TESIS – KI 42502

EKSTRAKSI FITUR *CONFLICT OF INTEREST* PADA ARTIKEL ILMIAH UNTUK MENENTUKAN KUALITAS *CITATION AUTHOR*

Akhmad Bakhrul Ilmi
NRP. 5116201046

DOSEN PEMBIMBING

Dr. Eng. Chastine Fatichah, M. Kom.
NIP: 197512202001122002

PROGRAM MAGISTER
DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M. Kom)
di

Institut Teknologi Sepuluh Nopember Surabaya

oleh:
AKHMAD BAKHRUL ILMI
Nrp. 5116201046

Dengan judul:
Ekstraksi Fitur Conflict of Interest Pada Artikel Ilmiah untuk Menentukan Kualitas Citation Author
Tanggal Ujian : 27 Juli 2018
Periode Wisuda : 2018 Gasal

Disetujui oleh:

1. Dr. Eng. Chastine Fatichah, M.Kom. ..... (Pembimbing I)
   NIP. 197512202001122002
2. Prof.Dr.Ir.Joko Lianto Buliali, M.Sc ..... (Penguji I)
   NIP. 196707271992031002
3. Dr. Eng. Darlis Heru Murti, S.Kom, M.Kom ..... (Penguji II)
   NIP. 197712172003121001
4. Dr. Eng. Nanik Suciati, S.Kom, M.Kom ..... (Penguji III)
   NIP. 197104281994122001

Dekan Fakultas Teknologi Informasi dan Komunikasi,

Dr. Agus Zainal Arifin, S.Kom, M.Kom
NIP. 19720809 199512 1 001

[Halaman ini sengaja dikosongkan]

Ekstraksi Fitur *Conflict of Interest* Pada Artikel Ilmiah untuk Menentukan Kualitas *Citation Author*

Nama Mahasiswa : Akhmad Bakhrul Ilmi

NRP : 5116 201 046

Pembimbing : Dr. Eng. Chastine Fatichah, M. Kom.

ABSTRAK

Sitasi pada publikasi ilmiah mempengaruhi kualitas artikel sehingga akan berpengaruh terhadap kredibilitas *author* (peneliti). Terdapat banyak cara untuk meningkatkan kredibilitas peneliti, salah satunya adalah dengan melakukan sitasi terhadap diri sendiri (*self citation*). Namun, proses *self citation* yang berlebihan mengurangi kualitas sitasi paper tersebut. Terdapat banyak penelitian yang membuat metode untuk mengukur kualitas self-citation yang tidak sesuai, salah satunya dengan menggunakan rasio self-citation pada jendela waktu. Akan tetapi, metode ini tidak mempertimbangkan kesesuaian topik penelitian paper utama terhadap paper yang mensitasinya. Sehingga diperlukan adanya penentuan kualitas sitasi pada author agar dapat diketahui apakah peneliti sering meggunakan citation yang tidak sesuai topiknya berdasarkan paper author dan paper sitasi.

Penelitian ini mengusulkan metode ekstraksi fitur *conflict of interest* untuk menentukan kualitas *citation* penulis artikel ilmiah. Hal ini dilakukan untuk mengetahui seberapa baik peneliti dalam menggunakan sitasinya. Terdapat 2 fitur yang diusulkan dalam penelitian ini. Pertama, fitur *confict of interest* yang didapatkan dari konflik kepentingan antara *author* paper dan *author* paper yang disitasi. Kedua, fitur similaritas konten yaitu fitur yang didapatkan dari kesamaan topik antar dokumen paper dan yang disitasinya. Metode similaritas yang digunakan adalah salah satu pendekatan deep learning yaitu *Siamese Neural Network* yang dikombinasikan dengan *Long Short Term Memory*. Kedua fitur ini selanjutnya diklasifikasi untuk menentukan kualitas *citation author*. Seluruh fitur akan diuji performanya pada proses klasifikasi. Hasil klasifikasi selanjutnya akan dihitung nilai akurasinya untuk mendapatkan performa fitur yang diusulkan. Hasil uji coba menunjukkan bahwa usulan fitur dapat digunakan untuk mengklasifikasi kualitas sitasi *author*. Hal ini ditunjukkan dengan nilai akurasi sebesar 66.67% pada klasifikasi *Random Forest* dan rata-rata akurasi sebesar 62% pada 3 klasifikasi yang digunakan.

Kata kunci*: Citation, Conflict of Interest, Ekstraksi Fitur, Deep Learning, Klasifikasi*

[Halaman ini sengaja dikosongkan]

Feature Extraction on Conflict of Interest for Detecting Quality of Scientific Author Citations

Student Name : Akhmad Bakhrul Ilmi

NRP : 5116 201 046

Supervisor : Dr. Eng. Chastine Fatichah, M. Kom.

ABSTRACT

Citation on scientific paper affect on article quality so that it will affect on author credibility. There are many ways to increase the credibility of researchers, one of them is to do a self-citation. However, this process makes the calculation in bibliometric becoming less accurate because it doesn't consider citation quality. There is some studies that proposed a method to measure an inappropriate self-citation, one of them is using self-citation ratio. But, this method doesnt consider topic relatedness between main paper and cited paper. So, its required to determine author's citation quality to know that author are using anomalous citation based on main paper and each cited paper.

This research proposed feature extraction conflict of interest to detect author's citation quality. It allows us to know how right an author use citation in publication. Two features are proposed in this research. First, conflict of interest feature, is obtained from interest conflict between paper author and citation's paper author. Second, content similarity feature, is obtained from the similarity between paper and cited papers of author. Deep learning approach is used to get the similarity of each document. Combination of Siamese neural network and Long Short-Term Memory can provide a better result on similarity based on training data. Last, all features will be combined with self-citation's count feature based on previous research and classified to detect author's citation quality. Features will be tested for its performance using classification. From the classification results, accuracy will be calculated to obtain the performance of the proposed feature. Based on the result, proposed feature can be used to classify author's citation quality. It is shown with 66,67% of accuracy by using Random Forest classification and 62% of average accuracy on 3 classifier.

***Keywords**: Citation, Conflict of Interest, Feature Extraction, Deep Learning, Classification*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

Segala puji dan syukur penulis panjatkan ke hadirat Allah Subhanahu wa ta'ala, atas segala rizki, berkah, nikmat serta karunia-Nya yang terlimpahkan kepada penulis, sehingga penulis akhirnya dapat menyelesaikan penelitian dengan judul "Ekstraksi Fitur *Conflict of Interest* Pada Artikel Ilmiah untuk Menentukan Kualitas *Citation Author*".

Penulis juga ingin mengucapkan banyak terimakasih kepada Disadari sepenuhnya bahwa tanpa bantuan dari berbagai pihak, penelitian ini tidak akan terselesaikan dengan hasil seperti sekarang ini. Oleh karena itu pada kesempatan ini penulis ingin menyampaikan terima kasih yang sebesar-besarnya, kepada:

1. Allah SWT atas limpahan nikmat iman, islam, kesehatan, waktu, serta berbagai kemudahan dari arah yang tidak pernah diduga sebelumnya, sehingga penulis dapat menyelesaikan penelitian ini dengan baik.
2. Keluarga tercinta yang tidak hentinya memberikan dukungan materil, do'a yang tulus, serta *belief* bahwa penulis harus senantiasa menjadi yang insan terbaik yang bermanfaat untuk orang sekitar dimanapun penulis berada.
3. Ibu Dr. Eng. Chastine Fatichah, M. Kom. selaku pembimbing dan Ibu Diana Purwitarasi S. Kom. M.Sc selaku pembimbing tak tertulis yang telah membantu, membimbing, dan memberikan motivasi kepada penulis dalam menyelesaikan Tugas Akhir ini dengan sabar.
4. Bapak Prof.Dr.Ir.Joko Lianto Buliali, M.Sc., Bapak Dr. Eng. Darlis Heru Murti, S. Kom, M. Kom, dan Ibu Dr. Eng. Nanik Suciati, S. Kom, M. Kom, selaku dosen penguji yang telah memberikan banyak saran dan arahan agar penulis mampu lebih baik dalam menyelesaikan penelitian.
5. Bapak Waskitho Wibisono, S.Kom., M.Eng. Ph.D., selaku Kaprodi S2 Teknik Informatika ITS Surabaya yang memfasilitasi mahasiswanya untuk belajar di Lab S2 hingga larut dalam rangka menyelesaikan penelitian.
6. Seluruh staf dosen, staf tata usaha dan karyawan perpustakaan Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember.
7. Tim Jajan; Amelia, Mbak Eva, Mbak Alifia, Adhi, Mbak Ulum, Kak Yaya, Ozzy, Kak Rozita, Mbak Myrna, Mbak Pipit, Kak Herna, Kak Udis, Mas Fatra dan Mas

Wahyu yang senantiasa menguatkan, membahagiakan, menyemangati dalam menyegerakan tesis, ibadah, juga makan tepat waktu dalam rangka menjaga kesehatan fisik, jasmani, dan tentunya rohani.

8. Geng Sosialita Informatika 2016, yang senantiasa memberikan *support* dan semangat dalam hal jasmani dan rohani terhadap penulis.
9. Rekan – rekan dan sahabat - sahabat S2 Teknik Informatika 2016 yang memberikan dorongan motivasi, bantuan, dan diskusi selama penelitian kepada penulis serta membuat hati senang dengan obrolan - obrolannya.
10. Teman – teman grup "Sarungandalan", Ampuh, Wahyu, Yudi, Habibi, dan Wiby yang senantiasa menghibur dengan lawakan – lawakannya dan bantuannya waktu pengerjaan tesis ini.
11. Semua pihak yang tidak dapat dituliskan satu per satu oleh penulis, terima kasih banyak atas doa dan dukungannya.

Semoga Allah SWT senantiasa menyayangi, menguatkan, memampukan, dan menunjukan jalan yang terbaik atas semua kebaikan yang telah diberikan. Penulis menyadari bahwa laporan penelitian ini tentunya masih jauh dari kesempurnaan. Oleh sebab itu, saran dan kritik sangat diharapkan untuk perbaikan dimasa yang akan datang. Semoga laporan penelitian ini dapat bermanfaat bagi penulis dan pembaca pada umumnya.

Surabaya, Agustus 2018

Akhmad Bakhrul Ilmi

DAFTAR ISI

DAFTAR GAMBAR

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

[Halaman ini sengaja dikosongkan]

BAB I
PENDAHULUAN

1.1. Latar Belakang

Pada setiap publikasi artikel ilmiah, sitasi sangat diperlukan untuk menghindari adanya plagiarisme dalam pembuatan artikel ilmiah [1]. Sitasi juga digunakan sebagai materi pendukung dalam penelitian yang sedang dilakukannya sehingga dapat membantu meyakinkan pembaca artikel ilmiah tersebut [2]. Sitasi dapat berasal dari beberapa sumber diantaranya buku, jurnal, koran, situs web, dan majalah [3]. Setiap artikel yang dikutip oleh peneliti, akan memberikan kredit terhadap peneliti sebelumnya.

Saat ini, terdapat peneliti yang menyalahgunakan sitasi dalam artikel yang dibuatnya seperti, *self-citations, negative citations, wrong citations, multi-authorship-biased citations, honorary citations, circumstantial citations, discriminatory citations, selective* dan *arbitrary citations.* Hal ini mengakibatkan pengukuran pada bibliometrik berbasis sitasi memiliki kekurangan yang fatal [4]. Pada dasarnya keilmuan memiliki peraturan masing-masing dan jurnal, peneliti dihormati berdasarkan performa sitasi yang didapatkan. Akan tetapi dengan adanya *self-citation* yang berlebihan mengakibatkan adanya kontradiksi antara peraturan pada perkebangan penelitian dan integritas dari jurnal atau peneliti tersebut [5].

Sitasi yang anomali yang berlebihan tersebut harus dianalisis terlebih dahulu agar hasil pengukuran tersebut menjadi tidak bias dan dapat dijadikan acuan. Oleh karena itu, diperlukan identifikasi adanya kecurangan terhadap penggunaan sitasi sebelum digunakan pengukuran tersebut. Dalam beberapa tahun terakhir, banyak penelitian yang meneliti tentang self-citation dan sebab-akibat dari *self-citation*. Akan tetapi, penelitian yang digunakan untuk menentukan peneliti yang abnormal berdasarkan *self-citation* yang berlebihan sangat sedikit. Maka dari itu penelitian ini bertujuan dapat mendeteksi, mengurangi kebiasaan dalam penggunakan *self-citation* yang berlebihan.

Pada penelitian sebelumnya, Tian Yu et. al. (2014) melakukan klasifikasi untuk mendeteksi adanya *self-citation* yang berlebihan pada jurnal menggunakan fitur jumlah sitasi dan *self-citation*nya [5]. Pada penelitian tersebut, beliau melakukan proses pengambilan fitur terhadap jurnal, akan tetapi untuk deteksi terhadap peneliti tidak terlalu banyak penelitian yang ada pada area tersebut. Feng Xia et. al. (2017) menyatakan bahwa

salah satu masalah yang terjadi pada bibliometrik berbasis sitasi adalah adanya kemungkinan perbedaan kepentingan [6]. Secara umum *Conflict of Interest* atau perbedaan kepentingan merupakan sebuah situasi dimana orang atau organisasi melakukan aksi perihal tujuan utama yang terpengaruhi oleh tujuan lainnya(pribadi, finansial)[7]. Salah satu contoh kasus adalah pada sitasi penelitian yang memiliki tujuan awal yakni melanjutkan penelitian yang sudah diteliti, akan tetapi digunakan untuk mencari keuntungan untuk pribadi atau orang lain dengan menaikkan nilai produktivitas dari peneliti tersebut. Maka dari itu diperlukannya deteksi adanya perbedaan kepentingan tersebut sebelum melakukan perhitungan bibliometriknya. Dari beberapa penelitian tersebut, saya mengusulkan sebuah metode ekstraksi fitur menggunakan fitur *conflict of interest* yang didapatkan dari beberapa cara yaitu perbedaan bidang minat pada peneliti dan sitasinya, dan kesamaan konten yang digunakan menggunakan pendekan deep learning.

Pada beberapa tahun terakhir, Deep Learning mampu menarik perhatian pada 20 sektor dari pengolahan bahasa manusia, yang secara umum terbagi ke dalam dua area besar [8]. Sektor pertama adalah mempelajari word embedding dengan melatikan training pada model bahasa [9] [10] [11], sedangkan sektor kedua adalah melakukan komposisi semantik untuk memperoleh tingkat representasi dari frasa atau kalimat [12]. Salah satu pendekatan deep learning yang mampu secara otomatis mempelajari fitur yang dideskripsikan dalam bentuk vektor adalah Recurrent Neural Networks (RNN). RNN dengan menggunakan word embedding mampu sukses diaplikasikan dalam kasus penggalian opini tanpa memerlukan modifikasi proses di dalamnya. Namun, RNN memiliki kelemahan dalam memahami keterhubungan dari suatu sequence yang terpisah dalam jarak yang cukup jauh. Long short term memory (LSTM) pada RNN didesain untuk mampu menutupi kekurangan RNN dalam memodelkan dependesi term yang cukup jauh [13]. Hal ini membuat metode ini sukses diterapkan pada berbagai sektor, diantaranya pemodelan bahasa, pengenalan suara, dan pemahaman bahasa yang diucapkan.

Deep Learning memiliki bermacam–macam arsitektur. Setiap arsitektur memiliki tujuan tersendiri. Pada kasus mempelajari similaritas antar dokumen dapat digunakan arsitektur *Siamese*. Arsiterktur ini menggunakan 2 model network, dimana hasil hidden layer dari 2 model tersebut dilakukan proses perhitungan similaritasnya. Penggunaan

arsitektur ini dapat memprediksi kedekatan antara 2 kalimat secara akurat [14]. Pada kasus dokumen artikel, data yang digunakan adalah abstrak dan judul yang memiliki kata yang cukup banyak dalam setiap dokumennya.

Pada penelitian ini bertujuan untuk mengembangkan metode untuk identifikasi peneliti yang melakukan kecurangan–kecurangan melalui sitasi yang digunakan menggunakan model klasifikasi yang berdasarkan dengan sitasi yang dilakukan oleh peneliti. Fitur yang digunakan pada proses pelatihan model adalah ekstraksi fitur dari analisis sitasi. Fitur pertama diekstrak dari bidang minat antara peneliti dan sitasi yang diteliti. Fitur kedua didapatkan dari similaritas konten dari artikel peneliti dengan artikel yang disitasi. Hasil similaritas tersebut didapatkan dengan menggunakan pendekatan *Deep Learning* yaitu menggunakan *Siamese Long Short Term Memory Network*. Data yang digunakan sebagai masukannya adalah pembobotan dari setiap judul dan abstrak artikel dari metode *embedding* kata. Fitur yang terakhir didapatkan dari relasi co-authorship antar peneliti. Semua fitur yang didapatkan, kemudian dilakukan penggabungan fitur. Setelah itu dilakukan proses pelatihan model berdasarkan fitur yang telah digabungkan.

1.2. Perumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut.

1. Bagaimana cara menentukan kualitas self-citation peneliti ilmiah berdasarkan sitasi secara presisi dan akurat?
2. Bagaimana cara mengekstrak fitur *conflict of interest* terhadap bidang penelitian *author*?
3. Bagaimana cara mengekstrak fitur *conflict of interest* terhadap similaritas konten menggunakan pendekatan *deep learning*?
4. Bagaimana cara mengevaluasi kinerja dari metode yang diusulkan?

1.3. Tujuan

Tujuan yang akan dicapai dalam pembuatan tesis ini adalah Melakukan ekstraksi fitur *conflict of interest* berdasarkan similaritas konten dan bidang penelitian pada artikel ilmiah untuk menentukan kualitas *self-citation author*.

1.4. Manfaat

Mengembangkan metode ekstraksi fitur dalam melakukan penentuan kualitas self-citation peneliti berdasarkan sitasi artikel ilmiah yang diharapkan dapat memudahkan pengguna dalam mengetahui kualitas sitasi peneliti setiap artikel ilmiah dan mengurangi adanya indikasi *self-citation* pada penelitian yang dilakukan.

1.5. Kontribusi Penelitian

Penelitian ini meiliki kontribusi pada ekstraksi fitur yang digunakan untuk menentukan kualitas *citation* peneliti, berdasarkan perbedaan bidang minat untuk setiap sitasi yang dilakukan, dan similaritas konten dari artikel yang disitasi.

