dapat diraih dengan menggunakan kombinasi jenis dan nilai *hyper-parameter* yang sudah disebutkan. Selain itu, jenis model yang digunakan ikut mempengaruhi hasil performa *Word2Vec* dalam menangkap relasi sematik atau relasi sintaksis.

2.2 Landasan Teori

Pada bagian Landasan Teori akan dipaparkan berbagai macam teori yang menjadi dasar penelitian yang akan dilakukan oleh penulis. Teori yang akan dipaparkan mencakup dasar teori *text mining*, *preprocessing* , *exploratory data analyis*, *word embedding*, dan metode pengukuran similaritas yang akan digunakan di dalam penelitian ini.

2.2.1 *Text mining*

Digitalisasi saat ini merambah berbagai macam bidang, tak terkecuali dalam bidang hukum. Dokumen hukum Undang-Undang di Indonesia dihimpun di dalam sebuah sistem informasi yang dikenal secara umum sebagai jaringan dokumentasi dan informasi hukum (JDIH). Didalam sistem informasi yang telah disebutkan diatas memuat semua dokumen undang-undang sampai dengan peraturan daerah. Data perundang-undangan di Indonesia yang berbentuk teks membutuhkan pendekatan *text mining* untuk dapat di lakukan analisis lebih lanjut.

Secara umum *text mining* dapat diterjemahkan menjadi sistem yang mengelola data teks untuk mendeteksi pola kebahasaan untuk mendapatkan informasi (Sebastiani, 2002). Untuk memperoleh informasi yang relevan dari sebuah dokumen, perlu untuk meninjau terlebih dahulu metode apa yang paling baik digunakan dengan objek data berupa dokumen Undang-Undang.

Secara garis besar *text mining* dibagi menjadi beberapa kategori berdasarkan tujuan diterapkan nya metode *text mining* tersebut. *Text mining* dapat dimanfaatkan untuk perangkuman teks, pengambilan dokumen yang relevan, similaritas dokumen, dan pengambilan informasi (Witten, 2004). Pada penelitian ini, pendekatan *text mining* akan digunakan adalah untuk mencari tingkat similaritas antara dokumen Undang-Undang atau Rancangan Undang-Undang.

2.2.2 *Preprocessing*

Sumber data yang digunakan pada penelitian ini menggunakan Undang-Undang. Terlebih dahulu sumber data ini perlu melalui tahap pertama di dalam *text mining* yakni *preprocessing* . Secara garis besar *preprocessing* dibagi menjadi dua yakni, *document triage* dan *text segmentation* (Indurkhya & Damerau, 2010). *Document triage* adalah proses mengkonversi kumpulan himpunan dokumen digital menjadi dokumen teks yang baik. *Document triage* didalam literatur lain (Vijayarani et al., 2015) disebut juga sebagai *extraction*.

Setelah dilakukan proses *document triage* maka langkah selanjutnya adalah *text segmentation*. *Text segmentation* (Indurkhya & Damerau, 2010) adalah sebuah proses untuk mengkonversi hasil dari *document triage (text corpus)* ke dalam komponen, yakni kata dan kalimat. Pada paragraf selanjutnya akan dijelaskan mengenai langkah-langkah *text segmentation* yang akan digunakan untuk penelitian ini.Pada sub-bab dibawah ini akan dijelaskan secara lebih mendetail mengenai metode *text* segmentation yang akan digunakan untuk *preprocessing* , yakni :

2.2.2.1 *Extraction*

Extraction adalah proses untuk mengambil data teks dari sebuah dokumen (Vijayarani et al., 2015). Dokumen Undang-Undang (UU) dan Rancangan Undang-Undang (RUU) yang ada di Indonesia tersedia dalam jaringan dokumentasi dan informasi hukum (JDIH) dalam format *portable document file (pdf)*. Untuk dapat di proses lebih lanjut, data teks yang terdapat di dalam dokumen dengan format (*.pdf*) terlebih dahulu harus melalui tahap *extraction*.

2.2.2.2 *Case Folding*

Didalam sebuah dokumen teks, biasanya akan ditemukan huruf kecil (*lowercase*) dan huruf kapital (*uppercase*). *Case folding* menjadi sebuah metode yang umum digunakan untuk mengatasi hal ini, yakni dengan cara mengubah semua huruf kapital di dalam sebuah dokumen teks menjadi huruf kecil (Manning et al., 2009).

2.2.2.3 *Filtering*

Filtering adalah proses yang dilakukan untuk menghilangkan karakter yang tidak dapat digunakan dalam pengolahan teks (Kulkarni & Shivananda, 2019). Contoh karakter yang tidak dapat digunakan lebih lanjut dalam pengolahan data teks adalah tanda baca, angka, dan karakter non-cetak *American Standard Code for Information Interchange* (*ASCII*).

2.2.2.4 *Stopwords Elimination*

Setelah melewati proses *case folding* atau membuat semua huruf di dalam dokumen teks menjadi huruf kecil, langkah yang akan dilakukan selanjutnya adalah *stopwords elimination*. *Stopwords* adalah semua kata yang tidak diperlukan di dalam proses analisis teks (Vijayarani et al., 2015), dikarenakan kata tersebut terlalu umum dalam sebuah data teks (Manning et al., 2009).

Disadur dari jurnal *An Algorithm For Suffix Stripping* (Porter, 2006) *stopwords* merupakan kata-kata yang bukan merupakan kata kunci di dalam analisis *text mining*. Alasan utama *stopwords* perlu dihilangkan adalah untuk mengurangi beban komputasi untuk kata-kata yang kurang memiliki arti.

2.2.3 *Exploratory Data Analysis (*EDA*)*

*Exploratory Data Analysis (*EDA*)* adalah tahap yang dilakukan untuk memahami lebih jauh tentang data yang dimiliki (Kulkarni & Shivananda, 2019). Dalam pengolahan data, tahap EDA dilakukan untuk mengetahui fakta tersembunyi di balik data. Tahap EDA berfokus pada data, struktur data, pencilan data, visualisasi, dan persiapan data untuk dapat digunakan dalam membentuk model (Mukhiya & Ahmed, 2020), dalam hal ini adalah model *Word2Vec*.

Berdasarkan *Hands-on Exploratory analisis with Python* (Mukhiya & Ahmed, 2020), tahap EDA memiliki empat bagian. Pertama, mendefinisikan permasalahan. Melakukan definisi permasalahan dalam tahap EDA diperlukan untuk menentukan desain analisis yang akan dilakukan terhadap data. Permasalahan yang dimaksud

adalah informasi atau fakta tersembunyi yang akan digali lebih lanjut dari data yang dimiliki. Dalam domain *text mining*, informasi atau fakta yang dicari lebih lanjut meliputi daftar kata yang terlalu umum, distribusi kata dalam setiap dokumen, dan visualisasi kata yang umum digunakan dalam sebuah dokumen.

Kedua, persiapan data. Dalam hal ini persiapan data dapat meliputi menentukan sumber daya, melakukan pembersihan data, dan menentukan format penyimpanan data. Dalam penelitian ini, tahap persiapan data untuk tahap EDA, terlebih dahulu sudah dilakukan dalam tahap *preprocessing*. Ketiga, analisis data. Analisis data meliputi mendeskripsikan data menggunakan metode statistik, menemukan korelasi antar data, dan mengelompokkan data hasil analisis sesuai dengan definisi permasalahan. Keempat. Pengembangan dan representasi hasil analisis. Setelah analisis data dilakukan, maka hasil analisis dapat disajikan dalam bentuk yang lebih mudah dibaca seperti tabel, grafik, dan diagram. Dalam EDA untuk data teks, representasi hasil analisis yang umum digunakan adalah *wordcloud*.

2.2.4 *Term Frequency Inverse Document Frequency* (*TF-IDF*) dan *Pivoted Normalization*

Beragamnya pemakaian kata di dalam sebuah dokumen teks, membuat semua kata di dalam teks tidak bisa dianggap sebagai sebuh kata yang penting. Penting atau tidak nya sebuah kata di dalam sebuah dokumen teks, akan berpengaruh dalam proses analisis yang akan dilakukan. Hal ini pertama kali dikemukakan oleh (Luhn, 1957), yang melakukan pendekatan statistik dengan cara memberikan bobot untuk setiap kata di dalam dokumen, dimana bobot yang dimaksudkan adalah jumlah kemunculan kata dalam sebuah dokumen yang disebut dengan *term frequency*.

Cara paling umum untuk menghitung *term frequency* adalah dengan metode pengitungan *document frequency*, yakni menghitung jumlah kemunculan kata pada setiap dokumen, bukan keseluruhan dokumen. Sehingga dengan cara ini akan didapatkan hasil penghitungan statistik pada tingkat dokumen yang lebih baik.Penggunaan *term frequency* tidak terlepas dari sebuah permasalahan, yakni semua kata dianggap sama penting kedudukannya di dalam dokumen. Hal ini

menyebabkan ketidak akuratan apabila digunakan dalam proses pengambilan informasi. Oleh karena itu tercetusnya konsep *inverse document frequency* oleh (Jones, 1972), dimana pembobotan berdasarkan jumlah kemunculan kata dilakukan secara terbailik. Nilai yang lebih tinggi, digunakan untuk kata dengan kemunculan rendah, dan nilai lebih rendah diberikan untuk kata dengan jumlah kemunculan yang tinggi. Perhitungan *inverse document frequency* ditunjukkan pada Persamaan 2.14.

$$df_t = \log \frac{N}{dft} \tag{2.14}$$

Persamaan 2.1 *Inverse Document Frequency*

Persamaan 2.1 merupakan perhitungan yang digunakan untuk mengubah bobot kata dengan jumlah kemunculan tinggi menjadi rendah, dan kata dengan kemunculan lebih rendah untuk memiliki bobot lebih tinggi.

Seiring dengan berkembang nya bidang *text mining*, jenis dokumen teks di proses semakin beragam. Hal ini menyebabkan timbulnya beberapa ketidakseimbangan perbandingan antar dokumen yang memiliki panjang yang berbeda. Dimana dokumen yang memiliki dimensi lebih besar, cenderung memiliki nilai *term frequency* lebih tinggi dan kata-kata yang akan mempengaruhi hasil dari pengambilan informasi yang akan dilakukan. Oleh karena itu, dibutuhkan metode untuk melakukan normalisasi untuk menyeragamkan semua bentuk dokumen yang ada, sehingga bias yang mungkin ditimbulkan karena perbedaan ukuran dokumen dapat dihindari.

Bentuk normalisasi yang hingga saat ini memiliki tingkat akurasi lebih baik dengan bentuk normalisasi lainnya disebut dengan *pivoted normalization* yang bekerja dengan cara menghitung normalisasi setiap dokumen, kemudian berdasarkan hasil normalisasi tersebut, dokumen dengan faktor normalisasi paling besar, akan memiliki kemungkinan terpilih semakin kecil (Singhal et al., 1995). Hal ini

ditunjukkan dalam Persamaan untuk dua jenis *pivoted normalization* yang lazim digunakan ditunjukkan pada Persamaan 2.2 dan Persamaan 2.3.

$$Pivoted\ Unique\ Normalization = \frac{\frac{1+\log(tf)}{1+\log(average\ tf)}}{(1-slope)\ x\ pivot + slope\ x\ \#\ of\ unique\ terms} \quad (1)$$

Persamaan 2.2 *Pivoted* Unique *Normalization*

$$Pivoted\ Lenght\ Normalization = (1-slope)\ x\ average\ byte\ size + slope\ x\ bytesize \quad (2)$$

Persamaan 2.3 *Pivoted Length Normalization*

Tujuan dari dilakukannya *pivoted normalization* adalah untuk meningkatkan akurasi hasil pengukuran relevansi dengan cara mengurangi bias yang disebabkan oleh perbedaan ukuran *corpus*.

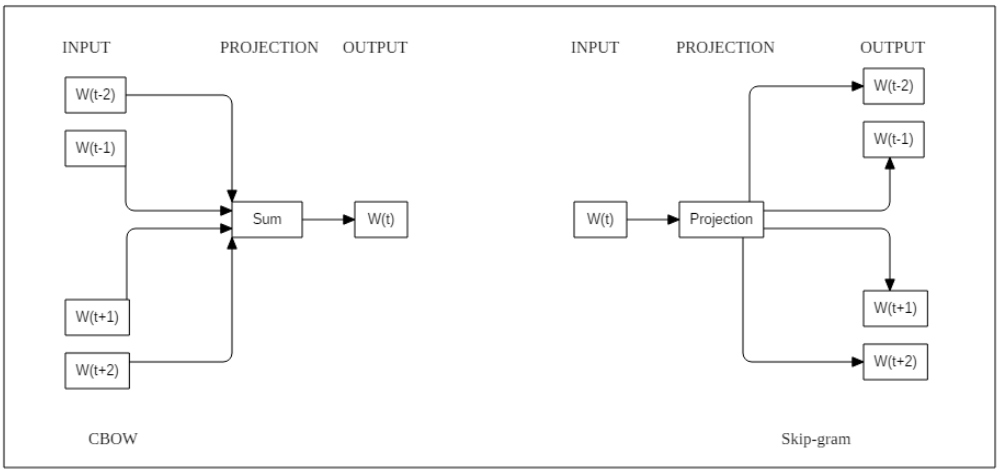
2.2.5 *Word2Vec*

Natural language processing atau pemrosesan bahasa alami mengalami banyak perkembangan sejak mulai ditemukan. Perkembangan ini ditandai dengan munculnya beragam model arsitektur yang menfasilitasi pemrosesan bahasa alami seperti *Feedforward Neural Net Language (NNLM)*, *Recurrent Neural Net Language Model (RNLMM)* dan *Parallel Training of Neural Network*. Namun, dari ketiga model yang sudah disebutkan ditemukan permasalahan yang sama yakni ketiga model tersebut membutuhkan daya komputasi yang tinggi dan mahal. Ketiga model tersebut memiliki kebutuhan komputasi yang tinggi dikarenakan ketiga model tersebut menerapkan konsep *hidden-layer* yang kompleks, sehingga untuk data yang berjumlah besar tentu ketiga model tersebut akan membutuhkan daya komputasi yang besar.

