TESIS – KI 42502

PELABELAN KLASTER ARTIKEL ILMIAH MENGGUNAKAN *TOPICRANK* DAN *MAXIMUM COMMON SUBGRAPH*

Adhi Nurilham
NRP. 5116201047

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 disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya

oleh:
Adhi Nurilham
NRP. 05111650010047

Dengan judul :
Pelabelan Klaster Artikel Ilmiah Menggunakan *TopicRank* dan *Maximum Common Subgraph*

Tanggal Ujian : 20 Juli 2018
Periode Wisuda : September 2018

Disetujui oleh:

Dr. Eng. Chastine Fatichah, M.Kom. NIP. 197512202001122002	..... (Pembimbing 1)
Prof. Dr. Ir. Joko Lianto Buliali NIP. 196707271992031002	..... (Penguji 1)
Dr. Agus Zainal Arifin, S.Kom., M.Kom. NIP. 197208091995121001	..... (Penguji 2)
Dr.Eng. Darlis Herumurti, S.Kom., M.Kom. NIP. 197712172003121001	..... (Penguji 3)

Dekan Fakultas Teknologi Informasi dan Komunikasi,

Dr. Agus Zainal Arifin, S.Kom., M.Kom.
NIP. 197208091995121001

[Halaman ini sengaja dikosongkan]

Pelabelan Klaster Artikel Ilmiah Menggunakan TopicRank dan Maximum Common Subgraph

Nama Mahasiswa : Adhi Nurilham
NRP : 5116 201 047
Pembimbing : Dr. Eng. Chastine Fatichah, M.Kom.

ABSTRAK

Metode klasterisasi dapat memudahkan pengelompokkan artikel ilmiah. Pelabelan klaster diperlukan untuk mengetahui frasa kunci yang merepresentasikan topik bahasan kelompok artikel ilmiah. Beberapa klaster artikel ilmiah perlu digabung karena masih memiliki kemiripan topik untuk memberikan hasil label klaster yang lebih baik. Kemiripan topik dapat diwakili dengan kesamaan relasi kata yang dimodelkan dengan graf. Penelitian ini memiliki usulan metode pelabelan klaster artikel ilmiah dengan proses penggabungan klaster berdasarkan kesamaan struktur graf representasi klaster.

Usulan metode terdiri dari : (1) Pengelompokkan artikel ilmiah menggunakan metode klasterisasi *K-Means++*. (2) Ekstraksi kandidat frasa menggunakan *Frequent Phrase Mining* (FPM). (3) Konstruksi graf menggunakan kata – kata pembentuk frasa sebagai *vertex* dan relasi kata sebagai *edge* berdasarkan *Word2Vec*. (4) Penggabungan klaster dengan pengukuran similaritas klaster berdasarkan struktur *Maximum Common Subgraph* (MCS). (5) Pelabelan klaster pada hasil penggabungan klaster menggunakan metode TopicRank.

Usulan metode dievaluasi pada 2 dataset artikel ilmiah yang memiliki variasi tingkat pemisahan dan kohesi klaster. Koherensi topik digunakan sebagai pengukuran evaluasi untuk mengukur tingkat keterkaitan topik label klaster pada sebuah klaster. Hasil pengujian menunjukkan bahwa dataset yang memiliki tingkat pemisahan dan kohesi klaster yang tinggi (homogen) menghasilkan koherensi topik label klaster gabungan yang lebih tinggi. Penggunaan relasi kata *co-occurrence* pada pembuatan graf representasi klaster menghasilkan koherensi topik yang lebih baik dibandingkan relasi kata *Word2Vec*. Hal ini disebabkan oleh relasi kata *co-occurrence* berbasis frekuensi sehingga merepresentasikan topik mayoritas klaster.

Kata kunci: *Pelabelan Klaster, Teori Graf, TopicRank, Frequent Phrase Mining, Maximum Common Subgraph*

[Halaman ini sengaja dikosongkan]

Cluster Labelling on Scientific Article Using TopicRank and Maximum Common Subgraph

Student Name : Adhi Nurilham

NRP : 5116 201 047

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

ABSTRACT

Unstructured scientific articles can benefited by clustering method to group scientific articles based on topic similarity. Cluster labeling on the yielded cluster is required to discover key phrases that best represent the topics covered. Several clusters still need to be bundled because they still have similar topics to give better cluster labels results. In addition to word occurences, the similarity of the topic can also be represented by word semantic relation that can be modeled with the graph. This research proposes labeling clusters of scientific articles with cluster merging as research contribution to provide a more representative label of cluster topics.

This research proposed cluster labeling method with cluster merging process using graph model. Graph model approach is choosen because it can map the relationship between words, hence representing text semantic information. There are several stages in the proposed method. First, *K-Means++* clustering method is applied on a collection of scientific articles. Second, for each cluster, phrase extraction is executed using *Frequent Phrase Mining* to get word tokens that capable to constitute representative phrase for cluster topics. Acquired word tokens used as input to constructing graph representation of a cluster. After that, cluster merging is done based on cluster graph similarity using *Maximum Common Subgraph* (MCS) method. Then, the cluster labeling process is performed on clusters that have been merged using the *TopicRank* method.

Proposed method evaluated on 2 dataset based on the merged cluster label topic coherence score, using Word2Vec-based graph model and co-occurence-based graph model. Result show that homogenous dataset 1 yield better result than heterogenous dataset 2. In addition, the use of co-occurence-based graph produce prefereable result on cluster merging process.

***Keywords**: Cluster Labeling, Graph Theory, TopicRank, Frequent Phrase Mining, Maximum Common Subgraph*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Alhamdulillahirabbil'alamin. Puji dan syukur penulis panjatkan kehadirat Allah SWT atas berkat, rahmat dan hidayah-Nya, penyusunan Tesis ini dapat diselesaikan. Tesis ini dibuat sebagai salah satu syarat dalam menyelesaikan Program Studi Magister di Institut Teknologi Sepuluh November Surabaya. Penulis menyadari bahwa Tesis ini dapat diselesaikan karena dukungan dari berbagai pihak, baik dalam bentuk dukungan moral dan material. Melalui kesempatan ini dengan kerendahan hati penulis mengucapkan terima kasih dan penghargaan setinggi-tingginya kepada semua orang untuk semua bantuan yang telah diberikan, antara lain kepada:

1. Ayahanda tercinta Bambang Soekanto dan Ibunda tercinta Enna Andari untuk semua doa, pengorbanan dan usaha yang tak kenal lelah telah mendidik dan membimbing dengan penuh ketulusan untuk keberhasilan penulis.
2. Nenek tercinta Toebijati Soetrisno atas dukungan dan doanya selama menempuh pendidikan di Kota Surabaya.
3. Ito Nurarief, Nahla Nur Ardhiani serta seluruh keluarga yang selalu berdoa dan memberikan dukungan.
4. Ibu Dr. Eng. Chastine Fatichah, S.Kom, M.Kom dan Ibu Diana Purwitasari, S.Kom, M.Sc. selaku pembimbing yang senantiasa memberikan arahan dan bimbingan kepada penulis. Semoga Allah SWT senantiasa merahmati Ibu dan keluarga.
5. Bapak Dr. Agus Zainal Arifin, S.Kom, M.Kom., Dr. Darlis Herumurti, S.Kom, M.Kom., dan Bapak Prof. Dr. Ir. Joko Liantio Buliali sebagai tim Penguji Tesis yang memberikan masukan dan kritik yang membangun untuk Tesis ini.
6. Seluruh dosen S2 Teknik Informatika yang telah memberikan ilmu dan pengetahuan kepada penulis selama menempuh studi.
7. Teman seperjuangan lainnya yang tidak dapat disebutkan satu persatu, terima kasih atas bantuan dan motivasi yang telah diberikan.

Akhirnya dengan segala kerendahan hati penulis menyadari masih banyak terdapat kekurangan pada Tesis ini. Oleh karena itu, segala tegur sapa dan kritik yang sifatnya membangun sangat penulis harapkan demi kesempurnaan Tesis ini. Penulis berharap bahwa perbuatan baik dari semua orang yang dengan tulus memberikan kontribusi terhadap penyusunan Tesis ini mendapatkan pahala dari Allah. *Aamiin Alluhamma Aamiin*.