1.6. Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Data dokumen yang digunakan adalah judul dan abstrak artikel ilmiah berbahasa Inggris.
2. Seluruh dataset yang digunakan berasal dari *a-miner* database.
3. Data peneliti yang digunakan sebagai *ground truth*, dianalisis secara semi-manual.

BAB II
KAJIAN PUSTAKA

Bab ini merupakan pembahasan dari referensi terkait yang telah dilakukan dalam menyelesaikan permasalahan sesuai dengan uraian pada latar belakang. Bab ini diawali dengan menjabarkan hal-hal yang diterapkan pada metode yang diusulkan, kelemahan yang terdapat pada penelitian sebelumnya, komparasi penelitian sebelumnya. Selanjutnya dilanjutkan dengan kelebihan dari metode yang akan digunakan untuk menyelesaikan permasalahan deteksi ketidaklengkapan referensi, metode usulan.

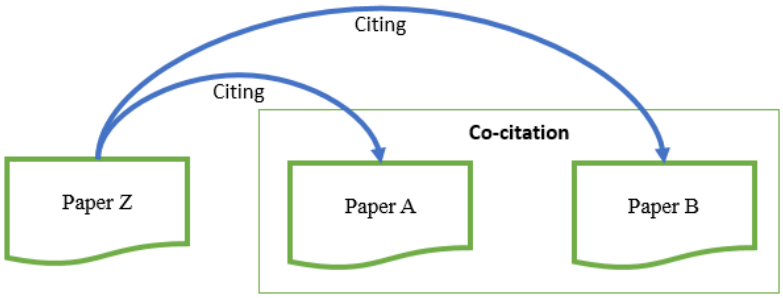
2.1. Kualitas *Citation Author*

Kualitas *citation author* adalah kualitas dari penggunaan sitasi oleh *author*. Kualitas citation *author* didapatkan dengan melakukan pencocokan setiap paper utama dengan paper sitasinya. Dengan adanya kualitas *citation author*, kualitas dari author dapat diketahui tidak hanya berdasarkan jumlah publikasi dan jumlah sitasi keseluruhan, tetapi juga keterhubungan dari topik penelitian yang disitasi. Sehingga, author tidak dapat melakukan manipulasi menggunakan *self-citation* atau kecurangan lainnya, seperti melakukan sitasi kepada penelitian yang berbeda dengan topik yang diajukan.

Pada penelitian sebelumnya, *h-index* digunakan untuk menentukan nilai produktifitas dan pengaruh publikasi peneliti berbasis sitasi [15]. Perhitungan nilai *h-index* didapatkan dengan menghitung jumlah sitasi berdasarkan jumlah publikasi yang dilakukan. H-index memiliki beberapa kelemahan, salah satunya adalah mudah untuk dimanipulasi dengan melakukan *self-citation* dikarenakan perhitungan yang hanya mengacu pada jumlah sitasi.

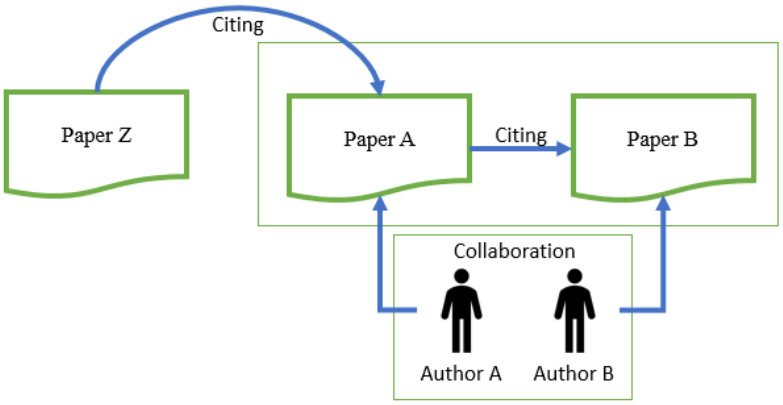
2.2. *Conflict of Interest* pada Penelitian

Conflict of Interest merupakan suatu kondisi dimana tidak terjadinya proses sitasi dari author paper lain terhadap dua paper yang memiliki keterkaitan topik (*co-citation*) [16].



Gambar 2.1 Illustrasi Co-citation

Berdasarkan Gambar 2.1, terdapat paper Z yang mensitasi dua paper, paper A dan paper B. Paper A dan paper B memiliki relasi *co-citation* karena kedua paper tersebut disitasi oleh paper yang sama yaitu paper Z.



Gambar 2.2 Illustrasi *Conflict of Interest*

Berdasarkan Gambar 2.2, author A dan author B pernah melakukan kolaborasi dan masing-masing memiliki paper. Namun, relasi antara paper A dan paper B memiliki relasi *Conflict of Interest* karena paper Z melakukan sitasi hanya ke paper A, sehingga relasi *co-citation* tidak terjadi antara paper A dan B.

Conflict of Interest merupakan perluasan lingkup dari *self-citation. Self-citation* sendiri merupakan sebuah proses sitasi yang dilakukan terhadap publikasi dari peneliti itu sendiri. *Self-citation* merupakan suatu hal yang wajar ketika paper yang disitasi berhubungan dengan paper publikasi dan tidak seritng dilakukan. Akan tetapi, beberapa peneliti melakukan proses *self-citation* dalam rangka mempromosikan dirinya [17]. Pengembangan dari *self-citation* yaitu perhitungan *conflict of interest* perlu dilakukan agar penggunaan sitasi tidak disalah gunakan untuk mempromosikan peneliti.

2.3. *Preprocessing* Teks

Tahap pra-pemrosesan teks bertujuan untuk membersihkan teks, sehingga teks hanya mengandung kata – kata yang diperlukan. Pada penelitian ini, pra-pemrosesan teks terdiri dari

- *Case Folding*

Merubah semua huruf pada dokumen menjadi huruf kecil

- Penghapusan Tanda Baca

Menghapus tanda baca yang ada pada dokumen.

- *Stopword Removal*

Menghapus kata – kata yang sering digunakan pada dokumen dimana tanpa kata – kata ini, dokumen tetap dapat memiliki arti yang sama.

- *Stemming*

Merubah seluruh term yang ada menjadi kata dasarnya.

2.4. K-Means Clustering

K-Means merupakan metode klasterisasi yang paling umum digunakan untuk mengelompokan dokumen [18]. Metode klasterisasi ini memiliki kekurangan dimana jumlah klaster perlu diketahui terlebih dahulu [19]. Pada umumnya K-Means menggunakan pendekatan VSM (*Vector Space Model*), dimana dokumen dimodelkan dalam vektor yang memiliki kata sebagai fitur. Setelah kumpulan dokumen telah melewati tahap pra-pemrosesan teks, seluruh kata pada kumpulan dokumen diekstrak untuk dijadikan fitur dokumen, pendekatan ini disebut juga *"bag-of-words model"*. Vektor dokumen dibentuk dengan menggunakan pembobotan Tf-Idf (*Term Frequency – Inverse Document Frequency*) pada fitur kata.

Pada pendekatan VSM, similaritas antar vektor dokumen dapat dihitung dengan menggunakan *cosine similarity*. Jika terdapat 2 vektor dokumen v_1^d dan v_2^d, maka nilai *cosine similarity* antara 2 vektor tersebut dihitung dengan menggunakan rumus (2.1), dimana $|v_1^d|$ merupakan nilai skalar dari v_1^d.

$$CosSim(v_1^d, v_2^d) = \frac{v_1^d \cdot v_2^d}{|v_1^d|_2 \ |v_2^d|_2} \quad (2.1)$$

Masukkan dari K-Means adalah kumpulan dokumen D dan parameter jumlah klaster k. Setelah vektor dokumen terbentuk, algoritma K-Means dilakukan seperti berikut:

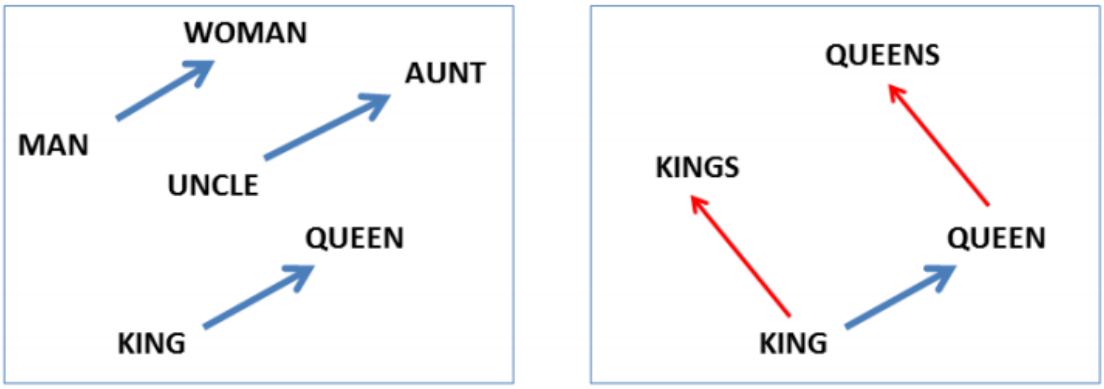
1. Pilih k dokumen secara acak sebagai *centroid* awal klaster $C = \{C_1, \dots, C_k\}$.
2. Hitung similaritas dokumen $d_i \in D$ dengan tiap *centroid* pada C menggunakan rumus (1).
3. Dokumen d_i akan menjadi anggota klaster yang memiliki nilai similaritas *centroid* tertinggi.
4. Ulangi langkah 2-3 untuk setiap dokumen dalam D.
5. Hitung ulang *centroid* untuk setiap $C_i \in C$.
6. Ulangi langkah 2-5, sampai tidak ada dokumen yang berpindah klaster.

2.5. Word Embedding

Word embedding merupakan kumpulan nama dari pemodelan bahasa dan teknik ekstraksi fitur pada natural language processing (NLP) dimana setiap kata atau phrasa dari suatu kosakata akan dipetakan menjadi vektor yang berupa bilangan real. Word embedding kerap digunakan dalam neural networks, reduksi dimensi pada matriks kemunculan kata, model probabilistik, dll. Metode word embedding ini juga digunakan sebagai input untuk meningkatkan performa pada pengolahan bahasa manusia seperti parsing sintaktik dan analisa sentimen.

Mikolov et. al. (2013) mengusulkan dua model log-linear untuk menghitung word embeddings dari suatu dataset secara efisien, yaitu bag-of-words dan skip gram [11]. Continous bag-of words (CBOW) model memprediksi kata saat ini berdasarkan konteks kata. Sedangkan skip-gram memprediksi kata-kata yang berada disekitar kata yang diberikan sesuai dengan kedekatan antara masing-masing vektor kata, seperti yang terlihat pada Gambar 2.3

Saat ini terdapat banyak sekali vektor *pre-trained,* yang dapat digunakan untuk mempercepat waktu perparasi pada *word embedding*. Yang sering digunakan adalah Word2Vec [11], dan GloVe [20]. Terdapat juga yang berbasis dependensi [21] dan lexicon [22].



Gambar 2.3 Contoh Word Embedding menggunakan Skip-gram

2.6. Long Short-Term Memory (LSTM)

LSTM adalah arsitektur recurrent neural network (RNN) yang didesain untuk memodelkan keterhubungan antara term yang memiliki interval yang jauh [13]. LSTM telah digunakan secara luas dalam pengolahan bahasa manusia seperti pada analisa sentimen, parsing sintaksis, kategorisasi dokumen yang memiliki ukuran yang panjang, dll. Secara umum, arsitektur dari LSTM digambarkan pada Gambar 2.4

LSTM terdiri dari empat elemen yaitu memory cell c, input gate i untuk mengontrol arus input yang masuk ke dalam neuron, output gate o untuk mengontrol efek dari aktivasi neuron pada neuron lainnya, dan forget gate f yang membuat neuron berada dalam status reset dari statusnya saat ini.

Secara umum, LSTM terdiri dari beberapa fungsi berikut:

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}) \quad (2.2)$$

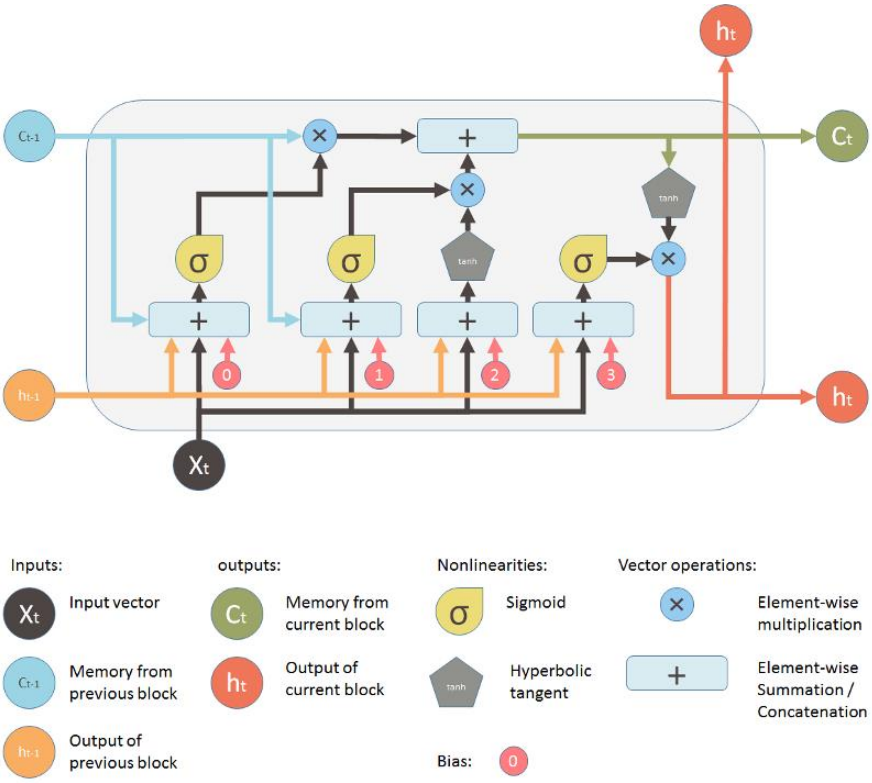
$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}) \quad (2.3)$$

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}) \quad (2.4)$$

$$u_t = \tanh(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}) \quad (2.5)$$

$$c_t = i_t \odot u_t + f_t \odot c_{t-1} \quad (2.6)$$

$$h_t = o_t \odot \tanh(c_t) \quad (2.7)$$



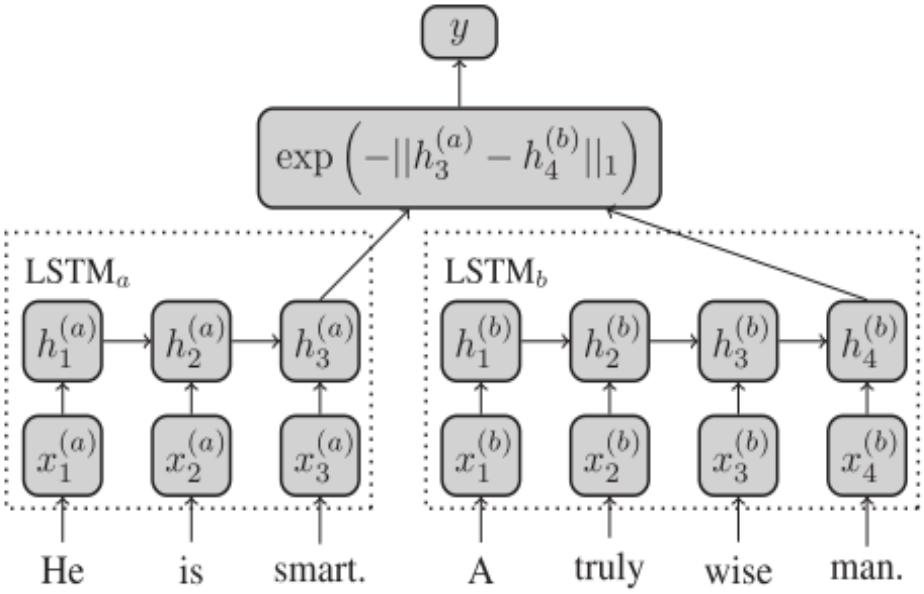
Gambar 2.4 Arsitektur LSTM

dimana W_k, U_k adalah bobot matrik antara dua hidden layer yang berurutan, antara input dan hidden layer, dan antara dua cell activation yang berurutan, masing-masing, terhubung dengan gate k (contoh: input, output, forget, dan cell), dan b_k adalah nilai vektor bias terkait. Simbol ⊙ menyatakan nilai produk untuk masing-masing elemen dari dua vektor. Nilai fungsi gate $\boldsymbol{\sigma}$ merupakan aktivasi sigmoid, dan g dan h adalah aktivasi dari cell input dan cell output, biasanya bernilai tanh.

2.7. *Siamese LSTM Network*

Siamese network adalah sebuah jaringan yang memiliki 2 atau lebih sub-jaringan yang identik. Jaringan ini dapat digunakan untuk pencarian similaritas secara *supervised* [14]. Kombinasi antara Siamese network dan LSTM dapat menghasilkan metode pencarian similaritas dalam teks, dimana pelatihan dilakukan pada LSTM. Siamese LSTM memiliki beberapa bagian yaitu word embedding, 2 sub-jaringan LSTM dan fungsi prediksi. Fungsi prediksi yang digunakan dalam penelitian ini adalah *cosine similarity*, dimana hasil yang diinginkan adalah nilai similaritas antar dokumen.

Pada Gambar 2.5 terdapat diagram proses *Siamese LSTM Network*. Dari kata yang sudah dilakukan pembobotan menggunakan word embedding. Input dari LSTM adalah data probabilitas dari dokumen 1 dan dokumen 2, yang kemudian dilakukan training. Bobot training yang ada pada LSTM digunakan bersama untuk kedua sub-jaringan. Output dari setiap LSTM yang berupa hidden vector LSTM, dimasukkan ke cosine similarity untuk mendapatkan nilai similaritasnya. Untuk mengetahui apakah model yang sudah dibuat dapat dipercaya, menggunakan MSE untuk evaluasinya.