Sedikit berbeda konsep dengan ketiga model yang sudah disebutkan, diusulkan sebuah model arsitektur pemrosesan bahasa alami yang lebih efisien dalam hal komputasi dikarenakan menghilangkan *hidden-layer non-linear* di dalam model *Word2Vec* yang di usulkan (Mikolov, Chen, et al., 2013).

Model yang diusuilkan disebut dengan *Word2Vec* yang memiliki dua jenis tipe arsitektur.Pertama, *Continuous Bag-of-Words Model (CBOW)*. Model arsitektur pertama yang diusulkan memiliki kemiripan dengan *feedforward NNLM*. Namun, model *CBOW* menghilangkan *hidden-layer non-linear* dan *layer projection* terbagi untuk semua kata konteks. Model ini bekerja dengan cara menerima input berupa kata-kata disekeliling kata target, kemudian model akan menghasilkan luaran berupa kata target. Kedua, *Continuous Skip-gram Model* yang memiliki kemiripan arsitektur dengan model *CBOW*, akan tetapi di dalam model *Skip-gram* terdapat perbedaan berupa model ini akan mengklasifikasi kata-kata yang berada di sekitar kata yang diinputkan. Kualitas hasil luaran *vector* yang dihasilkan oleh model *Skip-gram* dapat ditingkatkan dengan cara memperluas jangkauan klasifikasi kata di dalam model, akan tetapi akan meningkatkan kompleksitas komputasi.

Di dalam paper yang sama (Mikolov, Chen, et al., 2013) disebutkan bahwa selain memiliki tingkat kompleksitas yang lebih rendah, *Word2Vec* juga menghasilkan performa yang lebih baik dari beberapa arsitektur yang sudah ada sebelumnya. Performa ini ditunjukkan dengan hasil rata-rata dalam mencari *semantic relationship* dan *syntactic relationship*. Hal ini yang menjadikan penulis menjadikan *Word2Vec* sebagai metode *word embedding* yang dipilih.Berikut akan disajikan ilustrasi model *Word2Vec* pada Gambar 2.1.



Gambar 2.1 Ilustrasi Arsitektur *Word2Vec (CBOW & Skip-gram)*

Gambar 2.1 menjelaskan tentang perbedaan arsitektur *CBOW* dan *Skip-gram*. Dari gambar dapat dilihat bahwa model *CBOW* menghasilkan luaran berupa kata target dengan input kata di sekeliling target. S*eda*ngkan *Skip-gram* menghasilkan luaran berupa kata disekeliling kata target dengan diberikan input kata target.

2.2.5.1 Model *CBOW*

Model paling sederhana dari *Word2Vec* adalah *CBOW (Continuous Bag-of-Word)* yang memiliki cara kerja menerima input berupa kata disekeliling kata target, kemudian model akan menghasilkan luaran berupa kata target (Mikolov, Chen, et al., 2013). Model *Continuous Bag-of-Word* terbagi menjadi dua jenis, pertama yakni model *One-word context*, dan kedua adalah model *Multi-word context*. Untuk dapat menghitung dan membuat representasi vektor dari sebuah kompilasi Undang-Undang tentu model yang akan digunakan adalah *Multi-word context*.

Berdasarkan (Mikolov, Chen, et al., 2013) untuk kasus yang tidak memerlukan eksplorasi kata yang jarang di dalam sebuah teks, model *CBOW* lebih baik untuk digunakan. Konsep matematis model *Multi-word context CBOW* akan disajikan berdasarkan penelitian berjudul "*Word2Vec Parameter Learning Explained"* (Rong, 2014).

Pada fase *Forward Propagation* tahap pertama yang dilakukan adalah menghitung *output hidden-layer*. Pada *hidden layer* akan dihitung rata-rata vektor yang masuk ke dalam *input layer*. Persamaan dasar ouput *hidden-layer* ditunjukkan dalam Persamaan (2.4) .

$$h = \frac{1}{C}\left(V_{w_1} + V_{w_2} + \cdots + V_{w_C}\right)^T \tag{2.4}$$

Keterangan :

h = nilai bobot *hidden-layer*

C = jumlah kata input

V = representasi vektor dari kata input

Persamaan 2.4 Perhitungan Bobot *input layer* ke *hidden layer*

Penjelasan dari persamaan (2.4) adalah h merupakan *hidden-layer* ataupun *hidden weight*, Dimana *C* merupakan jumlah kata yang terdapat di dalam sebuah dokumen teks dan *Vw* merupakan vektor input dari sebuah kata *w* . Nilai *hidden layer* akan digunakan untuk membobotkan nilai vektor untuk setiap kata pada pada *input layer*. Persamaan pembobotan pada *hidden layer* di tunjukkan dalam Persamaan (2.5).

$$\mu = V'^{T}_{w_j} h \tag{2.5}$$

Keterangan :

μ = bobot *hidden layer* ke *output layer*

$V'^{T}_{w_j}$ = Vektor transpose *input layer*

h = nilai *hidden layer*

Persamaan 2.5 Perhitungan bobot *hidden layer* ke *output layer*

Persamaan 2.5 akan digunakan untuk membobotkan vektor untuk setiap kata dari *input layer* dengan nilai *hidden layer* yang didapatkan pada Persamaan 2.4. Setelah didapatkan bobot dari *hidden layer*, maka selanjutnya vektor akan masuk pada *output layer* yang menggunakan perhitungan *softmax function*. Perhitungan *softmax function* ditunjukkan pada Persamaan 2.6

$$p(w_j|w_I) = \frac{\exp\left(V'^{T}_{w_j} V_{w_I}\right)}{\Sigma^{V}_{j'} =_1 \exp\left(V'^{T}_{w_{j'}} V_{w_I}\right)} \tag{2.6}$$

Keterangan :

$p(w_j|w_I)$ = perhitungan *softmax function*

$V'^{T}_{w_j}$ = nilai vektor *hidden layer* ke *output layer*

V_{w_I} = nilai vektor *input layer* ke *hidden layer*

Persamaan 2.6 Perhitungan *Softmax function* pada *Output Layer*

Persamaan 2.6 akan memilih nilai vektor kata konteks yang dimasukkan ke dalam *hidden layer* secara acak pada *output layer*. Sehingga, terdapat probabilitas nilai vektor *hidden layer* ke *output layer* akan berbeda dengan nilai *output layer*. Perbedaan nilai vektor ini kemudian akan dihitung menggunakan *loss function*.

Persamaan untuk menghitung *error* ataupun *loss function* dalam pembentukan model disajikan dalam persamaan (2.6). Penghitungan ini dilakukukan dalam fase *learning weight matrices with backpropagation*.

$$E = -V'^{T}_{W_O}.h + log \sum_{j'=1}^{V} \exp(V'^{T}_{w_j}.h) \tag{2.7}$$

Keterangan :

$V'^{T}_{W_O}.h$ = nilai vektor *hidden layer* ke *output layer*

$V'^{T}_{w_j}.h$ = nilai *output layer*

Persamaan 2.7 Perhitungan *Loss function Hidden layer* ke *Output layer*

Hasil dari Persamaan 2.6 adalah nilai *error* yang dihasilkan antara nilai vektor *hidden layer* dan *output layer*. Apabila nilai vektor *hidden layer* lebih besar dari nilai *output layer*, hal ini disebut dengan *underestimated*, sedangkan apabila nilai vektor *hidden layer* lebih kecil dari nilai vektor *output layer*, hal ini disebut dengan *overestimated*. Sehingga, semakin kecil selisih antara nilai vektor *hidden-layer* dengan *output layer* maka nilai vektor akan semakin sedikit berubah. Persamaan yang digunakan untuk menghitung *update* vektor dari *hidden-layer* yang digunakan dalam *Multi-context word*s disajikan dalam persamaan (2.7).

$$V'^{(baru)}_{W_j} = V'^{(lama)}_{W_j} - \eta . e_j . h, \quad untuk\ j = 1,2, \dots, V \tag{2.8}$$

Keterangan :

$V'^{(lama)}_{W_j}$ = nilai *output layer* (lama)

η = nilai *learning rate* (0.25)

e_j = nilai *loss function output layer*

h = nilai bobot *hidden layer*

Persamaan 2.8 Perhitungan *Update* Vektor *Hidden layer* ke *Output layer*

Nilai yang dapat dalam pengitungan *loss function* akan digunakan untuk memperbarui nilai vektor *hidden layer* ke *output layer* seperti yang ditunjukkan

pada Persamaan 2.7 dan memperbarui nilai vektor *input layer* ke *hidden layer* yang akan ditunjukkan pada Persamaan 2.8.

Persamaan yang digunakan untuk menghitung *update input layer* ke *hidden-layer* yang digunakan dalam *Multi-context word*s disajikan dalam Persamaan 2.8.

$$V_{w_{I_{.c}}}^{(new)} = V_{w_{I_{.c}}}^{(old)} - \frac{1}{C} . \eta . EH^T \quad untuk\ c = 1,2, \dots, C \tag{2.9}$$

Keterangan :

$V_{w_{I_{.c}}}^{(old)}$ = nilai vektor setiap kata pada *input layer*

C = jumlah kata input

η = nilai *learning rate* (0.25)

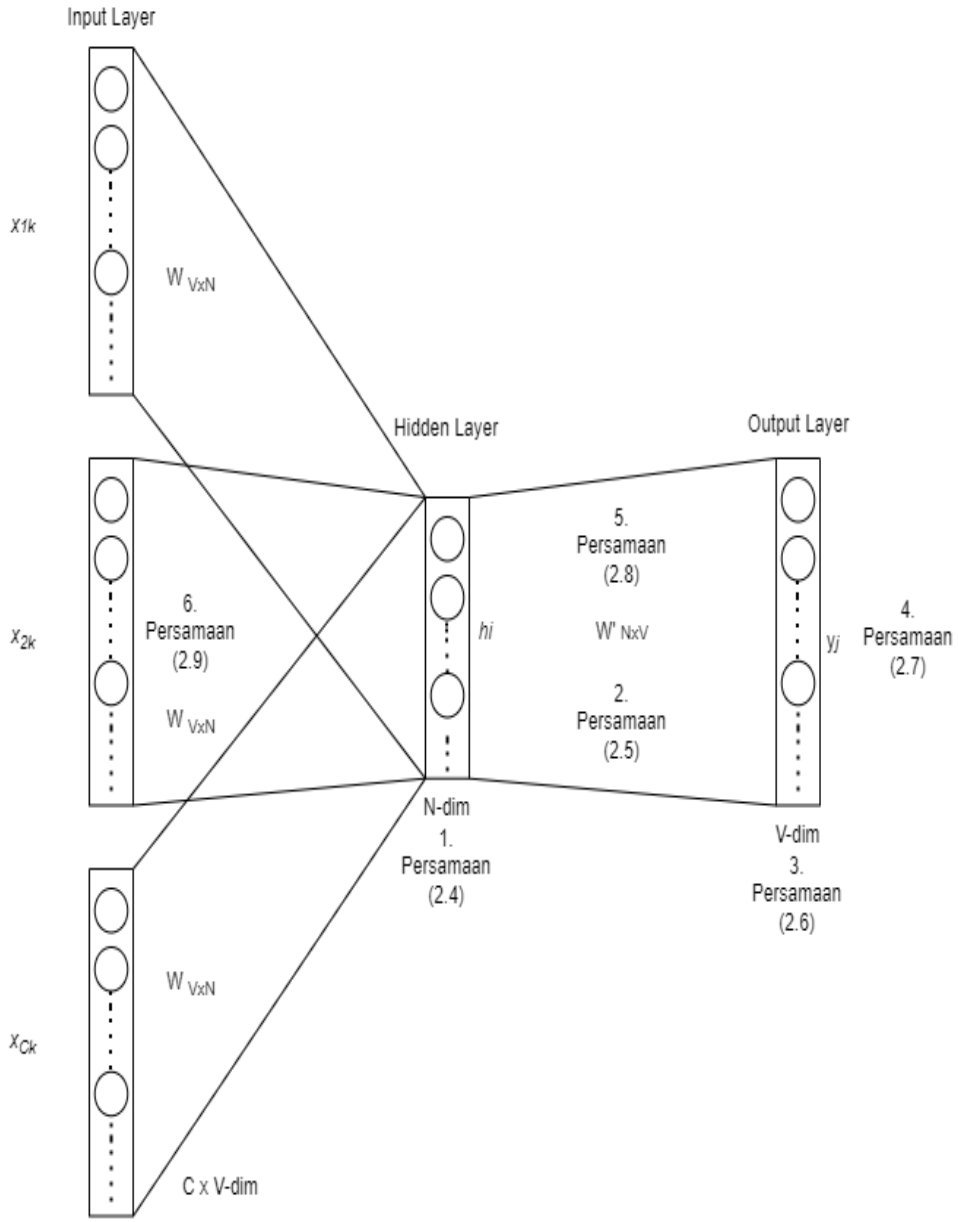
EH^T = jumlah nilai *output hidden layer*, dikalikan dengan nilai *loss function*

Persamaan 2.9 Perhitungan *Update* Vektor *Input layer* ke *Hidden layer*

Persamaan 2.4 hingga Persamaan 2.9 merupakan perhitungan yang dilakukan dalam membentuk model *Word2Vec*. Setiap kata akan direpresentasikan dalam one-hot vector atau vektor yang berisikan nilai 1 dan 0, dimana nilai 1 merepresentasikan sebuah kata dari himpunan kata dalam dokumen teks.