Surabaya,

Adhi Nurilham

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRAK iv
ABSTRACT vi
KATA PENGANTAR viii
DAFTAR ISI x
DAFTAR GAMBAR xii
DAFTAR TABEL xiv
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Perumusan Masalah 3
1.3 Tujuan 4
1.4 Manfaat 4
1.5 Batasan Masalah 4
BAB 2 KAJIAN PUSTAKA 5
2.1 Klasterisasi Dokumen 5
2.1.1 K-Means++ 5
2.1.2 Klasterisasi Hirarkikal 6
2.2 Pra-pemrosesan Teks 7
2.3 Perhitungan Similaritas Teks menggunakan Maximum Common Subgraph 7
2.4 Ekstraksi Frasa menggunakan *Frequent Phrase Mining* (FPM) 8
2.5 Representasi Teks dalam Graf 9
2.6 TopicRank 10
2.7 Koherensi Topik 12
BAB 3 METODOLOGI PENELITIAN 15
3.1 Studi Literatur 15
3.2 Dataset 16
3.3 Perancangan Sistem 17
3.3.1 Pra-pemrosesan Teks (P1 pada Gambar 3.3) 18
3.3.2 Klasterisasi Artikel Ilmiah (P2 pada Gambar 3.3) 19
3.3.3 Penggabungan Klaster (P3, P4, dan P5 pada Gambar 3.3) 19
3.3.3.1 Ekstraksi Frasa Topik dengan *Frequent Phrase Mining* (FPM) (P3 pada Gambar 3.3) 20

3.3.3.2 Konstruksi Graf Representasi Klaster (P4 pada Gambar 3.3) 22
3.3.3.3 Penggabungan Klaster dengan Pengukuran similaritas Maximum Common Subgraph (P5 pada Gambar 3.3) 23
3.3.4 Pelabelan Klaster (P6 pada Gambar 3.3) 26
3.4 Evaluasi 27
BAB 4 UJI COBA & ANALISIS HASIL 29
4.1 Lingkungan Uji Coba 29
4.2 Analisis Dataset 29
4.3 Skenario Uji Coba 30
4.4 Hasil Uji Coba 31
4.4.1 Skenario 1 31
4.4.2 Skenario 2 39
4.4.3 Skenario 3 43
BAB 5 PENUTUP 45
5.1 Kesimpulan 45
5.2 Saran 46
DAFTAR PUSTAKA 47
LAMPIRAN PENGGABUNGAN DAN PELABELAN KLASTER 51
LAMPIRAN DAFTAR RELASI KATA PADA GRAF MAXIMUM COMMON SUBGRAPH 63

DAFTAR GAMBAR

GAMBAR 3.1 ALUR METODOLOGI PENELITIAN 15
GAMBAR 3.2 SKEMA DATASET DARI BASIS DATA AMINER 16
GAMBAR 3.3 ALUR PROSES METODE USULAN 17
GAMBAR 3.4 PSEUDOCODE TAHAP EKSTRAKSI FRASA TOPIK (**P3**) 21
GAMBAR 3.5 PSEUDOCODE TAHAP KONSTRUKSI GRAF KLASTER (**P4**) 22
GAMBAR 3.6 VISUALISASI GRAF KLASTER 23
GAMBAR 3.7 PSEUDOCODE TAHAP PENGGABUNGAN KLATER (**P5.1)** 24
GAMBAR 3.8 PSEUDOCODE PERHITTUNGAN JARAK KLASTER BERBASIS GRAF (**P5.2**) 25
GAMBAR 3.9 PSEUDOCODE PEMBUATAN GRAF MCS (**P5.3**) 25
GAMBAR 3.10 VISUALISASI PENGGABUNGAN KLASTER SECARA HIERARKIKAL 26
GAMBAR 3.11 PSEUDOCODE TAHAP PELABELAN KLASTER (**P6**) 27
GAMBAR 4.1. ANALISIS *SILHOUETTE* DATASET ASLI, DATASET 1, DAN DATASET 2 30
GAMBAR 4.2. GRAF MCS KLASTER ASLI KA-4 DAN KA-8 PADA DATASET 1 SKENARIO 1 33
GAMBAR 4.3. GRAF MCS KLASTER ASLI KA-4 DAN KA-11 PADA DATASET 1 SKENARIO 1 33
GAMBAR 4.4. GRAF MCS KLASTER ASLI KA-8 DAN KA-11 PADA DATASET 1 SKENARIO 1 33
GAMBAR 4.5. GRAF MCS KLASTER ASLI KA-3 DAN KA-5 PADA DATASET 1 SKENARIO 1 34
GAMBAR 4.6. GRAF MCS KLASTER ASLI KA-2 DAN KA-11 PADA DATASET 2 SKENARIO 1 36
GAMBAR 4.7. GRAF MCS KLASTER ASLI KA-6 DAN KA-14 PADA DATASET 2 SKENARIO 1 38
GAMBAR 4.8. GRAF MCS KLASTER ASLI KA-3 DAN KA-5 PADA DATASET 1 SKENARIO 2 39
GAMBAR 4.9. GRAF MCS KLASTER ASLI KA-7 DAN KA-15 PADA DATASET 2 SKENARIO 2 41
GAMBAR 4.10. GRAF MCS KLASTER ASLI KA-7 DAN KA-14 PADA DATASET 2 SKENARIO 2 42
GAMBAR 4.11. GRAF MCS KLASTER ASLI KA-14 DAN KA-15 PADA DATASET 2 SKENARIO 2 43

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

TABEL 3.1 CONTOH DATA ARTIKEL ILMIAH PADA DATASET *AMINER* 16
TABEL 3.2 PENGGUNAAN PUSTAKA PENDUKUNG 18
TABEL 3.3 CONTOH HASIL PRA-PEMROSESAN TEKS 19
TABEL 3.4 CONTOH HASIL DATA TAHAP KLASTERISASI 20
TABEL 3.5 CONTOH KELUARAN TAHAP EKSTRAKSI FRASA 21
TABEL 3.6 CONTOH MODEL KATA WORD2VEC 22
TABEL 3.7 CONTOH HASIL GRAF KLASTER DALAM BENTUK CSV 23
TABEL 3.8 CONTOH HASIL MATRIK JARAK ANTAR KLASTER 25
TABEL 3.9 PERUBAHAN ID KLASTER SETELAH PENGGABUNGAN KLASTER 26
TABEL 4.1 SKENARIO PENGUJIAN USULAN METODE 31
TABEL 4.2. HASIL RATA - RATA KOHERENSI TOPIK SKENARIO 1 32
TABEL 4.3 CONTOH HASIL PENGGABUNGAN DAN PELABELAN KLASTER PERCOBAAN 1 SKENARIO 1 32
TABEL 4.4 CONTOH HASIL PENGGABUNGAN DAN PELABELAN KLASTER PERCOBAAN 2 SKENARIO 1 34
TABEL 4.5 CONTOH HASIL PENGGABUNGAN DAN PELABELAN KLASTER PERCOBAAN 3 SKENARIO 1 35
TABEL 4.6 CONTOH HASIL PENGGABUNGAN DAN PELABELAN KLASTER PERCOBAAN 4 SKENARIO 1 37
TABEL 4.7. HASIL RATA – RATA KOHERENSI TOPIK SKENARIO 2 38
TABEL 4.8 RELASI KATA PADA GRAF MCS KLASTER ASLI KA-3 DAN KA-5 DATASET 1 SKENARIO 2 39
TABEL 4.9 CONTOH HASIL PENGGABUNGAN DAN PELABELAN KLASTER PERCOBAAN 4 SKENARIO 2 40
TABEL 4.10 CONTOH HASIL PENGGABUNGAN DAN PELABELAN KLASTER PERCOBAAN 3 SKENARIO 2 40
TABEL 4.11 HASIL SKENARIO 3 43
TABEL 6.1 HASIL PENGGABUNGAN DAN PELABELAN KLASTER PADA PERCOBAAN 1 DALAM SKENARIO 1 51
TABEL 6.2 HASIL PENGGABUNGAN DAN PELABELAN KLASTER PADA PERCOBAAN 2 DALAM SKENARIO 1 52
TABEL 6.3 HASIL PENGGABUNGAN DAN PELABELAN KLASTER PADA PERCOBAAN 3 DALAM SKENARIO 1 53
TABEL 6.4 HASIL PENGGABUNGAN DAN PELABELAN KLASTER PADA PERCOBAAN 4 DALAM SKENARIO 1 55
TABEL 6.5 CONTOH ARTIKEL ILMIAH PADA KLASTER KA-4 DAN KA-8 DI DATASET 1 SKENARIO1 57
TABEL 6.6 CONTOH ARTIKEL ILMIAH PADA KLASTER KA-8 DAN KA-11 DI DATASET 1 SKENARIO1 57
TABEL 6.7 HASIL PENGGABUNGAN DAN PELABELAN KLASTER PADA PERCOBAAN 3 DALAM SKENARIO 2 58
TABEL 6.8 HASIL PENGGABUNGAN DAN PELABELAN KLASTER PADA PERCOBAAN 4 DALAM SKENARIO 2 60