Gambar 2.5 Diagram Proses Siamese LSTM Network

2.8. Evaluasi

Pada penelitian ini terdapat 3 jenis pengukuran evaluasi, yaitu pengukuran kualitas klaster data training deep learning, pengukuran error proses deep learning, dan pengukuran kehandalan klasifikasi dari fitur yang dihasilkan. Kualitas klaster data training deep learning diukur dengan *Silhouette Index* dan *Sum Squared Error*, dan kualitas klasifikasi diukur dengan *precision*, *recall*, dan *F-Measure*.

a. *Silhouette Index*

Pengukuran ini membandingkan similaritas data di dalam suatu klaster dengan klaster lain untuk mengevaluasi konsistensi data pada klaster. Jika terdapat beberapa klaster dimana data *i* merupakan anggota klaster *C*, dan kumpulan klaster lainnya dilambangkan

dengan *C'*. Nilai kualitas klaster didapatkan dengan menghitung rata – rata nilai *Silhouette Index* untuk setiap data pada klaster. Nilai *Silhouette Index* untuk data *i*, atau *s(i)* dihitung dengan cara:

1. Hitung *a(i)* yang merupakan nilai jarak rata – rata antara *i* dengan anggota lain di dalam klaster yang sama.
2. Untuk setiap $c \in C'$, hitung *d(i,c)* yang merupakan nilai jarak rata – rata antara *i* dengan anggota klaster *c*. Jarak antara *i* dan klaster terdekat didapatkan dengan *b(i) = min(d(i,c))*.
3. Sehingga, *s(i)* dapat dihitung dengan *s(i) = (b(i) – a(i)) / max(a(i), b(i))*.

b. *Sum Squared Error*

Pengukuran sum squared error digunakan untuk mengetahui error suatu hasil prediksi terhadap hasil kebenarannya. Pada kasus clustering, SSE digunakan untuk menentukan error dari setiap data dengan titik tengahnya. Rumus perhitungan SSE terdapat pada rumus (2.8)

$$SSE = \sum_{i=1}^{n}(y_i - f(x_i)) \quad (2.8)$$

c. *Precision*

Pengukuran *precision* digunakan untuk mengetahui persentase label yang benar dari label yang dihasilkan metode usulan, dihitung dengan rumus (2.9).

$$precision = \frac{|\{relevant\ labels\} \cap \{retrieved\ labels\}|}{|\{retrieved\ labels\}|} \quad (2.9)$$

d. *Recall*

Pengukuran *recall* digunakan untuk mengetahui persentasi label yang benar dari seluruh label benar yang ada, dihitung dengan rumus (2.10).

$$recall = \frac{|\{relevant\ labels\} \cap \{retrieved\ labels\}|}{|\{relevant\ labels\}|} \quad (2.10)$$

e. *F-Measure*

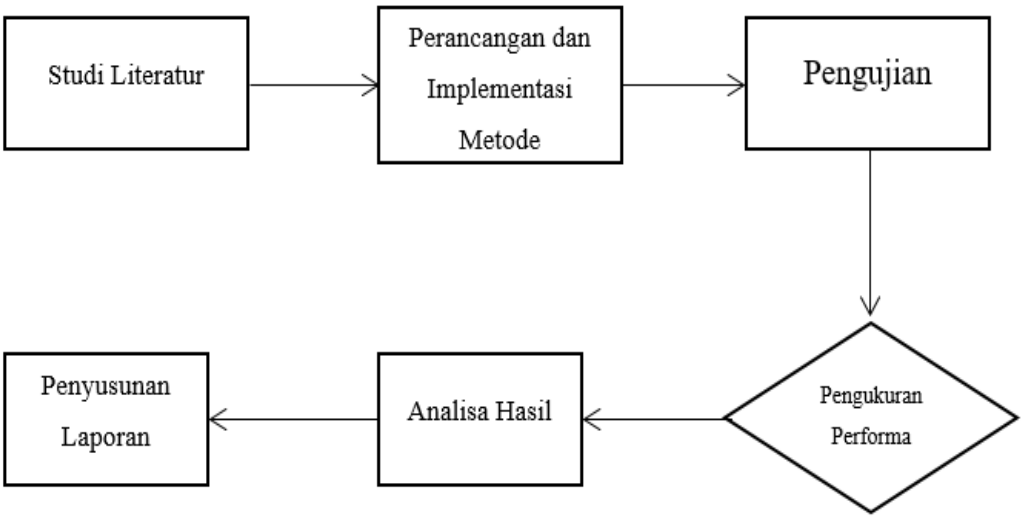
Pengukuran ini mengkombinasikan *precision* dan *recall* untuk mendapatkan keseimbangan dari kedua pengukuran tersebut, dihitung dengan rumus (2.11).

$$F = 2 * \frac{precision * recall}{precision + recall} \quad (2.11)$$

[Halaman ini sengaja dikosongkan]

BAB III
METODOLOGI PENELITIAN

Bab ini akan memaparkan tentang metodologi penelitian yang digunakan pada penelitian ini, yang terdiri dari (1) studi literatur, (2) desain dan implementasi, (3) pengujian, dan (4) dokumentasi dan pembuatan laporan. Ilustrasi alur metodologi penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Metodologi Penelitian

3.1. Studi Literatur

Tahap studi literatur bertujuan untuk mengumpulkan referensi - referensi yang dapat menunjang penelitian. Sumber referensi dapat berupa jurnal ilmiah atau buku teks. Referensi yang dikumpulkan berhubungan dengan metode ekstraksi fitur untuk menentukan kualitas self-citation author. Referensi tersebut digunakan untuk merumuskan permasalahan yang menjadi landasan dilakukannya penelitian ini dan solusi yang akan diusulkan. Berdasarkan studi literatur yang telah dilakukan, informasi yang berkaitan dengan penilitian yang dilakukan ini, seperti berikut:

1. Banyaknya author yang melakukan *self-citation* berlebihan menyebabkan *impact factor* jurnal ataupun h-index peneliti menjadi bias.

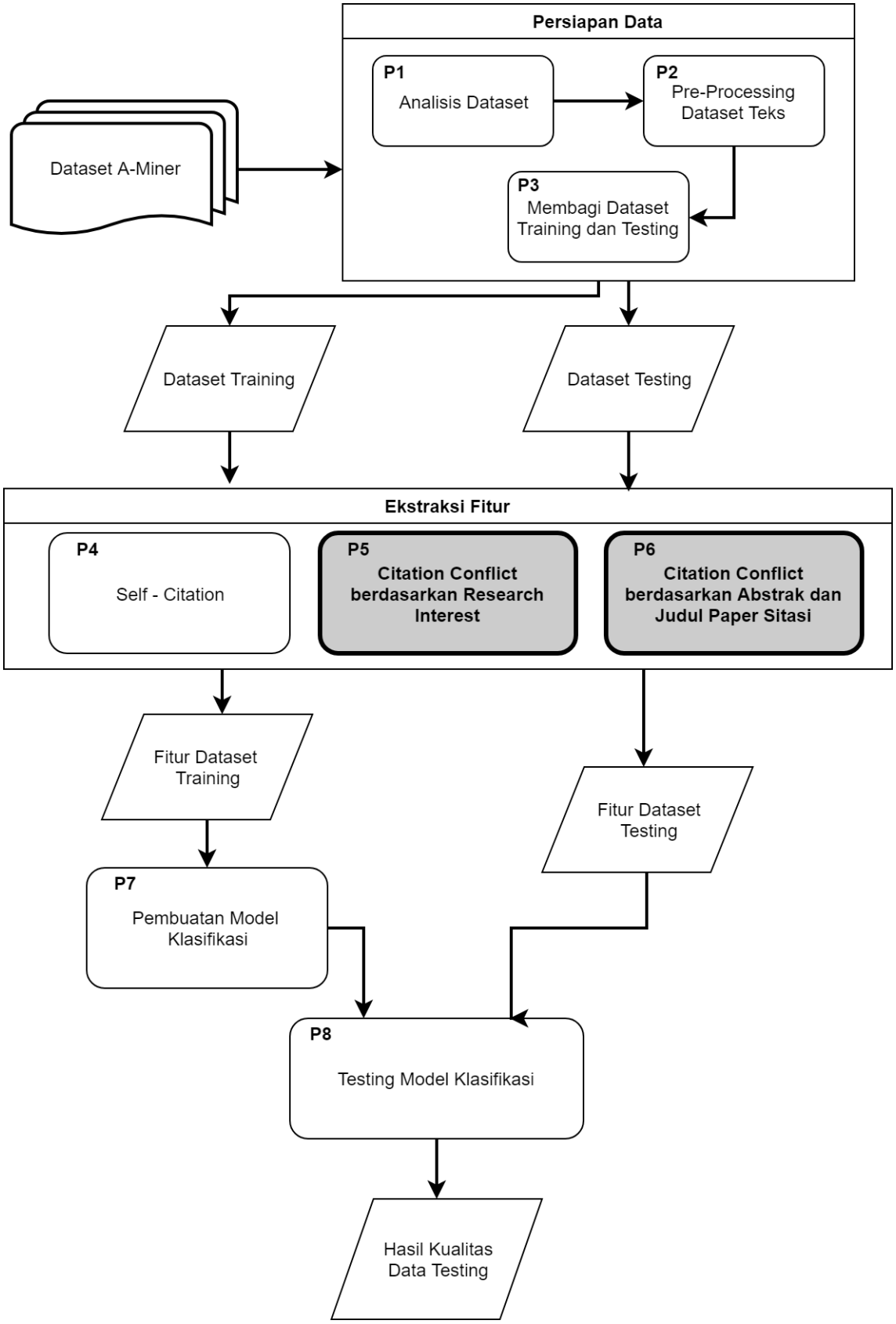
2. Pada sitasi memungkinan author untuk melakukan sitasi yang anomali, dimana sitasi tersebut mempunyai perbedaan relasi kepentingan pada paper yang disitasi.
3. Siamese neural network adalah ebuah jaringan yang memiliki 2 atau lebih sub-jaringan yang identik. Jaringan ini dapat digunakan untuk pencarian similaritas secara supervised.
4. Co-authorship antar author juga memiliki pengaruh pada proses terjadinya anomali sitasi.

3.2. Perancangan Sistem

Alur proses metode penentuan kualitas self-citation yang diusulkan terdiri dari beberapa tahap, seperti pada Gambar 3.2. Pertama, tahap persiapan data dilakukan analisis data pada dataset untuk mencari data yang dibutuhkan dan pencarian *groundtruth*, kemudian dilakukan preprocessing teks pada judul dan abstrak dari data yang digunakan. Kedua, dilakukan perhitungan adanya konflik kepentingan antar author dengan author yang disitasi. Ketiga, dilakukan perhitungan konflik kepentingan antar co-authorship pada jaringan co-authorshipnya. Selanjutnya, dilakukan perhitungan similaritas konten yang ada pada paper author dan paper yang disitasi. Perhitungan similaritas dilakukan dengan menggunakan Siamese neural network dan LSTM. Dari ketiga 3 perhitungan sebelumnya diambil fitur berdasarkan setiap perhitungannya dalam jendela waktu (2 tahun terakhir, 4 tahun terakhir, 6 tahun terakhir). Kemudian setiap fitur yang didapatkan digabung menjadi satu dan dilakukan proses klasifikasi. Hasil klasifikasi ini akan dilakukan evaluasi. Tahap – tahap proses metode usulan akan dijelaskan pada subbab berikut.

3.2.1. Persiapan Data

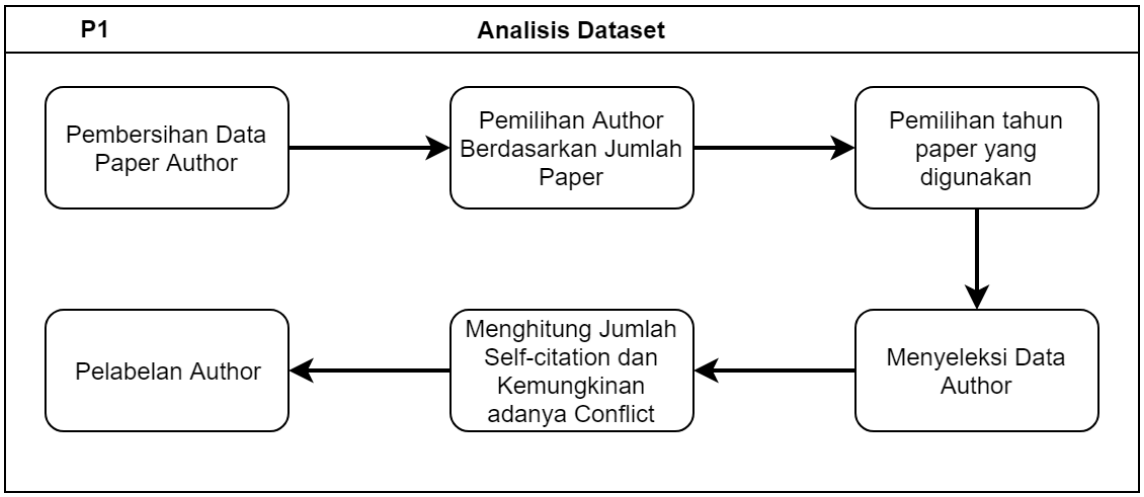
Pada modul ini akan dilakukan proses untuk mempersiapkan dan memastikan dokumen yang diproses telah siap untuk digunakan pada modul selanjutnya. Pada tahap ini dilakukan analisis dataset yang digunakan dan preprocessing data teks yang akan digunakan untuk tahap selanjutnya. Analisis data bertujuan untuk memilih data – data yang akan digunakan dan pengambilan *ground truth* sebagai penentu hasil evaluasi pada klasifikasi. Preprocessing data teks bertujuan untuk mengubah teks abstrak dan judul menjadi teks baku yang dapat diolah pada tahapan selanjutnya.



Gambar 3.2. Alur Proses Metode Usulan

a) Analisis Data

Dataset yang digunakan adalah dataset *Arnet Miner citation social network* [23]. Dataset tersebut memiliki jumlah sekitar 2.000.000 peneliti. Pada tahap ini dilakukan proses analisis data terhadap peneliti yang ada. Tahap ini digunakan untuk mendapatkan dataset groundtruth yang digunakan pada proses klasifikasi. Ada beberapa proses yang dilakukan pada tahap ini. Proses detail pada tahap ini terdapat pada Gambar 3.3



Gambar 3.3. Alur Proses Analisis Dataset

- Pembersihan Data Paper

  Pada tahap ini dilakukan proses pembersihan data paper yang tidak memiliki judul, abstrak, dan author. Hal ini dilakukan untuk menghindari adanya data kosong yang nantinya digunakan pada proses ekstraksi fitur. Dari sekitar 2.000.000 paper yang ada setelah dilakukan pembersihan, jumlah paper yang digunakan menjadi ±1.600.000 paper.

- Pemilihan Author Berdasarkan Jumlah Paper

  Pada Tahap ini dilakukan proses pemilihan author berdasarkan jumlah paper yang pernah diterbitkan yang ada pada dataset yang digunakan. Hal ini dilakukan untuk menghindari adanya author yang jarang menerbitkan hasil penelitiannya. Pada penelitian ini, jumlah paper yang dijadikan *threshold* adalah 50 paper. Dari sekitar 2.000.000 peneliti yang ada setelah dipilih, author yang memiliki publikasi lebih dari 50 paper sebanyak 9971 author.

- Pemilihan Tahun Paper

  Pada Tahap ini dilakukan proses pemilihan tahun berdasarkan jumlah paper pada setiap tahunnya. Sebelumnya, dilakukan proses rekapitulasi paper untuk 9971 author yang didapatkan sebelumnya. Hasil rekapitulasi jumlah paper untuk setiap tahunnya terdapat pada Gambar 3.4. Pada grafik tersebut didapatkan bahwa dari tahun 1980 hingga 2009 jumlah paper terbilang naik dan turun pada tahun 2010 dan seterusnya. Pemilihan tahun pertama diambil dari kenaikan yang cukup signifikan pada periodenya yaitu sekitar tahun 2001. Kemudian tahun terakhir didapatkan dari perbedaan jumlah paper yang tidak terpaut jauh pada tahun sebelumnya yaitu pada tahun 2012, dikarenakan pada tahun 2013 dapat dibilang terpaut jauh dengan jumlah paper pada tahun 2012.

Gambar 3.4. Grafik Jumlah Paper per Tahun

Dari hasil analisis diatas didapatkan bahwa paper yang digunakan pada penelitian ini adalah paper yang diterbitkan pada tahun 2001 – 2012. Dari 12 tahun tersebut data akan dilakukan pemisahan dalam beberapa periode untuk perlakuan skenario uji coba pada penelitian ini. Hal ini dilakukan untuk mengetahui pola adanya anomali pada sitasi untuk setiap periodenya. Periode yang digunakan pada penelitian ini adalah 4 tahun untuk setiap periodenya.

- Seleksi Data Author

  Dari paper yang ada pada tahun 2001 – 2012, kemudian diseleksi kembali author berdasarkan jumlah paper pada tahun 2001 - 2012. Karena author yang digunakan pada penelitian ini diambil kurang lebih adalah 200 author, maka author yang memiliki jumlah paper sedikit tidak dihiraukan. Proses ini diawali dengan melakukan perhitungan jumlah paper tiap author. Kemudian setiap author dimasukkan ke dalam variasi jumlah paper dengan jarak 50 (1–50, 51–100, 100–150, 151–200, 200+). Dari setiap variasi didapatkan yang paling mendekati 200 author dan memiliki cukup paper untuk setiap authornya adalah variasi 151-200 dengan jumlah author 187.