Kemudian, nilai *hidden layer* akan dihitung berdasarkan Persamaan 2.4. Hasil pembobotan vektor *input layer* dengan *hidden layer* dihitung menggunakan Persamaan 2.5. Dimana hasil dari Persamaan 2.5 akan dijadikan nilai input untuk menghitung nilai *softmax function* untuk pada *output layer* yang ditunjukkan pada Persamaan 2.6. Penghitungan *softmax function* akan menghasilkan nilai probabilitas untuk setiap kata, dan menyebabkan perbedaan nilai dengan nilai pembobotan dengan Persamaan 2.5. Hal ini membutuhkan penghitungan nilai *error* untuk mendapatkan nilai dengan probabilitas terbaik dengan menggunakan Persamaan 2.7. Nilai *error* yang didapatkan digunakan untuk memperbarui nilai pembobotan vektor *hidden layer* ke *output layer* dengan menggunakan Persamaan

2.8 dan nilai vektor pada *input layer* menggunakan Persamaan 2.9. Ilustrasi pembentukan model *Word2Vec* CBOW ditunjukkan pada Gambar 2.2.



Gambar 2.2 Ilustrasi Model CBOW

Ilustrasi Word2Vec model CBOW disajikan dalam Gambar 2.2. Dapat dilihat pada ilustrasi yang disajikan penggunan Persamaan 2.4 sampai dengan Persamaan 2.9 sesuai dengan layer pada model CBOW Word2Vec.

2.2.5.2 Model *Skip-gram*

Berbeda dengan model *CBOW*, prinsip kerja model *Skip-gram* berkebalikan dengan model *CBOW* (Mikolov, Chen, et al., 2013). Model ini akan mengklasifikasi kata-kata yang berada di sekitar kata yang diinputkan (kata target), sehingga model *Skip-gram* lebih baik digunakan untuk kasus yang membutuhkan representasi kata yang jarang di dalam sebuah dokumen teks. Dalam jurnal Distributed Representations of *Words and Phrases an their Composionality* (Mikolov, Sutskever, et al., 2013) dijelaskan bahwa model *skip-gram* dapat di kombinasikan dengan metode *negative sampling* yang disebut dengan *Skip-gram Negative Sampling* (SGNS), kombinasi ini membuat metode *skip-gram* memiliki efisiensi yang lebih baik dalam menemukan relasi semantik antar kata. Konsep persamaan model *Skip-gram* disajikan berdasarkan penelitian "*Word2Vec Parameter Learning Explained"* (Rong, 2014). Tahap pertama yang dilakukan adalah menghitung nilai *hidden-layer* (*Forward Propagation*), berikut akan disajikan persamaan *hidden-layer* model *Skip-gram* dalam persamaan (2.10).

$$h = W_{(k,.)}^{T} := V_{W_I}^{T}, \tag{2.10}$$

Keterangan :

$V_{W_I}^{T}$ = Nilai transpose dari baris vektor input

Persamaan 2.10 Perhitungan nilai *Hidden layer Skip-gram*

Kemudian, vektor dari *input layer* akan dibobotkan menggunakan bobot *hidden layer* untuk menjadi input pada *output layer*. Hal in ditunjukkan dalam persaman (2.11).

$$\mu_{c,j} = V_{w_j}'^{T} . h, untuk\ c = 1,2, \dots, C \tag{2.11}$$

Keterangan :

$\mu_{c,j}$ = bobot *hidden layer* ke *output layer*

$V_{w_j}'^{T}$ = Vektor transpose *input layer*

h = nilai *hidden layer*

Persamaan 2.11 Perhitungan bobot *hidden layer* ke *output layer*

Dikarenakan model *Skip-gram* ditujukan untuk memprediksi kata-kata yang ada disekitar kata yang diinputkan, maka pada *output layer* yang akan dihasilkan adalah distribusi multinomial dari *C* yang dalam hal ini adalah kata konteks. Setiap *output* kata menggunakan hasil yang sama dari *hidden layer*. Hal ini ditunjukkan dalam persamaan (2.12).

$$p(w_{c_{,j}} = wO, c \mid wI) = y_{c,j} = \frac{\exp\,(u_{c_{,j}})}{\sum_{j'=1}^{V} \exp(uj')} \tag{2.12}$$

Keterangan :

$u_{c_{,j}}$ = nilai vektor aktual dari *hidden layer* ke *output layer*

uj' = nilai vektor seluruh kata konteks dari *hidden layer* ke *output layer*

Persamaan 2.12 Perhitungan *Multinomial Softmax function Skip-gram* pada *output layer*

Persamaan 2.12 merupakan perhitungan *multinomial softmax function* untuk model *Skip-gram*, dimana perhitungan ini menghitung probabilitas *output* untuk kata input dengan kata konteks yang berada di sekitar kata input.Perhitungan *loss function* disajikan dalam persamaan (2.13)

$$E = \sum_{c=1}^{C} uj_c^* + C\,.\log \sum_{j'=1}^{V} \exp\,(uj') \tag{2.13}$$

Keterangan :

uj_c^* = penjumlahan nilai vektor aktual dari *hidden layer* ke *output layer* untuk setiap kata dalam dokumen

uj' = penjumlahan nilai vektor seluruh kata konteks dari *hidden layer* ke *output layer*

Persamaan 2.13 Persamaan *Loss function Skip-gram*

Persamaan 2.13 akan digunakan untuk menghitung nilai *loss function* vektor dari *hidden layer* ke *output layer*. Nilai yang didapatkan dari penghitungan ini kemudian akan digunakan untuk memperbarui nilai vektor *input layer* ke *hidden layer* dan

nilai vektor *hidden layer* ke *output layer*. Perhitungan untuk memperbarui nilai vektor *hidden layer* ke *output layer* ditunjukkan dalam Persamaan 2.14.

$$V'_{w_{ij}}{}^{(baru)} = V'_{w_{ij}}{}^{(lama)} - \eta \,.\, EI_j \,.\, h \quad untuk\ j = 1,2, \dots, V \tag{2.14}$$

Keterangan :

η = nilai *learning rate*

EI_j = Jumlah nilai *loss function* untuk setiap kata yang diinputkan

h = nilai bobot *hidden layer*

Persamaan 2.14 Perhitungan *Update* Vektor *Hidden layer* ke *Output layer Skip-gram*

Persamaan 2.14, akan memperbarui nilai vektor *hidden layer* ke *output layer*. Dapat dilihat pada Persamaan 2.14 bahwa *learning rate* memiliki peran dalam memperbarui nilai vektor. Dikarenakan jumlah *output layer* pada model *skip-gram* lebi dari satu panel dan melalui pembobotan hidden layer yang sama, untuk meringkas notasi dari setiap hasil vektor yang dikeluarkan output layer maka dirumuskan EI. EI merupakan jumlah prediksi eror yang dihasilkan oleh prediksi kata konteks dalam output layer. Persamaan untuk mendapatkan nilai EI dijabarkan dalam Persamaan 2.15.

$$EI_j = \sum_{c=1}^{C} e_{c,j} \tag{2.15}$$

Keterangan :

EI = Penjumlahan nilai *error* untuk setiap kata konteks pada output layer

C = Kata konteks yang terdapat pada output layer.

Persamaan 2.15 Persamaan Menghitung Nilai *Error* pada *Output Layer*

Persamaan 2.15 akan menjumlahkan setiap nilai *error* pada output layer pada model *skip-gram*. Nilai *error* tersebut merupakan selisih antara hasil prediksi kata konteks pada output layer model *skip-gram* dengan nilai vektor asli kata konteks.

Setelah itu, nilai vektor *input layer* ke *hidden layer* akan diperbarui dengan Persamaan 2.16.

$$V_{w\ I}^{(baru)} = V_{w\ I}^{(lama)} - \eta \,.\, EH^T \tag{2.16}$$

Keterangan :

$V_{w\ I}^{(lama)}$ = nilai vektor *input layer* ke *hidden layer*

η = nilai *learning rate*

EH^T = vektor *n-dimensional* antara jumlah nilai *loss function* dengan nilai vektor *hidden layer* ke *output layer*

Persamaan 2.16 Perhitungan *Update* Vektor *Input layer* ke *Hidden layer Skip-gram*

Persamaan 2.15 digunakan untuk memperbarui bobot vektor dari *input layer* ke *hidden layer*, dimana *EH* akan dijabarkan lebih lanjut dalam Persamaan 2.16.

$$EH_i = \sum_{j=1}^{V} EI_j \,.\, w'_{i\ j} \tag{2.17}$$

Keterangan :

EI_j= Penjumlahan nilai *error* untuk setiap kata konteks pada *output layer*

$w'_{i\ j}$ = vektor hidden ke output layer

Persamaan 2.17 Perhitungan Vektor *N-Dimensional Loss function*

Persamaan 2.17 akan menghitung vektor *n-dimensional* dimana *EI* adalah jumlah nilai *loss function* untuk setiap kata kontes yang ada di dalam dokumen dengan nlai vektor *hidden layer* ke *output layer*.

Persamaan yang telah disajikan pada Persamaan 2.10 sampai dengan Persamaan 2.17 merupakan persamaan yang akan digunakan apabila model *skip-gram* dilakukan secara normal tanpa adanya modifikasi terkait dengan metodologi yang digunakan dalam model *skip-gram*. Namun, akan berbeda apabila model *skip-gram* dilakukan dengan menggunakan modifikasi berupa *negative sampling*.

Negative sampling pada model *skip-gram* dilakukan dengan tujuan untuk meningkatkan kualitas representasi vektor yang dihasilkan. Hal ini, dilakukan dengan cara melakukan *subsampling* dan melakukan update vektor hanya berdasarkan nilai *negative sampling* yang didapatkan.

Berikut akan dijabarkan persamaan tambahan yang membentuk metode *skip-gram negative sampling* (SGNS), berdasarkan penelitian berjudul "*Distributed Representations of Words and Phrases and their Composionality*" (Mikolov, Sutskever, et al., 2013) dan "*Word2Vec Parameter Learning Explained*" (Rong, 2014). Persamaan 2.18 menunjukkan rumus yang digunakan untuk melakukan *subsampling*.

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}} \tag{2.18}$$

Keterangan :

t = Nilai ambang, biasanya memiliki nilai 10^{-5}

$f(w_i)$ = Frekuensi kemunculan kata *Wi*

Persamaan 2.18 Persamaan *Subsampling* Kata Model *Skip-gram*

Persamaan 2.18 menunjukkan persamaan yang digunakan untuk melakukan *subsampling*. *Subsampling* dilakukan dengan tujuan untuk menghindari komposisi data yang tidak seimbang yang berupa kata dengan frekuensi kemunculan tinggi,dengan kata dengan frekuensi kemunculan yang jauh lebih rendah. Penggunaan Persamaan 2.18 tidak dijabarkan secara detail, namun persamaan 2.18 diklaim dapat meningkatkan akurasi pembelajaran vektor untuk kata dengan frekuensi kemunculan yang rendah (Mikolov, Sutskever, et al., 2013).

Selain *subsampling*, terdapat beberapa persamaan yang akan berubah dikarenakan modifikasi model *skip-gram* menggunakan *negative sampling*. Persamaan 2.19 menyajikan lost function model *skip-gram* apabila menggunakan *negative sampling*.

$$E = -\log \quad \sigma\left(V'^{\,T}_{w_O} h\right) - \sum_{w_j \in W_{neg}} \log \quad \sigma(-V'^{\,T}_{w_j} h) \tag{2.19}$$

Keterangan :

$V'^{\,T}_{w_O}$ = Nilai transpose vektor output layer (Positive Sample)

h = Nilai hidden-layer

$w_j \in W_{neg}$ = Kata hasil sample menggunakan unigram distribution, untuk menghasilkan himpunan kata yang akan dijadikan *negative* sample.