TABEL 6.9 CONTOH ARTIKEL ILMIAH PADA KLASTER KA-7 DAN KA-15 DI DATASET 2 SKENARIO2 61
TABEL 7.1 50 RELASI KATA PADA GRAF MCS ANTARA KA-2 DAN KA-11 PADA DATASET 2 DI PERCOBAAN 3 SKENARIO 1 63
TABEL 7.2 50 RELASI KATA PADA GRAF MCS ANTARA KA-6 DAN KA-14 PADA DATASET 2 DI PERCOBAAN 3 SKENARIO 1 65
TABEL 7.3 50 RELASI KATA PADA GRAF MCS ANTARA KA-7 DAN KA-15 PADA DATASET 2 DI PERCOBAAN 3 SKENARIO 2 67
TABEL 7.4 50 RELASI KATA PADA GRAF MCS ANTARA KA-7 DAN KA-14 PADA DATASET 2 DI PERCOBAAN 3 SKENARIO 2 69
TABEL 7.5 50 RELASI KATA PADA GRAF MCS ANTARA KA-14 DAN KA-15 PADA DATASET 2 DI PERCOBAAN 3 SKENARIO 2 71

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Metode klasterisasi dokumen dapat digunakan untuk mengelompokkan artikel ilmiah berdasarkan kemiripan topik, namun klasterisasi hanya menghasilkan kelompok artikel ilimiah tanpa disertai label topik penelitian yang mewakili kelompok artikel ilmiah tersebut. Pelabelan klaster diperlukan untu mencari frasa representasi topik dari kelompok artikel ilmiah hasil klasterisasi.

Pada umumnya pelabelan klaster terbagi ke dalam 2 proses utama, yaitu proses ekstraksi frasa kandidat dan proses penilaian frasa. Pada proses ekstraksi frasa, Liao *et al* [1] mengimplementasi proses ekstraksi frasa berbasis aturan yang dibuat secara otomatis berdasarkan struktur *part-of-speech* dari kumpulan judul artikel. Carmel *et al* [2] menggunakan pengukuran *Jensen-Shannon Divergence* (JSD) untuk mencari kumpulan frasa yang memiliki jarak terjauh dengan klaster lainnya. Lopez *et al* [3] mengembangkan metode ekstraksi kata pada dokumen artikel ilmiah dengan menggunakan korpus eksternal GRISP dan *Wikipedia*. GRISP (*General Research Insight in Scientific and Technical Publication*) merupakan basis data terminologi dalam berbagai bidang ilmiah [4]. Kemunculan frasa pada basis data GRISP, membuat frasa tersebut lebih penting daripada frasa – frasa lainnya. El-Kishky *et al*. mengusulkan algoritma *Frequent Phrase Mining* (FPM) [5] yang terinspirasi dari algoritma *Apriori* [6]. Algoritma FPM telah dikembangkan untuk menemukan frasa konseptual yang merepresentasikan topik penelitan pada artikel ilmiah dengan menambahkan metode heuristik [7]. Penggunaan metode heuristik dalam ekstraksi frasa telah terbukti membersikan *recall* yang tinggi pada beberapa penelitian lainnya [8].

Pada proses penilaian frasa untuk pelabelan klaster, pendekatan yang paling banyak digunakan adalah pendekatan statistik. Pada pendekatan statistik, frekuensi kemunculan sebuah kata atau frasa dapat menjadi faktor dalam pemilihan label klaster dokumen. Aalla *et al*. menggunakan pembobotan IDF (*Inverse Document Frequency*) untuk mendapatkan frasa konseptual yang signifikan pada kumpulan artikel ilmiah [7]. Suadaa *et al.* menggunakan pembobotan TF-ICF (*Term Frequency – Inverse Cluster Frequency*) untuk memilih label klaster [9]. TF-ICF menghitung frekuensi kata pada dokumen dan frekuensi kebalikan kata pada klaster – klaster dokumen. Liao *et al*

menggunakan perhitungan berbasis *markov chain* untuk menilai frasa kandidat label klaster [1].

Metode penilaian frasa berbasis graf juga telah banyak diusulkan. Mihalcea *et al* [10] mengusulkan TextRank, metode penilaian frasa berbasis graf yang terinsiprasi dari algoritma PageRank [11]. CollabRank merupakan metode pengembangan TextRank yang menggunakan model graf yang dibuat dari beberapa dokumen dalam klaster yang sama untuk mengekstrasi frasa kata kunci pada sebuah dokumen [12]. Liu *et al* mengusulkan TopicRank yang menggunakan persebaran topik dalam perhitungan algoritma TextRank untuk meningkatkan cakupan topik yang direpresentasikan frasa kunci [13], dan telah dikembangkan oleh Sterckx *et al* untuk mengoptimalkan efisiensi waktu ekstraksi [14].

Label kelompok dokumen yang baik harus dapat dibedakan dengan label kelompok dokumen lainnya [15]. Namun terkadang proses klasterisasi menghasilkan hasil yang kurang optimal (*suboptimal*) sehingga terdapat beberapa kelompok dokumen yang memiliki kemiripan kontekstual [16]. Hal tersebut menyebabkan label kelompok dokumen yang dihasilkan sulit dibedakan. Hasil klasterisasi yang *suboptimal* disebabkan oleh beberapa faktor seperti pemilihan jumlah klaster, adanya dokumen yang bersifat derau (*outlier*), dan pemilihan fitur teks [17]. Oleh karena itu, diperlukan proses identifikasi dan penggabungan kelompok – kelompok artikel ilmiah yang memiliki kemiripan sebelum pelabelan klaster dilakukan.

Penggabungan klaster telah diusulkan pada beberapa penelitian untuk mengatasi berbagai macam permasalahan. Krauza et al. menggunakan penggabungan klaster untuk mengatasi hasil klaster yang tumpang tindih pada metode klasterisasi Fuzzy Gustafson-Kessel (FGK) [18]. Morsier et al. melakukan penggabungan klaster pada klaster yang tumpang tindih berdasarkan persebaran outlier antara klaster [19]. Czarnowski et al. mengusulkan penggabungan klaster berbasis konsensus dalam pemilihan data untuk proses pelatihan model klasifikasi [20].

Metode penggabungan klaster yang telah diusulkan pada umumnya memanfaatkan pendekatan *Vector Space Model* (VSM). Pada pendekatan VSM, jarak klaster, sebagai kriteria penggabungan klaster, dihitung dalam ruang fitur menggunakan perhitungan jarak seperti *euclidean distance*. Pada penggabungan artikel ilmiah pendekatan VSM menganggap bahwa setiap kata bersifat independen terhadap kata lainnya [21]. Sedangkan, sebuah artikel ilmiah tersusun atas kata – kata yang saling

berhubungan. Hubungan antar kata pada sebuah artikel ilmiah dapat menggambarkan unsur semantik yang terdapat pada artikel ilmiah.