- Perhitungan Jumlah Self-Citation dan Kemungkinan Conflict

  Dari 187 author yang didapatkan, dilakukan proses pengambilan jumlah kejadian *self-citation* dan kemungkinan konflik pada sitasi pada setiap author. Untuk penghitungan jumlah *self-citation*, dilakukan dengan menghitung adanya sitasi yang dilakukan oleh author terhadap dirinya sendiri dan *co-author*nya pada paper tersebut. Langkah-langkah yang dilakukan pada perhitungan jumlah *self-citation* terdapat pada Gambar Gambar 3.5.

```
1. Ambil data id author yang sudah dipilih pada proses analisis data
2. Definisi counter self-citation = 0
3. Ambil data paper untuk setiap author pada tiap periode
4. Untuk setiap sitasi pada paper, dilakukan pengecekan author paper
sitasinya
5. Jika pada paper sitasinya terdapat id author yang digunakan, maka
counter self-citation bertambah 1
6. Proses 2 - 6 dilakukan hingga seluruh author sudah dicari self-
citationnya
```

Gambar 3.5. *Pseudocode* pencarian *self-citation author*

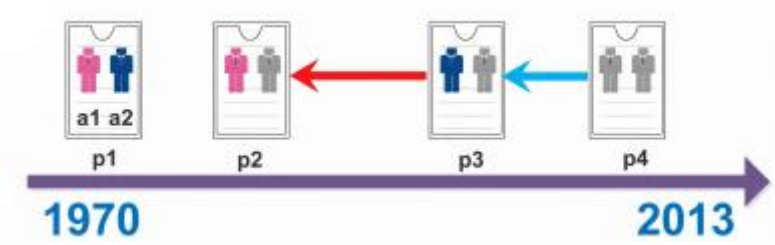
Untuk penghitungan kemungkinan konflik, dilakukan menggunakan konsep *co-citation* dan *co-authorship* [16]. Contoh cara mengetahui kemungkinan konflik terdapat pada Gambar 3.7. Misalkan pada suatu paper p1 author a1 dan a2 berkolaborasi, kemudian author a1 melakukan sitasi pada paper author a2. Jika

terdapat author lain yang melakukan sitasi terhadap paper author a1, maka akan dilakukan pengecekan apakah paper author lain tersebut melakukan sitasi terhadap paper author a2 yang disitais oleh author a1. Jika paper author lain melakukan sitasi juga terhadap paper author a2 yang disitasi oleh author a1 maka tidak terjadi kemungkinan konflik, akan tetapi sebaliknya jika tidak melakuakn sitasi maka dapat dikatakan bahwa paper author a1 terdapat kemungkinan konflik pada sitasinya. Langkah-langkah yang dilakukan pada proses perhitungan kemungkinan adanya konflik terdapat pada

Gambar 3.6.

```
1. Ambil data id author yang sudah dipilih pada proses analisis data
2. Untuk Setiap id author di cari kolaborasi berdasarkan paper yang
pernah dipublikasikan
3. Definisi counter conflict = 0
4. Ambil data paper A untuk setiap author pada tiap periode
5. Untuk setiap sitasi pada paper A, dilakukan pengecekan apakah
terdapat salah satu author yang pernah berkolaborasi
6. Jika terdapat sitasi yang berdasarkan tabel kolaborasi (paper B),
dilakukan proses pencarian paper lain yang pernah melakukan sitasi
terhadap paper author.
7. Untuk setiap paper lain (paper C) tersebut, dilakukan pengecekan
terhadap sitasinya
8.1. Jika paper C melakukan sitasi terhadap paper A dan paper B maka
relasi terhadap paper A dan paper B dianggap bukan konflik.
8.2. Jika paper C hanya melakukan sitasi terhadap paper A, maka
counter conflict bertambah 1.
9. Proses 2 - 8 dilakukan hingga seluruh author sudah dicari self-
citationnya
```

Gambar 3.6. *Pseudocode* pencarian kemungkinan adanya konflik



Gambar 3.7. Contoh Penggunakan Konsep *Co-citation* pada *Conflict of Interest*

- Pelabelan Kualitas *Citation Author*

  Setelah didapatkan jumlah *self-citation* dan jumlah kemungkinan konflik, dilakukan proses pelabelan kualitas *citation author*. Kualitas sitasi *author* ditentukan oleh dua kriteria yaitu jumlah *self-citation* dan jumlah kemungkinan konflik. Jika jumlah *self-citation* dan jumlah kemungkinan konflik tinggi, maka kualitas sitasi author tersebut adalah kurang baik. Sebaliknya jika jumlah *self-citation* dan jumlah kemungkinan konflik rendah, maka kualitas sitasi *author* tersebut adalah baik. Pseudocode untuk melakukan pelabelan kualitas sitasi *author* terdapat pada Gambar 3.8. Contoh hasil proses pelabelan kualitas sitasi *author* yang dilakukan terdapat pada Gambar 3.9

```
1.Hitung jumlah self-citation dan kemungkinan conflict tiap author
2.Hitung rasio self-citation dan rasio conflict dibandingkan
dengan jumlah sitasi.
3.Hitung rata-rata rasio self-citation dan rasio conflict.
4.Pembobotan setiap rasio untuk menentukan kualitas sitasi author
5.Hasil perkalian bobot dengan rata-rata rasio tersebut akan
dijadikan cut-off kelas baik dan kurang baik.
6.1 Jika melebihi nilai bobot, maka kualitas sitasi author
tersebut adalah kurang baik
6.2 Jika kurang dari nilai bobot, maka kualitas sitasi author
tersebut adalah baik
```

Gambar 3.8. *Pseudocode* penentuan kualitas *citation author* (*groundtruth*)

ID_Author	Self Citation	Positif CoI	Negatif CoI	Total Citation	Rasio A	Rasio B	Rasio C	Bobot	Label	Rata A	26.69753
427225	2288	263	201	8377	29.79768218	3.437438617	2.848163426	10.02148741	0	Rata B	2.003878
3601	1517	175	145	5091	26.49027734	0.541919031	1.179470832	11.05087409	0	Rata C	2.568226
75630	831	17	37	3137	24.03278689	3.245901639	3.442622951	8.645202423	1		
378168	2199	297	315	9150	24.42196532	0.674373796	1.469171484	9.580327869	1	A	0.3
47357	1014	28	61	4152	37.87635026	9.675936566	4.711560561	8.196050096	1	B	0.2
104908	3296	842	410	8702	34.54329775	2.325029656	2.894424674	15.65387267	0	C	0.5
1297710	1456	98	122	4215	21.50672646	0.269058296	1.757847534	12.27520759	0	Bobot	9.694147
7281	1199	15	98	5575	29.28325029	1.839785358	2.529704868	7.384753363	1		
87601	764	48	66	2609	27.98264642	0.911062907	2.472885033	10.41778459	0		

Gambar 3.9. Hasil proses pelabelan kualitas *citation author*

b) Praproses Teks

Praproses teks bertujuan untuk mempersiapkan data teks untuk dapat diproses pada tahap selanjutnya. Tahap ini terdiri dari *case folding*, penghilangan tanda baca, penghilangan *stopwords* dan *stemming*. Contoh dari tahap pra-pemrosesan teks dapat dilihat pada Gambar 3.10

Teks Awal	Teks Setelah *Preprocessing*
On the Multiple Implementation of Abstract Data Types Within a Computation' A fundamental step in the software design process is the selection of a refinement (implementation) for a data abstraction. This step traditionally involves investigating the expected performance of a system under different refinements of an abstraction and then selecting a single alternative which minimizes some performance cost metric. In this paper we reformulate this design step to allow different refinements of the same data abstraction within a computation.	multiple implementation abstract data type within computation fundamental step software design process selection refinement implementation data abstraction step traditionally involve investigate expect performance system different refinements abstraction select single alternative minimize performance cost metric paper reformulate design step allow different refinements data abstraction within computation

Gambar 3.10. Contoh Tahap Preprocessing Teks

c) Pembagian Data Latih dan Data Uji

Pada tahap ini dilakukan proses pembagian data latih dan data uji dari dataset yang sudah didapatkan pada tahap sebelumnya. Pembagian dilakukan dengan menggunakan random sampling dengan jumlah persentase untuk data latih lebih dari data ujinya. Prosentase yang digunakan pada penelitian ini adalah 70% untuk data latih dan 30% untuk data uji.

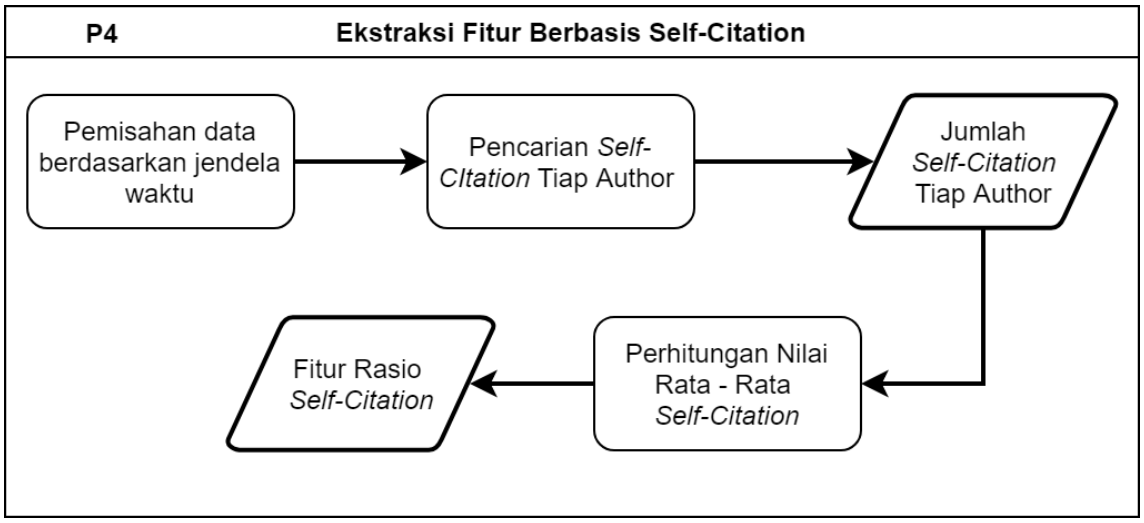
3.2.2. Ekstraksi Fitur Berbasis *Self-Citation*

Pada modul ini akan dilakukan proses untuk ekstraksi fitur menggunakan kerjadian *self-citation* pada author tersebut. Pada tahap ini memiliki masukan data peneliti, paper peneliti, dan sitasi setiap paper yang ditelitinya. Proses detail terdapat pada Gambar 3.12.

Pada proses pencarian *Self-Citation*, bertujuan untuk mendapatkan relasi paper dan sitasinya yang memiliki sitasi terhadap author itu sendiri. Hal ini dilakukan dengan melakukan pencarian author paper yang sama dengan author paper sitasi. Langkah-langkah yang dilakukan untuk mendapatkan author yang sama antara paper dan paper sitasi terdapat pada Gambar 3.11. Pseudocode pencarian *self-citation*

```
1. Ambil data id author yang sudah dipilih pada proses analisis
   data
2. Definisi counter self-citation = 0
3. Ambil data paper untuk setiap author pada tiap periode
4. Untuk setiap sitasi pada paper, dilakukan pengecekan author
   paper sitasinya
5. Jika pada paper sitasinya terdapat id author yang digunakan,
   maka counter self-citation bertambah 1
6. Proses 2 - 6 dilakukan hingga seluruh author sudah dicari
   self-citationnya
```

Gambar 3.11. Pseudocode pencarian *self-citation*



Gambar 3.12. Tahap Ekstraksi Fitur Berbasis *Self-Citation*

Perhitungan nilai rasio *self-citation* dilakukan untuk setiap author. Nilai *self-citation* dapat dihitung dengan formula 3.1. Contoh tahap perhitungan *fitur self-citation* terdapat pada Gambar 3.13.

$$SC = \frac{Jumlah\ Self-Citation}{Jumlah\ sitasi\ pada\ seluruh\ paper} \quad (3.1)$$

Tabel 3.1 Contoh data paper utama dan paper sitasi

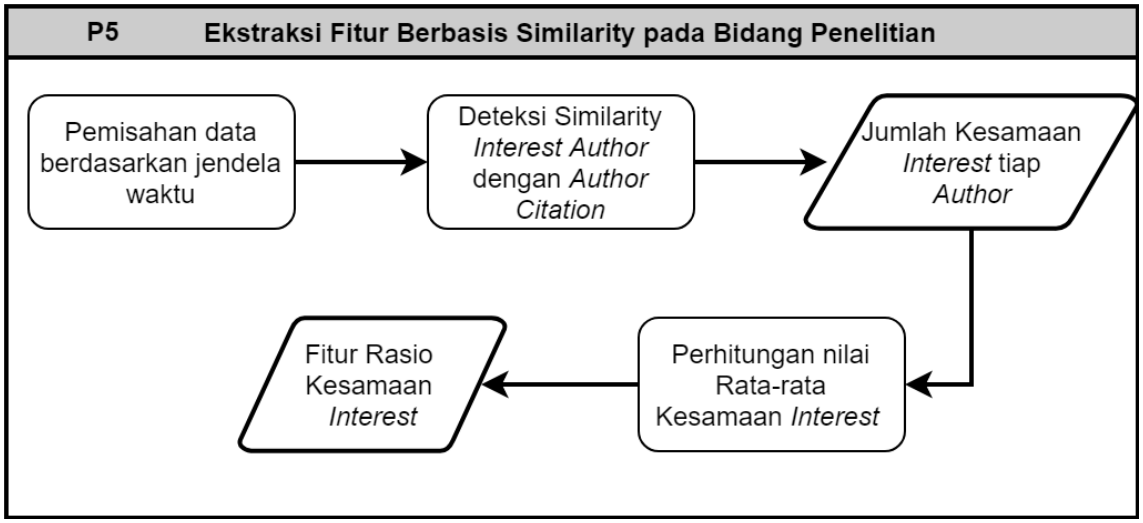
Author yang diteliti: Jason Cong Paper: *Performance-driven mapping for CPLD architectures*	
Judul Paper Sitasi	*Author* Paper Sitasi
Combinational logic synthesis for LUT based field programmable gate arrays	Jason Cong, Yuzheng Ding

Judul Paper Sitasi	*Author* Paper Sitasi
Technology mapping for large complex PLDs	Jason Helge Anderson, Stephen Dean Brown
Technology mapping for k/m-macrocell based FPGAs	Jason Cong, Hui Huang, Xin Yuan
Synthesis and Optimization of Digital Circuits	Giovanni De Micheli
Cross-talk immune VLSI design using a network of PLAs embedded in a regular layout fabric	Sunil P. Khatri, Robert K. Brayton, Alberto Sangiovanni-Vincentelli
DAG-Map: Graph-Based FPGA Technology Mapping for Delay Optimization	Kuang-Chien Chen, Jason Cong, Yuzheng Ding, Andrew B. Kahng, Peter Trajmar
A Fast Partitioning Method for PLA-Based FPGAs	Zafar Hasan, David Harrison, Maciej Ciesielski

Pada Tabel 3.1, terdapat contoh data paper yang digunakan. Dari 7 sitasi yang ada pada paper peneliti, terdapat 3 *self-citation*, maka dari itu pada paper tersebut rasio *self-citationnya*, adalah 3/7. Setelah didapatkan untuk satu paper, kemudian dilakukan untuk seluruh paper yang dimiliki oleh peneliti. Untuk contoh kasus diatas karena peneliti memiliki 1 paper maka hasil output dari fitur *self-citation* adalah 3/7 atau 0.429.

3.2.3. Ekstraksi Fitur Berbasis Conflict of Interest

Pada modul ini akan dilakukan proses untuk ekstraksi fitur menggunakan konflik kepentingan antar author dan author paper yang disitasi. Pada tahap ini memiliki masukan data peneliti, kepentingan setiap peneliti, dan hubungan referensi setiap paper peneliti. Proses detail terdapat pada Gambar 3.14.



Gambar 3.13. Tahap Ekstraksi Fitur Berbasis Conflict of Interest

Pada proses deteksi similaritas *interest* a*uthor* dilakukan dengan melakukan proses similaritas antar bidang penelitian *author* dengan *author citation*-nya. Kemudian hasil similaritas tersebut akan dibandingkan dengan threshold. Jika hasil similaritas lebih dari threshold, maka bidang penelitian author tersebut terdeteksi sama. Langkah-langkah yang dilakukan proses terdapat pada Gambar 3.14. *Pseudocode* proses deteksi similaritas *interest author*

```
1. Load Word2Vec yang sudah disediakan Google
2. Ambil data id author yang sudah dipilih pada proses analisa data
3. Definisi counter = 0
4. Ambil data paper untuk setiap author pada tiap periode
5. Untuk setiap sitasi pada paper, dilakukan pengecekan bidang
   penelitian author paper sitasinya
6. Untuk setiap bidang penelitian pada author dan bidang penelitian
   author sitasinya dilakukan proses similarity menggunakan cosine
   similarity
7. Jika terdapat salah satu bidang penelitian melebihi threshold,
   maka counter bertambah 1
8. Proses 2 - 7 dilakukan hingga seluruh author sudah didapatkan
   jumlah counternya
```

Gambar 3.14. *Pseudocode* proses deteksi similaritas *interest author*

Perhitungan nilai Conflict of Interest dilakukan untuk setiap paper. Nilai Conflict of Interest dapat dihitung dengan formula 3.2. Setelah nilai Conflict of Interest dihitung, dilakukan penjumlahan keseluruhan nilai setiap paper kemudian dibagi jumlah papernya.

$$CoI = \frac{Jumlah\ Conflict\ of\ Interest}{Jumlah\ Sitasi\ pada\ Paper} \tag{3.2}$$

Tabel 3.2 Contoh data paper utama, paper sitasi dan bidang penelitiannya

Author yang diteliti: Jason Cong Paper: *Performance-driven mapping for CPLD architectures* Bidang Penelitian: *resource utilization; polyhedral code generation; actual C code*			
No.	Judul Paper Sitasi	*Author* Paper Sitasi	Bidang Penelitian
1	*Combinational logic synthesis for LUT based field programmable gate arrays*	(1) Jason Cong, (2) Yuzheng Ding	(1) resource utilization; polyhedral code generation; actual C code (2) interval heap

	Judul Paper Sitasi	*Author* Paper Sitasi	Bidang Penelitian
2	*Technology mapping for large complex PLDs*	(1) Jason Helge Anderson, (2) Stephen Dean Brown	(1) FPGA architecture; FPGA hardware structure; dynamic power; glitch power (2) FPGA routing architecture; routing delay; cache design; FPGA CAD flow
3	*Synthesis and Optimization of Digital Circuits*	(1) Giovanni De Micheli	(1) finite state machine; optimal state assignment; computer program

Pada Tabel 3.2, terdapat contoh paper utama dan paper sitasinya. Berdasarkan contoh, didapatkan bahwa setiap author memiliki bidang penelitian yang berbeda-beda. Maka dari itu, untuk menentukan adanya kemiripan digunakan proses similaritas menggunakan word embedding dan *cosine similarity*. Proses dilakukan dengan melakukan pembandingan tiap bidang penelitian pada author yang diteliti dengan bidang penelitian author paper sitasi. Contoh, terdapat pada author Jason Cong dengan Yuzheng Ding, *resource utilizating* dibandingkan dengan *internal heap*, *polyhedral code generation* dibandingkan dengan *internal heap*, dan seterusnya. Setiap hasil similaritas akan dicocokkan dengan threshold yang ditentukan sebelumnya. Jika nilai similaritas melebihi threshold maka kedua bidang tersebut dianggap mirip.