$V'^{\,T}_{w_j} h$ = Nilai transpose vektor output layer (*Negative* sample)

Persamaan 2.19 Persamaan *Loss function Skip-gram Negative Sampling*

Persamaan 2.19 menunjukkan fungsi *loss function* yang digunakan apabila model *skip-gram* di modifikasi menggunakan *negative sampling*. Persamaan 2.19 memiliki peran untuk meningkatkan hasil pembelajaran representasi vektor yang dilakukan (Mikolov, Sutskever, et al., 2013). Hasil Persamaan 2.19 yang menghitung nilai error kemudian akan digunakan untuk memperbarui bobot matriks dari hidden layer ke output layer, ditunjukkan pada Persamaan 2.20.

$$V'^{(baru)}_{w_j} = V'^{(lama)}_{w_j} - \eta\left(\sigma\left(V'^{\,T}_{W_J} h\right) - t_j\right) h \tag{2.20}$$

Keterangan :

$V'^{(baru)}_{w_j}$ = Nilai vektor output layer yang sudah diperbarui

$V'^{(lama)}_{w_j}$ = Nilai vektor output layer sebelum diperbarui

η = Nilai learning rate

t_j = Label untuk kata pada output layer. t = 1 ketika wj adalah sampel positif, dan t = 0 ketika wj adalah sampel negatif.

Persamaan 2.20 Persamaan Update Vektor Hidden Layer ke Output Layer *Skip-gram Negative Sampling*.

Pada persamaan diatas, perbaruan nilai vektor pada *output layer* hanya akan dilakukan untuk kata yang termasuk dalam sampel, baik sampel positif atau sampel *negative*, dan bukan dilakukan untuk semua kata dalam vocabulary.

Setelah didapatkan hasil perbaruan nilai vektor dari hidden layer ke output layer menggunakan *skip-gram negative sampling* (SGNS), maka selanjutnya akan dilakukan perbaruan pada nilai vektor dari input layer ke hidden layer pada model *skip-gram negative sampling*. Secara umum, tidak ada perubahan yang siginfikan dalam persamaan yang digunakan untuk memperbarui nilai vektor dari input layer ke hidden layer sebagaimana yang disajikan pada Persamaan 2.16. Namun, terdapat perubahan pada definisi EH, yang akan ditunjukkan dalam Persamaan 2.21.

$$EH = \sum_{w_j \in \{wO\} \cup W_{neg}} \left(\sigma\left(V_{w_j}'^{T} h\right) - t_j\right) V_{w_j}' \tag{2.21}$$

Keterangan :

$w_j \in \{wO\} \cup W_{neg}$ = w_j merupakan kata output yang termasuk dalam kata yang menjadi sampel.

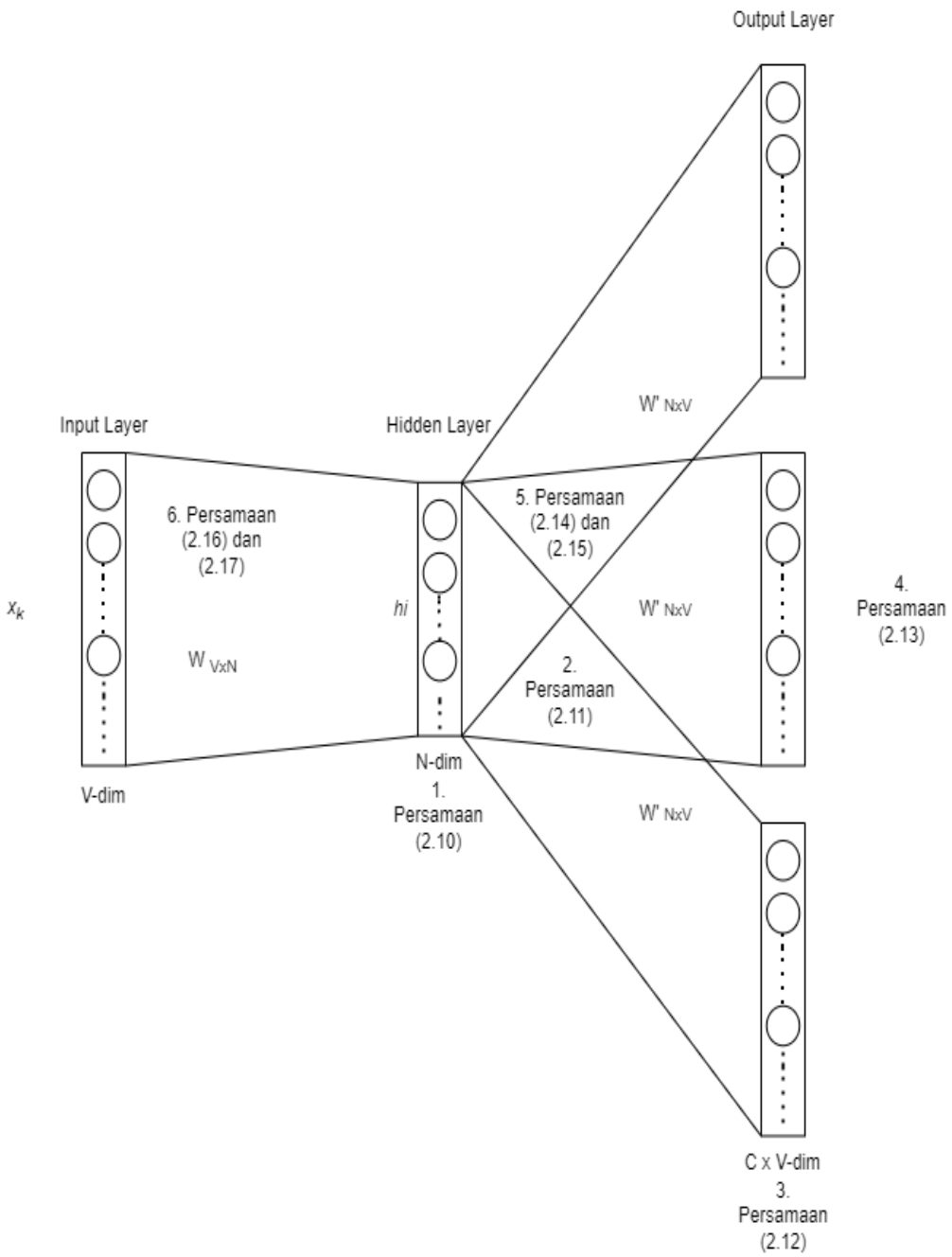
V_{w_j}' = Nilai output vektor

h = Nilai hidden layer

t_j = Label untuk kata pada output layer. t = 1 ketika wj adalah sampel positif, dan t = 0 ketika wj adalah sampel negatif.

Persamaan 2.21 Perhitungan Vekton N-dimensional *Loss function Skip-gram Negative Sampling*.

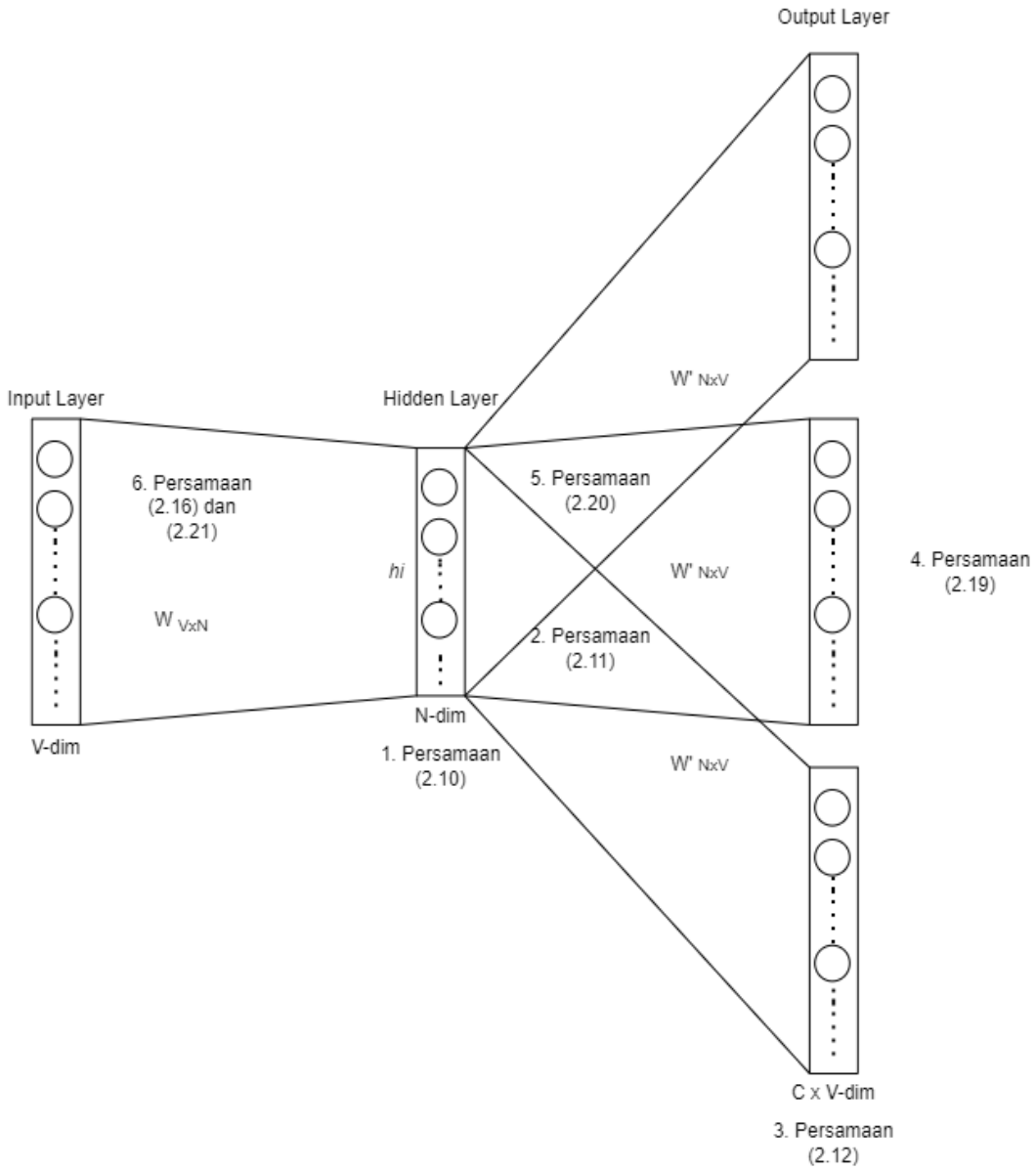
Berdasarkan Persamaan 2.10 sampai dengan Persamaan 2.17 yang menjabarkan model *skip-gram* tanpa *negative sampling*, dan Persamaan 2.18 sampai dengan Persamaan 2.21 yang merupakan persamaan yang digunakan untuk model *skip-gram* menggunakan *negative sampling*, akan diilustrasikan penggunaan Persamaan 2.10 sampai dengan 2.17 pada model *skip-gram*. Ilustrasi ditunjukkan pada Gambar 2.3.



Gambar 2.3 Ilustrasi Model *Skip-gram*

Gambar 2.3 mengilustrasikan penggunaan Persamaan 2.10 sampai dengan Persamaan 2.17 dalam setiap lapisan layer model *skip-gram* Word2Vec. Dimana penggunaan Persamaan 2.10 sampai dengan Persamaan 2.17 merupakan penerapan model *skip-gram* tanpa adanya modifikasi berupa *negative sampling*.

Sementara itu, Penggunaan Persamaan 2.19 sampai dengan Persamaan 2.21 dalam model *skip-gram* yang menggunakan modifikasi *negative sampling* di ilustrasikan dalam Gambar 2.4.



Gambar 2.4 Ilustrasi Model *Skip-gram Negative Sampling*

Gambar model *skip-gram* diatas seperti yang ditampilkan pada Gambar 2.4 menunjukkan ilustrasi penggunaan persaman model *skip-gram* dengan modifikasi *negative sampling*.

2.2.6 *Soft Cosine Similarity Measure*

Berdasarkan hasil yang didapatkan (Mikolov, Chen, et al., 2013), model arsitektur *Word2Vec* yang diusulkan yakni model *Skip-gram* memiliki hasil yang paling baik untuk menentukan hubungan semantik antar kata. Hubungan semantik antar kata yang dihasilkan oleh *Word2Vec* belum dapat digunakan untuk mengukur tingkat similaritas antar dokumen, oleh karena itu diperlukan tahapan lebih lanjut untuk dapat mengukur tingkat similaritas hubungan semantik yang dihasilkan. Salah satu metode yang sering digunakan adalah metode pengukuran *cosine similarity*.

Metode pengukuran *cosine similarity* di dalam bidang *text mining* cukup luas digunakan. *cosine similarity* menggunakan dasar perhitungan jarak *cosine* pada vektor (Salton & Buckley, 1988), sehingga linear untuk dipadukan dengan model arsitektur *Skip-gram* yang merepresentasikan kata dalam bentuk vektor. Di dalam pengukuran similaritas yang mengkombinansikan arsitektur *Skip-gram* dan *cosine similarity* sebuah teks ataupun dokumen direpresentasikan dalam bentuk vektor.

