

TUGAS AKHIR - IF184802

SISTEM REKOMENDASI KARYA ILMIAH BERDASARKAN *SEMANTIC SIMILARITY* MENGUNAKAN *FASTTEXT* DAN METODE *WORD MOVER'S DISTANCE*

NABIL HAIDARRAHMAN PRIBADI
NRP 05111640000185

Dosen Pembimbing
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.
Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



TUGAS AKHIR - IF184802

**SISTEM REKOMENDASI KARYA ILMIAH
BERDASARKAN SEMANTIC SIMILARITY
MENGUNAKAN FASTTEXT DAN WORD
MOVER'S DISTANCE**

NABIL HAIDARRAHMAN PRIBADI
NRP 05111640000185

Dosen Pembimbing
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.
Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - IF184802

PAPER RECOMMENDATION SYSTEM BASED ON SEMANTIC SIMILARITY USING FASTTEXT AND WORD MOVER'S DISTANCE

NABIL HAIDARRAHMAN PRIBADI
NRP 05111640000185

Supervisor

Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.
Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS ENGINEERING
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN
SISTEM REKOMENDASI KARYA ILMIAH
BERDASARKAN SEMANTIC SIMILARITY
MENGGUNAKAN FASTTEXT DAN WORD MOVER'S
DISTANCE

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer pada
Rumpun Mata Kuliah Manajemen Informasi
Program Studi S-1 Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh:

NABIL HAIDARRAHMAN PRIBADI

NRP: 05111640000185

Disetujui oleh Dosen Pembimbing Tugas Akhir

Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.

NIP: 19870103 201404 1 001

Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

NIP: 19860823 201504 1 004



(pembimbing 1)

(pembimbing 2)

SURABAYA
JANUARI 2020

[Halaman ini sengaja dikosongkan]

SISTEM REKOMENDASI KARYA ILMIAH BERDASARKAN SEMANTIC SIMILARITY MENGUNAKAN FASTTEXT DAN WORD MOVER'S DISTANCE

Nama Mahasiswa : NABIL HAIDARRAHMAN PRIBADI
NRP : 05111640000185
Jurusan : Informatika ITS
Dosen Pembimbing I : Prof. Drs.Ec.Ir. Riyanarto Sarno, M.Sc., Ph.D.
Dosen Pembimbing II : Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

Abstrak

Karya ilmiah digunakan oleh para dosen dan mahasiswa untuk dijadikan referensi dalam membuat tugas akhir atau riset. Tetapi, terdapat masalah dalam mencari karya ilmiah yang dicari, terutama dalam menentukan kata yang cocok. Karya ilmiah ini mengusulkan sistem rekomendasi berbasis semantic similarity. Studi kasus yang digunakan pada karya ilmiah ini adalah data dari website IEEE, ACM Digital Library, Science Direct, SpringerLink, dan Wiley Digital Library.

Implementasi pada tesis ini mengusulkan fastText untuk menghasilkan word embedding dan Word Mover's Distance untuk semantic similarity. Hasil yang diraih dari usulan tugas akhir ini menggunakan tujuh skenario dengan memasukkan query ke masing-masing skenario. Query tersebut adalah dengan menggunakan kata asli, kata asli ditambah kata yang ditambahkan oleh pengguna, dan kata asli ditambah kata yang memiliki makna yang mirip dengan kata asli dari sistem.

Dari skenario-skenario tersebut, menggunakan kata asli ditambah kata yang memiliki makna yang mirip dari sistem meraih hasil akurasi, precision, recall, dan f-1 score yang tertinggi dibandingkan dengan lainnya. Hasil ini membuktikan bahwa metode yang diusulkan dapat menangkap semantic yang lebih baik.

Kata kunci: Semantic similarity, fastText, Word Mover's Distance, Sistem Rekomendasi.

PAPER RECOMMENDATION SYSTEM BASED ON SEMANTIC SIMILARITY USING FASTTEXT AND WORD MOVER'S DISTANCE

Name : NABIL HAIDARRAHMAN PRIBADI
NRP : 05111640000185
Major : Informatika ITS
Supervisor I : Prof. Drs.Ec.Ir. Riyanarto Sarno, M.Sc., Ph.D.
Supervisor II : Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

Abstract

Scientific paper is used by lecturers and students to be used as a reference in making a final project or research. However, there are problems in finding the scientific work that is sought, especially in determining the appropriate word. This scientific work proposes a recommendation system based on semantic similarity. Case studies used in this scientific work are data from the IEEE website, ACM Digital Library, Science Direct, SpringerLink, and Wiley Digital Library.

Implementation in this thesis proposed fastText for produce word embedding and Word Mover's Distance for semantic similarity. The results obtained from this proposed method use seven scenarios by entering a query into each scenario. The query is to use the original word, the original word plus words added by the user, and the original word plus words that have meaning similar to the original word from system.

From these scenarios, using original words plus words that have similar meanings from system achieves the highest accuracy, precision, recall, and f-1 score compared to others. These results prove that the proposed method can capture semantics better

Keywords: Semantic similarity, fastText, Word Mover's Distance, Recommender System.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Segala puji dan syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga dapat diselesaikan tugas akhir ini yang berjudul **“Sistem Rekomendasi Karya Ilmiah Berdasarkan Semantic Similarity Menggunakan FastText dan Word Mover’s Distance”**.

Dalam pelaksanaan tugas akhir ini tentunya tidak dapat diselesaikan tanpa bantuan dari pihak lain. Tanpa mengurangi rasa hormat, diberikan penghargaan serta ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Keluarga penulis Bunda Retno Sulistyowati Rahayu, Ayah Wahyu Pribadi, Kakak Kemal Fadhlurrahman Pribadi, dan Adik Aliyahfitri Fadillah Pribadi yang telah memberikan dukungan secara lahir dan batin sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Bapak Prof. Drs. Ec. Ir. Rianarto Sarno, M.Sc., Ph.D. dan Ibu Adhatus Solichah Ahmadiyah, S.Kom., M.Sc. selaku pembimbing I dan pembimbing II yang telah membimbing, memberi nasehat, mendukung penulis dalam bentuk moral, serta memberikan waktu dan tenaga untuk membimbing dalam menyelesaikan Tugas Akhir Ini.
4. Dr. Eng. Darlis Herumurti, S.Kom., M.Kom. selaku Ketua Departemen Informatika ITS dan segenap dosen dan karyawan Departemen Informatika ITS yang telah memberikan ilmu dan pengalaman kepada penulis selama menjalani masa kuliah di Informatika ITS.
5. Elsa Siffana Hedianti selaku pasangan penulis yang telah membantu penulis dalam bentuk apapun sehingga penulis dapat menyelesaikan Tugas Akhir ini dan menemani penulis dalam menyelesaikan Tugas Akhir ini.
6. Said Fariz Hibban selaku senior dari penulis yang telah memberikan pembelajaran untuk penulis sehingga dapat

menjalani perkuliahan dari tahun pertama sampai tahun terakhir.

7. Rayhan Gemaruzman selaku kakak pendamping dari penulis selama masa kaderisasi tahun pertama penulis yang telah memberikan dukungan untuk penulis dalam menjalani apapun yang dijalani oleh penulis.
8. Gd. Wahyu Nugraha dan Salma Nurkhafidoh yang telah memberikan kesempatan kepada penulis untuk mempelajari makna kehidupan dengan menjadikan penulis Badan Pengurus Harian di Schematics REEVA 2017.
9. Vincent Marcello Dwi Tanujaya, Daniel Kurniawan, Dewi Sekarini, Khairunnisa' Rahma Mardiyani, Frandita Adhitama, Fandy Putra Mohammad, Ganendra Afrasya Salsabilla, Yolanda Hertita Pratama, Hafid Sriwijaya Bahrn, Desy Nilasari, Ibrahim Tamtama Adi, Almas Aqmarina, Naufal Andira Perdana, Elvega Dewangga, Vinsensius Indra Suryanto, Ivanda Zevi Amalia, Alifa Izzan Akhsani, Ismail Syarief, Diana Hudani Kisyono, Himawan Wijaya, Denise Sonia Rahmadina, Chendrasena Oemaryoga, Azkiatunnisa Rahma yang telah menjadi bagian dari Kabinet Himpunan Mahasiswa Teknik Computer-Informatika (HMTIC) "GARANG" 2018/2019 dan menemani penulis dalam memimpin satu periode kepengurusan himpunan.
10. Mochammad Isom Mufikri dan Satya Dharmawanto yang telah menemani dan mendukung penulis dalam menyelesaikan Tugas Akhir ini.
11. Faris Didin Andiyar dan Rahajeng Dwi Permatasari yang telah memberikan kesempatan kepada penulis untuk belajar banyak hal selama penulis menjadi Staff Kaderisasi dan Pemetaan (KDPM) HMTIC "KREASI" 2017/2018.
12. Alfindio Muhammad Abdallah dan Yuniar Permatasari yang telah memberikan kesempatan kepada penulis untuk belajar banyak hal selama penulis menjadi Staff *Internal Affair* BEM FTIK "SEMANGAT BERPADU" 2017/2018.

13. Subhan Maulana, Alfindio Muhammad Abdallah, Glenn Lucas Harryara, Gerald Parlindungan Salomo, Wahyu Ivan Satyagraha, Nanang Taufan Budianto, dan Rakhma Rufaida Hanum yang telah menemani dan mendukung penulis dalam menjalani perkuliahan.
14. Zahri Rusli, Djohan Prabowo, Tegar Satrio Utomo, Firman Aqil, Muhammad Illham Hanafi, Huda Fauzan Murthado, Muhammad Ichsan Sandi, Ilham Muhammad Misbahuddin, Satriyo Nugroho, Muhammad Adib Arinanda, Rafi R Ramadhan, Muhammad Pandu Praadha, Muhammad Adhitya Irvansyah, Pandito Hudiarso Abdul Rahim Setia Negara, Muhammad Faris Abdurrasyid, dan Maulana Sechan yang telah menjadi senior dan teman diskusi penulis selama menjalani masa perkuliahan.
15. Frandita Adhitama, Michael Julian Albertus. Dewi Ayu Nirmalasari, Fariz Maulana Purnomo, Alcredo Simanjuntak. Muhammad Fauzan, Rahadian Koesdijarto Putra, Dandy Naufaldi, Yolanda Hertita Pratama, Yoshima Syah Putri, Denny Rengganis, dan Aguel Satria yang telah menjadi teman diskusi penulis dan penyemangat dalam menyelesaikan tugas akhir ini.
16. Rayhan Calviandoro, Agung Nabawi Santoso, Putra Iranto, Maria Syauta, Izzan Aminul Majid, Rudy Hartono, Ari Krisna Putra, Mahanaim Nicolas Samuel, dan Calvin Prayandi yang telah menjadi teman diskusi penulis dalam membentuk pola pikir penulis.
17. Rahmatin Nadia, Kania Amalia, Anne Annisa Aulia, Nanang Taufan Budianto, Haidar Arya Prasetya, Unggul Widodo Wijayanto, Huda Fauzan Murthado, Ivan Fadhila, Narendra Haryo Bismo, Yolanda Wisdanita Samosir, Hilmi Raditya Prakoso, Azzam Jihad Ulhaq, Zahrul Zizki Dinanto, Yudhistiro Adi Nugroho, Karina Soraya Puspitasari, Isnaini Nurul Kurniasari yang telah menemani penulis selama menjadi Admin Lab Manajemen Informasi (MI).

18. Rani Aulia Hidayat, Muhammad Hilman, Afif Ridho Kamal Putra, Rizky Fenaldo Maulana, Tikva Imanuel Mooy, Zikrul Ihsan, dan Panji Rimawan selaku senior dari penulis yang telah memberikan ilmu-ilmu kepada penulis untuk menjalani perkuliahan dari tahun pertama sampai tahun terakhir.
19. Serta semua pihak yang tidak bias penulis sebutkan dan telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Diharapkan bahwa apa yang dihasilkan dari Tugas Akhir ini bisa memberikan manfaat bagi semua pihak. Penulis memohon maaf apabila terdapat kesalahan maupun kekurangan dalam Tugas Akhir.

Surabaya, Desember 2019

Nabil Haidarrahman Pribadi

DAFTAR ISI

Abstrak	vii
Abstract	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xviii
DAFTAR TABEL.....	xxi
DAFTAR KODE SUMBER	xxiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat.....	4
1.6 Metodologi	4
1.7 Sistematika Penulisan.....	6
BAB II TINJAUAN PUSTAKA.....	7
2.1 Sistem Rekomendasi	7
2.1.1 Collaborative Filtering.....	7
2.1.2 Content-based Filtering	8
2.1.3 Multi-criteria Recommender Systems	8
2.1.4 Risk-aware Recommender Systems	9
2.1.5 Mobile Recommender Systems.....	9
2.1.6 Hybrid Recommender Systems.....	9

2.2	<i>Semantic Similarity</i>	9
2.3	<i>FastText</i>	10
2.3.1	General Model.....	10
2.3.2	Subword Model.....	12
2.4	Model <i>Skip-Gram</i>	12
2.5	<i>Word Mover's Distance</i>	14
2.6	<i>Gensim</i>	15
2.7	<i>Python</i>	19
2.8	<i>Confusion Matrix</i>	20
2.9	<i>Cohen's Kappa Coefficient</i>	21
2.10	<i>Precision dan Recall</i>	22
2.11	<i>F-1 Score</i>	23
2.12	<i>Regular Expression</i>	23
2.13	<i>Multiprocessing</i>	23
BAB III ANALISIS DAN PERANCANGAN SISTEM		25
3.1	Analisis Metode Secara Umum.....	25
3.2	Data Preparation.....	31
3.3	<i>Data Preprocessing</i>	32
3.4	<i>Training Corpus</i>	34
3.5	<i>Semantic Search</i>	38
BAB IV IMPLEMENTASI.....		40
4.1	Lingkungan Implementasi	41
4.2	Implementasi Proses	43
4.2.1	<i>Data Preparation</i>	43
4.2.2	<i>Data Preprocessing</i>	48

4.2.3	<i>Training Korpus</i>	51
4.2.4	<i>Semantic Search</i>	53
BAB V PENGUJIAN DAN EVALUASI		57
5.1	Lingkungan Uji Coba	57
5.2	<i>Dataset</i> Uji Coba	57
5.3	Skenario Uji Coba	61
5.4	Hasil Evaluasi	67
5.5	Tampilan Aplikasi	69
BAB VI KESIMPULAN DAN SARAN		71
6.1	Kesimpulan	71
6.2	Saran	72
DAFTAR PUSTAKA		73
LAMPIRAN A : HASIL CRAWLING DATA		79
BIODATA PENULIS		87

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2. 1 Model <i>Skip-Gram</i> [18].....	13
Gambar 2. 2 Ilustrasi <i>Word Mover's Distance</i> [19]	14
Gambar 2. 3 <i>Confusion Matrix</i>	20
Gambar 3. 1 Diagram <i>use case</i>	26
Gambar 3. 2 <i>Prototype interface</i> tampilan awal.....	26
Gambar 3. 3 <i>Prototype interface</i> tampilan hasil rekomendasi	27
Gambar 3. 4 Bagan proses utama sistem	28
Gambar 3. 5 Diagram alir <i>data preparation</i>	32
Gambar 3. 6 Diagram alir <i>data preprocessing</i>	34
Gambar 3. 7 Diagram alir <i>training corpus</i>	37
Gambar 3. 8 Diagram alir <i>semantic search</i>	39
Gambar 3. 9 Hasil Keluaran <i>Data Preprocessing</i>	51
Gambar 5. 1 Grafik hasil performa iterasi	62
Gambar 5. 2 Grafik hasil perbandingan dimensi.....	63
Gambar 5. 3 <i>Screenshot</i> tampilan <i>interface</i>	69
Gambar 5. 4 <i>Screenshot</i> hasil rekomendasi <i>interface</i>	70

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 4. 1 Lingkungan Implementasi	41
Tabel 4. 2 <i>Tools</i> Pengerjaan.....	42
Tabel 5. 1 <i>Dataset</i>	58
Tabel 5. 2 Deskripsi Skenario	64
Tabel 5. 3 Hasil Evaluasi	67

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4. 1 Inisialisasi dasar	44
Kode Sumber 4. 2 Penyimpanan alamat <i>website</i>	44
Kode Sumber 4. 3 Ekstrasi Data.....	46
Kode Sumber 4. 4 Implementasi Proses	47
Kode Sumber 4. 5 Membaca masukkan data	48
Kode Sumber 4. 6 Melakukan <i>case folding</i>	49
Kode Sumber 4. 7 Menghapus <i>Noise</i>	49
Kode Sumber 4. 8 Menghapus <i>stopwords</i>	50
Kode Sumber 4. 9 Menyimpan <i>file CSV</i>	50
Kode Sumber 4. 10 Inisialisasi <i>library</i>	51
Kode Sumber 4. 11 Membaca <i>File CSV</i>	51
Kode Sumber 4. 12 Inisialisasi model	52
Kode Sumber 4. 13 Fungsi <i>Preprocessing</i>	52
Kode Sumber 4. 14 <i>Preprocessing Content</i>	53
Kode Sumber 4. 15 Model <i>fastText</i>	53
Kode Sumber 4. 16 Inisialisasi <i>Similarity</i>	54
Kode Sumber 4. 17 Inisialisasi <i>Query</i>	54
Kode Sumber 4. 18 <i>Similarity</i>	54
Kode Sumber 4. 19 Hasil Rekomendasi	55
Kode Sumber 5. 1 Parameter <i>fastText</i>	62
Kode Sumber 5. 2 <i>Query</i> makna kata.....	66

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Bab pendahuluan ini menjelaskan tentang pendahuluan pengerjaan tugas akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan tugas akhir, dan sistematika penulisan.

1.1 Latar Belakang

arya ilmiah adalah laporan tertulis dan diterbitkan dalam sebuah konferensi dengan memaparkan hasil dari penelitian atau pengkajian yang telah dilakukan oleh seseorang atau sebuah tim dengan memenuhi kaidah dan etika keilmuan yang dikukuhkan dan ditaati oleh para akademisi. Ada beberapa jenis karya ilmiah, antara lain laporan penelitian, makalah, seminar atau simposium, dan artikel jurnal lainnya. Untuk mendapatkan file-file dari karya ilmiah yang dipublikasi, biasanya tersedia di beberapa website yang tersedia di internet, antara lain *Google Scholar*, *ScienceDirect*, *IEEEExplore*, *Pubmed*, *Education Resources Information Center* (ERIC). Di perguruan tinggi, mahasiswa dilatih oleh para akademisi yang ada di perguruan tinggi tersebut dengan membuat karya ilmiah seperti makalah dan skripsi. Di Indonesia, terdapat berbagai perlombaan untuk menghasilkan karya ilmiah. Salah satunya adalah Program Kreativitas Mahasiswa (PKM) yang diselenggarakan Kementerian Riset Teknologi dan Perguruan Tinggi Republik Indonesia.