Pendekatan graf merupakan salah satu alternatif pendekatan VSM yang telah dijelaskan sebelumnya. Pendekatan graf dapat memetakan kata berdasarkan konteksnya dengan merepresentasikan kata sebagai *node* yang saling berkaitan [22]. Integrasi visualisasi pada graf dan metode statistik dapat membantu pencarian fitur penting pada data [23], tidak terkecuali fitur semantik teks. Jin *et al* [24] mengusulkan perhitungan similaritas teks berbasis graf menggunakan MCS (*Maximum Common Subgraph*). Perhitungan MCS mencari nilai similaritas antar teks dengan menghitung jumlah *node* dan *edge* yang sama antara graf representasi teks. Implementasi MCS pada proses penggabungan klaster dapat menjadi solusi alternatif penggunaan metode perhitungan berbasis VSM.

Pada penelitian ini diusulkan metode pelabelan klaster dengan proses penggabungan klaster menggunakan pendekatan graf. Pendekatan graf dilakukan karena dapat memetakan hubungan antar kata yang merepresentasikan informasi semantik teks. Pertama, metode klasterisasi k-means++ dilakukan pada kumpulan artikel ilmiah. Lalu, pada setiap klaster, ekstraksi frasa dilakukan untuk mendapatkan kata - kata yang merepresentasikan topik klaster. Setelah itu, penggabungan klaster dilakukan dengan pendekatan graf menggunakan metode *Maximum Common Subgraph* (MCS). Proses pelabelan klaster dilakukan pada kumpulan klaster yang telah digabung dengan menggunakan metode TopicRank [14]. Pelabelan klaster TopicRank digunakan karena metode ekstraksi frasa yang mengimplementasi klasterisasi topik terbukti menghasilkan hasil yang baik menggunakan data abstrak artikel ilmiah [8].

1.2 Perumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut.

1. Bagaimana cara merepresentasikan klaster artikel ilmiah dalam bentuk graf berdasarkan topik?
2. Bagaimana cara mengidentifikasi beberapa klaster artikel ilmiah yang memiliki kemiripan topik?
3. Bagaimana cara mengevaluasi hasil label klaster artikel ilmiah?

1.3 Tujuan

Tujuan yang akan dicapai dalam pembuatan tesis ini adalah pelabelan klaster artikel ilmiah menggunakan TopicRank dan *Frequent Phrase Mining* (FPM) dengan penggabungan klaster berbasis graf. Penggabungan klaster dilakukan menggunakan perhitungan similaritas graf representasi klaster berdasarkan struktur *Maximum Common Subgraph* antar graf klaster.

1.4 Manfaat

Manfaat dari penelitian ini adalah memudahkan pencarian artikel ilmiah yang diinginkan dengan membentuk kelompok – kelompok artikel ilmiah yang disertai oleh label frasa kunci yang representatif.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Data dokumen yang digunakan adalah abstrak artikel ilmiah berbahasa Inggris dari basis data *Aminer* pada publikasi *IEEE Transactions on Computer*.
2. Jumlah dataset asli adalah 12.000 artkel ilmiah yang dibagi ke dalam 2 jenis dataset dengan karakteristik yang berbeda.

BAB 2
KAJIAN PUSTAKA

Pada bab ini akan dijelaskan tentang pustaka yang terkait dengan landasan penelitian. Pustaka yang terkait dianalisa dan dirangkum dalam bentuk studi komparasi.

2.1 Klasterisasi Dokumen

Klasterisasi dokumen merupakan sebuah proses yang bertujuan untuk mengelompokkan dokumen. Pada penelitian ini akan digunakan dua metode klasterisasi dokumen yaitu K-Means++ dan Klasterisasi Hirarkikal. Kedua pendekatan klasterisasi dokumen tersebut dijelaskan pada subbab berikut ini.

2.1.1 K-Means++

K-Means++ merupakan metode pengembangan dari metode K-Means untuk mengelompokan dokumen [25]. Pada umumnya metode K-Means memilih data titik pusat klaster secara acak. Hal tersebut menyebabkan metode K-Means dapat menghasilkan solusi yang sub-optimal. Oleh karena itu, K-Means++ mengatasi permasalahan tersebut dengan mengoptimasi pemilihan titik pusat klaster awal sebelum metode K-Means dijalankan. Pada umumnya K-Means dan 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 (1), dimana $|v_1^d|$ merupakan nilai skalar dari v_1^d.

$$CosSim(v_1^d, v_2^d) = \frac{v_1^d \cdot v_2^d}{|v_1^d|_2 \ |v_2^d|_2} \quad (1)$$

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 sebuah data dokumen secara acak sebagai *centroid* awal klaster 1 yang dinotasikan sebagai C_1.
2. Untuk setiap data dokumen d_i hitung $D(d_i)$ yang merupakan jarak d_i ke *centroid* terdekat yang sudah terpilih.
3. Pilih *centroid* klaster selanjutnya menggunakan probabilitas $\frac{D(d_i)}{\sum D(d_i)}$
4. Ulangi langkah 2-3 sampai *centroid* untuk seluruh klaster telah terpilih.
5. Hitung similaritas dokumen d_i dengan tiap *centroid* pada C menggunakan rumus (1).
6. Dokumen d_i akan menjadi anggota klaster yang memiliki nilai similaritas *centroid* tertinggi.
7. Ulangi langkah 2-3 untuk setiap dokumen dalam D.
8. Hitung ulang *centroid* untuk setiap $C_i \in Centroids$.
9. Ulangi langkah 5-8, sampai tidak ada dokumen yang berpindah klaster.

2.1.2 Klasterisasi Hirarkikal

Klasterisasi hirarkikal dokumen teks merupakan metode pengelompokkan dokumen yang bekerja dengan membangun sebuah hirarki kelompok dokumen atau klaster. Hirarki klaster dapat disebut juga sebagai dendogram. Klasterisasi hirarkikal telah digunakan untuk pembentukan taksonomi konsep pada suatu teks [26]. Kelebihan klasterisasi hirarkikal dibandingkan dengan metode klasterisasi lain adalah tingkat hirarki yang dapat ditentukan sesuai kebutuhan [27].

Pada umumnya, pembangunan dendogram menggunakan pendekatan *bottom-up* yang dilakukan secara iteratif. Pada iterasi awal setiap dokumen memiliki klaster tersendiri, lalu klaster yang memiliki similaritas tertinggi akan digabung pada setiap iterasi selanjutnya. Iterasi akan berhenti, jika nilai similaritas tertinggi kurang dari nilai batas yang telah ditentukan. Similaritas antara dua buah dokumen diukur dengan menggunakan MCS (*Maximum Common Subgraph*) yang dijelaskan pada subbab 2.3. Setelah dua klaster digabung, nilai similaritas baru antara gabungan klaster tersebut dengan klaster lain ditentukan dengan rata – rata dari selisih nilai similaritas antara tiap anggota klaster (*average linked*).

Masukkan metode klasterisasi hirarkikal adalah matriks similaritas klaster yang dihitung dengan menggunakan MCS $cluster\_sim$ dan nilai batas similaritas minimum min_sim. Alur proses klasterisasi hirarkikal dokumen dijelaskan sebagai berikut :

1. Cari pasangan klaster a dan b yang memiliki nilai kedekatan tertingi, $[a, b] = max(cluster\_sim).index()$.
2. Jika nilai kedekatan tertinggi klaster a dan b kurang dari min_sim, maka proses dihentikan.
3. Gabung klaster a dan b menjadi klaster c.
4. Perbaharui nilai kedekatan klaster c dengan klaster lain pada matriks $cluster\_sim$.
5. Ulangi langkah 1-4.

2.2 Pra-pemrosesan 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 konversi ke huruf kecil, penghilangan tanda baca, dan penghilangan *stopwords*.

2.3 Perhitungan Similaritas Teks menggunakan Maximum Common Subgraph

Metode MCS (*Maximum Common Subgraph*) merupakan metode untuk mencari kesamaan sub-struktur yang optimal sama antara 2 graf. MCS dapat digunakan untuk menghitung similaritas antar teks berbasis graf [24]. Masukkan dari MCS adalah graf representasi teks G_1 dan G_2. Pembuatan graf representasi teks telah dijelaskan lebih detil pada subbab 2.5. Keluaran dari MCS adalah sub-graf optimal G'. Nilai similaritas graf representasi teks dapat dihitung dari hasil sub-graf. Alur metode MCS dapat digambarkan sebagai berikut :

1. Cari *node* yang sama antara G_1 dan G_2, dan tambahkan ke G'.
2. Ambil 2 *node* yang berbeda pada G'. Jika kedua *node* bersebelahan pada G_1 dan G_2, maka *edge* yang menghubungkan kedua *node* tersebut ditambahkan pada G'. Bobot *edge* terkecil antara bobot *edge* di G_1 dan G_2, untuk menjadi bobot *edge* pada G'.
3. Ulangi langkah 2, sampai tidak ada lagi *edge* yang bisa ditambahkan ke G'.