Berikut akan dijabarkan mengenai komponen dasar penghitungan *cosine similarity* (Rahutomo et al., 2012) :Representasi dokumen sebagai sebuah vektor ditunjukkan dalam Persamaan 2.15 :

$$\vec{d} = (Wd0, Wd1, \dots, Wdk) \tag{2.15}$$

Persamaan 2.22 Representasi Vektor Dokumen

Persamaan 2.15 menjelaskan representasi sebuah dokumen sebagai sebuah vektor. Di dalam sebuah dokumen tentu terdiri dari beberapa paragraf. Representasi dokumen yang dapat dipilih antara lain keseluruhan dokumen dan paragraf. Representasi kueri yang dimasukkan ditunjukkan dalam Persamaan 2.16 :

$$\vec{q} = (Wq0, Wq1, \dots, Wqk) \tag{2.16}$$

Persamaan 2.23 Representasi Vektor Kueri

Persamaan 2.16 menjelaskan representasi *querry/*kueri yang dimasukkan ke dalam perhitungan *cosine similarity*. Yang dimaksud dengan *querry* adalah input perintah untuk membandingkan sebuah dokumen dengan satu atau lebih dokumen lainnya. Similaritas diantara dua vektor dapat dihitung dengan persamaan (2.16) :

$$\mathrm{Sim}(\vec{q},\vec{d}) = \frac{\vec{q}\cdot\vec{d}}{|\vec{q}|\,|\vec{d}|} = \frac{\sum_{k=1}^{t} Wqk\ X\ Wdk}{\sqrt{\sum_{k=1}^{t}(Wqk)^2}\ \cdot\sqrt{\sum_{k}^{t}(Wdk)^2}} \tag{2.16}$$

Keterangan :

Wqk = representasi vektor kueri

Wdk = representasi vektor dokumen

Persamaan 2.24 Perhitungan *Cosine Similarity* Antara Dokumen dan Kueri

Persamaan (2.16) merupakan persamaan untuk menghitung *cosine similarity* antar dokumen yang ingin dibandingkan. Dokumen dilambangankan dengan “d” dan kueri yang diinputkan dilambangkan dengan “q”. Dengan perhitungan *cosine similarity* diatas, maka *output* yang kita dapatkan adalah tingkat similaritas antar dokumen, dikarenakan dokumen-dokumen yang diukur di representasikan dalam bentuk vektor.

Penghitungan similaritas menggunakan *cosine similarity* menganggap bahwa setiap fitur memiliki vektor yang berbeda. Sebagai contoh dengan menggunakan perhitungan *cosine similarity*, hasil untuk penghitungan vektor: (1) = [1,0,1,0] dengan vektor (2) = [0,1,0,1] akan menghasilkan nilai 0, padahal kedua vektor tersebut jika dilihat memiliki kemiripan. Oleh karena itu, (Sidorov et al., 2014) mengajukan sebuah perhitungan similaritas yang menyederhanakan *cosine similarity* dengan memperhatikan kemiripan vektor dari setiap vektor yang ada di dalam *vector* space model untuk sebuah dokumen. *Soft Cosine Measure (SCM)* bekerja dengan cara menghitung nilai *cosine similarity* dari vektor dokumen non-orthogonal, dimana sudut antara dua vektor diturunkan dari sudut antara *Word2Vec embedding* dan kata yang sesuai.

2.2.7 *Confusion Matrix*

Untuk mengetahui bahwa hasil pengukuran similaritas sesuai dengan yang diharapkan atau tidak, akan digunakan *confusion matrix*. *Confusion matrix* adalah metode yang digunakan untuk mengevaluasi akurasi sebuah model machine learning (Melita et al., 2018). *Confusion matrix* memiliki empat bagian klasifikasi yakni, *true positive (TP), false positive(FP), false negative(FN),* dan *true negative(TN)*.

Bernilai *true positive* apabila prediksi berdasarkan ambang batas yang telah ditentukan sesuai dengan kondisi aktual. *False positive* apabila prediksi berdasarkan nilai ambang yang telah ditentukan, prediksi bernilai relevan sementara kondisi aktual tidak relevan. *False negative* apabila prediksi berdasarkan nilai ambang yang telah ditentukan menyatakan tidak relevan, s*eda*ngkan kondisi aktual relevan. *True negative* apabila prediksi berdasarkan nilai ambang menyatakan tidak relevan dan kondisi aktual menyatakan tidak relevan.

Nilai yang dihasilkan dari *confusion matrix* menunjukkan perbandingan hasil dari prediksi yang dihasilkan model dengan kondisi aktual, sehingga nilai tersebut dapat dimanfaatkan sebagai bahan untuk melakukan evaluasi dan penarikan kesimpulan terhadap pengukuran similaritas yang dilakukan.

BAB 3 METODOLOGI PENELITIAN

3.1 Objek Penelitian

Pada pelaksanaan penelitian ini, yang menjadi fokus penelitian adalah optimalisasi *text mining* dengan menggunakan *word embedding* dan pengukuran similaritas antar dokumen Undang-Undang dan Rancangan Undang-Undang (RUU). Optimalisasi *text mining* di dalam proses pembangunan legislasi menjadi hal penting mengingat pembangunan legislasi adalah suatu hal yang kompleks dan akan selalu berubah sering dengan perkembangan sebuah negara.

Objek penelitian ini dipilih mengingat penerapan sistem yang cukup baru di dalam legislasi di Indonesia, yakni penerapan sistem *omnibus law*. Penerapan *omnibus law* mengalami tantangan tersendiri, mengingat Indonesia menganut sistem hukum *civil law*. Sementara *omnibus law* serupa dengan sistem pembentukan legislasi yang lazim dilakukan oleh negara yang menganut sistem hukum *common law*. Sekilas, penerapan sistem *omnibus law* bertentangan dengan kondisi Indonesia yang merupakan negara yang menganut sistem hukum *civil law*, namun, tentu ada cara untuk mengharmonisasikan kedua hal tersebut.

Penerapan metode *text mining* bertujuan untuk mencari tingkat similaritas produk legislasi berupa Undang-Undang untuk mendapatkan harmonisasi antar produk Undang-Undang yang dihasilkan. Penerapan *Word2Vec* sebagai metode pembobotan kata sudah sangat luas digunakan dikarenakan kehandalannya, kemudian dipadukan dengan metode pengukuran *Soft Cosine Similarity* yang memiliki basis matematis pengukuran jarak *cosine* antar fitur dalam vektor. Dengan menemukan tingkat similaritas antar dokumen Undang-Undang, diharapkan akan membantu memberikan dokumen Undang-Undang yang relevan.

3.2 Waktu dan Tempat Penelitian

Implementasi analisis *text mining* terhadap Undang-Undang dan Rancangan Undang-Undang dilakukan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang beralamatkan di jalan Soemantri Brojonegoro No.1 Gedong Meneng, Bandar Lampung. Penelitian ini akan dilakukan pada semester ganjil tahun akademik 2021/2022.

3.3 Data dan Perangkat Penelitian

Data dan perangkat yang digunakan dalam penelitian yang digunakan dijabarkan sebagai berikut :

3.3.1 Data

Data yang digunakan terdiri atas Undang-Undang (UU) dan Rancangan Undang-Undang (RUU). Data Undang-Undang yang digunakan adalah Undang-Undang dalam rentang tahun 1945-2020, yang sekaligus akan digunakan sebagai data latih untuk membangun model *Word2Vec*. Data uji terdiri data uji kueri dan data uji indeks pembanding. Data uji kueri terdiri dari Rancangan Undang-Undang (RUU) Tentang Layanan Publik dan Rancangan Undang-Undang (RUU) Tentang Pertambangan. Data uji indeks pembanding terdiri dari himpunan 15 dokumen Undang-Undang (UU). Data uji diberikan oleh mitra penelitian yang berasal dari Fakultas Hukum Universitas Lampung.

3.3.2 Perangkat Penelitian

Perangkat penelitian yang dimaksud adalah perangkat lunak yang berguna untuk menunjang pelaksanaan penelitian. Perangkat penelitian tersebut yaitu :

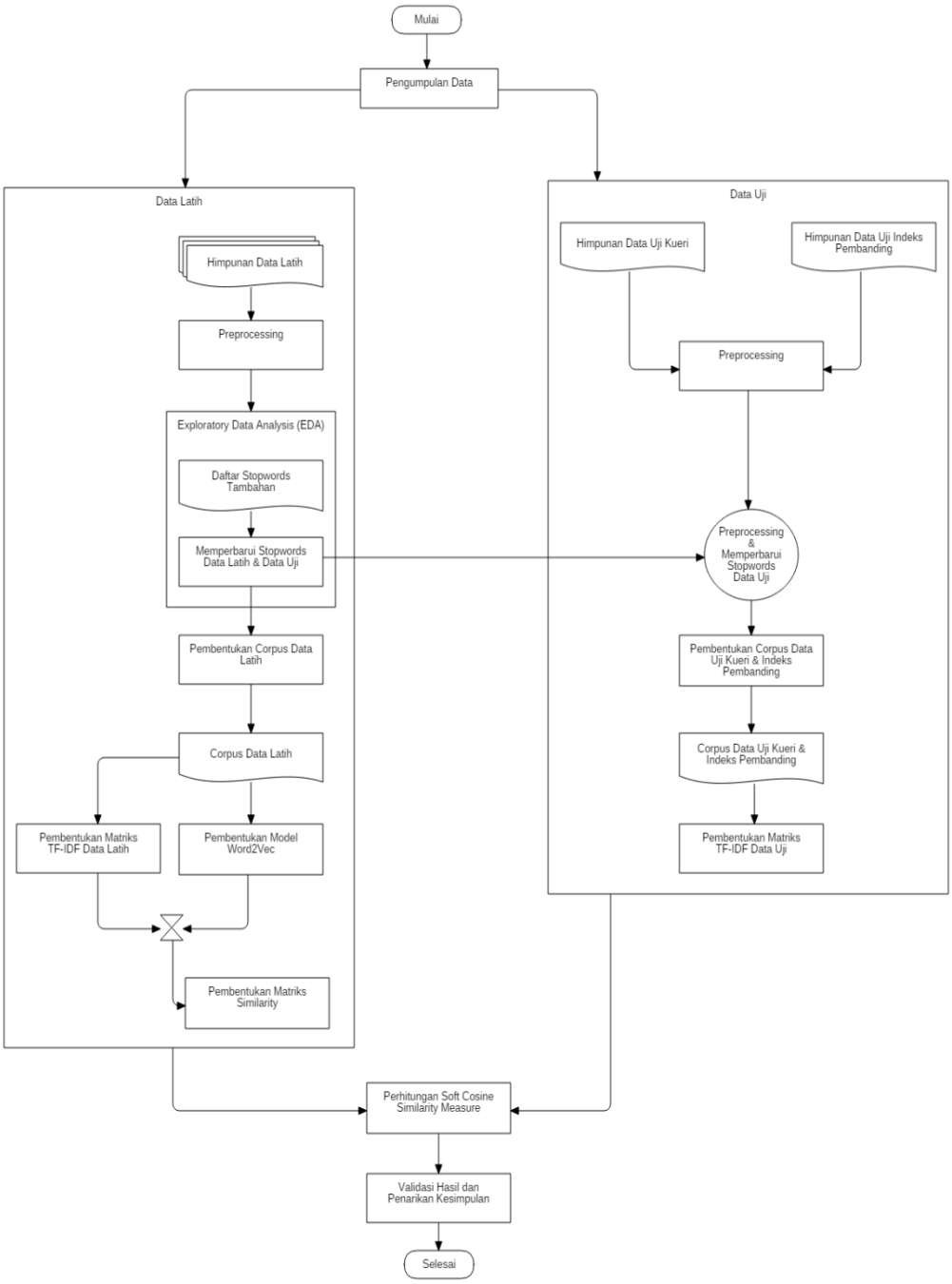
1. *Python 3.9.2.*
2. *Jupyter Notebook* versi 6.0.4.
3. *Library Gensim* versi 4.0.1.
4. *Library Natural language toolkit (NLTK)* versi 3.5.
5. *Library Scikit Learn* versi 0.23.2.
6. *Library Tika-Python* versi 2.14.

3.4 Desain Penelitian

Metode penelitian yang dilakukan di dalam penelitian menggunakan metode eksperimen kuantitatif. Pertama, penelitian berjenis eksperimen kuantitatif di pilih untuk mencapai tujuan penelitian yang membutuhkan pembentukan relasi semantik melalui model *Word2Vec* yang dihasilkan dengan menggunakan himpunan dokumen Undang-Undang dari tahun 1945 sampai dengan tahun 2020. Kedua, metode penelitian eksperimen digunakan karena di dalam penelitian ini peneliti akan melakukan pengukuran similaritas menggunakan kombinasi model *Word2Vec* dan *soft cosine similarity measure*.

Secara garis besar akan dilakukan tahapan penelitian terhadap dua jenis data yang akan digunakan dalam penelitian ini yakni data latih dan data uji. Terkhusus dalam tahap *exploratory data analysis* hanya akan dilakukan untuk data latih, hal ini mengingat data latih yang merupakan Undang-Undang dalam rentang tahun 1945-2020 merupakan representasi yang baik untuk mengetahui pola kebahasaan Undang-Undang atau Rancangan Undang-Undang.