Saat ini, terjadi masalah yang terjadi di kalangan mahasiswa ketika ingin mencari referensi dari karya ilmiah untuk membuat makalah atau skripsi. Masalah tersebut terjadi karena mahasiswa memiliki kesulitan untuk menemukan kata-kata yang penting dalam mencari karya ilmiah di *website-website* yang ada di internet. Dengan perkembangan teknologi yang pesat ini, dapat dilakukan berbagai macam cara untuk mengatasi permasalahan tersebut.

Salah satu cara untuk mengatasi masalah tersebut adalah dengan merancang sistem rekomendasi dalam studi kasus karya ilmiah. Sistem rekomendasi merupakan subkelas dari information filtering untuk memberikan rekomendasi kepada pengguna berdasarkan item yang dipilihnya [1]. Sistem rekomendasi memberikan kecepatan, kustomisasi otomatis, dan personalisasi pada *website*. Sistem rekomendasi pada *website* biasanya merekomendasikan *item* yang telah dilihat, *item* yang telah dibeli, ataupun rekomendasi dari pengguna lainnya. Permasalahan sistem rekomendasi dengan permasalahan yang dialami oleh para mahasiswa dalam mencari referensi karya ilmiah, sistem rekomendasi dapat memberikan rekomendasi karya ilmiah yang sesuai dengan preferensi dari mahasiswa yang mengakses sistem tersebut. Sistem ini diharapkan dapat menjadi salah satu solusi untuk mengatasi permasalahan yang dialami oleh para mahasiswa. Dengan adanya sistem ini dapat membantu mahasiswa untuk kemudahan dalam akses pencarian karya ilmiah dan mendapatkan karya ilmiah tersebut untuk dijadikan referensi dalam pembuatan makalah atau skripsi.

Tugas Akhir ini diharapkan dapat membangun sebuah sistem yang dapat memberikan rekomendasi yang sesuai dengan keinginan pengguna untuk mengatasi permasalahan yang dialami oleh mahasiswa untuk mencari referensi dalam pembuatan makalah atau skripsi. Tugas Akhir ini akan menggunakan metode *fastText* untuk mencari vektor representasi makna dari setiap kata yang terdapat pada *corpus* dan *Word Mover's Distance Similarity* untuk mencari kemiripan antara satu kalimat dengan kalimat lainnya dengan menggunakan *query* dari pengguna dan *metadata* karya ilmiah yang dikumpulkan. *Metadata* yang digunakan didapatkan dari data pengguna dan data informasi buku yang tersedia di website IEEE. Dari data tersebut akan digunakan untuk memberikan rekomendasi karya ilmiah. Selain itu, perhitungan akurasi akan dilakukan dengan menggunakan *dataset* dari *ground truth* berdasarkan *systematic literature review*.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut.

1. Bagaimana mengimplementasikan metode *fastText* dalam memetakan kata menjadi vektor?
2. Bagaimana mengimplementasikan metode *Word Mover's Distance Similarity* untuk mencari kemiripan dari kalimat yang dimasukkan oleh user dengan karya ilmiah yang tersedia?
3. Bagaimana hasil evaluasi yang didapatkan dengan mengimplementasikan model *fastText* dan *Word Mover's Distance Similarity*?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir memiliki beberapa batasan antara lain sebagai berikut:

1. Data yang digunakan pada Tugas Akhir ini adalah data informasi karya ilmiah. Informasi karya ilmiah yang berisi judul karya ilmiah, abstraksi, judul publikasi, *digital object identifier* (DOI), topik karya ilmiah, dan kata kunci dari karya ilmiah.
2. Bahasa pemrograman yang mengimplementasikan Tugas Akhir ini adalah *Python*.
3. *Library* yang digunakan untuk mengimplementasikan metode pada Tugas Akhir ini adalah *gensim*, *multiprocessing*, *time*, dan *pandas*.
4. Perangkat lunak yang digunakan pada Tugas Akhir ini adalah *Jupyter Notebook* untuk penerapan metode *fastText* dan *Word Mover's Distance*.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah untuk membangun sistem rekomendasi untuk karya ilmiah yang sesuai

dengan kalimat yang dituliskan oleh pengguna dengan menggunakan metode *fastText* dan *Word Mover's Distance* pada sistem rekomendasi, sehingga dapat memberikan rekomendasi berupa data dari karya ilmiah tersebut melalui kemiripan kalimat antara kalimat dari penulis dan data dari karya ilmiah yang tersedia.

1.5 Manfaat

Manfaat dari pembuatan tugas akhir ini antara lain sebagai berikut:

1. Membantu pengguna dalam menemukan karya ilmiah yang memiliki kemiripan secara *semantic* dengan *query* yang dilakukan oleh pengguna.
2. Hasil penelitian dapat menjadi bahan pembelajaran serta sebagai satu acuan dalam pengembangan penelitian selanjutnya.

1.6 Metodologi

Tahap yang dilakukan untuk menyelesaikan tugas akhir ini adalah sebagai berikut:

1. Penyusunan Proposal Tugas Akhir

Proposal tugas akhir ini terdiri dari tiga sub bab yaitu pendahuluan, tinjauan pustaka, metodologi dan jadwal pengerjaan tugas akhir. Pada pendahuluan akan dijelaskan mengenai hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Pada metodologi dijelaskan mengenai tahapan penyusunan tugas akhir. Pada jadwal kegiatan ini akan menjelaskan jadwal pengerjaan tugas akhir.

2. Studi literatur

Tahap ini merupakan tahap pengumpulan informasi dan pembelajaran yang akan digunakan pada tugas akhir ini. Studi literatur meliputi diskusi dan pemahaman terkait dengan tugas akhir ini, diantaranya mengenai :

1. *FastText*
2. *Word Mover's Distance*
3. *Gensim*
4. *Python*

3. Analisis dan Desain Perangkat Lunak

Pada tahap ini akan menganalisis metode yang akan diuji coba dalam pembuatan sistem rekomendasi berdasarkan studi literatur yang telah dilakukan. Adapun metode yang nantinya akan digunakan dalam pembuatan sistem rekomendasi ini adalah *fastText* dan *Word2vec*. Keduanya akan dibandingkan dengan melihat akurasi dari kedua metode tersebut untuk pembuatan sistem rekomendasi karya tulis ilmiah ini.

4. Pengujian dan Evaluasi

Penerapan dalam pembuatan sistem rekomendasi untuk karya tulis ini adalah dengan melakukan *word embedding* menggunakan *fastText* dari *library fasttext* yang tersedia di bahasa pemrograman *python* dan menerapkan algoritma *word mover's distance* dari *library gensim* yang tersedia di bahasa pemrograman *python*. Selain itu, dalam tugas akhir ini akan membandingkan implementasi *word embedding* dari *fastText* dan *Word2ve*.

5. Penyusunan Buku Tugas Akhir

Pada tahapan ini disusun buku tugas akhir yang akan menjadi dokumentasi mengenai pembuatan serta hasil dari tugas akhir yang dibuat.

1.7 Sistematika Penulisan

Buku tugas akhir ini terdiri atas beberapa bab yang tersusun secara sistematis, yaitu sebagai berikut.

1. Bab I. Pendahuluan

Bab pendahuluan berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan tugas akhir.

2. Bab II. Tinjauan Pustaka

Bab tinjauan pustaka akan menjelaskan mengenai landasan teori yang digunakan sebagai penunjang dalam pembuatan tugas akhir.

3. Bab III. Analisis dan Perancangan

Pada bab ini akan menjelaskan mengenai sistem yang dibangun, Adapun perancangan sistem ini meliputi perancangan data dan alur proses sistem.

4. Bab IV. Implementasi

Bab ini membahas mengenai implementasi dari sistem yang telah dibuat pada bab sebelumnya.

5. Bab V. Pengujian dan Evaluasi

Bab ini menjelaskan mengenai pengujian dari metode yang ditawarkan dalam tugas akhir untuk mengetahui kesesuaian metode dengan data yang ada.

6. Bab VI. Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang telah dilakukan. Bab ini juga membahas saran-saran untuk pengembangan sistem lebih lanjut.

BAB II

TINJAUAN PUSTAKA

Bab tinjauan pustaka ini berisi mengenai dasar teori yang digunakan dalam penelitian tugas akhir ini..

2.1 Sistem Rekomendasi

Sistem rekomendasi adalah subkelas dari *information filtering system* yang memiliki parameter untuk memprediksi *item* ke pengguna berdasarkan “*rating*” atau “*preferensi*” [1]. Saat ini, sistem rekomendasi sudah banyak digunakan pada situs-situs *website* yang tersedia di internet, contohnya adalah YouTube, Twitter, Amazon, Tokopedia, dan Bukalapak. Pada *website* YouTube, sistem rekomendasi digunakan untuk memberikan rekomendasi kepada pengguna berupa video yang memiliki *rating* atau *preferences* sesuai dengan video yang pengguna tonton. Pada *website* Twitter, sistem rekomendasi akan memberikan rekomendasi kepada pengguna berupa *content* [2]. Dengan beragamnya penerapan dari sistem rekomendasi ini, sistem ini juga diterapkan untuk mencari riset mengenai *experts* [3], *collaborators* [4], dan *financial services* [5].

Sistem rekomendasi memiliki beberapa pendekatan untuk memberikan rekomendasi kepada pengguna. Pendekatan-pendekatan tersebut antara lain adalah *collaborative filtering*, *content-based filtering*, *multi-criteria recommender systems*, *risk-aware recommender systems*, *mobile recommender systems*, dan *hybrid recommender systems*.

2.1.1 Collaborative Filtering

Collaborative filtering adalah salah satu dari beberapa pendekatan yang digunakan pada sistem rekomendasi untuk

memberikan rekomendasi berdasarkan asumsi dari pengguna yang pernah mengakses *item* pada masa lalu dan akan memberikan *item* yang mirip kepada pengguna yang baru [6]. Sistem rekomendasi dengan pendekatan ini akan memberikan rekomendasi berdasarkan informasi mengenai *profile* dari pengguna untuk pengguna atau *item* yang lain.

Contoh dari penerapan sistem rekomendasi ini antara lain sebagai berikut:

- Meneliti *item* yang pernah diakses oleh pengguna di dalam *online store*.
- Menganalisa jumlah akses dari *item* atau pengguna.
- Menyimpan *record* dari *item* yang pengguna melakukan transaksi secara *online*
- Mendapatkan kumpulan *item* yang pengguna dengar atau tonton dalam komputer.
- Menganalisis *social network* dari pengguna dan mengetahui *item* yang disukai atau tidak disukai.

2.1.2 Content-based Filtering

Content-based filtering adalah salah satu pendekatan dalam sistem rekomendasi berdasarkan *description* dari *item* dan *profile* dari preferensi pengguna [7]. Pendekatan ini adalah yang terbaik jika terdapat *metadata* yang diketahui dari *item* (nama, lokasi, deskripsi, dll.), tetapi bukan pada pengguna. Pendekatan ini dapat dikatakan sebagai *user-specific classification problem* dan mempelajari *classifier* dari *likes* dan *dislikes* pengguna berdasarkan fitur dari *item*.

2.1.3 Multi-criteria Recommender Systems

Multi-criteria recommender systems (MCRS) dapat didefinisikan sebagai sistem rekomendasi menggabungkan preferensi dari informasi pengguna atas kriteria yang banyak [8].

2.1.4 Risk-aware Recommender Systems

Mayoritas dari pendekatan yang dilakukan oleh sistem rekomendasi hanya terfokuskan pada merekomendasikan konten yang paling relevan kepada pengguna berdasarkan informasi secara kontekstual, namun tidak memperhatikan risiko yang dapat mengganggu pengguna dengan notifikasi yang tidak diinginkan. Pendekatan ini dilakukan pada sistem rekomendasi untuk mempertimbangkan risiko dari terganggunya pengguna dengan memberikan rekomendasi pada kondisi tertentu. Untuk mengatasi permasalahan ini, dapat diterapkan pendekatan ini [9].

2.1.5 Mobile Recommender Systems

Mobile recommender systems menggunakan *smartphone* untuk merekomendasikan rekomendasi yang memiliki *context-sensitive*. Ada tiga faktor yang dapat memberikan efek pada *mobile recommender systems* dan akurasi pada hasil prediksi, yaitu konteks, metode rekomendasi dan privasi [10].

2.1.6 Hybrid Recommender Systems

Mayoritas dari sistem rekomendasi yang telah diterapkan menggunakan pendekatan *hybrid*, dengan mengombinasikan *collaborative filtering*, *content-based filtering*, dan pendekatan lainnya. Pendekatan ini dapat mengatasi yang kerap muncul pada sistem rekomendasi seperti *cold start* dan *sparsity problem*, begitu juga dengan *knowledge engineering bottleneck* dalam *knowledge-based recommender systems* [11].

Pada Tugas Akhir ini, diusulkan metode *content-based filtering* dikarenakan *dataset* yang telah dikumpulkan cocok dengan parameter yang digunakan pada metode *content-based filtering*.

2.2 Semantic Similarity

Semantic similarity adalah sebuah *metric* yang didefinisikan dalam kumpulan dokumen, dimana ide dari jarak antara dokumen

tersebut berdasarkan kemiripan dari maknanya atau konten *semantic* yang berlawanan dengan *similarity* yang bisa dihitung berdasarkan representasi *syntactic* (atau format *string* dari dokumen tersebut). Metode ini adalah *mathematical tools* yang digunakan untuk mengestimasi kekuatan dari relasi *semantic* antara satuan dari bahasa, konsep, melalui deskripsi numerik yang didapatkan berdasarkan perbandingan antara informasi yang mendukung maknanya atau mendeskripsikan kata dasarnya. *Semantic similarity* biasanya dimiripkan dengan *semantic relatedness* [12]. *Semantic relatedness* memasukkan relasi antara dua istilah, dimana *semantic similarity* hanya memasukkan relasi “*is a*”. Sebagai contoh, “*car*” memiliki kemiripan dengan “*bus*”, tetapi memiliki relasi dengan “*road*” dan “*driving*” [13].

2.3 *FastText*

FastText adalah *library* untuk mempelajari *word embedding* dan klasifikasi teks yang dibuat oleh laboratorium *Facebook’s Artificial Intelligence Research* (FAIR). *FastText* dapat mengimplementasikan algoritma *unsupervised learning* atau *supervised learning* pada sebuah data yang dimiliki oleh user dan menghasilkan model yang berisi hasil dari vektor pada data yang dijadikan inputan ketika mengimplementasikan algoritma *unsupervised learning* atau *supervised learning*. Ada beberapa topik penelitian dari implementasi metode *fastText* ini, antara lain adalah mendeteksi komentar negatif di media sosial [14], klasifikasi teks [15], dan kategorisasi teks [16]. Pada Tugas Akhir ini, implementasi *fastText* yang digunakan adalah *unsupervised learning* untuk menghasilkan model vektor dari data corpus berbahasa Inggris.

Model *fastText* yang digunakan ini ada dua model, model-model tersebut adalah sebagai berikut [17]:

2.3.1 *General Model*

Model yang pertama adalah general model. Dalam model ini, diberikan kosa kata dalam besaran W , di mana kata tersebut

dapat diidentifikasi dari *index* $w \in \{1, \dots, W\}$. Tujuannya adalah mempelajari representasi vektor untuk setiap kata w . Secara jelasnya, diberikan corpus yang besar untuk dipelajari yang direpresentasikan sebagai kumpulan kata w_1, \dots, w_T , fungsi dari model skipgram ini untuk memaksimalkan seperti persamaan (2.1) yang ada di bawah ini:

$$\sum_{t=1}^T \sum_{c \in C_t} \log p(w_c | w_t), \quad (2.1)$$

Konteks C_t adalah set dari indeks yang berada di dalam kumpulan kata w_t . Untuk mendefinisikan peluang dari konteks kata tersebut adalah menggunakan softmax pada persamaan (2.2):

$$p(w_c | w_t) = \frac{e^{s(w_t, w_c)}}{\sum_{j=1}^W e^{s(w_t, j)}} \quad (2.2)$$

untuk konteks pada posisi c , menggunakan *binary logistic loss*, didapatkan negative log-likelihood seperti pada persamaan (2.3):

$$\log(1 + e^{-s(w_t, w_c)}) + \sum_{n \in \mathcal{N}_{t,c}} \log(1 + e^{s(w_t, n)}), \quad (2.3)$$

dengan mendenotasi fungsi logistic loss $\ell : x \mapsto \log(1 + e^{-x})$, dapat dituliskan seperti pada persamaan (2.4):

$$\sum_{t=1}^T \left[\sum_{c \in \mathcal{C}_t} \ell(s(w_t, w_c)) + \sum_{n \in \mathcal{N}_{t,c}} \ell(-s(w_t, n)) \right] \quad (2.4)$$

2.3.2 Subword Model

Dengan menggunakan representasi vektor yang berbeda untuk setiap kata, model skipgram mengabaikan struktur dari dalam kata itu sendiri. Dalam bagian ini, dilakukan fungsi skor s yang berbeda, untuk mengambil informasi dari kata.

Setiap kata w dapat direpresentasikan sebagai kumpulan karakter n -gram. Ditambahkan simbol spesial $<$ dan $>$ di awal dan akhir kata. Contoh, dalam kata *where* dan $n = 3$, dapat direpresentasikan sebagai berikut:

$$< wh, whe, her, ere, re >$$

dan urutannya :

$$< where >.$$

Perhatikan bahwa urutan $< \mathbf{her} >$, sesuai dengan kata *her* akan berbeda dari tri-gram *her* dari kata *where*.