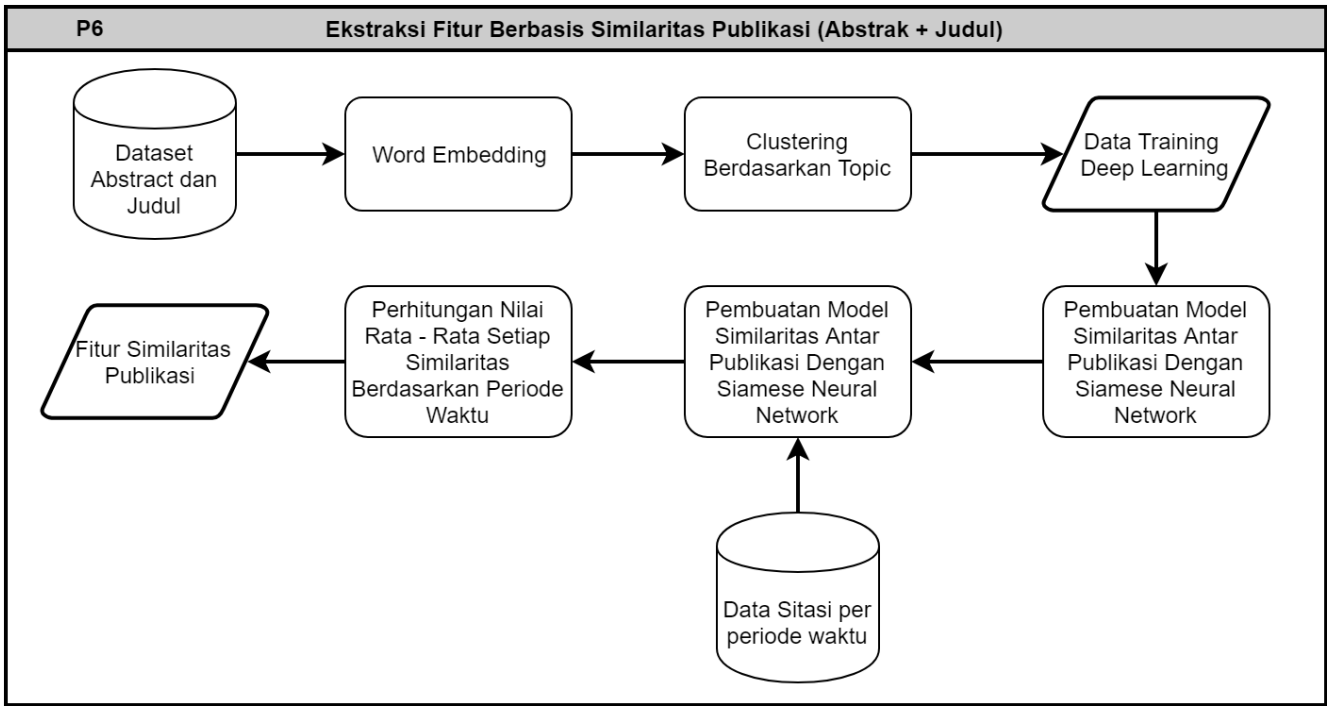
Pada Tabel 3.3, didapatkan hasil similaritas berdasarkan proses deteksi similaritas. Jika terdapat salah satu dari author dan author sitasi yang memiliki kemiripan, maka paper utama dianggap memiliki kemiripan terhadap paper sitasi. Berdasarkan Tabel 3.3 didapatkan bahwa hasil nilai fitur CoI untuk 1 paper adalah 2/3 atau 0,67 dengan nilai 2 adalah jumlah paper sitasi yang mirip dengan paper utama dan 3 adalah jumlah sitasi yang ada.

Tabel 3.3 Hasil deteksi similaritas *interest author*

No. Paper	*Author* Paper Sitasi	Hasil Similaritas
1	(1) Jason Cong, (2) Yuzheng Ding	(1) 1 (2) 0
2	(1) Jason Helge Anderson, (2) Stephen Dean Brown	(1) 0 (2) 1
3	(1) Giovanni De Micheli	(1) 0

3.2.4. Ekstraksi Fitur Berbasis Konten Publikasi

Pada modul ini akan dilakukan proses untuk ekstraksi fitur menggunakan similaritas antara paper dan referensinya. Pada tahap ini memiliki masukan data peneliti, abstrak paper, judul paper, dan hubungan referensi setiap paper peneliti. Proses detailnya terdapat pada Gambar 3.15.



Gambar 3.15. Tahap Ekstraksi Fitur Berbasis Konten Publikasi

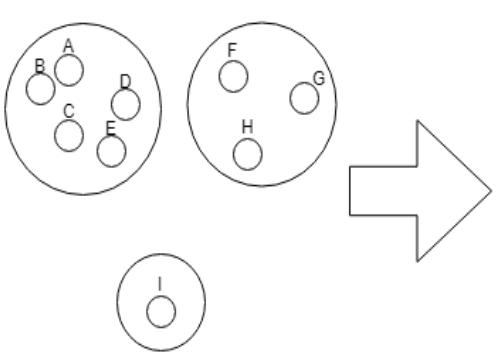
- *Clustering*

Tahapan ini bertujuan untuk membagi kumpulan dokumen artikel ilmiah ke beberapa klaster. Hasil dari klaster ini akan digunakan sebagai data inputan pada perhitungan Similaritas yang diolah terlebih dahulu sebelumnya. Hasil dari klaster ini diolah menjadi matrik jarak akan tetapi yang digunakan hanya yang ada pada klaster tersebut dan memenuhi *threshold* similaritas yang diberikan. Hal ini dilakukan agar data training yang digunakan pada *Siamese Neural Network* dapat memetakan kata – kata yang memiliki topik yang sama. Langkah-langkah yang dilakukan pada proses *Clustering* dengan pembentukan data latih terdapat pada Gambar 3.16

```
1. Untuk seluruh dataset artikel, dilakukan proses preprocessing
   terlebih dahulu
2. Data preprocessing, dilakukan proses klastering k-means
3. Untuk setiap klaster yang dihasilkan, dilakukan iterasi untuk
   mendapatkan data training
4. Pada satu klaster, dipilih data yang terdekat dari centroid dan
   terjauh dari centroidnya berdasarkan threshold
5. Setiap data yang didapatkan, dihitung cosine similarity
   terhadap seluruh data yang dipilih
6. Hasil similaritas kemudian disimpan dalam sebuah file, untuk
   digunakan pada proses selanjutnya
```

Gambar 3.16. *Pseudocode* proses klastering dan generate data latih SLSTM

Proses Klasterisasi yang digunakan adalah metode klasterisasi K-Means++ dengan *Cosine Similarity* dijelaskan pada subbab 2.2. Contoh proses perubahan hasil klaster menjadi dataset similaritas terdapat pada Gambar 3.17.



Paper 1	Paper 2	Similarity
A	B	0.8
A	C	0.64
A	D	0.55
A	E	0.49
B	C	0.72
....	....	....
G	H	0.59

Gambar 3.17. Tahap Perubahan Hasil *Clustering* Menjadi Dataset Similaritas

- Pelatihan Model Similaritas pada *Siamese Neural Network*

Tahapan ini bertujuan untuk melatih kumpulan data similaritas berdasarkan hasil klasterisasi pada tahapan sebelumnya. Hal ini dilakukan untuk menghasilkan nilai similaritas yang lebih baik. Seperti yang dijelaskan pada subbab 2.7, Siamese network memerlukan proses perhitungan similaritas hasil hidden tiap LSTM. Perhitungan nilai similaritas dilakukan pada proses pengeluaran nilai output. Input dari nilai similaritas didapatkan dari hidden units ke-n yang didapatkan pada proses pelatihan model LSTM. Pada kasus ini perhitungan similaritas yang digunakan adalah *Cosine Similarity*.

- Perhitungan Nilai Fitur Similaritas

Dari model similaritas yang didapatkan pada tahap sebelumnya, dalam hal ini model tersebut digunakan untuk menentukan similaritas 2 paper pada data uji. Untuk menghitung nilai similaritas, dilakukan dengan memasukkan 2 data teks ke dalam model LSTM yang sudah dibuat. Hasil similaritas didapatkan dari nilai *cosine similarity* antar 2 hidden dari data masukan. Setiap similarity pada setiap paper akan dijumlahkan dan dibagi dengan jumlah sitasinya. Kemudian untuk setiap peneliti akan dijumlah nilai rata-rata similaritas pada setiap paper dan dibagi dengan jumlah paper pada peneliti tersebut. Perhitungan fitur similaritas terdapat pada formula 3.3.

$$FSim = \frac{\sum_{i=1}^{n} \sum_{j=1}^{C_i} Sim(Pi,Pj)}{\sum_{i=1}^{n} C_i} \quad (3.3)$$

P_i = Publikasi Peneliti yang dianalisa

P_j = Publikasi yang dilakukan sitasi dari Publikasi P_i

n = Jumlah publikasi Peneliti yang dianalisa

C_i = Jumlah Sitasi yang ada pada Publikasi P_i

Tabel 3.4 Contoh data paper utama, paper sitasi beserta abstrak

Author yang diteliti: Jason Cong Paper: *Performance-driven mapping for CPLD architectures* Abstrak: *In this paper we present a performance-driven mapping algorithm, PLAmap, for CPLD architectures which consist of a large number of PLA-style logic cells. The primary goal of our mapping algorithm is to minimize the depth of the mapped circuit. ...*		
No.	Judul Paper Sitasi	Abstrak
1	*Combinational logic synthesis for LUT based field programmable gate arrays*	*The increasing popularity of the field programmable gate-array (FPGA) technology has generated a great deal of interest in the algorithmic study and tool development for FPGA-specific design automation problems. The most widely used FPGAs are LUT based FPGAs, in which the basic logic element is a K-input one-output lookup-table (LUT) that can implement any Boolean function of up to K variables. ...*
2	*Technology mapping for large complex PLDs*	*In this paper we present a new technology mapping algorithm for use with complex PLDs (CPLDs), which consists of a large number of PLA-style logic blocks. Although the traditional synthesis approach for such devices uses two-level minimization, the complexity of recently-produced CPLDs has resulted in a trend toward multi-level synthesis. ...*

No.	Judul Paper Sitasi	Abstrak
3	*Synthesis and Optimization of Digital Circuits*	*Synthesis and Optimization of Digital Circuits offers a modern, up-to-date look at computer-aided design (CAD) of very large-scale integration (VLSI) circuits. In particular, this book covers techniques for synthesis and optimization of digital circuits at the architectural and logic levels. …*

Pada Tabel 3.4, terdapat contoh paper dan abstrak dari paper utama dan sitasiya. Untuk mendapatkan nilai similaritas dari 2 paper, penelitian ini memerlukan model dari Siamese LSTM. Hasil dari similaritas 2 paper ini yang akan dipakai pada fitur yang diusulkan. Sebagai contoh pada Tabel 3.4, paper dan abstrak utama dibandingkan dengan paper dan abstrak sitasi pertama hingga ke 3. Hasil dari similaritas menggunakan Siamese LSTM terdapat pada Tabel 3.5.

Ketika seluruh hasil dari paper dan sitasinya didapatkan, proses yang dilakukan selanjutnya ialah merata-rata seluruh nilai similaritas untuk seluruh sitasi dan seluruh paper. Pada contoh sebelumnya terdapat 1 paper utama dan 3 paper sitasi, maka nila fitur yang dihasilkan berdasarkan konten similaritas adalah 0.80267 yang merupakan rata2 dari hasil yang ada pada Tabel 3.5

Tabel 3.5 Hasil similaritas paper utama dengan sitasinya

No. Paper	Judul Paper Sitasi	Hasil Similaritas
1	*Combinational logic synthesis for LUT based field programmable gate arrays*	0.730343
2	*Technology mapping for large complex PLDs*	0.88968
3	*Synthesis and Optimization of Digital Circuits*	0.787982

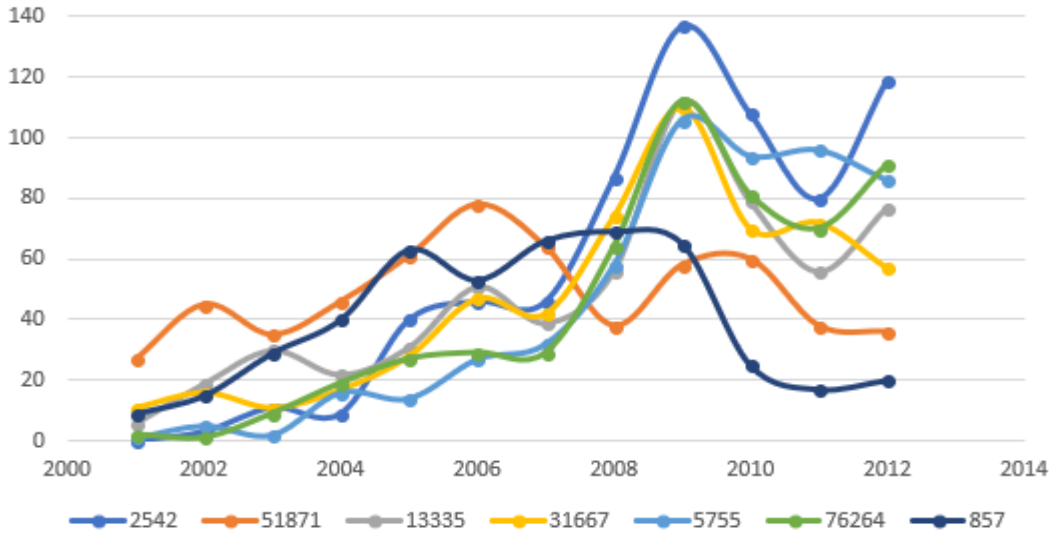
3.2.5. Klasifikasi

Pada modul ini akan dilakukan proses untuk melatih fitur yang diekstrak sebelumnya dan menentukan kualitas *citaion* peneliti. Input dari modul ini adalah fitur yang sudah dikombinasikan menjadi 1 matriks. Fitur yang dihasilkan adalah fitur dari *self-citation*, *Conflict of Interest* berdasarkan *Research Interest*, dan *Conflict of Interest* berdasarkan konten (Abstrak dan Judul). Fitur tersebut dipisahkan dengan periode yang digunakan. Contoh hasil data pada penggunaan periode 4 tahun terdapat pada Gambar 3.18. Pemisahan fitur berdasarkan periode ini bertujuan untuk mengetahui pola author dan sitasi anomali yang dilakukan oleh author tersebut. Pemisahan fitur ini berpengaruh

pada hasil pelatihan model klasifikasi dan pengujian data uji. Hal ini dikarenakan adanya pola keaktifan yang berbeda pada tiap periode. Hasil yang didapatkan pada kurun waktu 2 tahun bisa jadi berbeda dengan periode 4 tahun, begitu juga seterusnya. Pada Gambar 3.19, didapatkan bahwa pola publikasi setiap author berbeda-beda yang menghasilkan data dari fitur yang didapatkan pun berbeda yang nantinya mengubah data latih dan data uji yang digunakan pada proses klasifikasi.

Author	Self-citation (2001-2004)	Self-citation (2005-2008)	Self-citation (2009-2012)	Citation Conflict Interest (2001-2004)	Citation Conflict Interest (2005-2008)	Citation Conflict Interest (2009-2012)	Citation Conflict Content (2001-2004)	Citation Conflict Content (2005-2008)	Citation Conflict Content (2009-2012)	Kelas
A	0.5	0.2	0.4	0.42	0.2	0.1	0.7	0.4	0.6	1
B	0.6	0.7	0.6	0.5	0.47	0.4	0.3	0.1	0.05	0
C	0.3	0.17	0.24	0.35	0.24	0.19	0.6	0.5	0.5	1
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.

Gambar 3.18. Contoh Hasil Fitur yang dihasilkan



Gambar 3.19. Evolusi jumlah publikasi untuk setiap periode waktu

Ada 3 metode klasifikasi yang digunakan yaitu K-Nearest Neighbor, Decision Tree, dan Random Forest. Masing-masing akan dilakukan proses klasifikasi dan dilakukan perbandingan terhadap hasil yang didapatkan. Hasil klasifikasi pada tahap ini adalah berupa 2 kelas yaitu kelas positif (baik), dan negatif (kurang baik).

Proses pelatihan pada tiap metode klasifikasi memiliki cara yang berbeda-beda. *K-Nearest Neighbor* dapat dilakukan tanpa melakukan proses pelatihan atau sedikit pelatihan (*lazy learning*). *K-Nearest Neighbor* hanya mencari data input baru yang kemudian memasukkannya ke kelas yang terdekat sejumlah k. *Decision Tree* melakukan

pelatihan dengan cara membagi data latih menjadi sebuah *rule*, yang tiap rule diubah menjadi sebuah *tree*. Random Forest melakukan pelatihan dengan cara membuat beberapa pohon *rule*, yang nantinya pohon-pohon rule tersebut akan dijadikan model untuk mendapatkan hasil kelas dengan cara *voting*.

3.3. Pengujian

3.3.1. Dataset

Data artikel ilmiah yang digunakan bersumber dari dataset AMiner [23]. Dataset AMiner terdiri dari berbagai macam artikel ilmiah seperti *proceedings, journal papers*, dan tesis yang berkaitan dengan ilmu komputer. Sebelumnya dataset akan dilakukan proses pembersihan data. Data yang diambil hanyalah data yang memiliki abstrak, judul, dan referensi paper. Dataset artikel ilmiah tersusun atas judul, abstrak, author, dan referensinya. Data tabel yang digunakan terdapat pada Gambar 3.20. Spesifikasi tabel database *A-Miner* yang digunakan.