Berikut akan disajikan diagram alur penelitian pada Gambar 3.1. Penjabaran lebih lanjut mengenai desain penelitian akan dijabarkan dalam sub-bab yang terdapat di dalam sub-bab 3.4.



Gambar 3.1 Diagram Alir Penelitian

Gambar 3.1 merupakan diagram alir penelitian yang menjadi acuan penelitian.Berikut akan dijabarkan secara mendetail mengenai alur penelitian yang akan di tempuh dalam penelitian ini.

3.4.1 Pengumpulan dan Pengelompokkan Data

Tahap pengumpulan dan pengelompokkan data dalam penelitian ini dijabarkan sebagai berikut :

3.4.1.1 Pengumpulan dan Pengelompokkan Data Latih

Data latih adalah data yang akan digunakan sebagai input dalam membentuk model *Word2Vec* yang digunakan untuk pembobotan kata pada setiap dokumen. Dalam penelitian ini yang digunakan sebagai data latih adalah dokumen Undang-Undang (UU) dalam rentang tahun 1945-2020, tanpa memperhatikan status berlaku Undang-Undang tersebut. Tabel 3.1 akan menyajikan jumlah data Undang-Undang dalam rentang tahun 1945-2020 yang akan digunakan untuk membentuk model *Word2Vec*.

Tabel 3.1 Jumlah Data Latih

No	Tahun	Jumlah	No	Tahun	Jumlah
1	1945	1	39	1983	8
2	1946	23	40	1984	8
3	1947	41	41	1985	17
4	1948	35	42	1986	5
5	1949	12	43	1987	8
6	1950	14	44	1988	6
7	1951	24	45	1989	7
8	1952	24	46	1990	11
9	1953	37	47	1991	7
10	1954	76	48	1992	25
11	1955	12	49	1993	7
12	1956	35	50	1994	12
13	1957	80	51	1995	13
14	1958	87	52	1996	9
15	1959	29	53	1997	32
16	1960	9	54	1998	13
17	1961	22	55	1999	56
18	1962	19	56	2000	38
19	1963	14	57	2001	22

Tabel 3.1 (lanjutan)

No	Tahun	Jumlah	No	Tahun	Jumlah
20	1964	36	58	2002	32
21	1965	22	59	2003	41
22	1966	14	60	2004	41
23	1967	14	61	2005	14
24	1968	25	62	2006	23
25	1969	16	63	2007	46
26	1970	14	64	2008	56
27	1971	13	65	2009	53
28	1972	6	66	2010	13
29	1973	7	67	2011	24
30	1974	11	68	2012	24
31	1975	5	69	2013	24
32	1976	11	70	2014	42
33	1977	2	71	2015	13
34	1978	8	72	2016	20
35	1979	5	73	2017	17
36	1980	13	74	2018	13
37	1981	8	75	2019	24
38	1982	21	76	2020	13
Total Jumlah Data Latih					1682

Tabel 3.1 berisi jumlah dokumen Undang-Undang dalam rentang tahun 1945-2020. Dokumen Undang-Undang ini akan digunakan sebagai data latih. Penamaan dokumen dalam data latih akan mengikuti fomat penamaan (Tahun-Nomor UU). Penamaan dokumen sesuai format yang ditentukan akan mempermudah pengelolaan data dalam tahapan selanjutnya.

3.4.1.2 Pengumpulan dan Pengelompokkan Data Uji

Pada penelitian ini yang menjadi data uji adalah dokumen Undang-Undang (UU) dan Rancangan Undang-Undang (RUU) yang diberikan oleh mitra penelitian yang berasal dari Fakultas Hukum Universitas Lampung.Dalam hal ini, data uji akan terbagi menjadi dua. Pertama data uji yang akan digunakan sebagai data kueri. Kedua, data uji yang akan digunakan sebagai data indeks pembanding dari data kueri.

Pada Tabel 3.2 akan disajikan data uji yang akan digunakan sebagai data kueri.

Tabel 3.2 Data Uji Kueri

No	Nama Data Kueri	Tentang
1	180329_RUU_Pertambangan	Pertambangan,Mineral,Batubara
2	210615_RUU_PelayananPublik	Pelayanan Publik

Tabel 3.2 menyajikan data uji yang digunakan sebagai data uji kueri. Data yang tercantum dalam Tabel 3.2 akan menjadi data uji kueri yang akan dibandingkan dengan data uji indeks pembanding. Selain data uji kueri, terdapat data uji yang digunakan sebagai data indeks pembanding. Tabel 3.3 akan menyajikan daftar uji yang digunakan sebagai data indeks pembanding.

Tabel 3.3 Data Uji Indeks Pembanding

No	Nama Data Indeks Pembanding	Tentang
1	2008-14	Keterbukaan Informasi Publik
2	2008-39	Kementerian Negara
3	2009-25	Pelayanan Publik
4	2009-36	Kesehatan
5	2009-38	POS
6	2009-39	Kawasan Ekonomi Khusus
7	2011-4	Informasi Geospasial
8	2014-23	Pemerintahan Daerah
9	2014-30	Administrasi Pemerintahan
10	2017-1	Pengesahan Perjanjian Batas Laut RI dan Singapura
11	2019-2	Pengesahan Nota Kesepahaman Kemenhan RI dan Spanyol
12	2019-3	Pengesahan Nota Kesepahaman Pertahanan RI dan Serbia
13	2020-1	Pengesahan Kemitraan Ekonomi Komprehensif RI dan Australia
14	2020-11	Cipta Kerja
15	2020-4	Pengesahan Nota Kesepahaman Pertahanan RI dan Ukraina

Data pada Tabel 3.3 akan digunakan sebagai data yang digunakan sebagai data indeks yang dibandingkan dengan data uji kueri untuk mendapatkan similaritas.

3.4.2 *Preprocessing* Data

Setelah proses pengelompokkan dan pengumpulan data, langkah selanjutnya adalah tahap *preprocessing* data untuk untuk mengolah data yang telah dihimpun menjadi data siap proses untuk pembentukan model *Word2Vec* dan pengukuran similaritas. Penjabaran lebih lanjut tahap *preprocessing* dijabarkan sebagai berikut :

3.4.2.1 *Extraction*

Langkah *preprocessing* awal yang akan dilakukan adalah tahap *extraction*. Tahap *extraction* adalah tahapan untuk mengektraksi data teks di dalam dokumen dengan format *portable document file (.pdf)*. Tahap *extraction* akan dilakukan untuk data latih yang berjumlah 1682 dokumen maupun data uji menggunakan bantuan *library Tika python*.

3.4.2.2 *Case Folding*

Setelah tahap ekstraksi data dilakukan, data latih dan data uji akan berisi karakter huruf yang memiliki beragam bentuk. Untuk meminimalisir model *word embedding Word2Vec* mendeteksi perbedaan *uppercase* dengan *lowercase* memiliki bobot yang berbeda, langkah *case folding* dilakukan untuk membuat karakter kata menjadi seragam yaitu dalam bentuk karakter huruf kecil *(lowercase)*. Ilustrasi *Case folding* disajikan pada Tabel 3.4.

Tabel 3.4 Ilustrasi Tahap *Case folding*

Input	***Output Case folding***
Mengesahkan Persetujuan antara Pemerintah Republik Indonesia dan Pemerintah Kerajaan Swedia	mengesahkan persetujuan antara pemerintah republik indonesia dan pemerintah kerajaan swedia

Tabel 3.4 menyajikan ilustrasi proses *case folding* untuk menyeragamkan semua karakter huruf yang beragam menjadi karakter *lowercase*.

3.4.2.3 Filtering

Proses *filtering* di lakukan untuk menghilangkan karakter yang tidak dapat digunakan dalam tahap pengolahan data teks lebih lanjut. Karakter yang akan dihilangkan dalam tahap *filtering* antara lain, karakter numerik, karakter pungtuasi, karakter non-fungsional dari tahap ekstraksi, karakter *American Standard Code for Information Interchange* (*ASCII*) yang bersifat non-cetak atau tidak dapat tercetak. Selain itu, dalam tahap *filtering* akan dilakukan seleksi terhadap karakter yang memiliki panjang kurang dari tiga karakter, dan kata dengan frekuensi kemunculan kurang dari lima dalam setiap dokumen. Hal ini dilakukan untuk menghindari karakter yang tidak terekstraksi secara sempurna dan kata tipo yang terdapat dalam setiap dokumen.

Tabel 3.5 Ilustrasi *Filtering*

Input	***Output Filtering***
dalam kehidupan bernegara, aspek pertahanan, merupakan salah satu faktor	dalam kehidupan bernegara aspek pertahanan merupakan salah satu faktor

Tabel 3.5 menyajikan ilustrasi *Filtering* yang merupakan proses untuk menghilangkan karakter yang tidak dapat digunakan lebih lanjut dalam tahap pengolahan data teks.

3.4.2.4 Stopwords Elimination

Data teks yang sudah melewati tahap *case folding* dan *filtering* masih menyimpan banyak kata yang terlalu umum *(stopwords)*, yang tidak memiliki makna dalam keseluruhan dokumen. Dalam bahasa Indonesia kata hubung seperti “di”, “ke”, ”dan”,”dari”, termasuk ke dalam *stopwords*. *Stopwords* tidak memiliki makna yang signifikan di dalam hubungan semantik. Dikarenakan dampak yang tidak signifikan tersebut, *stopwords* dalam bahasa Indonesia perlu dihilangkan untuk efisiensi daya komputasi dan meningkatkan hasil *word embedding* untuk mendapatkan relasi semantik antar kata-kata yang memang memiliki dampak yang signifikan.

Tahap *stopwords elimination* akan dilakukan dengan bantuan *libray natural language toolkit (NLTK)*, yang memiliki koleksi *stopwords* dalam bahasa Indonesia. Ilustrasi *stopwords elimination* disajikan dalam Tabel 3.6.

Tabel 3.6 Ilustrasi *Stopwords Elimination*

Input	***Output Stopwords Elimination***
batubara adalah endapan senyawa organik karbonan yang terbentuk secara alamiah dari sisa tumbuh tumbuhan	batubara endapan senyawa organik karbonan terbentuk alamiah tumbuh tumbuhan

Tabel 3.6 menyajikan ilustrasi tahap *stopwords elimination* yang menghilangkan kata yang termasuk dalam daftar *stopwords*.

3.4.3 Exploratory Data Analysis (EDA)

Data teks yang telah melewati tahapan *stopwords elimination* kemudian akan melalui tahap *exploratory data analysis* (EDA), yakni tahap yang dilakukan untuk mengeksplorasi data teks untuk menemukan fakta dari data teks yang digunakan. Tahap EDA hanya dilakukan untuk data latih yang terdiri dari Undang-Undang dalam rentang tahun 1945-2020. Tahap EDA akan menggunakan format *document-term matrix* (*DTM*), oleh karena itu data latih yang sebelumnya dalam format dataframe akan di konversi terlebih dahulu ke dalam format DTM. Terdapat empat langkah yang akan dilakukan dalam tahap EDA.

Pertama, definisi permasalahan. Tujuan utama dari tahap EDA yang dilakukan pada penelitian ini adalah untuk menghasilkan daftar *stopwords* tambahan dan melakukan visualisasi terhadap dokumen Undang-Undang dalam rentang tahun 1945-2020. Kedua, persiapan data. Pada penelitian ini data yang akan digunakan adalah data dalam format *document-term matrix (dtm)* dari Undang-Undang dalam rentang 1945-2020 yang telah melewati tahap *preprocessing* yang sebelumnya tersimpan dalam format *dataframe*.

Ketiga, melakukan analisis data. Proses analisis data teks akan dilakukan dengan cara melakukan penghitungan terhadap frekuensi kemunculan setiap kata di dalam *document-term matrix*. Kemudian, melakukan klasifikasi terhadap kata dengan jumlah kemunculan lebih dari nilai ambang yakni 14% untuk dimasukkan ke dalam daftar *stopwords* tambahan. Daftar *stopwords* tambahan digunakan untuk memperbarui data latih dan data uji. Tahap keempat, melakukan visualisasi terhadap *document-term matrix* data latih yang sudah diperbarui dengan cara menghilangkan kata yang terdapat didalam daftar *stopwords* tambahan. Visualisasi setiap dokumen dalam data latih akan menggunakan *wordcloud*.

3.4.4 Pembentukan *Corpus* Data Latih,Data Uji Kueri & Data Uji Pembanding

Setelah diperbarui dengan hasil yang didapatkan pada tahap EDA, data latih dan data uji hanya terdiri dari koleksi kata *unigram* (satu suku kata). Dalam hal ini, *unigram* tidak dapat menjadi representasi makna dalam bentuk frasa yang memuat dua atau lebih suku kata. Tahap pembentukan *corpus* dilakukan untuk menambahkan *N-gram* ke dalam data latih dan data uji. *N-gram* yang dimaksudkan dalam tahap ini dibatasi untuk dua suku kata (*bigram*). Untuk melakukan pembentukan *corpus* data latih dan data uji akan menggunakan modul *phrase* dari *library* gensim, dengan nilai ambang untuk sebuah *bigram* dapat dimasukkan ke dalam corpus data latih dan data uji adalah sebanyak sepuluh kali kemunculan dalam total dokumen data latih dan data uji.