Fungsi untuk memberikan skor pada sebuah kata dapat didefinisikan oleh persamaan (2.5):

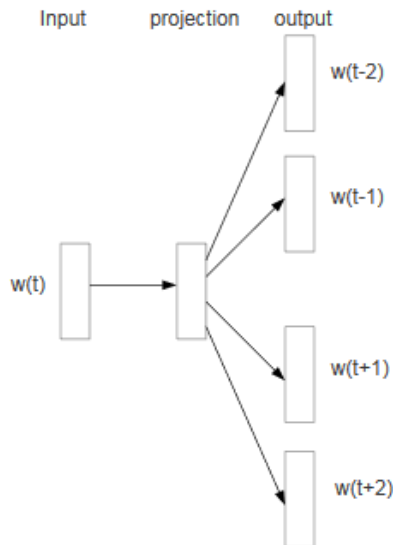
$$s(w, c) = \sum_{g \in \mathcal{G}_w} z_g^T v_c \quad (2.5)$$

2.4 Model Skip-Gram

Model *Skip-gram* merupakan metode yang efisien untuk mempelajari *high-quality* representasi vektor kata-kata dari data yang tidak terstruktur dalam jumlah yang banyak [18]. Tidak

seperti kebanyakan dari penggunaan arsitektur *neural network* sebelumnya dalam mempelajari vektor kata-kata, melatih model *skip-gram* tidak melibatkan multiplikasi *dense matrix*. Hal ini yang membuat pelatihan yang dilakukan model *skip-gram* sangat efisien: implementasi *single-machine* yang telah dioptimalisasi dapat melatih lebih dari 100 miliar kata dalam satu hari.

Dalam pengerjaan Tugas Akhir ini, akan digunakan model *skip-gram* untuk dilakukan penentuan model pada fastText. Arsitektur model *skip-gram* dapat diilustrasikan pada Gambar 2. 1.

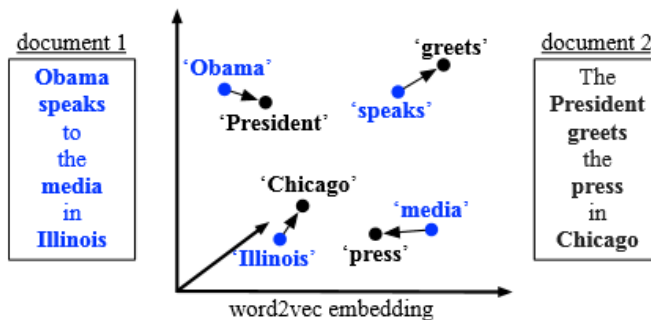


Gambar 2. 1 Model *Skip-Gram* [18]

Pada Gambar 2. 1, dijelaskan bahwa model *skip-gram* menargetkan konteks dari satu kata yang ditargetkan dengan mempelajari kata-kata yang berada di sekitar kata yang ditargetkan oleh model *skip-gram*. Hal ini yang membuat model ini dapat mempelajari banyak konteks dalam satu kata sehingga memperkaya representasi vektor kata-kata.

2.5 Word Mover's Distance

Word Mover's Distance (WMD) adalah fungsi jarak antara dokumen satu dan lainnya [19]. WMD menghitung ketidakmiripan antara dua dokumen teks sebagai jumlah minimum dari jarak sehingga kata-kata yang tertanam pada satu dokumen butuh untuk “pindah” untuk mencapai kata-kata yang tertanam pada dokumen lainnya. WMD sudah diterapkan pada beberapa studi kasus, antara lain analisis dari log jaringan [20] dan pemeringkatan dokumen [21]. Ilustrasi dari WMD dapat dilihat pada Gambar 2. 2.



Gambar 2. 2 Ilustrasi *Word Mover's Distance* [19]

Gambar 2. 2 mengilustrasikan kalkulasi dari WMD pada dua kalimat. Untuk membandingkan Kalimat 1 “*Obama speaks to the media in Illinois*” dengan Kalimat 2 “*The President greets the press in Chicago*”, pertama, temukan jarak minimum dari kata “Obama” bergerak ke semua kata yang ada di dokumen lain. Gambar 1 menampilkan bahwa “President” adalah kata terdekat dari “Obama”. Setelah itu, kata “speaks” memiliki kata terdekat “greets”. Terakhir, setelah memeriksa semua kata, jumlahkan nilai dari perpindahan kata sebagai jarak kalimat antara Kalimat 1 dan Kalimat 2.

Pada Tugas Akhir ini, diterapkan metode *WMD Similarity* (WMDS). Rumus dari WMDS dapat didefinisikan dalam persamaan (2.6):

$$WMDS = \frac{1}{1 + WMD} \quad (2.6)$$

Dari persamaan tersebut, WMDS menggunakan hasil dari WMD untuk dikalkulasikan dalam persamaan tersebut.

2.6 Gensim

Gensim adalah *open-source library* yang ada di bahasa pemrograman *python* untuk pemodelan ruang vektor dan toolkit dari pemodelan topik [22]. *Gensim* secara khusus ditujukan untuk menangani koleksi teks besar dengan menggunakan algoritma yang tersedia secara daring. *Gensim* mengimplementasikan *Latent Semantics Analysis* (LSA), *Latent Dirichlet Analysis* (LDA), *Word Mover's Distance* (WMD), dan lain-lain. Pada tugas akhir ini, *gensim* digunakan untuk mengimplementasikan *word2vec* dan WMD.

Parameter metode *fastText* yang diterapkan melalui *library gensim* dapat dilihat pada Tabel 2. 1.

Tabel 2. 1 Parameter *fastText*

Nama	Deskripsi
<i>sentences</i>	Bisa hanya <i>list</i> dari <i>list token</i> , tetapi untuk <i>corpus</i> yang lebih besar, pertimbangkan iterasi yang mengalirkan kalimat langsung dari <i>disk</i> / jaringan.
<i>corpus_file</i>	<i>Path</i> ke file <i>corpus</i> dalam format <i>LineSentence</i> . Dapat menggunakan argumen ini alih-alih menggunakan

Nama	Deskripsi
	<i>sentences</i> untuk mendapatkan peningkatan kinerja. Hanya satu kalimat atau argumen <i>corpus_file</i> yang perlu dilakukan (atau tidak satupun, dalam kasus itu, model dibiarkan tidak diinisialisasi).
<i>min_count</i>	Model ini akan mengabaikan semua kata yang memiliki frekuensi lebih rendah daripada yang ditentukan.
<i>size</i>	Besaran dimensi dari vektor.
<i>window</i>	Jarak maksimum antara satu kata dengan kata yang diprediksi dalam sebuah kalimat.
<i>workers</i>	Menentukan jumlah <i>threads</i> dalam melakukan <i>training</i> model <i>fastText</i> .
<i>alpha</i>	Inisialisasi <i>learning rate</i> .
<i>min_alpha</i>	<i>Learning rate</i> akan turun secara linier menuju <i>min_alpha</i> sebagai progress dari <i>training</i> .
<i>sg</i>	Algoritma dari <i>training</i> : <i>skip-gram</i> jika <i>sg</i> =1, selain itu CBOW.
<i>hs</i>	Jika <i>hs</i> =1, <i>hierarchical softmax</i> akan digunakan sebagai model <i>training</i> . Jika <i>hs</i> =0, dan negatif adalah <i>non-zero</i> , maka menggunakan <i>negative sampling</i> sebagai model <i>training</i> .

Nama	Deskripsi
<i>seed</i>	<i>seed</i> digunakan untuk menggenerasikan <i>random number</i> .
<i>max_vocab_size</i>	Limitasi <i>RAM</i> ketika melakukan <i>training vocabulary</i> .
<i>sample</i>	<i>Threshold</i> dalam mengkonfigurasi kata yang memiliki frekuensi yang tinggi akan secara <i>random</i> dilakukan <i>down-sampling</i> .
<i>ns_exponent</i>	<i>Exponent</i> yang digunakan untuk membentuk distribusi dari <i>negative sampling</i> .
<i>cbow_mean</i>	Jika 0, menggunakan jumlah dari vektor konteks kata. Jika 1, menggunakan rata-rata dari vektor konteks kata, dapat digunakan jika menggunakan CBOW.
<i>hashfxn</i>	Fungsi <i>hash</i> untuk menggunakan inisialisasi secara random dari bobot, untuk meningkatkan <i>training reproducibility</i> .
<i>iter</i>	Jumlah iterasi (<i>epochs</i>) dalam <i>corpus</i> .
<i>sorted_vocab</i>	Jika 1, melakukan pengurutan dari frekuensi yang paling rendah pada <i>vocabulary</i> sebelum menetapkan indeks dari kata.

Nama	Deskripsi
<i>trim_rule</i>	Aturan pemangkasan pada <i>vocabulary</i> , menentukan apakah kata-kata tertentu tetap berada pada <i>vocabulary</i> , dapat dipangkas, atau ditangani menggunakan <i>default</i> (dibuang jika jumlah kata $< min_count$).
<i>batch_words</i>	Menargetkan besaran (dalam kata-kata) untuk mengumpulkan contoh-contoh yang telah diloloskan ke <i>worker threads</i> .
<i>min_n</i>	Panjang minimum dari karakter <i>n-grams</i> yang akan digunakan untuk <i>training</i> representasi kata.
<i>max_n</i>	Panjang maksimum dari karakter <i>n-grams</i> yang akan digunakan untuk <i>training</i> representasi kata.
<i>word_ngrams</i>	Jika 1, memperkaya vektor kata dengan informasi <i>subword (n-grams)</i> .
<i>Bucket</i>	Karakter <i>n-grams</i> akan dilakukan <i>hash</i> menjadi angka yang telah ditetapkan dalam <i>buckets</i> .
<i>callbacks</i>	Kumpulan dari <i>callbacks</i> yang membutuhkan untuk dieksekusi/dijalankan di tahapan-tahapan yang spesifik dalam <i>training</i> .

Nama	Deskripsi
<i>compatible_hash</i>	Secara <i>default</i> , versi terbaru dari <i>Gensim's FastText</i> menggunakan fungsi <i>hash</i> yang 100% sesuai dengan <i>Facebook's FastText</i> .

Untuk penerapan metode WMD, dari *library gensim* memiliki parameter yang ditunjukkan pada Tabel 2. 2.

Tabel 2. 2 Parameter WMD

Nama	Deskripsi
<i>corpus</i>	Kumpulan dari dokumen, yang di dalamnya adalah kumpulan dari <i>token</i>
<i>model</i>	Model <i>word embedding</i> yang telah dilakukan <i>training</i> .
<i>num_best</i>	Jumlah yang akan dimunculkan
<i>normalize_w2v_and_replace</i>	Normalisasi pada vektor yang menjadikan panjangnya 1
<i>chunksizes</i>	Ukuran dari <i>chunk</i>

2.7 Python

Python adalah bahasa pemrograman yang interpretatif, *high-level*, dan *general-purpose*. Program *python* adalah kumpulan dari command untuk interpreter *python* mengeksekusi command tersebut. *Python* memiliki banyak *statements* untuk mengerjakan berbagai hal, seperti menampilkan keluaran di layar, mendapatkan

masukan dari pengguna, mengkalkulasi nilai dari ekspresi matematika, dan mengerjakan kumpulan dari *statements* secara terus-menerus [23]. *Python* memiliki banyak kegunaan, seperti *web development* [24], *data analysis* [25], *game development* [26], dan lain-lain.

2.8 Confusion Matrix

Confusion matrix, atau yang biasa disebut *error matrix* [27], adalah sebuah tabel yang menampilkan performansi dari sebuah algoritma, yang biasanya dipakai dalam *supervised learning* (Dalam *unsupervised learning*, biasanya disebut sebagai *matching matrix*). Setiap baris pada *matrix* merepresentasikan kelas *predicted*, sedangkan setiap kolom pada *matrix* merepresentasikan kelas *actual* [28]. Gambar 2.2 mengilustrasikan *confusion matrix*.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 2. 3 *Confusion Matrix*

TP = *True Positive*, data yang muncul dari uji coba di label positif

FN = *False Negative*, data yang tidak muncul dari uji coba di label positif

TN = *True Negative*, data yang tidak muncul dari uji coba di label negatif

FP = *False Positive*, data yang muncul dari uji coba di label negatif

2.9 Cohen's Kappa Coefficient

Cohen's Kappa Coefficient adalah metode statistik untuk menghitung *inter-rater reliability* (dan juga *intra-rater reliability*) untuk *item* yang kualitatif (kategorikal) [29]. Metode ini dikenal secara luas menjadi perhitungan yang lebih rumit dibandingkan dengan *simple percent agreement calculation*, karena koefisien memperhitungkan kemungkinan yang cocok dengan peluang.

Perhitungan akurasi menggunakan metode Kappa Cohen's. Metode tersebut memiliki persamaan (2.7).

$$\text{Cohen's Kappa Coefficient} = \frac{TP + TN}{(TP + FP + TN + FN)} \quad (2.7)$$

Cohen's Kappa Coefficient memiliki nilai dari 0 sampai 1, persebaran rentang tersebut dijelaskan pada tabel.

Tabel 2. 3 Tabel dari *Agreement*

Nilai Kappa	<i>Level of Agreement</i>	<i>% of Data that are Reliable</i>
0 – 0,2	<i>None</i>	0 – 4%
0,21 – 0,39	<i>Minimal</i>	4 – 15%

Nilai Kappa	<i>Level of Agreement</i>	<i>% of Data that are Reliable</i>
0,4 – 0,59	<i>Weak</i>	15 – 35%
0,6 – 0,79	<i>Moderate</i>	35 – 63%
0,8 – 0,9	<i>Strong</i>	64 – 81%
Above 0,9	<i>Almost Perfect</i>	82 – 100%

2.10 Precision dan Recall

Precision adalah rasio nilai TP dibagi dengan penjumlahan TP dan FP dimana TP adalah *True Positive* dan FP adalah *False Positive*. *Recall* adalah rasio nilai TP dibagi penjumlahan nilai TP dan FN dimana TP adalah *True Positive* dan FN adalah *False Negative* [30]. *Precision* dan *Recall* dapat diilustrasikan pada persamaan (2.8) dan persamaan (2.9).

$$Precision = \frac{TP}{TP + FP} \quad (2.8)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.9)$$

Dalam istilah yang sederhana, *precision* yang tinggi dapat diartikan bahwa suatu algoritma mengembalikan hasil yang jauh lebih relevan daripada yang tidak relevan, sedangkan *recall* yang tinggi dapat diartikan bahwa suatu algoritma mengembalikan sebagian hasil yang relevan.

2.11 *F-1 Score*

F-1 score adalah perhitungan dari akurasi tes. *F-1 score* menggunakan *precision* dan *recall* untuk dijadikan input dalam menghitung skor yang diraih.

F-1 score merupakan *harmonic mean* dari *precision* dan *recall* [31] yang dapat diilustrasikan pada persamaan (2.10).

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (2.1)$$

2.12 *Regular Expression*

Regular Expression, *regex* atau *regexp* (kadang-kadang disebut *rational expression*) adalah urutan dari *character* yang didefinisikan sebagai *search pattern* [32]. Biasanya, *regular expression* digunakan oleh algoritma *string searching* untuk “find” atau “find or replace” operasi dalam *string*, atau untuk validasi masukan. *Regular expression* sekarang diterapkan pada *search engine* dan digunakan dalam *text editor*.

Dalam Tugas Akhir ini, digunakan *regular expression* untuk mencari *string* dari *metadata* yang dibutuhkan pada Tugas Akhir ini. Implementasi dari *regular expression* pada Tugas Akhir ini adalah dengan memanggil *library re* dari bahasa pemrograman *python*.

2.13 *Multiprocessing*

Multiprocessing digunakan pada dua atau lebih *central processing units* (CPUs) dalam sistem komputer [33]. *Multiprocessing* juga diartikan sebagai kemampuan dari sistem untuk mendukung lebih dari satu prosesor atau kemampuan untuk mengalokasikan *tasks* di antara prosesor.

Dalam Tugas Akhir ini, *multiprocessing* digunakan untuk menambah *workers* untuk mengakses lebih dari satu *website* dalam satu waktu dalam melakukan *crawling data*. Implementasi dari *multiprocessing* pada Tugas Akhir ini adalah dengan memanggil *library multiprocessing* dari bahasa pemrograman *python*.

[Halaman ini sengaja dikosongkan]

BAB III

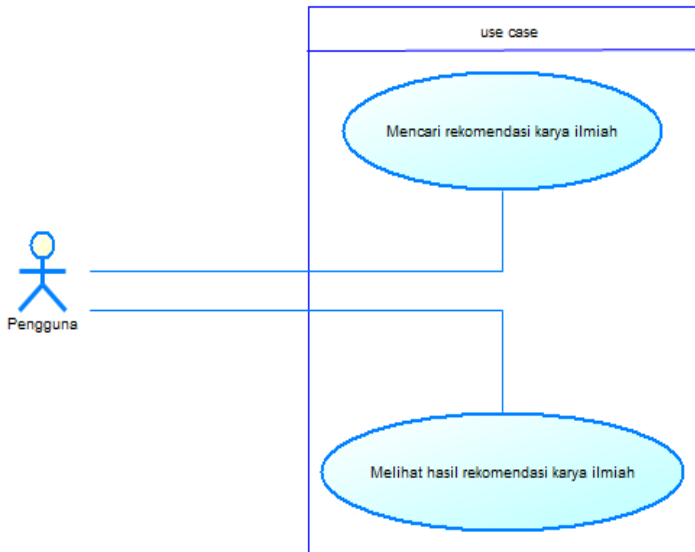
ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai analisis dan perancangan sistem tugas akhir.