Setelah sub-graf optimal G' didapatkan, similaritas antar graf representasi teks $S(G_1, G_2)$ dihitung dengan menggunakan rumus (2), dimana $N_{G'}$ merupakan jumlah *node*

pada G', $N_{\max(G_1,G_1)}$ merupakan nilai maksimal total *node* pada G_1 dan G_2, $E_{G'}$ merupakan jumlah *edge* pada G', dan $E_{\max(G_1,G_1)}$ merupakan nilai maksimal total *edge* pada G_1 dan G_2. Koefisien α merupakan nilai antara 0 dan 1 yang mewakili tingkat kepentingan *node* terhadap *edge* pada sub-graf.

$$S(G_1, G_2) = \alpha \frac{N_{G'}}{N_{\max(G_1,G_1)}} + (1-\alpha) \frac{E_{G'}}{E_{\max(G_1,G_1)}} \qquad (2)$$

2.4 Ekstraksi Frasa menggunakan *Frequent Phrase Mining* (FPM)

Pada umumnya dokumen memiliki 2 jenis frasa, yaitu frasa konseptual dan non-konseptual. Frasa yang merepresentasikan topik penelitian merupakan frasa konseptual. El-Kishky et al. mengusulkan penggunaan FPM (*Frequent Phrase Mining*) untuk mengekstraksi frasa dari sebuah korpus [5]. FPM terinspirasi salah satu metode penggalian *association rule* yaitu *frequent itemset mining*.

Metode FPM memiliki masukkan berupa kumpulan dokumen $D = [d_1, \dots, d_n]$ dan sebuah nilai batas minimum frekuensi c_min. Jika ukuran frasa yang akan diidentifikasi adalah n, maka pada iterasi awal nilai $n = 2$. Ukuran n akan bertambah seiring dengan bertambahnya iterasi. Metode FPM menghasilkan kumpulan kandidat frasa konseptual dan frekuensi kemunculannya. Alur metode FPM adalah sebagai berikut :

1. Ulangi jika $D \neq \emptyset$
   1.1. Ulangi untuk setiap dokumen $d \in D$
      1.1.1. Identifikasi seluruh $(n-1)$-gram pada dokumen d, dan simpan indeks aktif $(n-1)$-gram pada $A_{d,n-1}$. Indeks aktif merupakan indeks awal sebuah $(n-1)$-gram pada dokumen d. Sebuah $(n-1)$-gram tersusun atas kata w, $(n-1)$-gram $= [w_i, .., w_{i+(n-2)}]$, dimana i merupakan indeks aktif $(n-1)$-gram tersebut.
      1.1.2. Hitung frekuensi kemunculan setiap $(n-1)$-gram pada dokumen d.
      1.1.3. $(n-1)$-gram yang memiliki frekuensi kemunculan kurang dari -c_min, indeks aktifnya akan dikeluarkan dari $A_{d,n-1}$.
      1.1.4. Ulangi untuk setiap $i \in A_{d,n-1}$, jika $A_{d,n-1} \neq \emptyset$
         1.1.4.1. Jika $i + 1 \in A_{d,n-1}$ maka sebuah frasa P dibentuk oleh $[w_i, .., w_{i+(n-1)}]$.

1.1.4.2. Tambahkan penghitung frekuensi frasa P, $count(P) + +$.

1.1.4.3. Masukkan frasa P ke dalam n-gram, dan indeks awal frasa P ke $A_{d,n}$.

1.1.5. Hilangkan dokumen d dari D, jika $A_{d,n} = \emptyset$

1.2. $n + +$

2.5 Representasi Teks dalam Graf

Cara paling umum untuk merepresentasikan teks adalah dengan pendekatan VSM (*Vector Space Model*). Pendekatan VSM pada umumnya menggunakan frekuensi kata sebagai fitur teks. Namun, cara tersebut tidak mempertimbangkan informasi semantik dan struktur dari sebuah teks. Model graf dapat merepresentasikan teks secara matematis dengan tetap menjaga informasi semantik dan struktur teks [21].

Sumber teks dapat berupa satu atau banyak dokumen. Setelah melalui tahap pra-pemrosesan teks, kata – kata pada teks tersebut akan menjadi *node*. Relasi kemunculan bersama (*co-occurrence*) antar kata digunakan untuk membentuk *edge* pada graf. Relasi kemunculan bersama telah banyak digunakan untuk menggambarkan hubungan kontekstual antar kata pada teks [28] [24] [22]. Namun belakangan *Word2Vec* [29] banyak diusulkan untuk mencari similaritas antar kata seperti pada kasus pencarian sinonim. Oleh karena itu, pembentukan relasi antar kata pada graf ditentukan dengan 2 cara yaitu : a) Relasi kata menggunakan *Word2Vec*, b) Relasi kata menggunakan *Co-occurrence*.

Graf representasi teks dapat dinotasikan sebagai $G = (V, E, W)$, dimana V merupakan kumpulan *vertex/node*, E merupakan kumpulan *edge* dan W merupakan bobot untuk setiap *edge*. Jika $V_i \in V$ merepresentasikan *node* untuk kata t_i, dan $V_j \in V$ merepresentasikan *node* untuk kata t_j, maka sebuah *edge* $E_{ij} \in E$ akan dibentuk berdasarkan 2 cara yaitu relasi kata *Word2Vec* dan *Co-occurrence*.

a. Relasi Kata *Word2Vec*

Word2Vec [29] mempergunakan set teks sebagai masukkan dan memberikan keluaran berupa vektor yang merepresentasikan suatu kata melalui proses *training*. Pada pembuatan graf klaster artikel ilmiah, vektor kata yang dihasilkan *Word2Vec* digunakan untuk mencari similaritas antar kata. Similaritas kata tersebut dapat menggambarkan

suatu informasi semantik antar kata. Sehingga bobot *edge* E_{ij}, dinotasikan sebagai $w_{ij} \in W$ merupakan similaritas vektor *Word2Vec* antar kata *i* dan kata *j*.

b. Relasi Kata *Co-occurence*

Pada relasi kata *Co-occurence*, bobot *edge* E_{ij}, dinotasikan sebagai $w_{ij} \in W$, merupakan frekuensi kemunculan bersama yang telah dinormalisasi antara t_i dan t_j pada maksimum bentang 2 kata. Jika n_i merupakan frekuensi kata t_i pada teks, dan N merupakan jumlah kata pada teks, maka bobot kata w_i dapat dihitung menggunakan rumus (3). Bobot *edge* $w_{ij} \in W$ dapat dihitung menggunakan rumus (4), dimana $w_{i,j}$ frekuensi kemunculan bersama t_i dan t_j pada maksimum bentang 2 kata.

$$w_i = \frac{n_i}{N} \quad (3)$$

$$w_{ij} = \frac{w_{i,j}}{w_i + w_j - w_{i,j}} \quad (4)$$

2.6 TopicRank

TopicRank [14] merupakan metode ekstraksi kata atau frasa yang terinspirasi oleh algoritma penentuan peringkat PageRank [11]. TopicRank menerima masukkan teks yang telah dimodelkan dalam bentuk graf. Teks dapat bersumber dari satu atau banyak dokumen. Graf masukkan tersebut memiliki kata sebagai *node*, dan *edge* berupa relasi antar kata. Pada penelitian ini, *edge* pada graf masukkan berupa kemunculan bersama antar kata (*word co-occurrence*), dan jenis graf merupakan *undirected graph*. Konstruksi graf sebagai masukkan metode TopicRank dijelaskan lebih detil pada subbab 2.6.