3.4.5 Pembentukan Matriks *TF-IDF* Untuk Data Latih, Data Uji Kueri & Data Uji Pembanding

Corpus data latih dan data uji yang sudah terbentuk, kemudian akan digunakan untuk membentuk matriks *term frequency-inverse document matrix (TF-IDF)*. *TF-IDF* digunakan untuk memberikan bobot setiap kata berdasarkan nilai *inverse-document matrix* yang didapatkan berdasarkan frekuensi kemunculan kata untuk setiap dokumen dalam data latih dan data uji. Matriks *TF-IDF* data latih dan data uji akan digunakan untuk tahap pembentukan matriks *similarity*, dimana setiap

indeks kata dan bobot *TF-IDF* akan digunakan untuk pemetaan nilai vektor untuk setiap kata yang dihasilkan oleh model *Word2Vec*.

Untuk meningkatkan akurasi pengukuran similaritas antara data uji dan data latih, matriks *TF-IDF* akan disesuaikan dengan karakteristik data latih dan data uji, serta normalisasi berdasarkan (Manning et al., 2009; Salton & Buckley, 1988; Singhal et al., 1995). Tahap pembentukan matriks *TF-IDF* untuk data latih dan data uji akan menggunakan modul *TF-IDF* pada *library gensim*. Konfigurasi normalisasi matriks *TF-IDF* akan disajikan dalam Tabel 3.6 untuk kemudian akan dijabarkan alasan pemilihan konfigurasi tersebut.

Tabel 3.7 Konfigurasi Pembentukan Matriks *TF-IDF*

No	Kode Konfigurasi	Nama Konfigurasi
1	*l*	*Sub-linear logarithm*
2	*f*	*Inverse document matrix*
3	*b*	*Pivoted document-lenght normalization*

Tabel 3.7 menyajikan konfigurasi yang akan digunakan dalam tahap pembentukan matriks *TF-IDF* untuk data latih dan data uji.

Konfigurasi pertama, *sub-linear logarithm* (*l*) dipilih untuk dapat menangkap signifikansi kemunculan kata dari jumlah kemunculan kata tersebut di dalam sebuah dokumen. Konfigurasi kedua *inverse-document matrix* dipilih untuk memberi bobot pada setiap kata berdasarkan jumlah kemunculan kata tersebut dalam sebuah dokumen, sehingga kata yang umum muncul dalam sebuah dokumen memiliki bobot yang lebih rendah dan memiliki kemungkinan yang kecil untuk diambil sebagai informasi yang penting.

Konfigurasi ketiga, *pivoted document-length normalization (b)* merupakan konfigurasi yang digunakan untuk menjalankan normalisasi berdasarkan panjang atau ukuran setiap dokumen yang ada di dalam data latih dan data uji. Hal ini dilakukan untuk meminimalisir dampak dari kata yang tidak terekstraksi dengan baik akibat keterbatasan *library* yang digunakan dalam tahap *extraction*.

3.4.6 Pembentukan Model *Word2Vec*

Tahap pembentukan model *Word2Vec* akan dilakukan menggunakan data latih. Hasil dari pembentukan model *Word2Vec* adalah bobot vektor untuk setiap kata di dalam dokumen data latih. Pembentukan model *Word2Vec* akan dilakukan dengan menggunakan konfigurasi nilai *hyper-parameter* yang disajikan dalam Tabel 3.7, yang kemudian akan dijabarkan lebih lanjut alasan pemilihan nilai *hyper-parameter* tersebut.

Tabel 3.8 Konfigurasi Awal Pembentukan Model *Word2Vec*

No	Jenis *Hyper-parameter*	Besar Nilai
1	*Vector Size*	30
2	*Window Size*	3
3	*Skip-gram*	*True*
4	*Negative sampling*	5
5	*Ns_exponent*	0.75
6	*Sample*	0.1e-5
7	*Epochs*	90-200

Tabel 3.8 menyajikan konfigurasi jenis *hyper-parameter* dan besaran nilai yang akan digunakan untuk membentuk model *Word2Vec* menggunakan data latih.

Konfigurasi satu, *vector size* sebesar 30 dipilih untuk menyesuaikan pembangunan model *Word2Vec* yang menggunakan metode *skip-gram negative sampling*, yang mendapatkan hasil akurasi yang lebih baik apabila menggunakan ukuran vektor yang kecil (Mikolov, Sutskever, et al., 2013). Konfigurasi dua, *window size* sebesar tiga, dipilih berdasarkan hasil penelitian yang dilakukan untuk meneliti pengaruh pemilihan *hyper-parameter* untuk pembangunan model rekomendasi musik yang akurat oleh (*Case*lles-Dupré et al., 2018).

Konfigurasi tiga, merupakan jenis model *Word2Vec* yang akan digunakan, yakni model *skip-gram* yang memiliki akurasi lebih baik dibandingkan dengan model CBOW (Mikolov, Chen, et al., 2013). Konfigurasi empat, *negative sampling* sebesar 5, hal ini berfungsi untuk melakukan pemberian nilai negatif secara acak

terhadap kata yang ada dalam dokumen. Konfigurasi lima, *ns_exponent* sebesar 0.75, berfungsi untuk membentuk distribusi persebaran kata negatif untuk frekuensi kata dengan jumlah kemunculan lebih sedikit dominan dibandingkan dengan kata dengan jumlah frekunsi kemunculan lebih banyak (Mikolov, Sutskever, et al., 2013).

Konfigurasi enam, *sample* sebesar 0,1e-5 berfungsi untuk melakukan *subsampling* terhadap kata dengan frekuensi kemunculan tinggi (Mikolov, Sutskever, et al., 2013) Konfigurasi tujuh *epochs*, merupakan jumlah iterasi yang dilakukan untuk membentuk sebuah model *Word2Vec*. Jumlah iterasi yang digunakan akan mempengaruhi performa *Word2Vec* dalam membentuk relasi semantik antar kata yang terdapat di dalam dokumen (Di Gennaro et al., 2021). Model *Word2Vec* yang sudah dibentuk, akan dijadikan input untuk pembentukan matriks *sparse term similarity*.

3.4.7 Pembentukan Matriks *Similarity*

Matriks *similarity* digunakan untuk menyimpan indeks kata dan bobot untuk setiap kata. Pembentukan matriks *similarity* hanya dilakukan terhadap data latih. Hal ini dilakukan karena data latih digunakan sebagai input pembentukan model *Word2Vec* yang menghasilkan bobot vektor untuk setiap kata. Matriks *similarity* yang dibentuk menggunakan data latih akan menjadi referensi dalam setiap pengukuran similaritas antar dua buah *corpus* yang akan dilakukan.

Matriks *similarity* dibentuk menggunakan tiga komponen. Pertama, *dictionary* dari data latih yang terdiri atas Undang-Undang dalam rentang tahun 1945-2020. Kedua, bobot *term frequency-inverse document frequency* untuk setiap kata dalam kata latih. Ketiga, bobot vektor untuk setiap kata dalam data latih.

Urutan kolom akan berdasarkan bobot TF-IDF untuk setiap kata dalam data latih, dan baris akan berdasarkan dictionary data latih. Untuk nilai setiap sel pada matriks menggunakan bobot vektor dari model *Word2Vec* untuk setiap kata dalam data latih. Ilustrasi bentuk matriks *similarity* ditunjukkan pada Tabel 3.9.

Tabel 3.9 Ilustrasi Matriks *Similarity*

	1	2	3	4	5	6	7	8	Dictionary
0	0	0	0	0	0	0	0	0	
1	0	0,57	0	0	0	0	0	0	
2	0,87	0	0	0	0	0	0	0,37	
3	0	0	0,74	0	0	0	0	0	
4	0	0	0	0	0	0,34	0	0	
5	0	0,37	0	0	0	0,1	0	0	
6	0	0	0	0	0,33	0	0	0	
7	0	0	0	0,57	0	0	0	0	
8	0,68	0	0	0,43	0	0	0	0	
TF-IDF									

Tabel 3.9 menyajikan ilustrasi matriks *similarity* yang akan digunakan sebagai referensi untuk setiap skenario pengukuran yang akan dilakukan untuk dua buah corpus yang akan dibandingkan.

3.4.8 Pengukuran Similaritas

Pengukuran similaritas akan dilakukan untuk data uji kueri. Dalam hal ini, pengukuran similaritas dilakukan dalam dua skenario pengukuran. Skenario pengukuran pertama, dokumen 210615_RUU_PelayananPublik yang termasuk dalam data uji kueri, akan dibandingkan dengan 15 data uji indeks pembanding.

Nilai ambang yang ditetapkan untuk skenario pengukuran pertama menggunakan standar umum untuk penentuan relevansi informasi sebesar 50% (Manning et al., 2009). Apabila nilai pengukuran similaritas diatas 50% maka dokumen dianggap relevan, apabila nilai pengukuran dibawah 50% dokumen dianggap tidak relevan. Kemudian hasil pengukuran similaritas yang didapat, di validasi oleh ahli hukum apakah hasil pengukuran tersebut relevan atau tidak relevan. Skenario pengukuran kedua, dokumen 180329_RUU_Pertambangan yang termasuk dalam data uji kueri akan dibandingkan data latih. Akan diambil 15 dokumen dengan nilai pengukuran similaritas tertinggi.

Apabila setiap nilai pengukuran similaritas 15 dokumen tertinggi diatas 50%, maka akan di hitung rata-rata harmonik nilai 15 data tersebut, untuk kemudian dijadikan nilai ambang menentukan relevansi pengukuran oleh sistem.

Apabila lebih dari rata-rata harmonik yang didapat dokumen dianggap relevan, apabila dibawah rata-rata harmonik yang didapat dokumen dianggap tidak relevan. Kemudian, hasil pengukuran similaritas yang didapat, di validasi oleh ahli hukum apakah hasil pengukuran tersebut relevan atau tidak relevan. Rata-rata harmonik dipilih sebagai nilai ambang karena rata-rata harmonik mampu memberikan gambaran nilai tengah sebuah sampel data dengan adil tanpa terpengaruh oleh data pencilan. Sebagai catatan, apabila terdapat tujuh dokumen dengan nilai similaritas dibawah 50%, maka metode penentuan nilai ambang untuk skenario pengukuran kedua akan disamakan dengan skenario pengukuran pertama.

3.4.9 Validasi Hasil

Tahap validasi hasil pada penelitian ini dilakukan dengan melakukan perbandingan hasil pengukuran similaritas dengan penilaian oleh ahli hukum. Tahap validasi di lakukan untuk kedua skenario pengukuran similaritas mengunakan nilai ambang relevansi yang telah ditentukan untuk masing-masing skenario. Hasil yang akan didapatkan dalam tahap validasi hasil adalah *confusion matrix* yang akan digunakan lebih lanjut sebagai bahan penarikan kesimpulan pengukuran similaritas yang telah dilakukan. Tabel 3.9 akan menyajikan ilustrasi *confusion matrix* yang akan didapatkan, untuk kemudian dijabarkan lebih lanjut.

Tabel 3.10 Ilustrasi *Confusion Matrix* Validasi Hasil

		Nilai Similaritas *Word2Vec*	
		Relevan	*Non-Relevan*
Penilaian Ahli Hukum	*Relevan*	TP	FP
	Non-Relevan	FN	TN

Tabel 3.10 merupakan ilustrasi *confusion matrix* yang dihasilkan dengan membandingkan hasil pengukuran similaritas yang didapat dengan penilaian oleh mitra penelitian dari Fakultas Hukum Universitas Lampung.

Setelah dilakukan perbandingan antara hasil pengukuran similaritas dengan penilaian ahli hukum, akan didapatkan empat klasifikasi hasil pengukuran similaritas. *True positive* (TP), apabila pengukuran similaritas berdasarkan nilai ambang menyatakan bahwa sebuah dokumen relevan, dan penilaian ahli menyatakan bahwa dokumen tersebut relevan dengan data uji kueri. *False positive* (FP) apabila hasil pengukuran similaritas berdasarkan nilai ambang menyatakan bahwa dokumen tersebut tidak relevan, namun penilaian ahli menyatakan bahwa dokumen tersebut relevan. *False negative* (FN) apabila hasil pengukuran similaritas berdasarkan nilai ambang menyatakan bahwa dokumen tersebut relevan, namun penilaian ahli menyatakan bahwa dokumen tersebut tidak relevan. *True negative* (TN) apabila hasil pengukuran similaritas berdasarkan nilai ambang menyatakan sebuah dokumen tidak relevan, namun penilaian ahli menyatakan bahwa dokumen tersebut relevan.

Hasil dari tahap validasi hasil adalah satu *confusion matrix* untuk setiap skenario pengukuran similaritas. *Confusion matrix* yang di dapat, akan digunakan sebagai dasar untuk penarikan kesimpulan.

3.4.10 Penarikan Kesimpulan

Tahap penarikan kesimpulan dilakukan dengan menggunakan empat metriks evaluasi. Penarikan kesimpulan dilakukan untuk dua skenario pengukuran similaritas, yang dimaksudkan untuk mengetahui apakah ada perbedaan hasil penarikan kesimpulan dalam kedua skenario pengukuran similaritas.