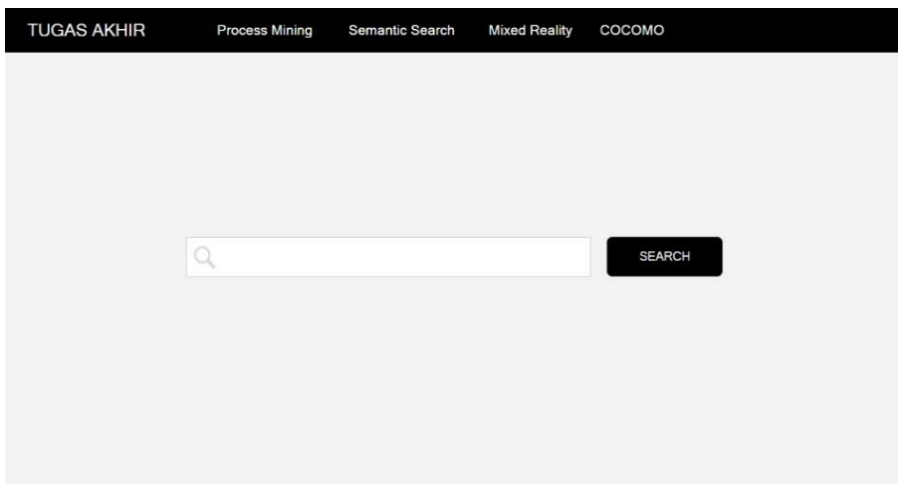
3.1 Analisis Metode Secara Umum

Pada tugas akhir ini dibangun suatu sistem yang dapat memberikan rekomendasi karya ilmiah menggunakan data karya ilmiah dari website *IEEEExplore*. Website *IEEEExplore* banyak digunakan oleh mahasiswa-mahasiswa yang berkuliah di Departemen Informatika ITS sebagai referensi untuk melakukan penulisan pada karya tulis, seperti hasil analisis untuk *final project* pada suatu mata kuliah dan penulisan buku Tugas Akhir. Selain itu, proses pengambilan *metadata* pada website *IEEEExplore* cukup lengkap sehingga dapat memudahkan pengambilan data yang dilakukan pada tugas akhir ini. Metode yang dikerjakan pada Tugas Akhir ini dibagi menjadi proses-proses. Pembagian proses-proses tersebut bertujuan untuk memudahkan pengerjaan pada Tugas Akhir ini. Gambar 3. 1 menunjukkan diagram *use case* yang bisa dilakukan pengguna ke sistem yang dibangun pada Tugas Akhir ini.

Tugas Akhir ini akan membangun *interface* yang akan menampilkan halaman awal dan halaman hasil rekomendasi. *Prototype interface* yang dibangun pada tugas akhir ini dapat dilihat pada Gambar 3. 2 dan Gambar 3. 3. Gambar 3. 2 menunjukkan tampilan awal untuk melakukan *query* dan Gambar 3. 3 menunjukkan tampilan hasil rekomendasi karya ilmiah.



Gambar 3. 1 Diagram *use case*

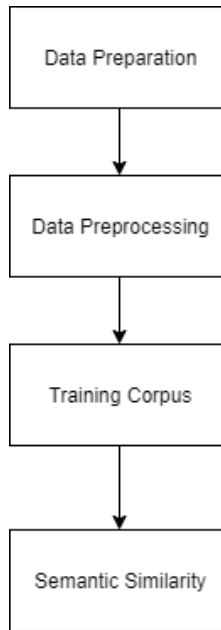


Gambar 3. 2 *Prototype interface* tampilan awal

TUGAS AKHIR		
Process Mining Semantic Search Mixed Reality COCOMO		
No	Title	Source
1	sample text	sample text
2	sample text	sample text
3	sample text	sample text
4	sample text	sample text
5	sample text	sample text
6	sample text	sample text

Gambar 3. 3 *Prototype interface* tampilan hasil rekomendasi

Proses-proses yang dilakukan dalam pengimplementasian sistem ini meliputi *data preparation*, *data preprocessing*, *training corpus* menggunakan konten dari keseluruhan data karya ilmiah meliputi *Title*, *Abstract*, *Topics*, dan *Keywords*, representasi kata menjadi vektor, dan *semantic similarity*. Penggunaan *Title*, *Abstract*, *Topics*, dan *Keywords* sebagai korpus untuk memperkaya representasi kata yang akan dijadikan vektor dengan menggunakan metode *fastText*. Proses-proses tersebut dilakukan secara berurutan. Gambaran secara umum untuk penerapan proses-proses yang dilakukan pada Tugas Akhir ini dapat diilustrasikan pada Gambar 3. 4. *Data preparation* dapat diilustrasikan pada Gambar 3. 5, *data preprocessing* dapat diilustrasikan pada Gambar 3. 6, *training corpus* dapat diilustrasikan pada Gambar 3. 7, dan *semantic similarity* akan dijelaskan pada Gambar 3. 8.



Gambar 3. 4 Bagan proses utama sistem

Pada tahap *data preparation* melakukan *crawling data* dari *website IEEEExplore*. Data yang dibutuhkan adalah *metadata* dari karya ilmiah yang tersedia pada *website IEEEExplore*. *Metadata* yang dibutuhkan pada karya ilmiah tersebut *Title, Abstract, Publication Name, DOI, Topics, dan Keywords*. *Metadata* tersebut digunakan untuk memberikan informasi-informasi secara lengkap pada karya ilmiah tersebut. Dari *website IEEEExplore*, tidak dapat diambil *metadata* dari isi karya ilmiah tersebut dikarenakan *metadata* tersebut tidak tersedia ketika diambil *source code* dari *website IEEEExplore*. *Metadata* yang diambil tersebut masih berupa data mentah. Oleh karena itu, diperlukan *data preprocessing* untuk menjadikan data tersebut menjadi data yang bersih.

Selanjutnya dilakukan *data preprocessing* pada hasil *data preparation*. *Data preprocessing* yang dilakukan pada tahapan ini untuk membersihkan *metadata* yang tidak diperlukan. Karena, hasil dari pengambilan data pada tahap *data preparation* masih memiliki *noise* dan terdapat data yang bukan data karya ilmiah. Tahapan ini dilakukan beberapa hal untuk melakukan *data preprocessing*. Hal-hal tersebut adalah melakukan *case folding (lowercase)*, menghapus *noise*, menghapus *stopwords* pada kolom *Topics*, memperbaiki *whitespace* pada *metadata* yang tersedia. *Case folding (lowercase)* dilakukan untuk mempermudah dalam menghilangkan *noise* yang terdapat pada *metadata* yang dikumpulkan pada tahap *data preparation*. Setelah dilakukan *case folding (lowercase)*, dilakukan penghapusan *noise* pada *metadata* yang dikumpulkan. *Noise* yang terdapat pada *metadata* yang dikumpulkan memiliki kriteria-kriteria sebagai berikut:

- Terdapat kolom yang tidak bernama karena hasil *crawling data*.
- *Metadata* dari artikel yang bukan karya ilmiah, seperti *author index*, *table of contents*, dan *keynotes*.
- Menghapus *metadata* yang memiliki nilai yang kosong pada salah satu kolom.
- Menghapus *metadata* yang terduplikasi.

Dari kriteria-kriteria tersebut dilakukan penghilangan *noise* sehingga membuat data menjadi bersih. Setelah itu, dilakukan kata *stopwords* Bahasa Inggris pada kolom *Topics*. *Stopwords* tersebut muncul karena tersedia dari website *IEEEExplore*. Terakhir, dilakukan penghilangan *whitespace* yang tidak diperlukan pada *metadata*. Penghilangan *whitespace* ini dikhususkan pada kolom *Topics* dan *Keywords* untuk merapikan isi dari kolom-kolom tersebut.

Pada tahap selanjutnya dilakukan *training corpus* untuk menghasilkan representasi vektor kata dan subkata menggunakan metode *fastText*. Metode *fastText* merupakan peningkatan dari metode *word2vec* dalam pembentukan vektor dari representasi

kata. Tahapan ini melakukan *data preprocessing* kembali. Perbedaan *data preprocessing* pada tahapan ini dibandingkan dengan tahapan sebelumnya adalah pada tahapan ini *data preprocessing* dilakukan hanya untuk membentuk representasi vektor kata dan subkata pada karya ilmiah.

Metadata yang diperlukan pada tahapan ini adalah *Title*, *Abstract*, *Topics*, dan *Keywords*. *Metadata-metadata* tersebut akan digabungkan dengan membuat kolom baru bernama *Content*. *Metadata* tersebut digunakan pada tahapan ini untuk menghasilkan konteks yang terkhususkan pada karya ilmiah yang terdapat pada data yang tersedia. Dari kolom ini, dilakukan *data preprocessing* yang memiliki tahapan-tahapan sebagai berikut:

- Melakukan *case folding* dengan mengubah huruf menjadi huruf kecil (*lowercase*).
- Menghapus tanda baca yang terdapat pada kolom tersebut.
- Menghapus angka yang terdapat pada kolom tersebut.
- Menghilangkan *stopwords* Bahasa Inggris pada kolom tersebut.
- Melakukan *lemmatization* untuk mengubah kata-kata yang terdapat pada kolom tersebut menjadi kata dasarnya.

Setelah selesai melakukan *data preprocessing*, dilakukan *training corpus* pada isi dari kolom *Content*. *Training corpus* dilakukan menggunakan *fastText* dari *library gensim* yang tersedia pada bahasa pemrograman *python*. Hasil *train* berupa kata beserta vektor kata sebesar 100 dimensi. Hasil tersebut akan disimpan dalam model *fastText* yang akan menjadi masukan pada tahapan selanjutnya.

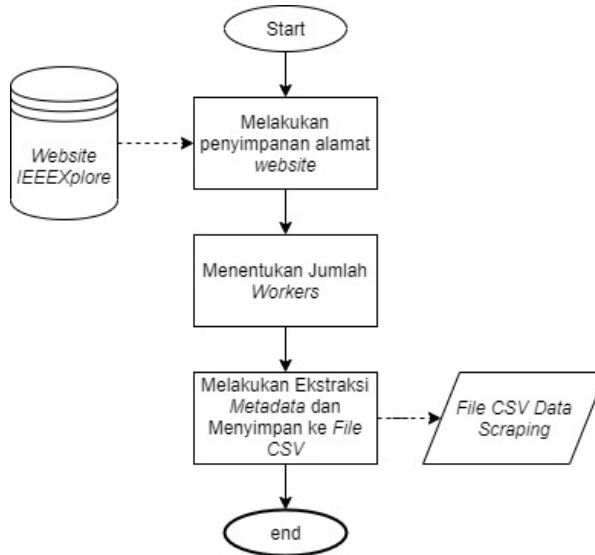
Tahapan terakhir ini adalah melakukan *semantic search* untuk memberikan rekomendasi karya ilmiah. Metode yang dilakukan pada tahapan ini adalah *Word Mover's Distance Similarity*. Metode ini merupakan metode yang dapat menangkap *semantic* dari makna kata. Metode ini merupakan peningkatan dari metode yang sudah menjadi standar pada metode *similarity*, yaitu

cosine similarity. Perbedaan dari metode ini dengan *cosine similarity* adalah metode *Word Mover's Distance Similarity* memindahkan kata di dalam sebuah kalimat ke kata di kalimat lain dan dihitung jarak perpindahannya. Karena memiliki cara kerja seperti itu, *Word Mover's Distance Similarity* dapat mengatasi permasalahan yang timbul dari limitasi *cosine similarity*. Limitasi tersebut adalah permasalahan sinonimitas. Oleh sebab itu, dengan menggunakan metode *Word Mover's Distance Similarity*, permasalahan sinonimitas dapat teratasi dengan baik. Selain itu, penerapan dari *Word Mover's Distance Similarity* menggunakan model vektor dari representasi makna kata, sehingga dapat menangkap similaritas secara semantic antar kata yang dibandingkan sesuai dengan vektor katanya. Penerapan metode *Word Mover's Distance Similarity* pada tugas akhir ini menggunakan data bersih karya ilmiah, model *fastText* dari data bersih karya ilmiah, dan *query* yang dilakukan oleh pengguna. Hasil dari tahapan ini adalah rekomendasi karya ilmiah yang sesuai dengan *query* yang dituliskan.

3.2 Data Preparation

Pada subbab 3.2 akan menjelaskan proses persiapan data. *Dataset* yang digunakan diambil dari *website IEEEExplore*. Data yang diambil berupa metadata yang tersedia di *website IEEEExplore*. Metadata yang disediakan di *website IEEEExplore* berjumlah cukup banyak, sehingga hanya diambil metadata yang penting dari karya ilmiah yaitu *Title*, *Abstract*, *Publication Name*, *DOI*, *Topics*, dan *Keywords*. Metadata yang tersedia dari *website IEEE* berbentuk JSON, sehingga hasil yang dibutuhkan untuk memasukkan data yang dikumpulkan ke tahap selanjutnya berbentuk CSV. Metadata diambil secara acak, sehingga tidak ditentukan harus berada di *topics* tertentu atau *keywords* tertentu, sehingga persebaran data cukup luas untuk dicari oleh pengguna. *Metadata* dari isi karya ilmiah tersebut tidak dapat diambil, karena tidak tersedianya *metadata* tersebut dari *source code website IEEEExplore* menggunakan *library* bahasa pemrograman *python*

yaitu *BeautifulSoup4* dan *requests*. Untuk mempercepat proses *crawling data*, digunakan *library multiprocessing* yang tersedia di bahasa pemrograman *python* untuk membuat *workers* yang dapat melakukan beberapa *crawling data* dalam waktu yang bersamaan. Untuk alur dari *data preparation* ini dapat diilustrasikan pada gambar yang ada di bawah ini. Gambar 3. 5 menunjukkan diagram alir dari tahapan proses *data preparation*.



Gambar 3. 5 Diagram alir *data preparation*

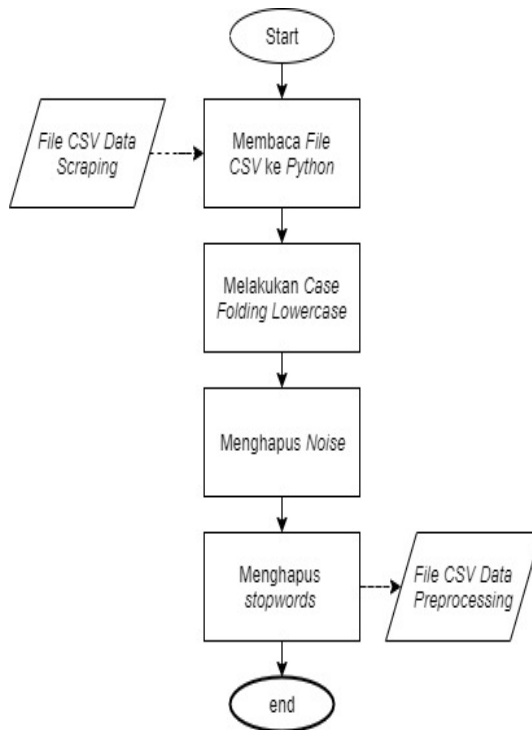
3.3 Data Preprocessing

Pada tahap ini dilakukan *data preprocessing* pada data karya ilmiah yang diambil dari *website IEEEExplore*. Proses ini meliputi pengecekan terlebih dahulu dari struktur data yang dilakukan apakah ada *noise* atau tidak. Dari hasil pengecekan, terdapat *noise* berupa data yang memiliki kolom yang tidak diketahui, isi dari data tersebut tidak sesuai dengan format, data yang kosong pada salah satu kolom, dan terdapat data yang tidak diinginkan masuk ke *dataset* yang dikumpulkan. Sehingga, dilakukan penghilangan

kolom yang tidak sesuai dengan format yang ditentukan, penghilangan kolom yang tidak diketahui, penghilangan data dengan isi yang tidak sesuai dengan format, penghilangan data yang memiliki data kosong pada salah satu kolomnya, dan penghilangan *row* data yang tidak dibutuhkan. Dapat dilihat pada contoh gambar yang ada di bawah ini.

Selanjutnya, dilakukan *case folding* yaitu membuat huruf menjadi huruf kecil (*lowercase*) pada semua kolom kecuali kolom DOI, dilakukan penghilangan *stopwords* berbahasa inggris pada kolom *topics*, dan melakukan *lemmatization* pada kolom *topics*. Terakhir, memperbaiki struktur kata pada kolom dengan melakukan *string join* pada data di setiap kolom sehingga dapat menghilangkan *whitespace* yang tidak perlu.

Input dari *data preprocessing* adalah *file CSV*. *Output* dari *data preprocessing* juga berbentuk *file CSV*. Gambar 3. 6 menunjukkan diagram alir dari tahapan proses *data preprocessing*.



Gambar 3. 6 Diagram alir *data preprocessing*

3.4 *Training Corpus*

Training corpus dilakukan untuk mencari vektor dari representasi kata. *Training corpus* yang dilakukan pada Tugas Akhir ini menggunakan data bersih karya ilmiah dari tahapan sebelumnya yaitu *data preprocessing*. Langkah pertama yang dilakukan adalah menggabungkan *metadata* yang diperlukan untuk melakukan *training corpus*. *Metadata* tersebut adalah *Title*, *Abstract*, *Topics*, dan *Keywords*. Hasil dari penggabungan *metadata* tersebut akan dibentuk kolom baru bernama *Content*. Kolom *Content* ini masih berbentuk data mentah yang belum dilakukan *data preprocess*. Data mentah ini akan *data*

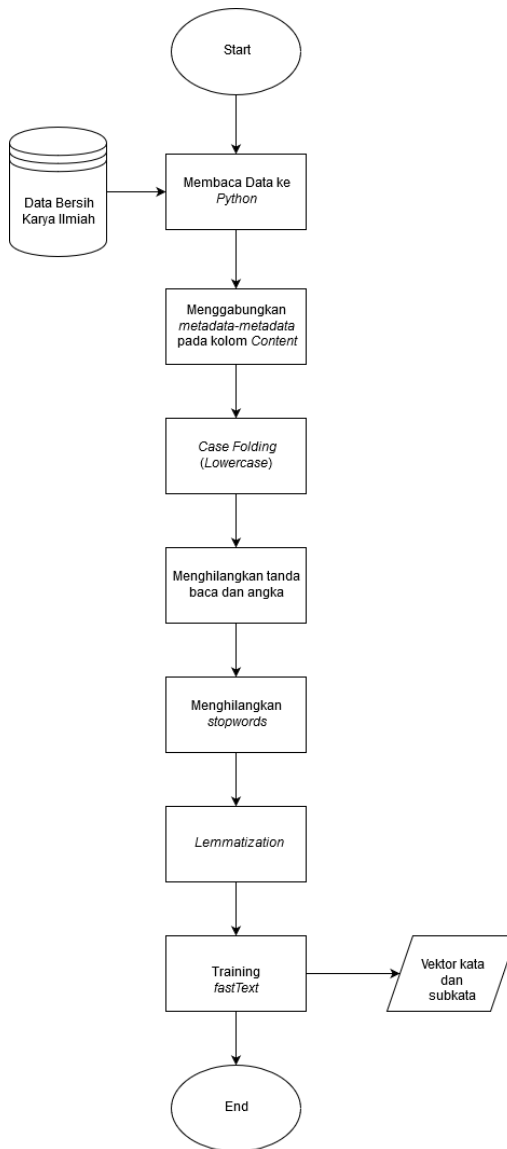
preprocessing kembali untuk menjadi masukkan dari *training corpus*. Kebutuhan yang dibutuhkan pada *data preprocessing* yang dilakukan pada tahapan ini meliputi:

1. Melakukan *case folding* dengan mengubah huruf menjadi huruf kecil (*lowercase*).
2. Menghapus tanda baca yang terdapat pada kolom tersebut.
3. Menghapus angka yang terdapat pada kolom tersebut.
4. Menghilangkan *stopwords* Bahasa Inggris pada kolom tersebut.
5. Melakukan *lemmatization* untuk mengubah kata-kata yang terdapat pada kolom tersebut menjadi kata dasarnya.