Pada dasarnya metode TopicRank menentukan tingkat kepentingan sebuah node kata berdasarkan informasi global pada struktur graf dan persebaran topik dokumen. Dalam hal ini, seluruh teks artikel ilmiah pada sebuah klaster disatukan dan dianggap sebagai sebuah dokumen. Metode ini menjalankan PageRank sebanyak jumlah topik yang ada. Untuk setiap perhitungan skor setiap *node*, pembobotan *node* dilakukan dengan menghitung *cosine similarity* antara vektor probabilitas kata terhadap topik dan vektor probabilitas topik terhadap dokumen. Bobot *node* tersebut digunakan untuk mengganti probabilitas *random walk* pada algoritma PageRank. Vektor probabilitas *node* kata V_i dinotasikan dengan $\vec{P}(V_i \mid Z) = [\, P(V_i \mid z_1), \ldots, P(V_i \mid z_n) \,]$, dimana $P(V_i \mid z_i)$

merupakan probabilitas kata V_i pada topik z_i. Vektor probabilitas topik pada dokumen dinotasikan dengan $\vec{P}(Z \mid d) = [\, P(z_1 \mid d), \ldots, P(z_n \mid d)\,]$, dimana $P(z_i \mid d)$ merupakan probabiltas topik z_i pada dokumen *d*. Bobot *node* V_i dinotasikan dengan w_i^n dihitung dengan rumus (5).

$$w_i^n = \frac{\vec{P}(V_i \mid Z) \,.\vec{P}(Z \mid d)}{||\vec{P}(V_i \mid Z)|| \,.\, ||\,\vec{P}(Z \mid d)||} \qquad (5)$$

Jika sebuah graf dilambangkan sebagai $G = (V, E, W^n, W^e)$, dimana V adalah kumpulan *vertex/node*, E adalah kumpulan *edge*, W^n adalah kumpulan bobot *node*, dan W^e adalah kumpulan bobot *edge*. Notasi *N* merepresentasikan jumlah *node* yang terdapat pada graf. Untuk setiap *node* $V_i \in V$, notasi $In(V_i)$ merupakan kumpulan *node* yang mengarah ke *node* V_i, dan notasi $Out(V_i)$ merupakan kumpulan *node* tujuan *node* V_i. Relasi antara *node* V_i dan V_j memiliki bobot *edge* yang dilambangkan dengan $w_{i,j}^e \in W^e$. Bobot *node* V_i dinotasikan dengan $w_i^n \in W^n$. Maka, skor *node* V_i pada topik *z*, dinotasikan dengan $R_z(V_i)$, akan diperbaharui setiap iterasi dengan rumus (6).

$$R_z(V_i) = (1-d)\frac{w_i^n}{\sum_{k \in |W^n|} w_k^n} + d * \sum_{V_j \in In(V_i)} \frac{w_{j,i}^e}{\sum_{V_k \in Out(V_j)} w_{j,k}^e} R_z(V_j) \qquad (6)$$

Pada rumus (6), d merupakan koofisien *damping*, yang memiliki nilai antara 0 dan 1. Koefisien *damping* mewakili kemungkinan loncatan dari sebuah *node* ke *node* acak. Dalam konteks pe nelusuran web, koefisien *damping* menggambarkan probabilitas, sebesar d, pengguna untuk memilih *link* yang tersedia pada halaman tersebut, dan probabilitas, sebesar $(1-d)$, pengguna untuk pergi ke halaman web yang benar – benar acak. Implementasi koefisien *damping* dapat disebut juga *"Random Surfer Model"*.

Pasca-pemrosesan akan dilakukan setelah konvergensi tercapai, yaitu jika nilai bobot *node* $WS(V_i)$ sudah tidak banyak mengalami perubahan. Pada pasca-pemrosesan, sejumlah *n* kata yang memiliki skor bobot *node* terbesar akan dipilih. Setiap *n* kata akan diperiksa kumpulan *in-degree node* dan *out-degree node* milik *node* kata tersebut, untuk mencari kata lain yang terletak bersebelahan pada dokumen asal. Jika ditemukan, maka

kata tersebut akan digabung menjadi sebuah frasa. Keluaran tahap pasca-pemrosesan adalah kumpulan kata dan frasa yang dianggap merepresentasikan korpus sumber.

2.7 Koherensi Topik

Koherensi topik mengukur interpretabilitas topik set label frasa yang dihasilkan pelabelan klaster artikel ilmiah. Suatu set label frasa koheren secara topikal jika antara satu label dengan label lainnya memiliki keterkaitan kontekstual. Pada penelitian ini set label frasa yang dihasilkan oleh pelabelan klaster untuk merepresentasikan topik bahasan klaster suatu artikel ilmiah dievaluasi menggunakan koherensi topik. Roder *et al* telah mengusulkan pengukuran koherensi topik yang memiliki performa lebih baik dibandingkan usulan metode koherensi topik yang telah ada sebelumnya dalam hal korelasinya dengan penilaian manusia [30].

Misal proses pelabelan klaster menghasilkan sebanyak *n* label frasa yang merepresentasikan topik bahasan klaster artikel ilmiah ke-*i* yang dinotasikan sebagai $CL_i = [l_1, \ldots., l_n]$. Setiap label frasa $l_k \in CL_i$ direpresentasikan ke dalam vektor ruang frasa $\overrightarrow{vl_k}$ yang memiliki dimensi sebesar jumlah kata *m*. Elemen vektor $\overrightarrow{vl_k}$ ke-*j* yang dinotasikan dengan vl_{kj} dihitung dengan (9) yang merupakan perhitungan *Normalized Pointwise Mutual Information* (NPMI). Notasi ϵ pada (7) merupakan nilai bias yang ditentukan sendiri. $P(l_i)$ merupakan probabilitas kemunculan frasa l_i pada rentang jendela kata (*sliding window*) yang dinotasikan dengan n_{win}. $P(l_i, l_j)$ merupakan probabilitas kemunculan bersama frasa l_i dan l_j pada n_{win}. Probabilitas kemunculan frasa dihitung berdasarkan korpus *Wikipedia* yang terdiri dari artikel *Wikipedia* dari tahun 2009. Nilai n_{win} dan ϵ yang digunakan adalah 110 dan 1 sesuai dengan penelitian pada [30].

$$PMI(l_k, l_j) = log \frac{P(l_k, l_j) + \epsilon}{P(l_k).P(l_j)} \quad (7)$$

$$NPMI(l_k, l_j) = \frac{PMI(l_k, l_j)}{-log(P(l_k, l_j) + \epsilon)} \quad (8)$$

$$vl_{kj} = NPMI(l_k, l_j) \quad (9)$$

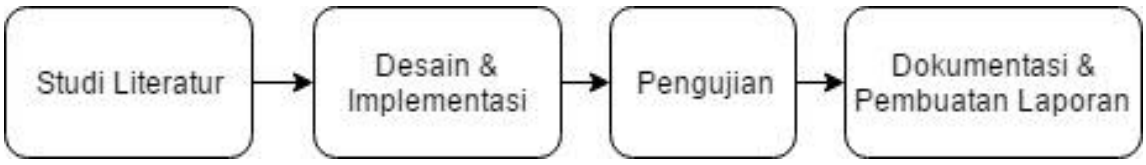
Jika kumpulan vektor konteks frasa $VL = [\overrightarrow{vl_1}, \ldots, \overrightarrow{vl_q}]$, maka kombinasi pasangan vektor konteks kata disimpan dalam $qC_2(VL)$. Koherensi topik set label klaster L dihitung dengan (10) dimana $CosSim(S)$ merupakan cosine similarity dari pasangan vektor konteks kata $S \in qC_2(VL)$.

$$Coh(L) = \frac{\sum_{S \in qC_2(VL)} CosSim(S)}{|qC_2(VL)|} \quad (10)$$

[Halaman ini sengaja dikosongkan]

BAB 3
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

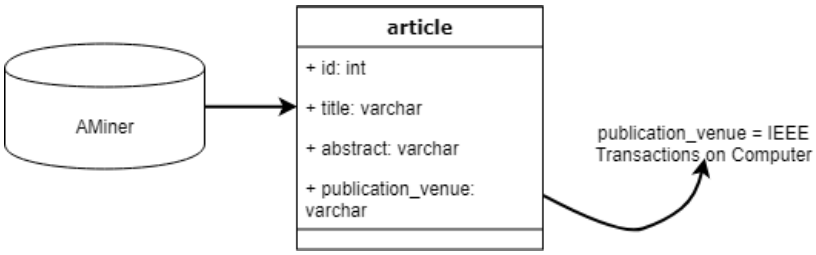
3.1 Studi Literatur

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 pemrosesan teks yang dapat dipakai untuk pelabelan klaster khususnya pada dokumen artikel ilmiah. 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 penelitian yang dilakukan ini, seperti berikut :

1. Frasa lebih deskriptif dibandingkan kata dalam melabeli klaster dokumen.
2. Hasil dari proses klasterisasi mempengaruhi proses pelabelan klaster.
3. Pendekatan Vector Space Model pada proses klasterisasi tidak mempertimbangkan hubungan antar kata pada teks.
4. Graf dapat memberikan informasi yang lebih mendalam mengenai hubungan antar kata di dalam sebuah dokumen.