Metriks pengukuran yang akan digunakan adalah *precision*, *recall*, *F-β measure*, dan *accuracy* (Rijsbergen, 1979; Vanderbeck et al., 2011). *Precision* adalah nilai yang menunjukkan berapa banyak dokumen yang dianggap relevan, adalah benar dokumen yang relevan.

Recall adalah nilai yang menunjukkan jumlah dokumen relevan berdasarkan penilaian ahli yang dianggap relevan berdasarkan pengukuran similaritas. *Accuracy* menunjukkan nilai pembagian semua prediksi yang benar dengan semua nilai prediksi yang diberikan. Nilai *accuracy* tidak dapat menunjukkan peran *precision* dan *recall* dalam menentukan performa pengukuran similaritas yang dilakukan, oleh karena itu dibutuhkan metriks pengukuran lain yakni *F-β measure* yang merupakan rata-rata harmonik antara *precision* dan *recall*.

Persamaan (3.1) menunjukkan rumus menghitung tingkat *Precision*.

$$Precision = \frac{(TP)}{(TP + FP)} \tag{3.1}$$

Keterangan :

TP = *true positive*

FP = *false positive*

Persamaan 3.1 *Precision*

Persamaan (3.2) menunjukkan rumus menghitung *Recall*.

$$Recall = \frac{TP}{(TP + FN)} \tag{3.2}$$

Keterangan :

TP = *true positive*

FN = *false negative*

Persamaan 3.2 *Recal*

Persamaan (3.3) menunjukkan persamaan untuk menghitung *Accuracy*.

$$Accuracy = \frac{TP + TN}{(TP + FP + TN + FN)} \tag{3.3}$$

Keterangan :

TP = *true positive*

TN = *true negative*

FP = *false positive*

FN = *false negative*

Persamaan 3.3 *Accuracy*

Persamaan (3.4) menujukkkan rumus menghitung *F-β measure.*

$$F_{\beta=1} = \frac{(\beta^2+1)PR}{\beta^2\ P+R} = \frac{2PR}{(P+R)}; P = \frac{(TP)}{(TP+FP)}; \quad (3.4)$$

$$R = \frac{TP}{(TP+FN)}$$

Keterangan :

β = nilai beta (beta=1)

P = precision

R = *Recall*

TP = *true positive*

TN = *true negative*

FP = *false positive*

FN = *false negative*

Persamaan 3.4 Perhitungan *F-β measure*

BAB 5 SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil analisis dan pembahasan dalam bab empat mengenai pengukuran similaritas dokumen yang dilakukan terhadap Rancangan Undang-Undang Pelayanan Publik dan Rancangan Undang-Undang Pertambangan dalam dua skenario pengukuran. Didapatkan simpulan sebagai berikut:

1. Dokumen Undang-Undang (UU) tidak dapat diperlakukan sebagaimana dokumen *for*mal lain, hal ini di karenakan pengkategorian Undang-Undang tidak selalu selaras dengan isi Undang-Undang tersebut.
2. Tahap *preprocessing* dan *exploratory data analysis* (EDA) dokumen Undang-Undang (UU) dan Rancangan Undang-Undang (RUU) penting untuk dilakukan. Hal ini disebabkan karena dalam dokumen UU dan RUU masih terdapat banyak kata yang terlalu umum (*stopwords*) yang perlu dihilangkan karena dapat mempengaruhi pembobotan kata yang ada di dalam dokumen.
3. Dibutuhkan pemahaman yang cukup mengenai struktur UU dan RUU. Dikarenakan hal ini penting untuk proses analisis lebih lanjut.
4. Pemilihan *hyper-parameter* dan jenis model dalam pembentukan model *Word2Vec* penting untuk dilakukan untuk meningkatkan performa pembobotan kata.
5. Dalam proses pengukuran similaritas menggunakan *soft cosine similarity measure*, membutuhkan dua komponen utama. Pertama, model Wor2Vec. Kedua, matriks vektor *term frequency-inverse document matrix* untuk setiap *corpus* yang akan dibandingkan.
6. Penting untuk melakukan normalisasi terhadap setiap matriks vektor *TF-IDF* yang dibentuk untuk menghindari bias yang diakibatkan oleh perbedaan ukuran dokumen dalam setiap *corpus*.

7. Hasil pengukuran skenario pertama menggunakan dokumen RUU Pelayanan Publik dan dokumen Undang-Undang yang terlebih dahulu sudah melalui proses kurasi oleh ahli hukum mendapatkan nilai *precision* sebesar 67%, *recall* sebesar 80%, *F-β measure* sebesar 73%, dan *accuracy* sebesar 80%.
8. Hasil pengukuran skenario kedua menggunakan dokumen RUU Pertambangan dan 1681 dokumen Data latih mendapatkan nilai *precision* sebesar 67%, *recall* sebesar 67%, *F-β measure* sebesar 67%, dan *accuracy* sebesar 73%.
9. Perbedaan performa pengukuran similaritas pada skenario pertama dan skenario kedua disebabkan oleh perbedaan ukuran *corpus* yang dibandingkan. Hal ini di validasi dan disajikan dalam Tabel 4.21.
10. Pengukuran similaritas menggunakan kombinasi model *Word2Vec*, *TF-IDF*,dan metode pengukuran *soft cosine similarity* disimpulkan reliabel berdasarkan hasil evaluasi untuk pengukuran similaritas skenario pertama dan skenario kedua.

5.2 Saran

Berikut saran yang dapat diberikan untuk penelitian pengukuran similaritas antar dokumen UU atau RUU yang menggunakan basis *corpus* dan pembobotan kata :

1. Penelitian ini dapat dikembangkan lebih lanjut dengan menggunakan tahapan ekstraksi struktur dokumen UU dan RUU yang lebih mendalam menggunakan knowledge-based *extraction*.
2. Untuk metode pembobotan kata atau dokumen yang digunakan dapat menggunakan algoritma lain seperti *Doc2Vec* dan *Fast-Text*.
3. Untuk mengembangkan penelitian ini lebih lanjut, dapat dilakukan untuk merubah representasi dokumen yang pada penelitian ini berupa keseluruhan dokumen, menjadi representasi berdasarkan struktur dokumen UU dan RUU. Semisal Bab, Pasal, dan Ayat

DAFTAR PUSTAKA

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

Black, H. C., & Frey, A. H. (1934). Black's Law Dictionary. *University of Pennsylvania Law Review and American Law Register*, *82*(8), 886. https://doi.org/10.2307/3308065

Caselles-Dupré, H., Lesaint, F., & Royo-Letelier, J. (2018). Word2vec applied to Recommendation: Hyperparameters matter. *RecSys 2018 - 12th ACM Conference on Recommender Systems*, 352–356. https://doi.org/10.1145/3240323.3240377

Di Gennaro, G., Buonanno, A., & Palmieri, F. A. N. (2021). Considerations about learning Word2Vec. *Journal of Supercomputing*, *March*. https://doi.org/10.1007/s11227-021-03743-2

Hantoro, N. M., Harefa, M., Alhusain, A. S., Retnaningsih, H., Nainggolan, P. P., & Prayudi. (2020). *Parliamentary Review : Omnibus law* (Vol. 2, Issue 1). Pusat Penelitian Badan Keahlian DPR RI. https://berkas.dpr.go.id/puslit/files/parliamentary_review/Parliamentary Review-II-1-M-2020.pdf

Hartono, D., & Hardiwinoto, S. (2018). Legal Perspective on Asean Economic Community. *Diponegoro Law Review*, *3*(2), 199. https://doi.org/10.14710/dilrev.3.2.2018.199-222

Indurkhya, N., & Damerau, F. J. (2010). Handbook of Natural Language Processing : Second Edition. In *Handbook of Natural Language Processing*. https://doi.org/10.1201/9781420085938

Jones, K. S. (1972). *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*.

Kulkarni, A., & Shivananda, A. (2019). *Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python*.

Kumar, S., Reddy, P. K., Reddy, V. B., & Singh, A. (2011). Similarity analysis of legal judgments. *Compute 2011 - 4th Annual ACM Bangalore Conference*, 3–6. https://doi.org/10.1145/1980422.1980439

Kumar, S., Reddy, P. K., Reddy, V. B., & Suri, M. (2013). *under Common Law System*. 103–116.

Luhn, H. P. (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development*, *October*, 309–317. https://doi.org/10.1147/rd.14.0309

Malau, M. T. M. (2014). Aspek Hukum Peraturan Dan Kebijakan Pemerintah

Indonesia Menghadapi Liberalisasi Ekonomi Regional : Masyarakat Ekonomi ASEAN 2015. *Rechtsvinding*, *1*(2 Agustus), 375–395.

Mandal, A., Chaki, R., Saha, S., Ghosh, K., Pal, A., & Ghosh, S. (2017). Measuring similarity among legal court case documents. *ACM International Conference Proceeding Series*, 1–9. https://doi.org/10.1145/3140107.3140119

Manning, C. D., Raghavan, P., & Schutze, H. (2009). *An Introduction to Information Retrieval*.

Melita, R., Amrizal, V., Suseno, H. B., & Dirjam, T. (2018). PENERAPAN METODE TERM FREQUENCY INVERSE DOCUMENT FREQUENCY (TF-IDF) DAN COSINE SIMILARITY PADA SISTEM TEMU KEMBALI INFORMASI UNTUK MENGETAHUI SYARAH HADITS BERBASIS WEB (STUDI KASUS: SYARAH UMDATIL AHKAM). *Jurnal Teknik Informatika*, *11*(Oktober), 149–164.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1–12.

Mikolov, T., Sutskever, I., Chen, K., Corrado, & Dean, J. (2013). *Distributed Representations of Words and Phrases and their Compositionality*. *October*, 1–9.

Mukhiya, S. K., & Ahmed, U. (2020). *Hands-On Exploratory Data Analysis with Python*.

Novotná, T. (2020). Document similarity of Czech supreme court decisions. *Masaryk University Journal of Law and Technology*, *14*(1), 105–122. https://doi.org/10.5817/MUJLT2020-1-5

Peñas, A., Hovy, E., Forner, P., Rodrigo, Á., Sutcliffe, R., & Morante, R. (2013). QA4MRE 2011-2013: Overview of question answering for machine reading evaluation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8138 LNCS*, 303–320. https://doi.org/10.1007/978-3-642-40802-1_29

Porter, M. F. (2006). An algorithm for suffix stripping. *Program*, *40*(3), 211–218. https://doi.org/10.1108/00330330610681286

Rahutomo, F., Kitasuka, T., & Aritsugi, M. (2012). Semantic Cosine Similarity. *Semantic Scholar*, *2*(4), 4–5.

Ramadhanti, N. R., & Mariyah, S. (2019). Document Similarity Detection Using Indonesian Language Word2vec Model. *ICICOS 2019 - 3rd International Conference on Informatics and Computational Sciences: Accelerating Informatics and Computational Research for Smarter Society in The Era of Industry 4.0, Proceedings*, 1–6. https://doi.org/10.1109/ICICoS48119.2019.8982432

Rijsbergen, C. J. van. (1979). *Information Retrieval* (2nd ed.). Butterworth-Heinemann.

Rong, X. (2014). *word2vec Parameter Learning Explained*. 1–21. http://arxiv.org/abs/1411.2738

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

Sadiawati, D. (2015). *Strategi Nasional Reformasi Regulasi : Mewujudkan Regulasi Yang Sederhana dan Tertib*. Kementrian PPN / Bappenas.

Salton, G., & Buckley, C. (1988). Term-Weighting Approaches In Automatic Text Retrieval. *Information Processing and Management*, *24*(5), 513–523.

Sebastiani, F. (2002). *Machine Learning in Automated Text Categorization*. *34*(1), 1–47.

Setiadi, W. (2020). SIMPLIFIKASI REGULASI DENGAN MENGGUNAKAN METODE PENDEKATAN *OMNIBUS LAW*. *Jurnal Rechts Vinding: Media Pembinaan Hukum Nasional*, *9*(April), 39–52.

Sidorov, G., Gelbukh, A., Gómez-Adorno, H., & Pinto, D. (2014). Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computacion y Sistemas*, *18*(3), 491–504. https://doi.org/10.13053/CyS-18-3-2043

Singhal, A., Buckley, C., Mitra, M., & Salton, G. (1995). *Pivoted Document Lenght Normalization*.

Vanderbeck, S., Bockhorst, J., & Oldfather, C. (2011). A Machine Learning Approach to Identifying Sections in Legal Briefs. In S. Visa, A. Inoue, & A. Ralescu (Eds.), *Proceedings of the Twenty-second Midwest Artificial Intelligence and Cognitive Science Conference* (pp. 16–21). Omnipress-Madison,WISCONSIN.

Vijayarani, S., Ilamathi, M. J., Nithya, M., Professor, A., & Research Scholar, M. P. (2015). Preprocessing Techniques for *Text mining* -An Overview. *International Journal of Computer Science & Communication Networks*, *5*(1), 7–16.

Witten, I. H. (2004). *Text mining*. *The Practical Handbook of Internet Computing*, 14-1-14–22. https://doi.org/10.1201/9780203507223

Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., & Sun, M. (2020). *How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence*. 5218–5230. https://doi.org/10.18653/v1/2020.acl-main.466