Case folding dilakukan untuk mengubah huruf menjadi huruf kecil (*lowercase*). Langkah ini dilakukan untuk menghindari *case sensitive* antar kata, sehingga tidak terjadinya duplikasi makna kata. Selanjutnya adalah menghapus tanda baca yang terdapat pada kolom tersebut. Tanda baca tidak dibutuhkan dalam melakukan *training corpus*, jadi dilakukan penghapusan tanda baca tersebut untuk meraih hasil representasi yang sesuai. Setelah itu, dilakukan penghapusan angka yang terdapat pada kolom tersebut. Angka tidak dibutuhkan juga pada *training corpus*, karena dikhawatirkan akan mempengaruhi vektor dari representasi makna kata pada *training corpus*. Begitu juga dengan *stopwords* Bahasa Inggris, dilakukan penghapusan karena frekuensi dari *stopwords* tersebut dapat mempengaruhi vektor representasi makna kata. Terakhir, dilakukan *lemmatization* untuk mengubah kata-kata yang terdapat pada kolom tersebut menjadi kata dasarnya. Langkah ini dilakukan karena terdapat kata-kata yang memiliki makna yang sama tetapi dalam bentuk yang berbeda. *Lemmatization* yang diterapkan pada *data preprocessing* ini menggunakan Bahasa Inggris. Sehingga, tidak terjadinya duplikasi makna pada kata yang sama dalam bentuk yang berbeda.

Setelah dilakukan *data preprocessing* pada tahapan ini, maka selanjutnya akan melakukan *training corpus* dengan menggunakan metode *fastText*. Metode ini diimplementasikan dengan

menggunakan *library gensim* dari bahasa pemrograman *python*. *Training* yang dilakukan menggunakan iterasi sebanyak lima kali untuk menghasilkan vektor representasi kata yang lebih lengkap. Metode ini menggunakan model *Skip-Gram*. Hasil dari *training corpus* pada tahapan ini adalah vektor kata dan subkata dari kolom *Content* pada langkah sebelumnya. Kolom *Content* digunakan sebagai masukan dari metode *fastText* ini untuk memberikan makna yang sesuai dengan karya ilmiah tersebut. Jika menggunakan *corpus* yang lebih umum, hasilnya akan menjadi lebih rendah dibanding menggunakan *corpus* dari kolom *Content*. Hasil dari *training corpus* ini akan menjadi masukan dari tahapan selanjutnya untuk mencari rekomendasi dari *query* yang dimasukkan oleh pengguna. Gambar 3. 7 menunjukkan diagram alir dari tahapan proses *training corpus*.



Gambar 3. 7 Diagram alir *training corpus*

3.5 *Semantic Similarity*

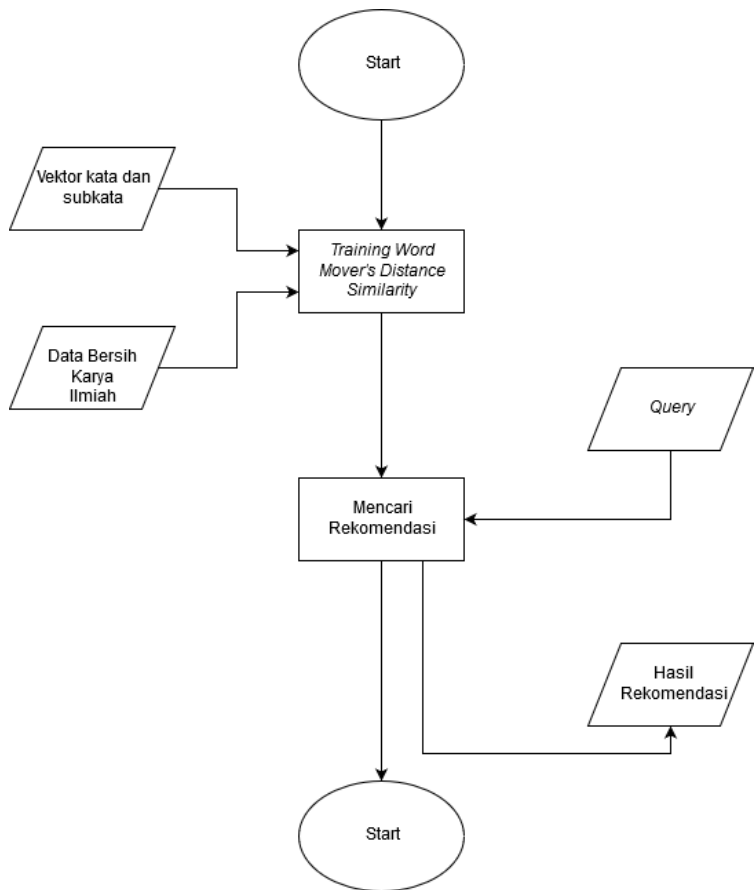
Tahap terakhir ini adalah melakukan metode *semantic search* untuk menghasilkan rekomendasi karya ilmiah pada pengguna dengan menggunakan *query* yang dimasukkan.

Langkah awal dari tahapan ini adalah melakukan *training* metode *Word Mover's Distance Similarity*. Masukkan yang diperlukan pada langkah *training* ini adalah data bersih karya ilmiah dari tahapan *data preprocessing* dan vektor kata dan subkata dari tahapan *training corpus*. Selain itu, menentukan berapa jumlah rekomendasi yang dikeluarkan dari metode ini. Hasil *training* pada langkah ini berbentuk model yang akan dijadikan fungsi untuk menghasilkan rekomendasi karya ilmiah yang sesuai.

Selanjutnya, dapat memasukkan *query* yang dimasukkan oleh pengguna. *Query* tersebut akan dilakukan *data preprocessing* yang sama seperti pada tahapan *training corpus* untuk mencari kata-kata yang sesuai. *Data preprocessing* yang dilakukan pada tahapan ini adalah sebagai berikut:

1. Melakukan *case folding* dengan mengubah huruf menjadi huruf kecil (*lowercase*).
2. Menghapus tanda baca yang terdapat pada *query*.
3. Menghapus angka yang terdapat pada *query*.
4. Menghapus *stopwords* Bahasa Inggris yang terdapat pada *query*.
5. Melakukan *lemmatization* untuk mengubah kata-kata yang terdapat pada *query* menjadi kata dasarnya.

Setelah dilakukan *data preprocessing*, *query* tersebut akan menjadi masukkan ke model yang dibentuk pada langkah sebelumnya. Hasil dari langkah ini adalah rekomendasi karya ilmiah. Tahapan ini dapat diilustrasikan pada gambar yang ada di bawah ini. Gambar 3. 8 menunjukkan diagram alir dari tahapan proses *semantic search*.



Gambar 3. 8 Diagram alir *semantic search*

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Bab ini menjelaskan tentang implementasi pada tugas akhir ini, yaitu tahapan – tahapan dalam proses pengerjaan tugas akhir.

4.1 Lingkungan Implementasi

Lingkungan implementasi sistem yang digunakan untuk mengembangkan tugas akhir memiliki spesifikasi perangkat keras dan perangkat lunak seperti yang ditampilkan pada Tabel 4. 1.

Tabel 4. 1 Lingkungan Implementasi

Perangkat	Spesifikasi
Perangkat Keras	A. Laptop <ul style="list-style-type: none">• Prosesor: Intel® Core™ i7• Memori: 16GB B. Komputer <ul style="list-style-type: none">• Prosesor: Intel® Core™ i7• Memori: 24GB
Perangkat Lunak	A. Sistem Operasi Windows B. Jupyter C. <i>Terminal</i> D. Microsoft Word E. Microsoft Excel

Lingkungan implementasi untuk mengerjakan tugas akhir ini menggunakan *Windows 10* untuk melakukan semua kegiatan, dimulai dari pembuatan program sampai dengan dokumentasi tugas akhir. Adapun pembuatan program dilakukan dengan menggunakan *tools* Jupyter dengan menggunakan bahasa pemrograman *Python*. Dokumentasi dari tugas akhir ini akan menggunakan *Microsoft Word* dan *Microsoft PowerPoint*. Adapun

tools yang akan digunakan dalam menjalankan *script* pembuatan model dari tugas akhir ini terdapat pada Tabel 4.2.

Tabel 4. 2 *Tools* Pengerjaan

No	<i>Tools</i>	Deskripsi
1	<i>Python 3.7</i>	Bahasa Python digunakan sebagai bahasa pemrograman untuk membangun sistem.
2	<i>gensim</i>	<i>Library</i> Python untuk pemodelan topik yang digunakan untuk <i>unsupervised topic modelling</i> dan <i>natural language processing</i> , menggunakan metode statistika modern <i>machine learning</i> . <i>Library</i> ini digunakan untuk mengimplementasi metode representasi kata dan <i>semantic similarity</i> .
3	<i>BeautifulSoup4</i>	<i>Library</i> Python yang digunakan untuk <i>crawling dataset</i> dari website <i>IEEEExplore</i> .
4	<i>multiprocessing</i>	<i>Library</i> yang didukung untuk membuat proses-proses dengan menggunakan API yang mirip dengan modul <i>threading</i> . <i>Library</i> ini digunakan untuk membuat banyak proses dalam melakukan <i>crawling data</i> .
4	<i>requests</i>	<i>Library requests</i> adalah <i>library</i> yang digunakan pada bahasa pemrograman <i>python</i> untuk membuat HTTP <i>requests</i> .
5	<i>re</i>	<i>Library re</i> adalah <i>library</i> yang digunakan pada bahasa pemrograman <i>python</i> untuk

No	Tools	Deskripsi
		menspesifikasikan kumpulan dari <i>string</i> dan menyesuaikannya.
	<i>string</i>	<i>Library string</i> adalah <i>library</i> yang digunakan pada bahasa pemrograman <i>python</i> yang menampung kelas dan fungsi yang berguna untuk melakukan metode <i>string</i> .
	<i>nltk</i>	<i>Library nltk</i> adalah <i>library</i> yang digunakan pada bahasa pemrograman <i>python</i> untuk melakukan komputasi linguistic pada bahasa pemrograman <i>python</i> .
	<i>pandas</i>	<i>Library pandas</i> adalah <i>library</i> yang digunakan pada bahasa pemrograman <i>python</i> untuk memanipulasi dan analisis data.

4.2 Implementasi Proses

Implementasi proses merupakan tahapan implementasi dari perancangan proses yang telah dijelaskan sebelumnya pada bab analisis dan perancangan sistem.

4.2.1 Data Preparation

Subbab ini akan membahas implementasi tahapan untuk *data preparation*, yaitu mengambil data dari *website IEEEExplore* menggunakan *library BeautifulSoup4*, *request*, *time*, *string*, *re*, dan *multiprocessing*. Kode Sumber 4. 1 menunjukkan inisialisasi *library BeautifulSoup4*, *requests*, *time*, *string*, *re*, dan *multiprocessing* beserta alamat *website* yang dituju dan iterasi yang akan dilakukan. *Library requests* digunakan untuk mengambil *source code* dari alamat *website* yang dimasukkan. *Library BeautifulSoup4* digunakan untuk membaca hasil *source*

code yang didapatkan melalui *library requests*. Selanjutnya, *library re* digunakan untuk mengekstraksi fitur dari *metadata* yang dibaca oleh *library BeautifulSoup4*. *Library time* dan *library string* merupakan *library* pelengkap untuk menghitung berapa lama eksekusi waktu yang dibutuhkan dalam menjalankan *code* yang dibangun dan mengaktifkan fungsi *string* pada bahasa pemrograman *python*.

```

1. from bs4 import BeautifulSoup
2. from multiprocessing import Pool
3. import multiprocessing
4. import requests as r
5. import time
6. import re
7. import string
8.
9. all_urls = list()
10. first = 6000000
11. second = 9000001
12. # x = 0
13. url = "https://ieeexplore.ieee.org/document/"

```

Kode Sumber 4. 1 Inisialisasi dasar

Informasi karya ilmiah yang diperoleh dengan mencantumkan *id* dari karya ilmiah tersebut dalam rentang 6,000,000 — 9,000,000. Kode Sumber 4. 2 menunjukkan pembuatan dari fungsi memasukkan *id* dari karya ilmiah ke alamat *website*.

```

1. def generate_url():
2.     # loop =
3.     for x in range(first,second):
4.         all_urls.append(url + str(x))

```

Kode Sumber 4. 2 Penyimpanan alamat *website*

Iterasi yang dilakukan dalam rentang 6,000,000 – 9,000,000 akan dimasukkan ke alamat website dengan membuat *list* alamat website tersebut untuk dilakukan *data scraping*. Kode Sumber 4.3 menunjukkan pembuatan fungsi untuk mengekstraksi fitur yang terdapat pada website *IEEEExplore*.

```

1. def go_scrapp(url):
2.     try:
3.         page = r.get(url)
4.         pages = page.content
5.         soup = BeautifulSoup(pages, 'html.parser')
6.         # print(soup.prettify())
7.         title = soup.find("title").get_text()
8.         if title != ' - ':
9.             try:
10.                data = soup.find_all("script")[9].g
11.                et_text()
12.                # print(data)
13.                publications = re.findall('"display
PublicationTitle": "(.+?)"', data)
14.                publication = publications[0].repla
ce(",", "")
15.                topics = re.findall('"name": "(.+?)"
', data)
16.                topics = topics[-3:]
17.                topic = " ".join(topics)
18.                topic = topic.replace(',', ' ')
19.                dois = re.findall('"doi": "(.+?)"',
data)
20.                doi = dois[0]
21.                doi = doi.replace(',', ' ')
22.                titles = re.findall('"title": "(.+?)
"', data)
23.                title = titles[0].replace(",", "")
24.                keywords = re.findall('"kwd": "(.+?)'
', data)
25.                keyword = " , ".join(keywords)
26.                keyword = keyword.translate(str.mak
etrans(string.punctuation, '' * len(string.punctuati
on)))

```

```

26.         abstracts = re.findall("abstract":
27.             "(.+)\"", data)
28.         abstract = abstracts[1].replace(", "
29.             , "")
30.         csv = title + "," + abstract + ","
31.         + publication + "," + doi + "," + topic + "," + key
32.         word
33.         f = open('Hasil2.csv', 'a+', encodi
34.             ng="utf-8")
35.         f.write(csv + '\n')
36.         f.close()
37.     except:
38.         pass
39.     else:
40.         pass
41. except:
42.     print('Error!')

```

Kode Sumber 4. 3 Ekstrasi Data

Fungsi yang dibangun akan mengekstraksi metadata yang dibutuhkan untuk pembangunan sistem rekomendasi pada tugas akhir ini. *Metadata* yang didapatkan dari alamat *website* tersebut akan ditunjukkan pada lampiran yang menampilkan salah satu contoh artikel yang telah berhasil dilakukan *crawling data* menggunakan fungsi tersebut. Dari *metadata* yang diekstraksi dari *website IEEEExplore* adalah *title*, *abstract*, *publication name*, *DOI*, *topics*, dan *keywords*. Data yang diambil menggunakan *library requests* dan *BeautifulSoup4* dalam bentuk *file* JSON. Sehingga, setelah diekstraksi data dari *file* JSON, data yang menjadi *output* dari fungsi tersebut akan ditulis dalam bentuk *file* CSV. Kode Sumber 4.4 akan menunjukkan fungsi *main* untuk menjalankan *script* secara keseluruhan.

```

1. if __name__ == '__main__':
2.     start_time = time.time()
3.     generate_url()
4.     # print(all_urls)
5.     # for x in all_urls:

```

```

6.     #     go_scrapp(x)
7.     #     print(str(first) + " dari " + str(second-
      1))
8.     #     first = first + 1
9.     #     y = y + 1
10.    # print(all_urls)
11.    p = Pool(multiprocessing.cpu_count()-1)
12.    p.map(go_scrapp, all_urls)
13.    p.close()
14.    p.join()
15.    print("%s seconds of execute time" % (time.time
      ()-start_time))

```

Kode Sumber 4. 4 Implementasi Proses

Fungsi *main* menghitung waktu yang dieksekusi dalam menjalankan *script* untuk *data scraping* dari *website IEEEExplore*. Setelah itu, memanggil fungsi *generate_url()* untuk melakukan *input* alamat *website* dan iterasi yang dilakukan dalam rentang 6,000,000 – 9,000,000. Selanjutnya, memakai fungsi *Pool* dari *library multiprocessing* untuk menciptakan *workers* yang bekerja dalam melakukan *data scraping*. Jumlah dari *workers* yang tersedia adalah banyaknya jumlah *cores* dalam CPU dikurangi dengan satu. Tahap akhir dari *script* ini adalah menjalankan *workers* yang sudah diinisiasi dalam tahap sebelumnya dan menjalankan fungsi *go_scrapp* untuk melakukan pengambilan *metadata* dari *website IEEEExplore*. Salah satu contoh hasil dari fungsi tersebut dapat dilihat pada Gambar 4. 1 yang ada di bawah ini

	Title	Abstract	Publication Name	DOI	Topics	Keywords
0	Numerical Simulation of the Chua's Oscillator ...	The article presents the results of numerical ...	2018 9th International Conference on Ultrawide...	10.1109/UWBUSIS.2018.8520001	Fields Waves and Electromagnetics Geoscience S...	Oscillators MOSFET Mathematical model ...