3.2 Dataset

Data artikel ilmiah yang digunakan untuk tahap pengujian adalah dataset *Citation* dari basis data publikasi ilmiah *AMiner*[1]. Seterusnya dataset tersebut disebut sebagai dataset asli. Dataset asli tersusun atas artikel ilmiah dari jurnal *IEEE Transaction of Computers* beserta artikel ilmiah kutipannya dan berjumlah ±12.000 artikel ilmiah. Skema basis data *AMiner* yang merupakan sumber dataset dapat dilihat pada Gambar 3.2.



Gambar 3.2 Skema Dataset dari Basis Data AMiner

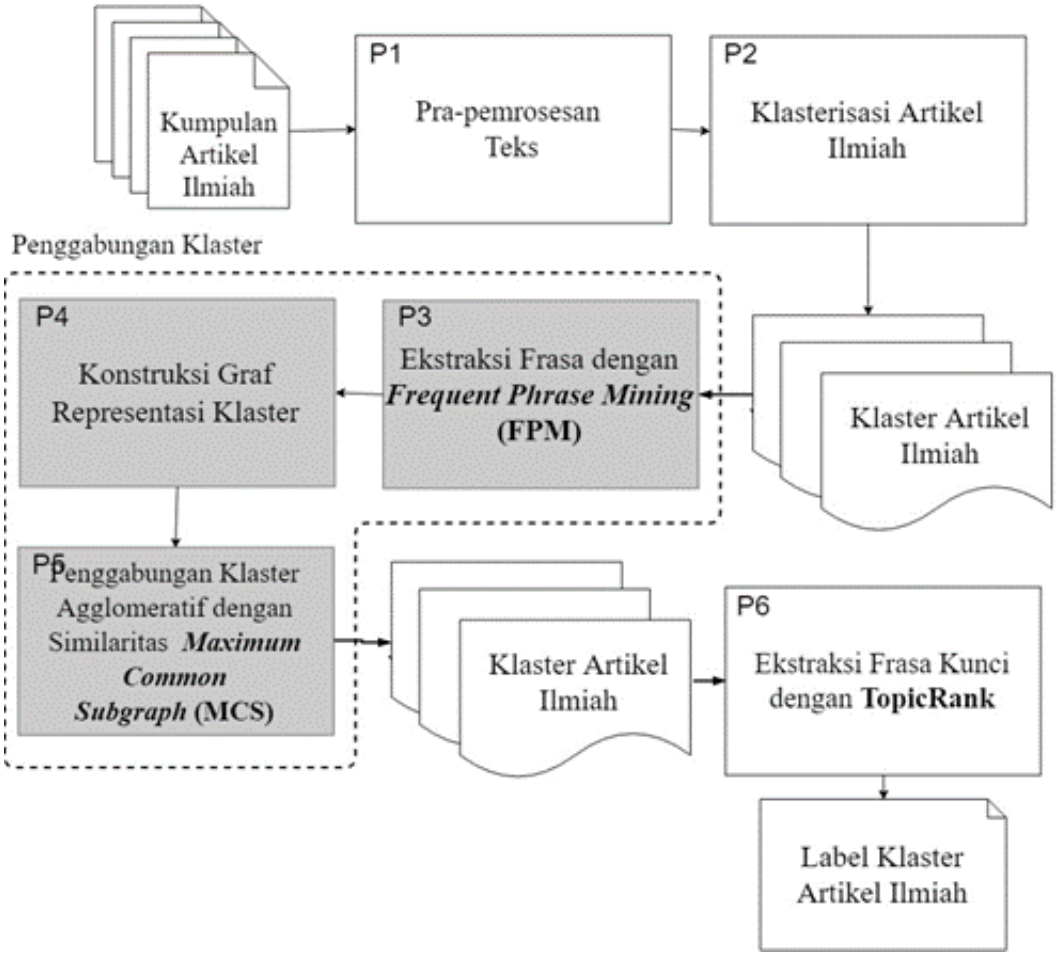
Dataset asli akan dibagi menjadi dua dataset yaitu dataset 1 dan dataset 2 yang memiliki karakteristik data yang berbeda. Dataset 1 memiliki jumlah yang lebih kecil dengan kualitas pemisahan antar klaster yang lebih baik daripada dataset 2. Pembagian dataset ini bertujuan untuk pengamatan performa usulan metode dalam penggunaan dataset dengan variasi karakteristik. Contoh data artikel ilmiah pada dataset tertera pada Tabel 3.8. Usulan metode hanya memerlukan kolom judul dan abstrak artikel ilmiah untuk diproses selanjutnya. Rata – rata jumlah kata pada dataset artikel ilmiah adalah 137.2 kata.

Tabel 3.1 Contoh Data Artikel Ilmiah pada Dataset *AMiner*

ID Artikel Ilmiah	**Judul**	**Abstrak**	**Publikasi**
19452	Pseudorandom Testing	Pseudorandom Testing : Algorithmic test generation for high fault coverage is an expensive and time-consuming process. As an alternative, circuits can be tested by applying pseudorandom patterns generated by a linear feedback shift register (LFSR). Although no fault simulation is needed, analysis of pseudorandom testing requires the circuit detectability profile.	*IEEE Transaction on Computer*

[1] https://aminer.org/

3.3 Perancangan Sistem



Gambar 3.3 Alur Proses Metode Usulan

Alur proses metode pelabelan klaster usulan terdiri dari beberapa tahap, seperti pada Gambar 3.3. Pertama, tahap pra-pemrosesan teks dilakukan pada kumpulan artikel ilmiah sebelum proses klasterisasi dokumen dilakukan. Kedua, pada setiap klaster artikel ilmiah, ekstraksi frasa dilakukan untuk mendapatkan frasa – frasa yang merepresentasikan topik. Penggabungan klaster dilakukan pada setiap klaster melalui 2 tahap, yaitu konstruksi graf representasi klasters dan penggabungan klaster menggunakan pengukuran similaritas dengan *Maximum Common Subgraph* (MCS). Pada tahap konstruksi graf representasi klaster, untuk setiap klaster artikel ilmiah, kata – kata yang menyusun frasa – frasa dari keluaran tahap ekstraksi frasa digunakan untuk menjadi *node*, dan relasi kemunculan bersama (*co-occurrence*) antar kata digunakan untuk menghitung bobot *edge*. Perhitungan similaritas dilakukan menggunakan metode MCS pada graf –

Tabel 3.2 Penggunaan Pustaka Pendukung

Nama Pustaka/ Aplikasi	Penggunaan	Versi Python	Dokumentasi Pustaka
skelarn	Implementasi klasterisasi K-Means++ pada tahap P2 (lihat Gambar 3.3)	3.6	http://scikit-learn.org/stable/ documentation.html
Gensim	Implementasi pembuatan model Word2Vec pada tahap P4 (lihat Gambar 3.3)	3.6	https://code.google.com/ archive/p/word2vec/
networkx	Penggunaan struktur data graf pada tahap P4 dan P5 (lihat Gambar 3.3)	3.6	https://networkx.github.io/ documentation/stable/
Gephi	Visualisasi Graf	-	https://gephi.org/
Matplotlib	Visualisasi Dendogram Penggabungan Klaster pada P5 (lihat Gambar 3.3)	3.6	https://matplotlib.org/
pke	Implementasi Pelabelan Klaster pada tahap P6 (lihat Gambar 3.3)	2.7	https://github.com/boudinfl/pke
palmetto	Implementasi metode evaluasi koherensi topik pada P6 (lihat Gambar 3.3)	2.7	https://github.com/earthquakesan/ palmetto-py

graf klaster yang telah dibuat. Klaster – klaster yang memiliki nilai similaritas MCS yang lebih dari nilai batas akan digabungkan pada tahap penggabungan klaster. Pelabelan klaster dilakukan untuk setiap klaster hasil proses penggabungan klaster. Konstruksi graf representasi klaster akan dilakukan kembali menggunakan klaster hasil penggabungan untuk menjadi masukkan metode ekstraksi frasa kunci *TopicRank*. Keluaran dari proses ekstraksi frasa kunci adalah *n* label frasa terbaik untuk merepresentasikan klaster artikel ilmiah. Tahap – tahap proses metode usulan akan dijelaskan pada subbab berikut.