Dataset *AMiner* memiliki ±2.000.000 artikel ilmiah. Akan tetapi pada penelitian ini data yang dibutuhkan adalah abstrak dan judulnya pada proses pemrosesan teks. Dari keseluruhan dataset, tidak semua data memiliki abstrak dan atau *list author*. Maka dari itu dataset dilakukan pembersihan untuk data yang tidak memiliki abstrak, dan *list author*. Pada akhirnya dataset yang digunakan pada penelitian ini adalah ±1.600.000 artikel ilmiah.

3.3.2. Skenario Pengujian

Performa metode usulan dievaluasi dalam 2 skenario pengujian. Skenario pertama, mengevaluasi keefektifan tahap perbaikan klaster pada metode usulan. Skenario kedua, performa metode usulan dibandingkan dengan metode lain.

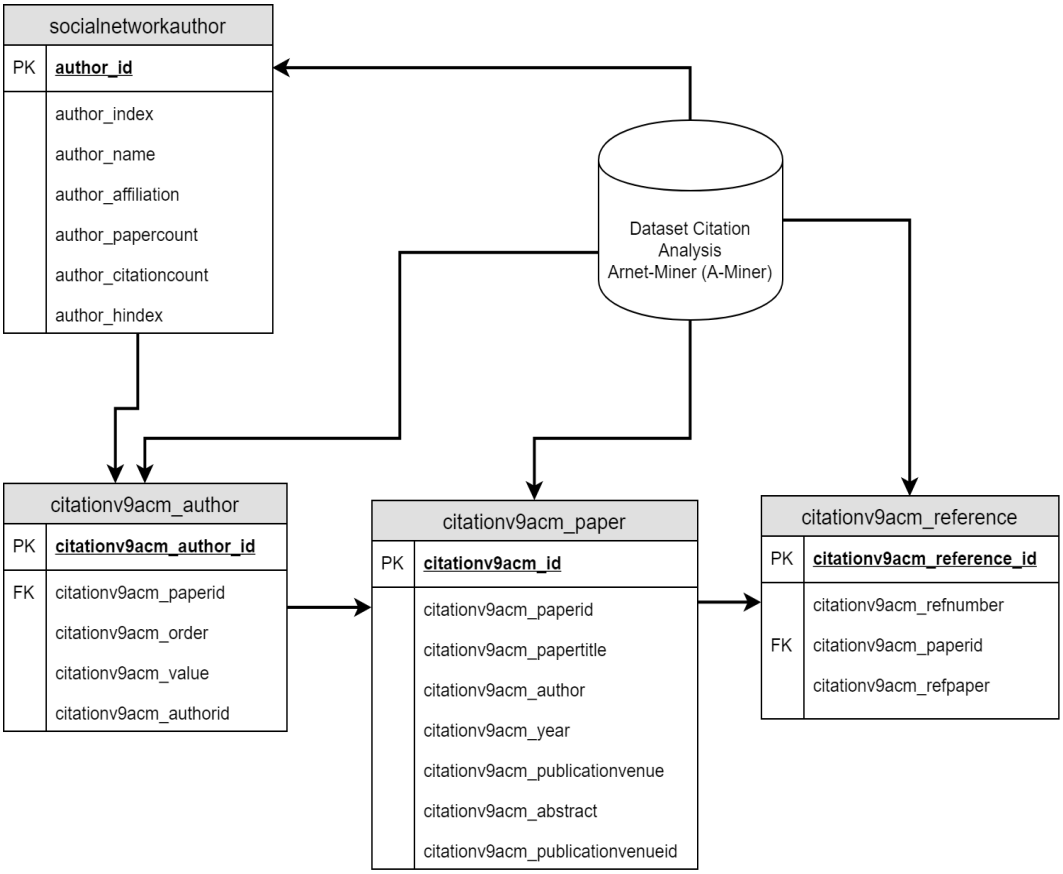
a. Pengujian Hasil Data Training pada Siamese Neural Network

Skenario ini bertujuan untuk mengetahui keefektifan data training pada Siamese neural network pada ekstraksi fitur usulan. Kualitas klaster dokumen akan dibandingkan berdasarkan jumlah klaster yang diinginkan. Evaluasi klaster dokumen akan menggunakan pengukuran *Silhouette Index*.

b. Pengujian Perbandingan Performa Fitur yang diusulkan.

Skenario ini berfungsi untuk mengetahui performa fitur yang diusulkan dibandingkan dengan fitur pada penelitian lain untuk jurnal. Fitur yang dijadikan pembanding adalah

fitur *cosine similarity* dan *jaccard coefficient*. Evaluasi metode akan menggunakan pengukuran *Akurasi dari klasifikasi*



Gambar 3.20. Spesifikasi tabel database *A-Miner* yang digunakan

BAB IV
HASIL DAN PEMBAHASAN

Pada pembahasan ini diberikan pemaparan mengenai implementasi sistem serta pengujian dari sistem berdasarkan skenario yang telah dirancang pada Bab Tiga. Proses implementasi dilakukan berdasarkan tahapan yang telah diberikan pada pembahasan sebelumnya. Selanjutnya pengujian sistem dilakukan dengan beberapa kondisi yang disesuaikan dengan skenario pengujian. Dari hasil pengujian yang telah didapatkan, selanjutnya diberikan pembahasan dan analisa dari setiap pengujian yang dilakukan dengan tujuan untuk mendapatkan hasil dari penelitian, sehingga mendapatkan kesimpulan yang diberikan pada pembahasan selanjutnya.

4.1. Perangkat Implementasi

Dalam tahapan implementasi, penelitian tesis membutuhkan beberapa perangkat, diantaranya komputer dengan spesifikasi processor Intel(R) Core (TM) i5-7400 CPU @ 3.0GHz (4 CPU), memory (RAM) 8.00 GB (gigabyte), HD 1,024 GB. Perangkat lunak yang digunakan dalam pengujian tesis adalah sistem operasi yang digunakan windows 64-bit operating system, Python 3.6, Spyder 3.2.4, Microsoft Office 365 64-bit.

4.2. Implementasi Sistem

4.2.1 Deskripsi Data Uji

Dataset yang digunakan pada proses pengujian adalah data publikasi dan sitasi dari setiap author yang sudah dilabeli sebelumnya. Data publikasi dan sitasi yang digunakan adalah judul dan abstraksi. Dokumen yang dianalisa berbahasa inggris dan memiliki tahun publikasi dari 2001 hingga 2012. Data sebelumnya akan dibersihkan dari judul, abstrak dan author yang kosong yang awalnya dari 2.000.000 paper menjadi 1.600.000 paper. Contoh data paper yang digunakan terdapat pada Tabel 4.1.

Pada tahap analisis data, dilakukan pelabelan *ground truth* dari data yang memiliki jumlah paper sekitar 150 hingga 200. Dari 180 data yang ada, dilakukan proses pemilihan data berdasarkan standar deviasi terhadap setiap tahunnya. 80 Data terpilih karena memiliki standar deviasi diatas rata – rata dari keseluruhan data. 80 data peneliti tersebut memiliki 14832 paper, dan 431589 jumlah sitasi. 80 data tersebut dibagi menjadi 2 kelas yaitu kelas positif dan negatif. Komposisinya adalah 48 kelas positif dan 32 kelas negative. Komposisi kelas pada dataset terdapat Tabel 4.2.

Tabel 4.1 Contoh dataset yang digunakan

ID Paper	Judul Paper	Abstrak Paper	*Author*
45	On the Multiple Implementation of Abstract Data Types Within a Computation	A fundamental step in the software design process is the selection of a refinement (implementation) for a data abstraction. This step traditionally involves investigating the expected performance of a system under different refinements of an abstraction ...	J. R White
102	Smalltalk-80: the language and its implementation	From the Preface (See Front Matter for full Preface) Advances in the design and production of computer hardware have brought many more people into direct contact with computers. …	Adele Goldberg, David Robson
118	Algorithms for trie compaction	The trie data structure has many properties which make it especially attractive for representing large files of data. These properties include fast retrieval time, quick unsuccessful search determination, and …	M. Al-Suwaiyel, E Horowitz
134	Logical, internal, and physical reference behavior in CODASYL database systems	This work investigates one aspect of the performance of CODASYL database systems: the data reference behavior. We introduce a model of database traversals at three levels: the logical, internal, and physical levels. …	Wolfgang Effelsberg, Mary E. S. Loomis
135	A parallel pipelined relational query processor	This paper presents the design of a relational query processor. The query processor consists of only four processing PIPEs and a number of random-access memory modules. Each PIPE processes tuples of relations in a bit-serial, …	Won Kim, Daniel Gajski, David J. Kuck

Tabel 4. 2 Komposisi Kelas Dataset

Dataset	Jumlah Paper	Jumlah Citation	Jumlah Kelas 1 (Baik)	Jumlah Kelas 0 (Kurang)
79 Data	14832	431589	47	32

4.2.2 Ekstraksi Fitur Self-Citation

Pada tahap ekstraksi fitur *self-citation*, memiliki 2 tahapan yaitu tahap deteksi adanya self-citation, dan tahap perhitungan nilai rasionya. Tahap deteksi self-citation dilakukan dengan melihat apakah pada paper yang dia miliki terdapat sitasi yang menuju dirinya sendiri. Tahapan ini dilakukan terhadap setiap periode waktu yang digunakan. Kemudian tahap perhitungan nilai rasio dilakukan dengan menghitung jumlah self-citation dibagi dengan jumlah sitasi yang dilakukan pada periode tersebut. Contoh hasil fitur self-citation terdapat pada Tabel 4.3.

Tabel 4.3 Hasil Fitur Self-Citation

ID Author	*Author Name*	*Self Citation*	*Total Citation*	*Ratio Self Citation*
427225	Jason Cong	514	1724	0.298143852
3601	Kwan-Liu Ma	587	1836	0.319716776
75630	Lei Liu	27	131	0.20610687
378168	Tat-Seng Chua	336	1046	0.321223709
47357	Qi Zhang	58	228	0.254385965
104908	Gonzalo Navarro	745	1766	0.421857305
1297710	Massoud Pedram	396	983	0.402848423

4.2.3 Ekstraksi Fitur Conflict of Interest Berbasis *Research Interest*

Pada tahap ekstraksi fitur *conflict of interest* berbasis *research interest*, memiliki 2 tahapan, yaitu tahapan pertama adalah deteksi conflict of interest dari *research interest* menggunakan similaritas dari *research interest* peneliti dan *research interes*t sitasinya. Pada tahap ini, data masukan berawal dari data bidang penelitian author dengan sitasinya. Contoh data masukan pada tahap ini terdapat pada Tabel 4.4. Dari data masukan yang berupa data seluruh bidang penelitian, kemudian dilakukan pemisahan tiap bidang penelitian sehingga pada saat melakukan perhitungan similartas bidang penelitan dapat dilakukan setiap 1 bidang. Jika salah satu bidang dari *author* berhubungan dengan salah

satu bidang dari *author* sitasinya, maka nilai similaritas adalah 1. Untuk menentukan adanya hubungan antar bidang, nilai similaritas yang dihasilkan akan dicocokkan dengan *threshold*. Jika melebihi *threshold*, maka kedua bidang tersebut akan dianggap sama. Hasil proses similaritas terdapat pada Tabel 4.5.

Tabel 4.4 Contoh data masukan deteksi *conflict of interest* dari *research interest*

ID Paper 1	*ID Paper 2*	*Interest Author 1*	*Interest Author 2*
320841	294826	*EDA community; potential impact; software patent*	*data mining;macroeconomic analysis;Web Services;important factor;data interface;data update;macroeconomic data;*
320843	109706	*EDA community;potential impact;software patent*	*Data Collection Project;C programming;LinuxThe author;Symphony Network Cards;end to end process;serial deviPCI*
329314	594630	*EDA community;potential impact;software patent*	*dedicated channelsHierarchical organization;hierarchical use;packet switching communication system*
737544	252836	*EDA community;potential impact;software patent*	*bner base;exponential space computation*
855736	1837915	*EDA community;potential impact;software patent*	*software patent , Desktop computer*

Tabel 4.5 Hasil nilai hubungan antar paper berbasis bidang penelitian

ID Paper 1	ID Paper 2	Hasil Similaritas
320841	294826	1
320843	109706	1
329314	594630	1

ID Paper 1	ID Paper 2	Hasil Similaritas
737544	252836	0
855736	1837915	0

Tahapan kedua yaitu perhitungan nilai rasio dari conflict of interest dengan menghitung jumlah konflik yang dideteksi dibagi dengan total sitasi pada setiap periodenya. Contoh hasil fitur *conflict of interest* berbasis *research interest* terdapat pada Tabel 4.6.

Tabel 4.6 Hasil Fitur *Conflict of Interest* Berbasis *Research Interest*

ID Author	*Author Name*	Rata-Rata Conflict
427225	Jason Cong	0.298143852
3601	Kwan-Liu Ma	0.319716776
75630	Lei Liu	0.20610687
378168	Tat-Seng Chua	0.321223709
47357	Qi Zhang	0.254385965
104908	Gonzalo Navarro	0.421857305
1297710	Massoud Pedram	0.402848423

4.2.4 Ekstraksi Fitur Conflict of Interest Berdasarkan Data Publikasi

Pada tahap ekstraksi fitur *conflict of interest* berbasis data publikasi, memiliki beberapa tahapan, yaitu tahapan pertama adalah proses pembuatan data latih untuk proses *deep learning*. Proses ini dilakukan dengan cara melakukan klasterisasi pada data artikel ilmiah. Bobot yang digunakan pada proses klasterisasi berasal dari bobot *Word2Vec*, yang sudah dilatih dari seluruh artikel. Bobot tersebut kemudian dilakukan proses pembuatan matriks untuk setiap artikelnya berdasarkan kata-kata yang ada dan telah dilakukan proses *preprocessing*. Metode klasterisasi yang digunakan adalah *K-Means++*. Hasil klaster yang didapatkan terdapat pada Tabel 4.7. Setelah dilakukan klasterisasi, kemudian dilakukan pembuatan data training dengan menggunakan label klaster yang didapatkan. Untuk setiap klaster, diambil data yang memiliki kedekatan dengan *centroid* dan yang paling jauh dari *centroid*. Hal ini dilakukan dengan cara memberi *threshold* pada jarak antar paper dengan *centroid*-nya untuk data yang jauh dan dekat. Untuk setiap data yang memenuhi *threshold*, dilakukan perhitungan *cosine similarity*. Hasil dari *similarity* akan dijadikan data latih untuk *deep learning*. Contoh hasil data latih terdapat pada Tabel 4.8

Tahapan selanjutnya adalah proses *deep learning* menggunakan *Siamese LSTM*. Pada proses ini, dilakukan pelatihan data dari data latih yang dihasilkan sebelumnya. Parameter yang digunakan pada *deep learning* ini terdapat pada Tabel 4.8. Setelah dilakukan pelatihan terhadap data latih, kemudian dilakukan proses penentuan similaritas berdasarkan model yang dihasilkan. Hasil dari similaritas ini, terdapat pada Tabel 4.9. Nilai similaritas yang dihasilkan, kemudian dilakukan perhitungan rata-rata terhadap seluruh author, yang kemudian dijadikan sebagai fitur pada penelitian ini. Hasil nilai rata-rata yang dihasilkan terdapat pada Tabel 4.10

Tabel 4.7 Hasil label klaster yang didapatkan dari klasterisasi

ID Paper	Label Klaster
320841	1
294826	1
320843	1
109706	1
329314	10
594630	8

Tabel 4.8 Parameter yang digunakan pada proses *Siamese LSTM*

Parameter	Nilai Parameter
Embedding_dim	100
Hidden Layer	50
Batch_size	256
epoch	5

Tabel 4.9 Hasil similaritas antar paper menggunakan *Siamese LSTM*

ID Paper 1	ID Paper 2	Hasil Similaritas
320841	294826	0.7822359
320843	109706	0.9168026
329314	594630	0.847425
737544	252836	0.851128
855736	1837915	0.868899

Tabel 4.10 Hasil perhitungan *conflict of interest* berdasarkan data publikasi

ID Author	*Author Name*	Rata-Rata Similaritas Sitasi
427225	Jason Cong	0.716702306
3601	Kwan-Liu Ma	0.70488597
75630	Lei Liu	0.730141455
378168	Tat-Seng Chua	0.765984567
47357	Qi Zhang	0.76488921
104908	Gonzalo Navarro	0.747264678

4.2.5 Klasifikasi

Proses ini akan mengelola input yang diperoleh dari fitur yang diekstrak sebelumnya. Data yang diperoleh akan dilakukan klasifikasi menggunakan SVM. Pada penelitian ini, data akan diklasifikasi apakah ia termasuk klaster tertentu atau tidak. Hal ini dikenal dengan nama *binary classification*. Pada kasus ini, setiap data akan diklasifikasikan menjadi 2 kelas yaitu (1) kelas positif (Baik), dan (2) kelas negatif (Kurang).