Gambar 4. 1 Tampilan Hasil Ekstraksi *Metadata*

4.2.2 Data Preprocessing

Data preprocessing dilakukan dengan langkah pertama yaitu mengecek apakah terdapat noise pada data yang telah diambil. Pada Kode Sumber 4.5 untuk menunjukkan inisialisasi library pandas untuk membaca file CSV dari hasil data scraping.

```
1. import pandas as pd
2. df = pd.read_csv('Scraping IEEE.csv', encoding = "ISO-8859-1")
```

Kode Sumber 4. 5 Membaca masukkan data

Hasil dari kode sumber tersebut dapat dilihat pada Gambar 4. 2 yang ada di bawah ini.

	title	abstract	publication name	DOI	topics	keywords	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	...
0	adaptive visual symbols for personal health re...	as a hub of information controlled by the pati...	2011 15th international conference on informat...	10.1109/IV/2011.87	computing and processing	visualization medical services adaptation mode...	NaN	NaN	NaN	NaN	...
1	the hilbert transform of the magnetic anomalie...	usually the field measurements of magnetic pro...	2011 international conference on multimedia te...	10.1109/ICMT.2011.6002002	signal processing and analysis communication n...	transforms geology magnetic field measurement ...	NaN	NaN	NaN	NaN	...
2	the simulation of direct current control about...	the further development of power electronics p...	2011 2nd international conference on artificia...	10.1109/AIMSEC.2011.6010010	signal processing and analysis computing and p...	harmonic analysis power system harmonics react...	NaN	NaN	NaN	NaN	...
3	reduced complexity online sparse signal recons...	this paper presents a novel online method for ...	2011 17th international conference on digital ...	10.1109/ICDSP.2011.6005005	signal processing and analysis communication n...	current measurement convergence computational ...	NaN	NaN	NaN	NaN	...
4	a monitoring and audit logging architecture fo...	current cloud infrastructures have opaque serv...	2011 ieee international symposium on parallel ...	10.1109/IPDPS.2011.304	computing and processing communication network...	monitoring security ip networks cloud computin...	NaN	NaN	NaN	NaN	...

Gambar 4. 2 Tampilan Dari Hasil Crawling Data

Selanjutnya, Kode Sumber 4.6 menunjukkan pengimplementasian case folding membuat huruf menjadi huruf kecil (lowercase).

```
1. list = ['title', 'abstract', 'publication name', 't
      topics', 'keywords']
```

```

2. for x in list:
3.     df[x] = df[x].str.lower()

```

Kode Sumber 4. 6 Melakukan *case folding*

Setelah itu, Kode Sumber 4. 7 menunjukkan untuk pengecekan apakah terdapat *noise* atau tidak. Dari hasil tersebut, terdapat *noise* berupa data yang memiliki kolom yang tidak diketahui, isi dari data tersebut tidak sesuai dengan format, data yang kosong pada salah satu kolom, dan terdapat data yang tidak diinginkan masuk ke *dataset* yang telah dikumpulkan.

```

1. df.head()
2. df.info()
3. df = df.loc[:, ~df.columns.str.contains('^Unnamed')
   ]
4. df = df[~df.keywords.str.contains(';', na=False)]
5. df.dropna(axis = 0, how = 'any', inplace=True)
6. df = df[~df.title.str.contains("\[", na=False)]
7. df = df[~df.title.str.contains("keynotes", na=False
   )]
8. df = df[~df.title.str.contains("author index", na=F
   alse)]
9. df = df[~df.title.str.contains("table of", na=False
   )]
10. df.count()

```

Kode Sumber 4. 7 Menghapus *Noise*

Pada Kode Sumber 4. 8, ditunjukkan cara penghapusan *stopwords* berbahasa inggris pada kolom *topics* dan *keywords*.

```

1. from nltk.corpus import stopwords
2.
3. stopword = stopwords.words('english')
4. stopword = set(stopword)
5.
6. list_stopwords = []
7. for x in stopword:

```

```

8.     list_stopwords.append(x)
9.
10. len(df.topics)
11.
12. for x in range(len(df['topics'])):
13.     df['topics'].iloc[x] = ' '.join([word for word
    in df['topics'].iloc[x].split() if word not in list
    _stopwords])
14.
15. for x in range(len(df['keywords'])):
16.     df['keywords'].iloc[x] = ' '.join([word for wor
    d in df['keywords'].iloc[x].split() if word not in
    list_stopwords])

```

Kode Sumber 4. 8 Menghapus *stopwords*

Dari kode sumber tersebut dilakukan pemanggilan *library nltk* untuk melakukan *import stopwords*. Selanjutnya, melakukan penentuan *stopwords* Bahasa Inggris. Setelah itu, dilakukan penghapusan *stopwords* Bahasa Inggris pada kolom *topics* dan *keywords*. Pada Kode Sumber 4. 9, menunjukkan untuk menyimpan hasil *data preprocessing* menjadi *file* CSV.

```

1. df.to_csv('Scraping IEEE Cleaned.csv', index=False)

```

Kode Sumber 4. 9 Menyimpan *file* CSV

Gambar 3. 9 menampilkan hasil akhir keluaran dari *data preprocessing* sebelum disimpan dalam *file* CSV.

	title	abstract	publication name	DOI	topics	keywords
0	adaptive visual symbols for personal health re...	as a hub of information controlled by the pati...	2011 15th international conference on informat...	10.1109/IV.2011.87	computing processing	visualization medical services adaptation mode...
1	the hilbert transform of the magnetic anomalie...	usually the field measurements of magnetic pro...	2011 international conference on multimedia te...	10.1109/ICMT.2011.6002002	signal processing analysis communication netwo...	transforms geology magnetic field measurement ...
2	the simulation of direct current control about...	the further development of power electronics p...	2011 2nd international conference on artificia...	10.1109/AIMSEC.2011.6010010	signal processing analysis computing processin...	harmonic analysis power system harmonics react...
3	reduced complexity online sparse signal recons...	this paper presents a novel online method for ...	2011 17th international conference on digital ...	10.1109/ICDSP.2011.6005005	signal processing analysis communication netwo...	current measurement convergence computational ...
4	a monitoring and audit logging architecture fo...	current cloud infrastructures have on-line serv...	2011 ieee international symposium on parallel ...	10.1109/IPDPS.2011.304	computing processing communication networking	monitoring security ip networks cloud computin...

Gambar 3. 9 Hasil Keluaran *Data Preprocessing*

4.2.3 Training Korpus

Pada subbab ini, dilakukan *training* korpus dari metadata yang sudah dilakukan *data preprocessing*. Pada Kode Sumber 4. 10, untuk menunjukkan inisialisasi *library pandas* dan *time* untuk membaca *file CSV* hasil dari *data preprocessing* dan menampilkan waktu hasil eksekusi dari *script* yang dijalankan.

```
1. import pandas as pd
2. from time import time
```

Kode Sumber 4. 10 Inisialisasi *library*

Pada Kode Sumber 4. 11, untuk menunjukkan pembacaan dari *file CSV*.

```
1. path = "../Preprocessing/Scraping IEEE Cleaned.csv"
2. df = pd.read_csv(path, encoding='ISO-8859-1')
```

Kode Sumber 4. 11 Membaca *File CSV*

Pada Kode Sumber 4. 12, untuk menunjukkan inisialisasi *fastText* dari *library gensim*.

```
1. from gensim.models import FastText as ft
```

Kode Sumber 4. 12 Inisialisasi model

Pada Kode Sumber 4. 13, menunjukkan untuk membuat fungsi *preprocessing* dan *lemmatization* untuk melakukan *preprocessing* dan *lemmatization* pada korpus dan *query*.

```
1. import nltk
2. from nltk import word_tokenize
3. from nltk.corpus import stopwords
4. from nltk.stem import WordNetLemmatizer
5.
6. lemmatizer=WordNetLemmatizer()
7. stop_words = stopwords.words('english')
8.
9. def preprocess(doc):
10.     doc = doc.lower() # Lower the text.
11.     doc = word_tokenize(doc) # Split into words.
12.     doc = [w for w in doc if not w in stop_words]
13.     # Remove stopwords.
14.     doc = [w for w in doc if w.isalpha()] # Remove
15.     # numbers and punctuation.
16.     return doc
17.
18. def lemmatize(doc):
19.     doc_lemma = list()
20.     for word in doc:
21.         doc_lemma.append(lemmatizer.lemmatize(word))
22.     return doc_lemma
```

Kode Sumber 4. 13 Fungsi *Preprocessing*

Pada Kode Sumber 4. 14 dilakukan pembuatan kolom baru bernama *content* yang isinya berupa penggabungan *metadata* dari

title, *abstract*, *topics*, dan *keywords*. Selanjutnya dilakukan *preprocessing* dan *lemmatization* pada isi kolom *content*.

```

1. df['content'] = df[['title', 'abstract', 'topics',
    'keywords']].apply(lambda x: ' '.join(x), axis = 1)
2. content = df['content'].to_list()
3. count = 0
4.
5. for text in content:
6.     corpus.append(preprocess(text))
7.     count = count + 1
8.     print(count)
9.
10. for text in content:
11.     corpus.append(preprocess(text))
12.     count = count + 1
13.     print(count)

```

Kode Sumber 4. 14 *Preprocessing Content*

Pada Kode Sumber 4. 15, dilakukan *training* korpus dengan menggunakan korpus yang telah dilakukan *preprocessing* dan *lemmatization* pada tahap sebelumnya.

```

1. model_ft= ft(corpus_lemma, sg=1, workers=multiproc
    ssing.cpu_count()-
    1, size=100, iter=15, min_count=10)
2.

```

Kode Sumber 4. 15 Model *fastText*

4.2.4 *Semantic Search*

Subbab ini menjelaskan tentang pengimplementasian algoritma untuk mencari similarity antara *query* dan dokumen yang telah dilatih dengan model training korpus.

Pada Kode Sumber 4. 16, dilakukan pembangunan algoritma *Word Mover's Distance Similarity* untuk mencari similarity antara *query* dan dokumen karya ilmiah dari hasil *output data preprocessing* pada subbab sebelumnya. Algoritma *Word Mover's Distance Similarity* diimplementasikan melalui *library gensim*.

```
1. from gensim.similarities import WmdSimilarity
2. instance_ft = WmdSimilarity(corpus_lemma, model_ft_
   no_freq, num_best=10)
```

Kode Sumber 4. 16 Inisialisasi *Similarity*

Pada Kode Sumber 4. 17, dimasukkan *query* untuk dilakukan *data preprocessing* terlebih dahulu sebelum dilakukan untuk mencari *similarity* antara *query* dan dokumen.

```
1. sentence = input()
2. query = preprocess(sentence)
3. query = lemmatize(query)
```

Kode Sumber 4. 17 Inisialisasi *Query*

Pada Kode Sumber 4. 18, dimasukkan *query* pada algoritma *Word Mover's Distance Similarity* untuk mencari *similarity* antara *query* dan dokumen agar menghasilkan rekomendasi dokumen dengan *similarity* yang tinggi dari *query*.

```
1. sims_w2v = instance_w2v[query]
2. sims_ft = instance_ft[query]
```

Kode Sumber 4. 18 *Similarity*

Pada Kode Sumber 4. 19, dapat ditampilkan hasil rekomendasi dari *query* yang dituliskan oleh pengguna yang memiliki *semantic similarity* dari *query* yang dituliskan oleh pengguna.

```

1. print('Query:')
2. print(sentence)
3. for i in range(num_best):
4.     print(df.iloc[sims_ft[i][0]])

```

Kode Sumber 4. 19 Hasil Rekomendasi

Kode Sumber 4. 19 menampilkan hasil rekomendasi dari *query* yang diinputkan oleh pengguna. Keluaran yang dihasilkan berupa *metadata* dari karya ilmiah dikumpulkan pada *data preparation*. Kode tersebut menggunakan *index* dari karya ilmiah yang berada pada hasil *semantic similarity* kemudian dicocokkan dengan *index* dari *metadata* karya ilmiah tersebut yang disimpan dalam *dataframe*. Hasil tersebut dapat dilihat pada Gambar 4. 3.

	title	abstract	publication name	DOI	topics	keywords	content
275769	from big data to big service	big service-the convergence and collaboration ...	computer	10.1109/MC.2015.182	computing processing	big data cloud computing meteorology convergen...	from big data to big service big service-the C...
130351	privacy-preserving big data stream mining. opp...	this paper explores recent achievements and no...	2017 ieee international conference on data min...	10.1109/ICDMW.2017.140	computing processing general topics engineers	big data data privacy privacy publishing traje...	privacy-preserving big data stream mining. opp...
141497	privacy-preserving big data stream mining. opp...	this paper explores recent achievements and no...	2017 ieee international conference on data min...	10.1109/ICDMW.2017.140	computing processing general topics engineers	big data data privacy privacy publishing traje...	privacy-preserving big data stream mining. opp...
130351	privacy-preserving big data stream mining. opp...	this paper explores recent achievements and no...	2017 ieee international conference on data min...	10.1109/ICDMW.2017.140	computing processing general topics engineers	big data data privacy privacy publishing traje...	privacy-preserving big data stream mining. opp...
141497	privacy-preserving big data stream mining. opp...	this paper explores recent achievements and no...	2017 ieee international conference on data min...	10.1109/ICDMW.2017.140	computing processing general topics engineers	big data data privacy privacy publishing traje...	privacy-preserving big data stream mining. opp...

Gambar 4. 3 Hasil Rekomendasi

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini akan menjelaskan hasil pengujian dan evaluasi sistem yang telah dirancang untuk mengetahui kinerja sistem.

5.1 Lingkungan Uji Coba

Lingkungan pengujian sistem pada pengerjaan tugas akhir ini dilakukan pada lingkungan dan kaskas bantu sebagai berikut:

- Prosesor Intel® Core™ i7
- *RAM 24GB*
- Jenis *Device Personal Computer* (PC)
- Sistem Operasi Windows 10

Selain itu, lingkungan pengujian sistem pada pengerjaan tugas akhir ini juga dilakukan pada lingkungan dan kaskas bantu sebagai berikut:

- Prosesor Intel® Core™ i7
- *RAM 16GB*
- Jenis *Laptop*
- Sistem Operasi Windows 10

5.2 Dataset Uji Coba

Dataset yang dilakukan pada tahapan uji coba ini menggunakan dataset karya ilmiah yang dicari berdasarkan *query* pada *website-website* yang menyediakan data karya ilmiah secara daring. Hal ini dilakukan karena pada tahap sebelumnya tidak memiliki *ground truth* pada setiap karya ilmiah yang dikumpulkan. Maka, pada tahap uji coba ini dibuat *dataset* baru untuk menentukan akurasi dari sistem yang dibangun pada Tugas Akhir ini.

Website yang digunakan untuk mencari data karya ilmiah tersebut adalah sebagai berikut:

- *ACM Digital Library*
- *IEEEExplore Digital Library*
- *Springer Link*
- *Science Direct*
- *Wiley Online Library*

Query yang dilakukan pada *website-website* tersebut adalah sebagai berikut:

- *COCOMO (Constructive Cost Model)*
- *Process Mining*
- *Semantic Search*
- *Mixed Reality*

Pembagian data yang dilakukan pada tahapan uji coba ini yang memiliki karya ilmiah yang positif membahas mengenai *query* tersebut dan negatif yang membahas mengenai *query* tersebut. Persebaran data yang dilakukan pada tahapan uji coba ini dapat dijelaskan pada Tabel 5. 1 yang ada di bawah ini.

Tabel 5. 1 *Dataset*

<i>Query</i>	Positif	Negatif
COCOMO	105	23
<i>Process Mining</i>	42	14
<i>Semantic Search</i>	46	16
<i>Mixed Reality</i>	43	9

Label positif dan negatif yang terdapat pada *metadata* tersebut ditentukan menggunakan *ground truth* dari data yang telah dikumpulkan oleh salah satu dosen yang melakukan penelitian secara manual mengenai salah satu *query* tersebut yaitu COCOMO dan *systematic literature review* yang tersedia mengenai *Process Mining* [34], *Semantic Search* [35], *Mixed Reality* [36]. Untuk *metadata* dari *query Process Mining*, *Semantic Search*, dan *Mixed Reality* yang diambil dari *website-website* yang telah disebutkan

sebelumnya, akan dicocokkan berdasarkan kriteria-kriteria yang terdapat pada *systematic literature review*. Setelah itu, diberikan label positif bagi *metadata* karya ilmiah yang sesuai dengan kriteria-kriteria dari *systematic literature review* dan label negatif yang tidak sesuai dengan kriteria-kriteria dari *systematic literature review*. Pemberian label ini diperlukan untuk membagi jumlah data yang akan dibagi menjadi data yang positif dan data yang negatif.

Kriteria-kriteria yang terdapat pada *systematic literature review* pada *query process mining*, *semantic search*, dan *mixed reality* untuk penentuan label positif dan negatif pada *dataset* yang dikumpulkan adalah sebagai berikut:

- *Process Mining*:
 - Melakukan *query “process mining”* pada *website-website* penyedia karya ilmiah secara daring dalam *advanced search* dengan pengecekan terhadap semua konten dari karya ilmiah tersebut.
 - Melakukan pengecekan pada *title* dari karya ilmiah yang diambil apakah memiliki topik mengenai *process mining*. Jika iya, maka akan dimasukkan ke tahap selanjutnya. Jika tidak, maka akan diberi label negatif.
 - Melakukan pengecekan pada *abstract*, apakah membicarakan tentang *process mining*. Jika iya, maka akan dimasukkan ke tahap selanjutnya. Jika tidak, maka akan diberi label negatif.
 - Melakukan pengecekan keseluruhan teks apakah memiliki topik *process mining*. Jika iya maka akan diberi label positif. Jika tidak maka akan diberi label negatif.
- *Semantic Search*
 - Melakukan *query “semantic search”* pada *website-website* penyedia karya ilmiah secara daring dalam *advanced search* dengan pengecekan terhadap semua konten dari karya ilmiah tersebut.

- Melakukan pengecekan pada *title* dari karya ilmiah yang diambil apakah memiliki topik mengenai *semantic search*. Jika iya, maka akan dimasukkan ke tahap selanjutnya. Jika tidak, maka akan diberi label negatif.
- Melakukan pengecekan pada *abstract*, apakah membicarakan tentang *semantic search*. Jika iya, maka akan dimasukkan ke tahap selanjutnya. Jika tidak, maka akan diberi label negatif.
- Melakukan pengecekan keseluruhan teks apakah memiliki topik *semantic search*. Jika iya maka akan diberi label positif. Jika tidak maka akan diberi label negatif.
- *Mixed Reality*
 - Melakukan *query* “*mixed reality*” pada *website-website* penyedia karya ilmiah secara daring dalam *advanced search* dengan pengecekan terhadap semua konten dari karya ilmiah tersebut.
 - Melakukan pengecekan pada *title* dari karya ilmiah yang diambil apakah memiliki topik mengenai *mixed reality*. Jika iya, maka akan dimasukkan ke tahap selanjutnya. Jika tidak, maka akan diberi label negatif
 - Melakukan pengecekan pada *abstract*, apakah membicarakan tentang *mixed reality*. Jika iya, maka akan dimasukkan ke tahap selanjutnya. Jika tidak, maka akan diberi label negatif.
 - Melakukan pengecekan keseluruhan teks apakah memiliki topik *mixed reality*. Jika iya maka akan diberi label positif. Jika tidak maka akan diberi label negatif.