4.3. Hasil pengujian dan Analisis

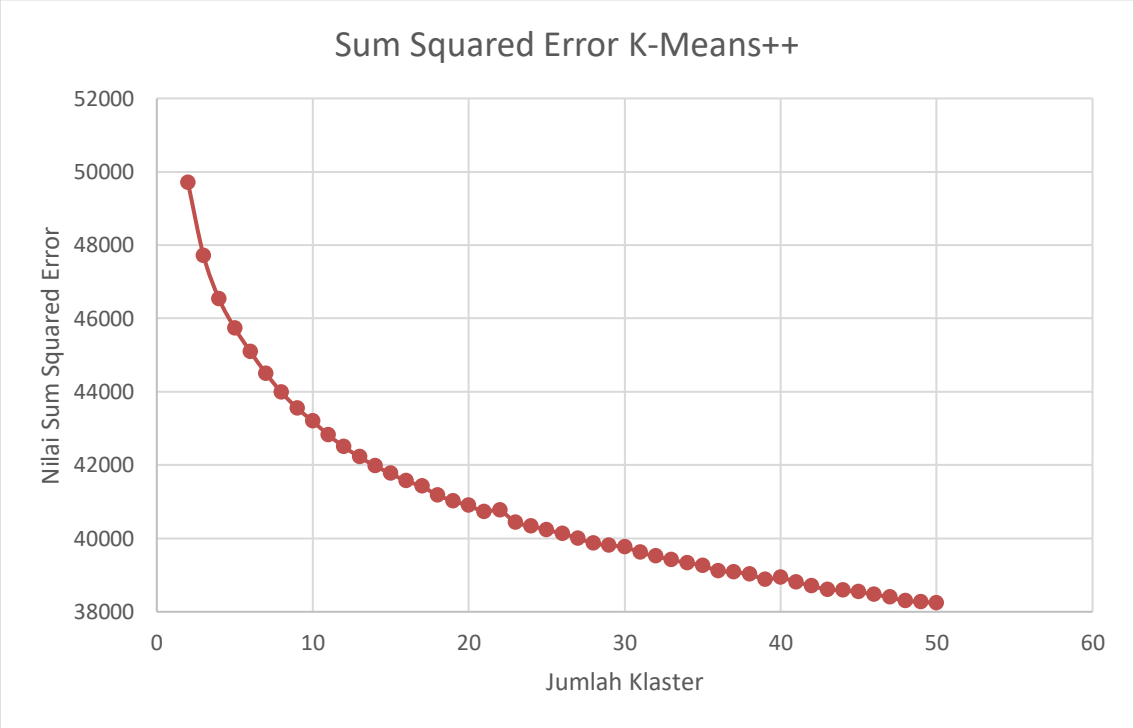
Performa metode usulan dievaluasi dalam 2 skenario pengujian. Skenario pertama, mengevaluasi keefektifan tahap perbaikan klaster pada metode usulan. Skenario kedua, performa metode usulan dibandingkan dengan metode lain. Adapun kombinasi scenario yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Penggunaan nilai K mulai K = 2 hingga K=50, untuk mengetahui performa klaster yang dihasilkan pada proses klastering.
2. Penggunaan threshold pada proses deteksi kesamaan antar bidang penelitian. Threshold mulai dari 0,3 hingga 0,5 dari proses similaritas terhadap bidang penelitian untuk berbagai macam metode klasifikasi.
3. Penggunaan Salah satu fitur yang diusulkan pada penelitian ini yaitu Siamese LSTM yang dibandingkan dengan *cosine similarity*, dan *jaccard coefficient*.

4.3.1 Pengujian Nilai K pada Klaster untuk Data Training Deep Learning

Skenario ini bertujuan untuk mengetahui keefektifan data training pada Siamese neural network pada ekstraksi fitur usulan. Kualitas klaster dokumen akan dibandingkan berdasarkan jumlah klaster yang diinginkan. Evaluasi klaster dokumen akan menggunakan pengukuran *Silhouette Index* dan *Sum Squared Error (*SSE*)*

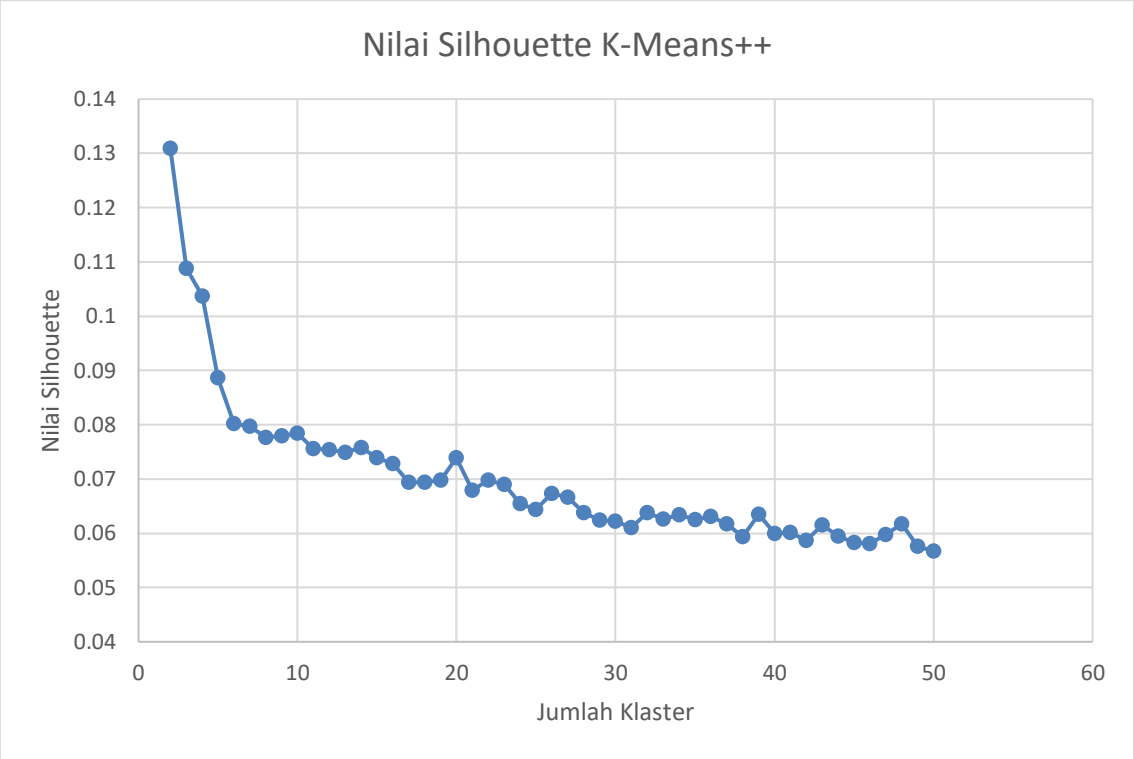
Pada Gambar 4.1, didapatkan bahwa nilai SSE mengalami penuruna setiap nilai-k bertambah. Hal ini disebabakan karena semakin kecil nilai k yang digunakan, maka semakin besar pula cakupan hasil klasternya. Untuk menentukan nilai k dengan menggunakan SSE, dapat dilakukan menggunakan *elbow's method*. Metode ini dilakukan dengan cara mencari titik siku pada grafik SSE. Titik siku didapatkan dari nilai k yang nilai error selanjutnya tidak menambah jumlah error yang besar pada hasil akhirnya (landai). Dari grafik diatas, dapat ditentukan bahwa nilai k yang memiliki nilai error setelahnya landai, terdapat pada nilai k antara 8 hingga 12.



Gambar 4.1. Grafik nilai *Sum Squared Error* pada tiap k – klaster

Untuk mendapatkan hasil nilai K terbaik, maka dilakukan analisis menggunakan nilai siluetnya. Dari nilai k 8 hingga 12 yang didapatkan dari analisis sebelumnya, dilihat nilai siluetnya. Dari Gambar 4.2, didapatkan bahwa dari nilai k 8 hingga 12, nilai k yang

memiliki nilai siluet tertinggi adalah k = 10. Maka dari itu pada penelitian ini, pada proses klasterisasi dokumen digunakan nilai k = 10 pada metode k-means++.



Gambar 4.2. Grafik nilai *Silhouette* pada tiap k – klaster

4.3.2 Pengujian Analisa Penggunaan Fitur dan Perbandingan dengan Fitur Terhadap proses klasifikasi

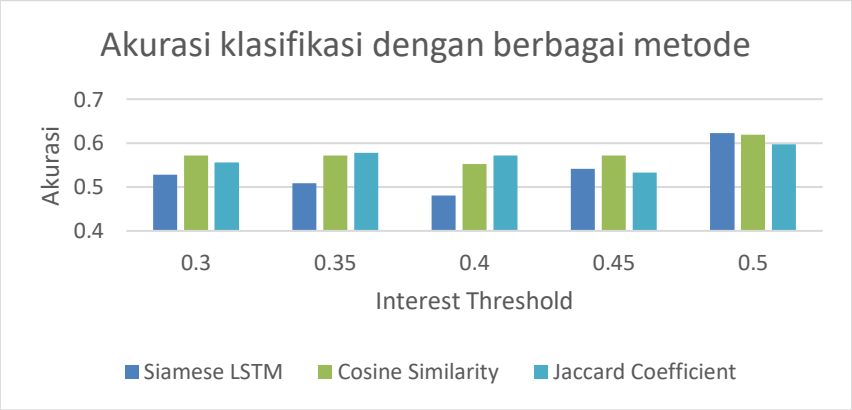
Skenario ini berfungsi untuk mengetahui performa fitur yang diusulkan dibandingkan dengan fitur pada penelitian lain untuk jurnal. Fitur yang dijadikan pembanding yaitu *cosine similarity* dan *jaccard coefficient*. Evaluasi metode akan menggunakan pengukuran nilai Akurasi dari klasifikasi. Pengujian dilakukan dengan melakukan klasifikasi terhadap 3 *classifier*

Pada Tabel 4.11. didapatkan bahwa, nilai akurasi terbesar ada pada *Deep Learning* dan *Cosine Similarity* dengan nilai akurasi 0.6667. Klasifikasi yang mendapatkan nilai akurasi terbesar adalah Klasifikasi *Random Forest*. Meskipun mempunyai nilai akurasi terbesar yang sama, akan tetapi fitur yang dibuat dengan deep learning mampu mendapatkan rata-rata akurasi yang lebih tinggi dari pada metode pembandingnya. Hal ini dipengaruhi oleh adanya fitur yang lebih merepresentasikan terhadap kelasnya dibandingkan dengan fitur lainnya.

Tabel 4.11 Nilai akurasi dari pengujian fitur usulan

Similarity Content	*Threshold Interest*	Klasifikasi KNN	Klasifikasi Decision Tree	Klasifikasi Random Forest	Rata-Rata Akurasi
Deep Learning dan *Cosine Similarity*	0.3	*0.525*	*0.525*	*0.533333*	*0.527778*
	0.35	*0.533333*	*0.441667*	*0.55*	*0.508333*
	0.4	*0.525*	*0.466667*	*0.45*	*0.480556*
	0.45	*0.541667*	*0.491667*	*0.591667*	*0.541667*
	0.5	*0.641667*	*0.558333*	*0.666667*	*0.622222*
Cosine Similarity	0.3	*0.558333*	*0.591667*	*0.566667*	*0.572222*
	0.35	*0.558333*	*0.591667*	*0.566667*	*0.572222*
	0.4	*0.516667*	*0.541667*	*0.6*	*0.552778*
	0.45	*0.525*	*0.575*	*0.616667*	*0.572222*
	0.5	*0.575*	*0.616667*	*0.666667*	*0.619444*
Jaccard Coefficient	0.3	*0.55*	*0.491667*	*0.625*	*0.555556*
	0.35	*0.616667*	*0.575*	*0.541667*	*0.577778*
	0.4	*0.541667*	*0.533333*	*0.641667*	*0.572222*
	0.45	*0.475*	*0.5*	*0.625*	*0.533333*
	0.5	*0.591667*	*0.55*	*0.65*	*0.597222*

Pada proses similaritas terhadap bidang penelitian, memerlukan adanya *threshold* untuk mengetahui apakah antar bidang minat memiliki keterkaitan. Dari 5 threshold yang diuji coba, didapatkan bahwa threshold 0.5 menjadi yang terbaik karena pada Gambar 4.3, ditunjukkan bahwa ketiga metode klasifikasi threshold 0.5 memiliki nilai akurasi tertinggi. Hal ini disebabkan oleh karena fitur yang didapat semakin tersaring dengan baik. Dengan begitu hasil akurasi yang diberikan semakin baik. Akan tetapi terlalu tinggi nilai threshold akan juga mempengaruhi tingkat penyaringan nilai fiturnya.



Gambar 4.3. Grafik akurasi klasifikasi dengan berbagai metode

4.3.3 Pengujian Analisa Penggunaan dan Perbandingan proses Similaritas pada artikel ilmiah

Skenario ini berfungsi untuk mengetahui performa fitur yang diusulkan dibandingkan dengan fitur pada penelitian lain untuk jurnal. Fitur yang dijadikan pembanding yaitu *cosine similarity* dan *jaccard coefficient*. Evaaluasi dilakukan dengan cara membandingkan secara manual, hasil dari tiap fitur dan nilai similaritasnya dan kemudian dianalisa terhadap semantik yang didapatkan setelah membaca dan melihat paper yang dilakukan proses similaritas.

Tabel 4.12 Hasil Salah satu klasifikasi dengan *groundtruth*-nya.

ID Author	Author Name	Deep Learning	Cosine Similarity	Jaccard Coefficient	Ground Truth
587770	Rajkumar Buyya	0	0	1	1
10858	Hao Chen	1	1	1	1
427225	Jason Cong	0	1	1	0

Pada Tabel 4.13 diketahui dari hasil klasifikasi menggunakan *Random Forest* dan *threshold* sebesar 0.5, didapatkan bahwa terdapat beberapa kesalahan pada metode pembanding dengan metode usulan. Salah satunya ada pada ID Author 587770 terdapat kesalahan klasifikasi terhadap metode usulan dan metode *cosine similarity*, dan ID Author 427225 terdapat kesalahan klasifikasi terhadap metode pembanding. Untuk salah satu contoh hasil benar untuk ketiga metode. Untuk mengetahui hubungan antara hasil similaritas dengan hasil klasifikasi akan dilakukan pembedahan terhadap hasil similaritasnya.

Tabel 4.13 Hasil Similaritas antara 2 paper berdasarkan 3 metode similaritas

ID Author	ID Paper 1	ID Paper2	Siamese LSTM	Cosine Similarity	Jaccard Coefficient
10858	782606	725313	0.442904	0.061718	0.070513
	1393882	499896	0.912291	0.555375	0.086207
427225	769264	737551	0.996504	0.798263	0.36
	342936	239627	0.177248	0.01593	0.063492
587770	782128	590325	0.943122	0.145124	0.106383
	435616	391643	0.356697	0.013326	0.011173

Pada kasus pertama, saat seluruh fitur usulan dan pembanding memiliki hasil klasifikasi yang benar. Pada Tabel 4.14 terdapat salah satu hasil similaritas terhadap 3 *author* sebelumnya. Untuk author 10858 baris pertama, Kedua paper memiliki background yang sama yakni rekayasa perangkat lunak. Akan tetapi banyak kata-kata yang digunakan tidak berhubungan atau memiliki kesamaan antar paper tersebut. Hal ini menyebabkan kesalahan pada Cosine Similarity yang menilai bahwa kedua paper tersebut tidak berhubungan dengan nilai 0.062. Berbeda dengan Cosine Similarity, Siamese LSTM mampu menganalisis hasil kesamaan walaupun hanya sebatas background dengan nilai similaritas 0.443. Untuk author 10858 baris kedua, Kedua paper memiliki topik yang sama yakni "*Intrusion Detection*". Setelah dibaca secara manual, dapat disimpulkan bahwa paper 1 menggunakan paper 2 sebagai kombinasi dengan metode lain. Pada Siamese LSTM didapatkan bahwa nilai similaritas tinggi, begitu juga dengan *cosine similarity*. Jaccard tidak mampu dalam mendeteksi kata-kata yang tidak sama persis antara 2 paper tersebut karena paper 2 memiliki banyak sekali kata dibandingkan dengan paper 1.

Pada kasus kedua, saat fitur usulan dan pembanding, *cosine similarity* mengalami kesalahan klasifikasi. Untuk author 587770 baris pertama, Kedua paper memiliki keterkaitan terhadap proses simulasi pada sistem terdistribusi dan memiliki tujuan yang sama. Meskipun memiliki keterkaitan, *cosine similarity* dan *jaccard coefficient* tidak mampu dalam mengenali keterkaitan tersebut dengan nilai similaritas sebesar 0.145 dan 0.106. Untuk Siamese LSTM mendapatkan nilai similaritas yang cukup tinggi dan dapat mengenali adanya keterkaitan tersebut. Berbeda dengan kasus sebelumnya pada author 587770 baris kedua, ketiga metode usulan mendapatkan nilai similaritas yang relative kecil. Topik yang dikerjakan pada paper 1 dan paper 2 memiliki kesamaan yakni pada peer-to-peer dan komputasi grid. Salah satu masalah pada kasus ini adalah sedikitnya data abstrak yang dimiliki paper 1 yang menyebabkan kurangnya penjelasan terhadap paper yang dilakukan perbandingan.

Pada kasus terakhir, saat seluruh fitur usulan hasil klasifikasi yang benar dan pembanding memiliki hasil klasifikasi yang salah. Untuk author 427225 baris pertama, Kedua paper memiliki keterkaitan yang cukup erat karena sama-sama membahas *low-optimal circuit pada FPGA Architecture*. Seluruh metode similaritas mampu mendapatkan hasil yang cukup tinggi sesuai dengan keterkaitan kedua paper tersebut.

Untuk kasus kedua pada author 427225, didapatkan bahwa seluruh metode similaritas mampu mendapatkan nilai similaritas yang rendah karena kedua paper tersebut tidak berkaitan dalam abstraknya dan paper rujukannya memiliki banyak kesalahan pengetikan yang menyebabkan nilai similaritas semakin rendah.

Dari kasus-kasus diatas, didapatkan bahwa hasil Siamese LSTM mampu mengungguli *cosine similarity* pada beberapa teks yang memiliki keterkaitan terhadap topik akan tetapi tidak memiliki kesamaan terhadap kata-kata yang digunakan.

[Halaman ini sengaja dikosongkan]

BAB V
KESIMPULAN DAN SARAN

Pada bab terakhir ini, ditarik beberapa kesimpulan yang didapat dari hasil penelitian dan saran-saran yang dapat digunakan sebagai bahan pertimbangan untuk pengembangan atau riset selanjutnya

5.1 Kesimpulan

Berdasarkan metode yang telah diimplementasikan dan hasil uji coba yang diperoleh, maka dapat ditarik beberapa kesimpulan:

1. Nilai K terbaik yang digunakan pada proses klasterisasi dokumen menggunakan metode K-Means++ pada penelitian ini adalah K = 10
2. Penggunaan fitur similaritas deep learning, dapat meningkatkan sedikit rata-rata akurasi dibandingkan dengan fitur lainnya.
3. Hasil klasifikasi terbaik didapatkan dengan threshold fitur bidang penelitian 0,5 dan fitur similaritas paper dengan SLSTM dengan akurasi 0.667 pada klasifikasi *Random Forest* dan rata-rata klasifier 0,62.
4. Similaritas paper dengan SLSTM mampu mengungguli *cosine similarity* dan *jaccard coefficient* dalam mengetahui kalimat secara semantiknya.