Data karya ilmiah yang dibutuhkan pada tahapan uji coba ini memiliki *metadata* yang mirip seperti pada tahap implementasi Tugas Akhir ini. *Metadata-metadata* tersebut adalah *Title*, *Abstract*, dan *Keywords*. *Metadata* dari *Topics* tidak dimasukkan

karena tidak didapatkannya *metadata* tersebut dari *website-website* seperti *ACM Digital Library*, *Springer Link*, *Science Direct*, dan *Wiley Online Library*. *Metadata* tersebut akan dilakukan *data preprocessing* yang dilakukan sebagai berikut:

- Menggabungkan *metadata-metadata* dari *Titles*, *Abstract*, dan *Keywords* pada data karya ilmiah tersebut dan menyimpannya pada kolom baru yaitu kolom *Content*.
- Melakukan *case folding* dengan mengubah huruf menjadi huruf kecil (*lowercase*).
- Menghapus tanda baca yang terdapat pada kolom tersebut.
- Menghapus angka yang terdapat pada kolom tersebut.
- Menghilangkan *stopwords* Bahasa Inggris pada kolom tersebut.
- Melakukan *lemmatization* untuk mengubah kata-kata yang terdapat pada kolom tersebut menjadi kata dasarnya

5.3 Skenario Uji Coba

Subbab ini akan menjelaskan skenario uji coba yang telah dilakukan. Terdapat beberapa skenario uji coba yang dilakukan. Pada masing-masing skenario, dilakukan uji coba dengan menentukan *query* yang dilakukan.

Uji coba yang dilakukan pada subbab ini dilakukan oleh salah satu dosen yaitu:

- Nama : Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.
- Waktu : Minggu, 20 Januari 2020
- Lingkungan : Laptop

Sebelum melakukan skenario, dilakukan *training corpus* untuk mencari vektor representasi makna kata dari *metadata* karya ilmiah yang telah dikumpulkan. *Training corpus* menggunakan metode *fastText* yang dipanggil melalui *library gensim* yang tersedia di bahasa pemrograman *python*. Parameter *fastText* yang

diterapkan pada Tugas Akhir ini dapat dilihat pada Kode Sumber 5. 1.

```
1. from gensim.models import FastText as ft
2. model_ft = ft(corpus_lemma, sg=1, workers=multiproc
   essing.cpu_count()-
   1, size=100, iter=5, min_count=5, window=2)
```

Kode Sumber 5. 1 Parameter *fastText*

Parameter yang ditentukan menggunakan model *skip-gram*, menentukan jumlah *workers* dengan menghitung jumlah CPU yang digunakan, besaran vektor sebesar 100, menggunakan iterasi sebanyak lima kali, dan jumlah minimum kata dalam *corpus* adalah 5. Minimum kata yang ditentukan pada tahap uji coba ini ditentukan oleh *expert*. Parameter tersebut digunakan karena memiliki performansi yang lebih baik dibandingkan parameter-parameter yang dicoba sebelumnya. Perbandingan hasil performansi ditunjukkan pada Gambar 5. 1.

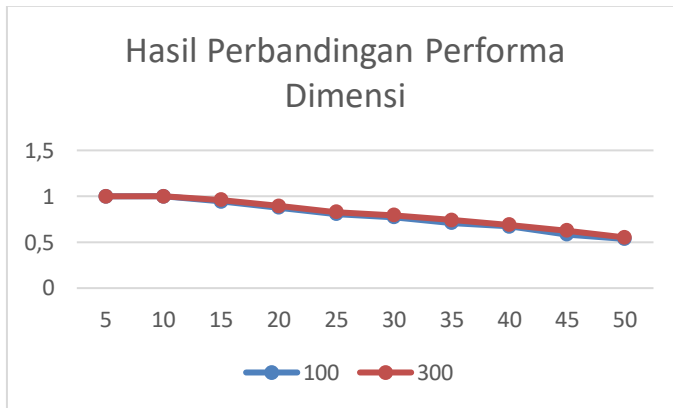


Gambar 5. 1 Grafik hasil performa iterasi

Gambar tersebut menunjukkan penentuan performa iterasi yang dilakukan pada uji parameter. Uji parameter ini dilakukan untuk mencari makna kata yang ada di dalam *corpus* yang dikumpulkan. Kata yang dicari maknanya pada *corpus* adalah COCOMO. Iterasi yang dilakukan pada uji parameter ini dimulai

dari rentang 5 sampai 50. Hasil yang ditunjukkan bahwa, semakin banyak iterasi maka performansi dari model tersebut semakin menurun. Sehingga, performansi terbaik yang akan diterapkan pada bab ini adalah dengan menggunakan iterasi sebanyak 5.

Selanjutnya, dilakukan uji coba parameter pada dimensi yang akan diterapkan. Uji coba yang dilakukan akan membandingkan dimensi 100 dan 300. Perbandingan hasil performansi ditunjukkan pada Gambar 5. 2.



Gambar 5. 2 Grafik hasil perbandingan dimensi

Gambar tersebut menunjukkan perbandingan performa uji parameter berdasarkan dimensi. Hasil tersebut menunjukkan bahwa, dengan performansi yang diperoleh dari hasil uji parameter 100 dimensi dan 300 dimensi, vektor dengan ukuran 300 dimensi memiliki performansi yang lebih baik dibandingkan dengan 100 dimensi. Tetapi, dikarenakan perbedaan tidak cukup signifikan, maka digunakan 100 dimensi untuk diterapkan pada Tugas Akhir ini.

Tabel 5. 2 Deskripsi skenario

Skenario	Query	Pengujian
1	Kata asli	User
2	Kata asli ditambah satu kata yang diinput secara manual oleh user	
6	Kata asli ditambah dua kata yang diinput secara manual oleh user	
3	Kata asli ditambah satu kata yang paling mirip ke-1 oleh sistem	Sistem
4	Kata asli ditambah satu kata yang paling mirip ke-10 oleh sistem	
5	Kata asli ditambah satu kata yang paling mirip ke-20 oleh sistem	
7	Kata asli ditambah dua kata yang paling mirip ke-1 dan ke-10 oleh sistem	

Sebelum melakukan skenario pengujian, dibagi *dataset* yang telah dikumpulkan pada tahapan sebelumnya masing-masing menjadi dua *dataset*. *Dataset* tersebut adalah *dataset* yang memiliki label positif dengan *query* dan label negatif dengan *query*. *Dataset* tersebut dilakukan skenario uji coba dengan *query* seperti pada Tabel 5. 2

Pada Kode Sumber 5. 2, didapatkan kata-kata yang memiliki makna yang paling mirip oleh sistem. Untuk rincian kata-kata yang digunakan, akan dijelaskan sebagai berikut:

- Skenario Pengujian 1 :
 - COCOMO
 - *Process Mining*
 - *Semantic Search*
 - *Mixed Reality*
- Skenario Pengujian 2 :
 - COCOMO : menambahkan kata “*cost*”
 - *Process Mining* : menambahkan kata “*business*”
 - *Semantic Search* : menambahkan kata “*query*”
 - *Mixed Reality* : menambahkan kata “*virtual*”
- Skenario Pengujian 3 :
 - COCOMO : menambahkan kata “*computing*”.
 - *Process Mining* : menambahkan kata “*processing*”
 - *Semantic Search* : menambahkan kata “*searching*”.
 - *Mixed Reality* : menambahkan kata “*interaction*”
- Skenario Pengujian 4 :
 - COCOMO : menambahkan kata “*measurement*”.
 - *Process Mining* : menambahkan kata “*representation*”
 - *Semantic Search* : menambahkan kata “*relation*”.
 - *Mixed Reality* : menambahkan kata “*visualization*”
- Skenario Pengujian 5 :
 - COCOMO : menambahkan kata “*modifiability*”.
 - *Process Mining* : menambahkan kata “*organizational*”
 - *Semantic Search* : menambahkan kata “*searchable*”.

- *Mixed Reality* : menambahkan kata “*reconstruction*”
- Skenario Pengujian 6 :
 - COCOMO : menambahkan kata “*cost constructive*”.
 - Process Mining : menambahkan kata “*business techniques*”
 - Semantic Search : menambahkan kata “*query engine*”
 - Mixed Reality : menambahkan kata “*virtual realization*”
- Skenario Pengujian 5 :
 - COCOMO : menambahkan kata “*computing measurement*”
 - Process Mining : menambahkan kata “*processing representation*”
 - Semantic Search : menambahkan kata “*searching relation*”.
 - Mixed Reality : menambahkan kata “*interaction visualization*”

```

1. similar = model_ft.most_similar(positive=query)
2. similar = similar[0][0]
3. similar

```

Kode Sumber 5. 2 *Query* makna kata

Hasil dari skenario-skenario yang dilakukan pada tahapan evaluasi dan uji coba ini merupakan berapa jumlah data yang tertangkap pada sistem yang dibangun untuk Tugas Akhir ini. Skenario-skenario ini dilakukan pada data yang memiliki label positif dan juga label negatif.

Dari skenario-skenario yang dilakukan pada tahapan ini, hasil akhir dari skenario-skenario tersebut berupa perhitungan *cohen's kappa coefficient*, *precision*, *recall*, dan *f-1 score*.

5.4 Hasil Evaluasi

Pada subbab ini akan dijelaskan mengenai hasil evaluasi yang telah dilakukan di setiap uji coba dan skenario-skenario pada tugas akhir ini.

Hasil pada skenario-skenario yang telah dilakukan di setiap uji coba dapat dilihat pada Tabel 5. 3. Tabel tersebut menunjukkan hasil *True Positive*, *False Positive*, *True Negative*, dan *False Negative*. Selain itu, dapat dilihat juga hasil akurasi *cohen's kappa coefficient*, *precision*, *recall*, dan *f-1 score*. Dari setiap skenario yang dilakukan, hasil yang diraih pada skenario 3, skenario 4, skenario 5, dan skenario 7 menunjukkan nilai yang lebih tinggi dibandingkan dengan skenario-skenario lain.

Tabel 5. 3 Hasil Evaluasi

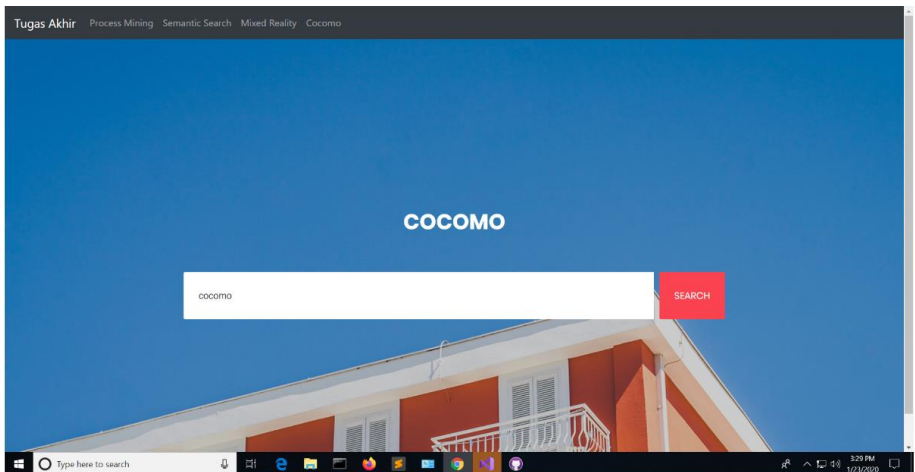
	Scena rio	1	2	3	4	5	6	7
COCO MO	TP	96	96	97	97	97	93	97
	FP	0	0	0	0	0	0	0
	TN	23	23	23	23	23	23	23
	FN	9	9	8	8	8	12	8
	Accur acy	0,9 30	0,9 30	0,9 38	0,9 38	0,9 38	0,8 98	0,9 38
	Precisi on	1	1	1	1	1	1	1
	Recall	0,9 14	0,9 14	0,9 24	0,9 24	0,9 24	0,8 76	0,9 24
	F-1 Score	0,9 55	0,9 55	0,9 60	0,9 60	0,9 60	0,9 34	0,9 60

	Scena rio	1	2	3	4	5	6	7
<i>Process Mining</i>	TP	36	36	38	38	38	36	38
	FP	0	0	0	0	0	0	0
	TN	16	16	14	14	14	16	14
	FN	6	6	4	4	4	6	4
	Accur acy	0,8 93	0,8 93	0,9 29	0,9 29	0,9 29	0,8 93	0,9 29
	Precisi on	1	1	1	1	1	1	1
	Recall	0,8 57	0,8 57	0,9 05	0,9 05	0,9 05	0,8 57	0,9 05
	F-1 Score	0,9 23	0,9 23	0,9 50	0,9 50	0,9 50	0,9 23	0,9 50
<i>Semant ic Search</i>	TP	41	41	42	42	42	41	42
	FP	0	0	0	0	0	0	0
	TN	16	16	16	16	16	16	16
	FN	5	5	4	4	4	5	4
	Accur acy	0,9 19	0,9 19	0,9 35	0,9 35	0,9 35	0,9 19	0,9 35
	Precisi on	1	1	1	1	1	1	1
	Recall	0,8 91	0,8 91	0,9 13	0,9 13	0,9 13	0,8 91	0,9 13
	F-1 Score	0,9 43	0,9 43	0,9 55	0,9 55	0,9 55	0,9 43	0,9 55
<i>Mixed Reality</i>	TP	38	39	39	39	39	39	39
	FP	0	0	0	0	0	0	0
	TN	9	9	9	9	9	9	9
	FN	5	4	4	4	4	4	4

	Scenar rio	1	2	3	4	5	6	7
	Accur acy	0,9 04	0,9 06	0,9 06	0,9 06	0,9 06	0,9 06	0,9 06
	Precisi on	1	1	1	1	1	1	1
	Recall	0,8 84	0,9 07	0,9 07	0,9 07	0,9 07	0,9 07	0,9 07
	F-1 Score	0,9 38	0,9 51	0,9 51	0,9 51	0,9 51	0,9 51	0,9 51

5.5 Tampilan Aplikasi

Pada subbab ini menampilkan hasil implementasi sistem pada website berupa *interface* yang dapat digunakan oleh pengguna. Gambar 5. 3 menampilkan tampilan awal dari *interface* yang dibuat. Pada tampilan tersebut, pengguna dapat melakukan *query* untuk mencari rekomendasi karya ilmiah yang diinginkan. Uji coba yang dilakukan pada data COCOMO.



Gambar 5. 3 Screenshot tampilan *interface*

Gambar 5. 4 adalah hasil dari rekomendasi karya ilmiah yang ditampilkan untuk pengguna. Hasil yang ditampilkan berupa *Title* karya ilmiah dan *source* karya ilmiah tersebut.

COCOMO		
	Title	Source
20	Software cost estimation for component based fourth-generation language software applications	IEEE
53	A PSO-based model to increase the accuracy of software development effort estimation	Springer Link
87	A simulation model for strategic management process of software projects	Science Direct
31	Cuckoo search-based hybrid models for improving the accuracy of software effort estimation	Springer Link
45	Transfer learning in effort estimation	Springer Link
111	Evaluating filter fuzzy analogy homogenous ensembles for software development effort estimation	Wiley
100	Functional networks as a novel data mining paradigm in forecasting software development efforts	Science Direct
70	Determining relevant training data for effort estimation using Window-based COCOMO calibration	Science Direct
109	Investigating the use of duration-based windows and estimation by analogy for COCOMO	Wiley
97	Analogy-based software effort estimation using Fuzzy numbers	Science Direct
48	Neural network-based models for software effort estimation: a review	Springer Link
78	Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles	Science Direct
8	Using static and dynamic impact analysis for effort estimation	IEEE
23	Optimizing design parameters of fuzzy model based COCOMO using genetic algorithms	Springer Link
90	A contingency estimation model for software projects	Science Direct
94	Systematic literature review of machine learning based software development effort estimation models	Science Direct
107	Handling imprecision and uncertainty in software development effort prediction: A type-2 fuzzy logic-based framework	Science Direct
1	Recent Methods for Software Effort Estimation by Analogy	ACM
104	GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation	Science Direct
25	A hybrid model of wavelet neural network and metaheuristic algorithm for software development effort estimation	Springer Link
81	An empirical evaluation of ensemble adjustment methods for analogy-based effort estimation	Science Direct
83	Analyzing and handling local bias for calibrating parametric cost estimation models	Science Direct
103	Predicting software project effort: A grey relational analysis-based method	Science Direct

Gambar 5. 4 Screenshot hasil rekomendasi *interface*

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan yang diperoleh selama pengerjaan tugas akhir ini. Selain itu, juga terdapat saran terhadap tugas akhir ini yang dapat dijadikan masukan pengembangan penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan yang didapatkan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Untuk mengimplementasikan metode *fastText* dalam memetakan kata menjadi vektor, digunakan parameter iterasi sebanyak lima kali, besaran dimensi sebesar 100, model yang dipakai adalah model *skip-gram*, dan jumlah *window* yang dipakai sebesar 2.
2. Untuk mengimplementasikan metode *Word Mover's Distance Similarity* untuk mencari kemiripan dari kalimat yang dimasukkan oleh *user* dengan karya ilmiah yang tersedia, dilakukan *training* metode yang menggunakan parameter masukan seperti data karya ilmiah yang dipakai dan vektor dari model *fastText*.
3. Hasil evaluasi yang didapatkan dengan mengimplementasikan model *fastText* dan *Word Mover's Distance Similarity* adalah dengan menggunakan *query* yang dari kata asli yang menjadi masukan dan dengan menambahkan satu atau dua kata yang memiliki makna yang mirip ke-1, ke-10, atau ke-20 dari kata aslinya, dapat memberikan hasil akurasi *cohen's kappa coefficient*, *precision*, *recall*, dan *f-1 score* yang lebih tinggi dibandingkan dengan skenario-skenario lainnya yang dapat dilihat pada Tabel 5. 3.