5.2 Saran

Beberapa saran atas pengerjaan tesis ini guna pengembangangan lebih lanjut diantaranya adalah:

1. Penentuan data latih pada penelitian ini, dapat menggunakan proses pengecekan terhadap data yang seharusnya / data real.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

[1] MIT Academic Integrity., "Citing Your Source." [Online]. Available: https://integrity.mit.edu/handbook/citing-your-sources/avoiding-plagiarism-cite-your-source.

[2] Association of Legal Writing Directors & Darby Dickerson, *ALWD Citation Manual: A Professional System of Citation*, 4th ed. 2010.

[3] Citation Machine, "Cite a Book," 2000. [Online]. Available: http://www.citationmachine.net/apa/cite-a-book.

[4] Khaled Moustafa, "Aberration of the Citation," *Account. Res.*, vol. 23, no. 4, pp. 230–244, 2016.

[5] T. Yu, G. Yu, and M. Y. Wang, "Classification method for detecting coercive self-citation in journals," *J. Informetr.*, vol. 8, no. 1, pp. 123–135, 2014.

[6] F. Xia, W. Wang, T. M. Bekele, and H. Liu, "Big Scholarly Data: A Survey," *IEEE Trans. Big Data*, vol. 3, no. 1, pp. 18–35, 2017.

[7] Institute of Medicine, *Conflict of interest in medical research, education and practice*. 2009.

[8] P. Wang, "Semantic Expansion using Word Embedding Clustering and Convolutional Neural Network for Improving Short Text Classification," *J. Neurocomputing*, vol. 174, pp. 806–814.

[9] Y. Bengio, "A Neural Probabilistic Language Model," *J. Mach. Learn. Res.*, pp. 1137–1155, 2003.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Work. ICLR*, 2013.

[11] H. Mikolov, T., Sutskever, "Distributed Representations of Words and Phrases and their Compositionality," *Adv. Neural Inf. Process. Syst. 26*, pp. 1–9, 2013.

[12] R. Collobert, "Natural Language Processing (Almost) from Scratch," *J. Mach. Learn. Res.*, pp. 2493–2537, 2011.

[13] S. Hochreiter and J. Schmidhuber, "Long Short Term Memory," *Neural Comput.*, pp. 1735–1780, 1997.

[14] J. Mueller, "Siamese Recurrent Architectures for Learning Sentence Similarity," *Proc. 30th Conf. Artif. Intell. (AAAI 2016)*, no. 2012, pp. 2786–2792, 2016.

[15] J. E. Hirsch, "An index to quantify an individual's scientific research output," vol.

102, no. 46, pp. 16569–16572, 2005.

[16] X. Bai, I. Lee, Z. Ning, A. Tolba, and F. Xia, “The Role of Positive and Negative Citations in Scientific Evaluation,” *IEEE Access*, vol. 5, no. usually 1, pp. 17607–17617, 2017.

[17] P. O. Seglen, “The skewness of science,” *Sci. J. Am. Soc. Inf.*, vol. 43, no. 9, pp. 628–638, 1992.

[18] H. G. and R. Srivastava, “K-means Based Document Clustering with Automatic ‘ K ’ Selection and Cluster Refinement,” *Int. J. Comput. Sci. Mob. Appl*, 2014.

[19] V. K. D. N. Y. Saiyad, H. B. Prajapati, “A Survey of Document Clustering using Semantic Approach,” *Int. Conf. Electr. Electron. Optim. Tech*, 2016.

[20] C. D. M. Jeffrey Pennington, Richard Socher, “GloVe: Global Vectors for Word Representation,” *Empir. Methods Nat. Lang. Process.*, 2014.

[21] O. Levy and Y. Goldberg, “Dependency-Based Word Embeddings,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014, vol. 52, pp. 302–308.

[22] A. S. M. Idiart and A. Villavicencio, “Matrix Factorization using Window Sampling and Negative Sampling for Improved Word Representations,” *Proc. 54th Annu. Meet. Assoc. Comput. Linguist.*, 2016.

[23] J. Tang and J. Zhang, “ArnetMiner : Extraction and Mining of Academic Social Networks,” in *KDD ’08 Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 990–998.

LAMPIRAN

Tabel 6.1 data yang digunakan pada perbandingan hasil similaritas (Bab 4.3.3)

ID Author	ID Paper 1	Judul Paper1	Abstrak Paper 1	ID Paper 2	Judul Paper 2	Abstrak Paper 2
10858	782606	Using build-integrated static checking to preserve correctness invariants	A key missing link in the creation of secure and robust systems is finding a ost effective way to demonstrate and preserve correspondence between a software design and its implementation. This paper explores the use of software model checking techniques to validate selected design invariants in the EROS operating system kernel. Several global consistency policies in the EROS kernel can be expressed as finite state automata. Using the MOPS static hecker, we have been able to validate the EROS kernel implementation against these automata. In the process, we have confirmed the practical utility of the basic verification technique, identified a number of desirable enhancements in MOPS, and located bugs in the EROS implementation. A key contribution of this paper is establishing that it is practical to integrate software model checking into normal development lifestyle. Model hecking is efficient enough that it does not add noticeably to our build times. This allows us to view it as a tool for error prevention rather than detection. Our work with EROS and MOPS suggests that domain specific application of software model hecking is a practical and powerful technique for software assurance and maintenance.	725313	MECA: an extensible, expressive system and language for statically checking security properties	This paper describes a system and annotation language, MECA, for checking security rules. MECA is expressive and designed for checking real systems. It provides a variety of practical constructs to effectively annotate large bodies of code. For example, it allows programmers to write programmatic annotators that automatically annotate large bodies of source code. As another example, it lets programmers use general predicates to determine if an annotation is applied; we have used this ability to easily handle kernel backdoors and other false-positive inducing constructs. Once code is annotated, MECA propagates annotations aggressively, allowing a single manual annotation to derive many additional annotations (e.g., over one hundred in our experiments) freeing programmers from the heavy manual effort required by most past systems. MECA is effective. Our most thorough case study was a user-pointer checker that used 75 annotations to check thousands of declarations in millions of lines of code in the Linux system. It found over forty errors, many of which were serious, while only having eight false positives.
10858	1393882	A risk-sensitive intrusion detection model	Intrusion detection systems (IDSs) must meet the security goals while minimizing risks of wrong detections. In this paper, we study the issue of building a risk-sensitive intrusion detection model. To determinate whether a	499896	A New Intrusion Detection Method based	There have been two well-known models for intrusion detection. They are called Anomaly Intrusion Detection (AID) model and Misuse Intrusion Detection (MID) model. The former model analyzes user behavior and the statistics of a process in

ID Author	ID Paper 1	Judul Paper1	Abstrak Paper 1	ID Paper 2	Judul Paper 2	Abstrak Paper 2
			system calls sequence is normal or not, we consider not only the probability of this sequence belonging to normal sequences set or intrusion sequences set, but also the risk of a false detection. We define the risk model to formulate the expected risk of an intrusion detection decision, and present risk-sensitive machine learning techniques that can produce detection model to minimize the risks of false negatives and false positives. Meanwhile, this model is a hybrid model that combines misuse intrusion detection and anomaly intrusion detection. To achieve a satisfying performance, some techniques are applied to extend this model.		on Process Profiling	normal situation, and it checks whether the system is being used in a different manner. The latter model maintains datab ase of known intrusion technique and detects intrusion by comparing a behavior against the database. An intrusion detection method based on an AID model can detect a new intrusion method, however it needs to update the data describing users behavior and the statistics in normal usage. We call these information profiles. There are several problems in AID to be addressed. The profiles are tend to be large. Detecting intrusion needs a large amount of system resource, like CPU time and memory and disk space. An MID model requires less amount of system resource to detect intrusion. However it cannot detect new, unknown intrusion methods. Our method solves these problems by recording system calls from daemon processes and set fluid programs. We improved detection accuracy by adopting a DP matching scheme.
427225	769264	Delay optimal low-power circuit clustering for FPGAs with dual supply voltages	This paper presents a delay optimal FPGA clustering algorithm targeting low power. We assume that the configurable logic blocks of the FPGA can be programmed using either a high supply voltage (high-Vdd) or a low supply voltage (low-Vdd). We carry out the clustering procedure with the guarantee that the delay of the circuit under the general delay model is optimal, and in the meantime, logic blocks on the non-critical paths can be driven by low-Vdd to save power. We explore a set of dual-Vdd combinations to find the best ratio between low-Vdd and high-Vdd to achieve the largest power reduction. Experimental results	737551	Low-power technology mapping for FPGA architectures with dual supply voltages	In this paper we study the technology mapping problem of FPGA architectures with dual supply voltages (Vdds) for power optimization. This is done with the guarantee that the mapping depth of the circuit will not increase compared to the circuit with a single Vdd. We first design a single-Vdd mapping algorithm that achieves better power results than the latest published low-power mapping algorithms. We then show that our dual-Vdd mapping algorithm can further improve power savings by up to 11.6% over the single-Vdd mapper. In addition, we investigate the best low-Vdd/high-Vdd ratio for the largest power reduction among several dual-Vdd combinations. To our

ID Author	ID Paper 1	Judul Paper1	Abstrak Paper 1	ID Paper 2	Judul Paper 2	Abstrak Paper 2
			show that our clustering algorithm can achieve power savings by 20.3% on average compared to the clustering result for an FPGA with a single high-Vdd. To our knowledge, this is the first work on dual-Vdd clustering for FPGA architectures.			knowledge, this is the first work on dual-Vdd mapping for FPGA architectures.
427225	342936	SPFD-based global rewiring	This paper presents the theory and algorithm for SPFD-based global rewiring (SPFD-GR). SPFD-GR allows us to globally replace a target wire with some alternative wire possibly far away from the target. It successfully overcomes the limitations of the existing SPFD-based local rewiring algorithm (SPFD-LR), which can only replace a wire with another wire that has the same destination node. In order to perform SPFD-based global rewiring, we developed the theory and algorithm for solving a fundamental problem in SPFD-based rewiring: Given the in-pin functions of a node and the SPFD at the node's out-pin, is there a way to modify the node's internal function so that the SPFD at the node's out-pin can be satisfied? Combined with a state-of-the-art partitioning algorithm, SPFD-GR scales well to large circuits with good synthesis quality. Our SPFD-based rewiring algorithm is ideal for LUT-based FPGAs, where the node's internal function can be changed freely without any area or delay penalty. Extensive experimental results show that for LUT-based FPGAs, the rewiring ability of SPFD-GR (in terms of the number of wires that have alternative wires) is 1.45, and 3 times that of SPFD-LR and an ATPG-based rewiring algorithm (with a preliminary experimental flow),	239627	Post-layout logic restructuring for performance optimization	We propose a new methodology based on incrementallogic restructuring for post-layout performance improvement.The new post-layout logic restructuring techniqueallows to use accurate interconnection delays for performanceoptimization, while the incremental nature of thetechnique guarantees convergence between logic synthesisand layout. The technique can be further integrated withother post-layout optimization techniques such as gate sizingand buffer insertion. Experimental results show that thistechnique combined with post-layout buffer insertion canachieve an additional 15% improvement in performancecompared to designs produced by timing-driven logic optimizationfollowed by pre-layout buffer insertion followedby timing-driven physical design.

ID Author	ID Paper 1	Judul Paper1	Abstrak Paper 1	ID Paper 2	Judul Paper 2	Abstrak Paper 2
			respectively, while the run time is quite acceptable. When applied to the post-mapping area reduction for large LUT-based FPGAs under circuit depth restriction, SPFD-GR achieves 17.1% average area reduction, with no or little delay increase.			
587770	782128	A taxonomy of computer-based simulations and its mapping to parallel and distributed systems simulation tools	In recent years, extensive research has been conducted in the area of simulation to model large complex systems and understand their behavior, especially in parallel and distributed systems. At the same time, a variety of design principles and approaches for computer-based simulation have evolved. As a result, an increasing number of simulation tools have been designed and developed. Therefore, the aim of this paper is to develop a comprehensive taxonomy for design of computer-based simulations, and apply this taxonomy to categorize and analyze various simulation tools for parallel and distributed systems.	590325	Simgrid: A Toolkit for the Simulation of Application Scheduling	Advances in hardware and software technologies have made it possible to deploy parallel applications over increasingly large sets of distributed resources. Consequently, the study of scheduling algorithms for such applications has been an active area of research. Given the nature of most scheduling problems one must resort to simulation to effectively evaluate and compare their efficacy over a wide range of scenarios. It has thus become necessary to simulate those algorithms for increasingly complex distributed, dynamic, heterogeneous environments. In this paper we present Simgrid, a simulation toolkit for the study of scheduling algorithms for distributed application. This paper gives the main concepts and models behind Simgrid, describes its API and highlights current implementation issues. We also give some experimental results and describe work that builds on Simgrid's functionalities.
587770	435616	Weaving Computational Grids: How Analogous Are They with Electrical Grids?	Can computational grids make as great an impact in the 21st century as electrical grids did in the 20th? A comparison of the two technologies and their deployment histories could provide clues about how to make computational grids pervasive, dependable, and convenient.	391643	Peer-to-Peer: Harnessing the Power of Disruptive Technologies	From the Publisher: Upstart software projects Napster, Gnutella, and Freenet have dominated newspaper headlines, challenging traditional approaches to content distribution with their revolutionary use of peer-to-peer file-sharing technologies. Reporters try to sort out the ramifications of seemingly ungoverned peer-to-peer networks. Lawyers, business leaders, and social commentators debate the virtues and evils of these

ID Author	ID Paper 1	Judul Paper1	Abstrak Paper 1	ID Paper 2	Judul Paper 2	Abstrak Paper 2
						bold new distributed systems. But what's really behind such disruptive technologies -- the breakthrough innovations that have rocked the music and media worlds? And what lies ahead? In this book, key peer-to-peer pioneers take us beyond the headlines and hype and show how the technology is changing the way we communicate and exchange information. Those working to advance peer-to-peer as a technology, a business opportunity, and an investment offer their insights into how the technology has evolved and where it's going. They explore the problems they've faced, the solutions they've discovered, the lessons they've learned, and their goals for the future of computer networking. Until now, Internet communities have been limited by the flat interactive qualities of email and network newsgroups, where people can exchange recommendations and ideas but have great difficulty commenting on one another's postings, structuring information, performing searches, and creating summaries. Peer-to-peer challenges the traditional authority of the client/server model, allowing shared information to reside instead with producers and users. Peer-to-peer networks empower users to collaborate on producing and consuming information, adding to it, commenting on it, and building communities around it. This compilation represents the collected wisdom of today's peer-to-peer luminaries. It includes contributions from Gnutella's Gene Kan, Freenet's Brandon Wiley, Jabber's Jeremie Miller, and many others -- plus serious discussions of topics ranging from accountability and trust to security and performance. Fraught

ID Author	ID Paper 1	Judul Paper1	Abstrak Paper 1	ID Paper 2	Judul Paper 2	Abstrak Paper 2
						with questions and promise, peer-to-peer is sure to remain on the computer industry's center stage for years to come.

Tabel 6.2 Groundtruth yang dihasilkan pada penelitian ini

ID Author	Nama Author	Kelas *GroundTruth*
427225	Jason Cong	0
3601	Kwan-Liu Ma	1
75630	Lei Liu	1
378168	Tat-Seng Chua	1
47357	Qi Zhang	0
104908	Gonzalo Navarro	0
1297710	Massoud Pedram	1
7281	Laurence T. Yang	0
87601	Min Wu	0
182532	Jeng-Shyang Pan	1
400710	Shojiro Nishio	1
22496	Wei Jiang	1
29557	Hong Chen	1
35161	Qiang Wang	1
227169	Jian Pei	0
411898	Horst bunke	0
1377987	Rachid Guerraoui	1
5150	Hong Jiang	0
45158	Anil K. Jain	0
53742	Yao-Wen Chang	0
73523	Xin Yao	0
577970	Nikil Dutt	0
826792	Thomas A. Henzinger	1
1119936	Wouter Joosen	1
12337	Liang Zhan	1
57403	Luc Van Gool	0
61775	Hong Shen	1
71908	Dong Wang	0
155863	Viktor K. Prasanna	1
8023	Jun Xu	0
109428	Xian-Seng Hua	1
155536	Hanqing Lu	1
1008	Wei Xu	0
124292	Michael R. Lyu	1
126858	Jack Dongarra	0
604732	Gerhard Weikum	1
7940	Jing Yang	0
10858	Hao Chen	0
18082	Xiaoou Tang	1
720460	C Lee Giles	0

ID Author	Nama Author	Kelas *GroundTruth*
795174	Francky Catthoor	1
60542	Zheng Chen	1
89120	Ge Yu	1
132363	Jun Ma	0
832210	Nikos E. Mastorakis	1
1865	Hong Mei	1
1905	Zhen Liu	1
16507	Min Chen	0
58061	Bing Liu	1
12607	Hyunseung Choo	1
23426	Wei Huang	0
6252	Li Yang	1
189396	Qi Tian	0
1178362	John Mylopoulos	1
39	Zhaohui Wu	0
20834	Hong Zhang	0
28033	Ning Zhong	0
33275	Michel Raynal	1
8254	Jeffrey Xu Yu	1
30469	Yu Liu	0
235267	Elizabeth Chang	0
315502	Noga Alon	0
783282	Bart Preneel	1
6889	Chuang Lin	0
100513	Kang G. Shin	0
578328	Sushit Jajodia	0
8215	Wei Sun	1
51270	Lei Guo	0
587770	Rajkumar Buyya	1
2916	Bo Wang	0
40821	Ping Wang	0
50303	Dacheng Tao	1
577920	Mateo Valero	1
9456	Sungyoung Lee	0
54753	Ping Li	1
11000	Lei Li	1
20040	Hui Liu	1
30104	Jiang Chen	1
90724	Feng Liu	0

BIODATA PENULIS

Penulis dilahirkan di Sidoarjo, 13 Mei 1995, merupakan anak ketiga dari 3 bersaudara. Penulis telah menempuh pendidikan formal yaitu TK Tunas Islam (1999-2001), MI Pucang Sidoarjo (2001-2006), SMP Negeri 5 Sidoarjo (2006-2009), SMA Negeri 1 Sidoarjo (2009-2011), mahasiswa S1 Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya dan mahasiswa S2 Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Selama kuliah di teknik informatika ITS, penulis mengambil bidang minat Komputasi Cerdas Visi (KCV). Komunikasi dengan penulis dapat melalui email: abakhrul.ilmi@gmail.com