6.2 Saran

Adapun saran terkait tugas akhir ini antara lain adalah sebagai berikut.

1. Untuk meningkatkan hasil akurasi menjadi lebih baik, disarankan untuk mencoba metode *word embedding* yang terbaru selain metode *fastText*. Metode tersebut merupakan penelitian terbaru dan bernama *elMo*.

DAFTAR PUSTAKA

- [1] F. Ricci, L. Rokach and B. Shapira, Introduction to Recommender Systems Handbook, 2010.
- [2] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang and R. Zadeh, "WTF: the who to follow service at Twitter," *WWW '13: Proceedings of the 22nd international conference on World Wide Web*, pp. 505-514, 2013.
- [3] H.-H. Chen, A. G. O. II and C. L. Giles, "ExpertSeer: a Keyphrase Based Expert Recommender for Digital Libraries," 2015.
- [4] H.-H. Chen, L. Gou, X. (. Zhang and C. L. Giles, "CollabSeer: A Search Engine for Collaboration Discovery," *Proceedings of the 2011 Joint International Conference on Digital Libraries*, 2011.
- [5] A. Felfernig, K. Isak, K. Szabo and P. Zachar, "The VITA Financial Services Sales Support Environment," *IAAI'07: Proceedings of the 19th national conference on Innovative applications of artificial intelligence*, vol. 2, pp. 1692-1699, 2007.
- [6] J. S. Breese, D. Heckerman and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *UAI'98: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 43-52, 1998.
- [7] C. C. Aggarwal, Recommender Systems, 2016.
- [8] K. Lakiotaki, N. F. Matsatsinis and A. Tsoukias, "Multicriteria User Modeling in Recommender Systems," *IEEE Intelligent Systems*, vol. 26, no. 2, pp. 64-76, 2011.
- [9] D. Bouneffouf, "DRARS, A Dynamic Risk-Aware Recommender System," 2013.

- [10] E. Pimenidis, N. Polatidis and H. Mouratidis, "Mobile recommender systems: Identifying the major concepts," vol. 45, no. 3, pp. 387-397, 2018.
- [11] R. Hoekstra, "The Knowledge Reengineering Bottleneck," *Semantic Web*, pp. 111-115, 2010.
- [12] S. Harispe, S. Ranwez, S. Janaqi and J. Montmain, *Semantic Similarity from Natural Language and Ontology Analysis*, Morgan & Claypool, 2015.
- [13] A. Ballatore, M. Bertolotto and D. C. Wilson, "An evaluative baseline for geo-semantic relatedness and similarity," *GeoInformatica*, vol. 18, no. 4, pp. 747-767, 2014.
- [14] N. I. Pratiwi, I. Budi and I. Alfina, "Hate Speech Detection on Indonesian Instagram Comments using FastText Approach," *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2018.
- [15] A. Alessa, M. Faczipour and Z. Alhassan, "Text Classification of Flu-related Tweets Using FastText with Sentiment and Keyword Features," *2018 IEEE International Conference on Healthcare Informatics*, 2018.
- [16] H. Y. Erdinc and A. Guran, "Semi-supervised Turkish Categorization with Word2Vec, Doc2Vec and Fasttext Algorithms," *2019 27th Signal Processing and Communications Applications Conference (SIU)*, 2019.
- [17] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, "Enriching Word Vectors with Subword Information," vol. 5, pp. 135-146, 2017.
- [18] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems*, vol. 2, pp. 3111-3119, 2013.

- [19] M. J. Kusner, Y. Sun, N. I. Kolkin and K. Q. Weinberger, "From Word Embeddings To Document Distances," *ICML'15 Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 957-966, 2015.
- [20] R. Chen, Q. Gao, W. Ji, F. Long and Q. Ling, "Network Log Analysis based on the Topic Word Mover's Distance," *2018 Chinese Control And Decision Conference*, 2018.
- [21] C. Yongkiatpanich and D. Wichadakul, "Extractive Text Summarization Using Ontology and Graph-Based Method," *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, 2019.
- [22] K. A. Sekarwati, L. Y. Banowosari, I. M. Wiryana and D. Kerami, "Pengukuran Kemiripan Dokumen dengan Menggunakan Tools Gensim," vol. 1, 2015.
- [23] J. M. Zelle, *Python Programming An Introduction To Computer Science*, Portland: Franklin, Beedle & Associates Incorporated, 2016.
- [24] M. Anders, *Python 3 Web Development Beginner's Guide*, Birmingham: Packt Publishing Ltd., 2011.
- [25] W. McKinney, *Python for Data Analysis*, Sebastopol: O'Reilly Media, Inc, 2013.
- [26] S. Riley, *Game Programming With Python (Game Development Series)*, Charles River Media, 2003.
- [27] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sensing of Environment*, vol. 62, no. 1, pp. 77-89, 1997.
- [28] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," 2008.
- [29] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia Medica*, pp. 276-282, 2012.

- [30] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," *Proceedings of the 23rd International Conference on Machine Learning*, 2006.
- [31] S. Yutaka, "The Truth of the F-Measure," 2007.
- [32] R. Mitkov, *The Oxford Handbook of Computational Linguistics*, Oxford: Oxford University Press, 2005.
- [33] R. Rajagopal, *Introduction to Microsoft Windows NT Cluster Server: Programming and Administration*, CRC Press, 1999.
- [34] R. Kelemen, "Systematic Review on Process Mining and Security," *Central and Eastern European e/Dem and e/Gov Days 2017*, 2017.
- [35] J. M. Vidal and A. Melgar, "Research on Proposals and Trends in the Architectures of Semantic Search Engines: A Systematic Literature Review," *2017 Federated Conference on Computer Science and Information Systems*, 2017.
- [36] C. M. Y. Rasimah, M. Nurazean, M. D. Salwani, M. Z. Norziha and I. Roslina, "A Systematic Literature Review of Factors Influencing Acceptance on Mixed Reality Technology," *ARNP Journal of Engineering and Applied Sciences*, vol. 10, pp. 18239-18246, 2015.
- [37] C. Zhang, X. Wang, S. Yu and Y. Wang, "Research on Keyword Extraction of Word2vec Model in Chinese Corpus," *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, 2018.
- [38] M. Al-Amin, M. S. Islam and S. D. Uzzal, "Sentiment Analysis of Bengali Comments With Word2Vec and Sentiment Information of Words," *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2017.
- [39] C. Xia, T. He, W. Li, Z. Qin and Z. Zou, "Similarity Analysis of Law Documents Based on Word2vec," *2019 IEEE 19th*

- International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 354-357, 2019.
- [40] M. A. Ghazal, O. Ibrahim and M. A. Salama, "Educational Process Mining: A Systematic Literature Review," *2017 European Conference on Electrical Engineering and Computer Science (EECS)*, 2017.
- [41] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *Advances in neural information processing systems* 26, pp. 1-9, 2013.

[Halaman ini sengaja dikosongkan]

LAMPIRAN A : HASIL CRAWLING DATA

A. 1 Hasil Crawling Data

Hasil Crawling Data
<pre>global.document.metadata={"userInfo":{"customerNameRaw":"Institut Teknologi Sepuluh Nopember","institutionName":"Institut Teknologi Sepuluh Nopember","institute":true,"member":false,"individual":false,"guest":false,"subscribedContent":false,"fileCabinetContent":false,"fileCabinetUser":false,"institutionalFileCabinetUser":false,"instType":"Academic","userIds":[2122785],"showPatentCitations":true,"showGet802Link":true,"openUrlImgLoc":"/assets/img/btn.find-in-library.png","openUrlLink":"NA","showOpenUrlLink":false,"marketingInfoCaptured":false,"tracked":false,"ringGoldId":"106242","delegatedAdmin":false,"isInstitutionDashboardEnabled":false,"isInstitutionProfileEnabled":false,"isRoamingEnabled":false,"isDelegatedAdmin":false},"authors":[{"name":"Andriy Semenov","affiliation":"Vinnytsia National Technical University, Dept. of Radio-Frequency Engineering, Vinnytsia, Ukraine","firstName":"Andriy","lastName":"Semenov","affiliation":"Vinnytsia National Technical University, Dept. of Radio-Frequency Engineering, Vinnytsia, Ukraine","id":"37410734000"},{"name":"Anton Savytskyi","affiliation":"Vinnytsia National Technical University, Dept. of Radio-Frequency Engineering, Vinnytsia, Ukraine","firstName":"Anton","lastName":"S</pre>

```

avytskyi","affiliation":"Vinnytsia
National Technical University, Dept. of
Radio-Frequency Engineering, Vinnytsia,
Ukraine","id":"37086213285"},{"name":"Olen
a Semenova","affiliation":"Vinnytsia
National Technical University, Dept. of
Telecommunication Systems and Television,
Vinnytsia,
Ukraine","firstName":"Olena","lastName":"S
emenova","affiliation":"Vinnytsia National
Technical University, Dept. of
Telecommunication Systems and Television,
Vinnytsia,
Ukraine","id":"38244610900"},{"name":"Maks
ym Huz","affiliation":"Vinnytsia National
Technical University, Dept. of
Telecommunication Systems and Television,
Vinnytsia,
Ukraine","firstName":"Maksym","lastName":"
Huz","affiliation":"Vinnytsia National
Technical University, Dept. of
Telecommunication Systems and Television,
Vinnytsia,
Ukraine","id":"37085899995"}],"isbn":[{"fo
rmat":"Electronic ISBN","value":"978-1-
5386-2468-5"},{"format":"USB
ISBN","value":"978-1-5386-2467-
8"},{"format":"Print on Demand(PoD)
ISBN","value":"978-1-5386-2469-
2"}],"articleNumber":"8520001","dbTime":"3
ms","metrics":{"citationCountPaper":0,"cit
ationCountPatent":0,"totalDownloads":27},"
pdfUrl":"/stamp/stamp.jsp?tp=&arnumber=852
0001","purchaseOptions":{"showOtherFormatP
ricingTab":false,"showPdfFormatPricingTab"
:true,"pdfPricingInfoAvailable":true,"othe

```

```

rPricingInfoAvailable":false,"mandatoryBun
dle":false,"optionalBundle":false,"pdfPric
ingInfo":[{"memberPrice":"$14.95","nonMemb
erPrice":"$33.00","partNumber":"8520001","
type":"PDF/HTML"}]],"formulaStrippedArticl
eTitle":"Numerical Simulation of the
Chua's Oscillator Based on a MOSFET
Structure with a Cubic
Nonlinearity","getProgramTermsAccepted":fa
lse,"sections":{"abstract":"true","authors
":"true","figures":"true","multimedia":"fa
lse","references":"true","citedby":"false"
,"keywords":"true","definitions":"false","
algorithm":"false","supplements":"false","
footnotes":"false","disclaimer":"false","m
etrics":"true"},"title":"Numerical
Simulation of the Chua's Oscillator Based
on a MOSFET Structure with a Cubic
Nonlinearity","abstract":"The article
presents the results of numerical modeling
of Chua oscillator based on a metal-oxide-
semiconductor field-effect transistors
(MOSFET) structure with a cubic non-
linearity. An overview of the Chua's
oscillator circuitry basis on MOSFET
structures. The approximation equation of
static current-voltage characteristic with
a cubic non-linearity is presented, as
well as oscillator mathematical model. The
phase portraits of the Chua oscillator
with cubic non-linearity, time diagrams
and amplitude-frequency spectra of
generated oscillations are obtained. The
graphs of Lyapunov exponents are
presented, depending on the coefficients
change in the mathematical model of the

```

```

Chua oscillator with cubic non-linearity,
as well as the Poincaré
map.", "allowComments": false, "keywords": [{
  "type": "IEEE
Keywords", "kwd": ["Oscillators", "MOSFET", "M
athematical model", "Current-voltage
characteristics", "Dynamics", "Ultra
wideband technology"]}, {"type": "INSPEC:
Controlled Indexing", "kwd": ["approximation
theory", "Chua's
circuit", "MOSFET", "numerical
analysis", "oscillators", "Poincare
mapping"]}, {"type": "INSPEC: Non-Controlled
Indexing", "kwd": ["cubic
nonlinearity", "MOSFET
structure", "oscillator mathematical
model", "numerical simulation", "metal-
oxide-semiconductor field-effect
transistors structure", "Chua oscillator
circuitry", "approximation
equation", "static current-voltage
characteristic", "time
diagrams", "amplitude-frequency
spectra", "Lyapunov exponents", "Poincaré
map"]}, {"type": "Author Keywords
", "kwd": ["oscillator", "Chua", "MOSFET
structure", "chaos", "phase portrait", "cubic
non-linearity"]}], "publicationTitle": "2018
9th International Conference on
Ultrawideband and Ultrashort Impulse
Signals
(UWBUSIS)", "rightsLink": "http://s100.copyr
ight.com/AppDispatchServlet?publisherName=
ieee&publication=proceedings&title=Numeric
al+Simulation+of+the+Chua%27s+Oscillator+B
ased+on+a+MOSFET+Structure+with+a+Cubic+No

```

```

nlinearity&isbn=978-1-5386-2468-
5&publicationDate=September+2018&author=An
driy+Semenov&ContentID=10.1109/UWBUSIS.201
8.8520001&orderBeanReset=true&startPage=14
4&endPage=149&proceedingName=2018+9th+Inte
rnational+Conference+on+Ultrawideband+and+
Ultrashort+Impulse+Signals+%28UWBUSIS%29",
"startPage":"144","endPage":"149","display
PublicationTitle":"2018 9th International
Conference on Ultrawideband and Ultrashort
Impulse Signals
(UWBUSIS)","pdfPath":"/iel7/8496384/851996
3/08520001.pdf","doi":"10.1109/UWBUSIS.201
8.8520001","standardTitle":"Numerical
Simulation of the Chua's Oscillator Based
on a MOSFET Structure with a Cubic
Nonlinearity","pubLink":"/xpl/conhome/8496
384/proceeding","issueLink":"/xpl/tocresul
t.jsp?isnumber=8519963","isJournal":false,
"isConference":true,"isBook":false,"dateOf
Insertion":"05 November
2018","isGetArticle":false,"isGetAddressIn
foCaptured":false,"isMarketingOptIn":false
,"xploreDocumentType":"Conference
Publication","applyOUPFilter":false,"pubTo
pics":[{"name":"Communication, Networking
and Broadcast
Technologies"}, {"name":"Components,
Circuits, Devices and
Systems"}, {"name":"Engineered Materials,
Dielectrics and Plasmas"}, {"name":"Fields,
Waves and
Electromagnetics"}, {"name":"Geoscience"}, {"
name":"Signal Processing and
Analysis"}],"publisher":"IEEE","conference
Date":"4-7 Sept.

```

```

2018","isNotDynamicOrStatic":false,"isFree
Document":false,"isSMPTE":false,"isSAE":fa
lse,"isNow":false,"isCustomDenial":false,"
isStandard":false,"isACM":false,"isEarlyAc
cess":false,"isChapter":false,"isStaticHtm
l":false,"htmlLink":"/document/8520001/","
publicationDate":"September
2018","accessionNumber":"18243959","isOpen
Access":false,"isProduct":false,"isEphemer
a":false,"isMorganClaypool":false,"persist
entLink":"https://ieeexplore.ieee.org/serv
let/opac?punumber=8496384","chronOrPublica
tionDate":"4-7 Sept.
2018","htmlAbstractLink":"/document/852000
1/","isPromo":false,"isOUP":false,"isDynam
icHtml":true,"startPage":"144","openAccess
Flag":"F","ephemeraFlag":"false","chorusFl
ag":"false","title":"Numerical Simulation
of the Chua's Oscillator Based on a MOSFET
Structure with a Cubic
Nonlinearity","confLoc":"Odessa,
Ukraine","accessionNumber":"18243959","htm
l_flag":"false","ml_html_flag":"true","pro
moFlag":"false","sourcePdf":"144-149-
PID5592321.pdf","content_type":"Conference
s","mlTime":"PT0.062822S","chronDate":"4-7
Sept. 2018","xplore-pub-
id":"8496384","pdfPath":"/iel7/8496384/851
9963/08520001.pdf","isNumber":"8519963","r
ightsLinkFlag":"1","dateOfInsertion":"05
November
2018","contentType":"conferences","publica
tionDate":"September
2018","publicationNumber":"8496384","xplore-
issue":"8519963","articleId":"8520001","pu

```



```

blicationTitle":"2018 9th International
Conference on Ultrawideband and Ultrashort
Impulse Signals
(UWBUSIS)","sections":{"abstract":"true","
authors":"true","figures":"true","multimed
ia":"false","references":"true","citedby":
"false","keywords":"true","definitions":"f
alse","algorithm":"false","supplements":"f
alse","footnotes":"false","disclaimer":"fa
lse","metrics":"true"},"onlineDate":"","co
nferenceDate":"4-7 Sept.
2018","publicationYear":"2018","subType":"
IEEE
Conference","_value":"IEEE","lastupdate":"
2019-12-
09","mediaPath":"/mediastore_new/IEEE/cont
ent/media/8496384/8519963/8520001","endPag
e":"149","displayPublicationTitle":"2018
9th International Conference on
Ultrawideband and Ultrashort Impulse
Signals
(UWBUSIS)","doi":"10.1109/UWBUSIS.2018.852
0001"};

```

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Nabil Haidarrahan Pribadi, lahir di Jakarta pada tanggal 15 Agustus 1999. Penulis menempuh pendidikan mulai dari TK Tadika Puri Ciputat (2003-2005), SD Pembangunan Jaya (2005-2011), SMPN 11 Jakarta (2011-2013), SMAN 70 Jakarta (2013-2016), dan sekarang sedang menjalani pendidikan S1 Informatika di ITS. Penulis aktif dalam organisasi dan kepanitiaan Himpunan Mahasiswa Teknik Computer-Informatika (HMTc), Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi dan Komunikasi (BEM FTIK), FTIF FESTIVAL, Schematics. Di antaranya adalah menjadi Staff Departemen Kaderisasi dan Pemetaan (KDPM) HMTc ITS 2017-2018, Staff Departemen Internal Affairs (IA) BEM FTIK ITS 2017-2018, Ketua Himpunan HMTc ITS 2018-2019, Badan Pengurus Harian Schematics REEVA 2017, Staff Ahli REEVA 2018, Staff Keamanan dan Perizinan FTIF FESTIVAL 2017. Komunikasi dengan penulis dapat melalui telepon: +6287888880014 dan *email*: nabilhpribadi@gmail.com.