Implementasi usulan metode menggunakan bahasa pemrograman *python2.7* dan *python3.6*. Beberapa pustaka (*library*) pendukung digunakan dalam merangcang sistem usulan metode dan proses evaluasi. Detil penggunaan pustaka pendukung dalam setiap tahap perancangan sistem dapat dilihat pada Tabel 3.2.

3.3.1 Pra-pemrosesan Teks (P1 pada Gambar 3.3)

Tahap pra-premosesan teks bertujuan untuk mempersiapkan data teks untuk dapat diproses pada tahap selanjutnya. Tahap pra-pemrosesan teks terdiri dari konversi ke huruf

kecil, tokenisasi, penghilangan tanda baca, dan penghilangan *stopwords*. Contoh dari tahap pra-pemrosesan teks dapat dilihat pada Tabel 3.3.

Tabel 3.3 Contoh Hasil Pra-pemrosesan Teks

Teks Artikel Ilmiah (article_id = 19452)
Pseudorandom Testing : Algorithmic test generation for high fault coverage is an expensive and time-consuming process. As an alternative, circuits can be tested by applying pseudorandom patterns generated by a linear feedback shift register (LFSR). Although no fault simulation is needed, analysis of pseudorandom testing requires the circuit detectability profile.
Hasil Pra-pemrosesan Teks
['pseudorandom', 'testing', 'algorithmic', 'test', 'generation', 'high', 'fault', 'coverage', 'expensive', 'time', 'consuming', 'process', 'alternative', 'circuits', 'tested', 'applying', 'pseudorandom', 'patterns', 'generated', 'linear', 'feedback', 'shift', 'register', 'lfsr', 'although', 'fault', 'simulation', 'needed', 'analysis', 'pseudorandom', 'testing', 'requires', 'circuit', 'detectability', 'profile']

3.3.2 Klasterisasi Artikel Ilmiah (P2 pada Gambar 3.3)

Tahap klasterisasi artikel ilmiah bertujuan untuk membagi kumpulan dokumen artikel ilmiah ke beberapa klaster. Pada penelitian ini, metode K-Means++ akan digunakan untuk klasterisasi dokumen artikel ilmiah. Pengukuran similaritas antar dokumen akan menggunakan *Cosine Similarity*. Penjelasan metode klasterisasi K-Means ++ dengan *Cosine Similarity* dijelaskan pada subbab 2.1.1.

Keluaran dari tahap ini adalah data artikel ilmiah, terdiri dari teks dan hasil pra-pemrosesan teks, yang diberikan label klaster berupa angka, seperti pada Tabel 3.4. Data artikel ilmiah hasil klasterisasi disimpan dalam bentuk (*Comma-seperated Value*). Klasterisasi K-Means++ dilakukan dengan bantuan pustaka *sklearn* pada *python3.6*.

3.3.3 Penggabungan Klaster (P3, P4, dan P5 pada Gambar 3.3)

Proses penggabungan klaster bertujuan untuk menggabungkan klaster yang memiliki kemiripan kontekstual dengan pendekatan graf. Pertama – tama ekstraksi frasa kandidat dilakukan untuk memilih kata – kata yang berpotensi membentuk frasa topik, yang akan dijadikan sebagai *vertex* pada graf klaster. Lalu graf representasi klaster dibuat

Tabel 3.4 Contoh Hasil Data Tahap Klasterisasi

ID Klaster	ID Artikel Ilmiah	Data Tersimpan	Isi Data
4	19452	**Teks**	Pseudorandom Testing : Algorithmic test generation for high fault coverage is an expensive and time-consuming process. As an alternative, circuits can be tested by applying pseudorandom patterns…
		Hasil Pra-pemrosesan	['pseudorandom', 'testing', 'algorithmic', 'test', 'generation', 'high', 'fault', 'coverage', 'expensive', 'time', 'consuming', 'process', 'alternative', 'circuits', 'tested', 'applying', 'pseudorandom', 'patterns', 'generated', ….]
4	40420	**Teks**	Pseudoexhaustive Test Pattern Generator with Enhanced Fault Coverage : A method of pseudoexhaustive test pattern generation is proposed that is suitable above all for circuits using random access scan…..
		Hasil Pra-pemrosesan	['pseudoexhaustive', 'test', 'pattern', 'generator', 'enhanced', 'fault', 'coverage', 'method', 'pseudoexhaustive', 'test', 'pattern', 'generation', 'proposed', 'suitable', 'circuits', 'using', 'random', 'access', 'scan',…….]

dengan memodelkan relasi antar kata. Graf klaster tersebut akan digunakan dalam perhitungan jarak klaster untuk penggabungan klaster. Tahap – tahap pada proses penggabungan klaster dijelaskan pada subbab berikut.

3.3.3.1 Ekstraksi Frasa Topik dengan *Frequent Phrase Mining* (FPM) (P3 pada Gambar 3.3)

Tahap ekstraksi frasa topik bertujuan untuk mengurangi jumlah kata pada setiap klaster artikel ilmiah dengan mengidentifikasi kata – kata yang berpotensi untuk menyusun frasa topik. Tahap ini memiliki masukkan berupa seluruh teks yang terdapat pada klaster artikel ilmiah dan memprosesnya menjadi daftar kata – kata pembentuk frasa yang merepresentasikan topik.

Identifikasi frasa topik dilakukan menggunakan metode FPM. Metode FPM terinspirasi dari algoritma *apriori* pada *frequent item set mining*. Metode FPM bekerja dengan mengidentifikasi *frasa* dimulai dari nilai ukuran frasa *n* terkecil yaitu $n = 1$ sampai *n* maksimal yang masih memenuhi nilai batas yang telah ditentukan. Alur proses metode FPM telah dijelaskan pada subbab 2.4.

Pseudocode modul program FPM pada Gambar 3.4 menunjukkan bahwa masukkan metode FPM memiliki masukkan berupa kumpulan artikel ilmiah pada sebuah

```
1  def frequentPhraseMining(masukan=(klaster, minsupport)) :
2      list_frasa = list()
3      list_kata = list()
4      indeks_aktif = list()
5      kumpulan_dokumen = [seluruh dokumen.teks di dalam klaster]
6      panjang_frasa = 1
7
8      #IDENTIFIKASI FRASA
9      while jumlah(kumpulan_dokumen) not 0 :
10         for dokumen.teks in kumpulan_dokumen :
11             indeks_aktif[panjang_frasa] = seluruh indeks awal frasa dengan
12                 ukuran = panjang_frasa, yang memiliki frekuensi > minsupport
13
14             for indeks in indeks_aktif :
15                 if (indeks in indeks_aktif) and (indeks + 1 in indeks_aktif) :
16                     frasa_baru = dokumen.teks[indeks] + ' ' + dokumen.teks[indeks + 1]
17                     list_frasa.tambahkan(frasa_baru)
18                     indeks_aktif[panjang_frasa + 1].tambahkan(indeks)
19
20             if jumlah(indeks_aktif[panjang_frasa + 1]) is 0 :
21                 kumpulan_dokumen.hilangkan(dokumen_teks)
22
23         panjang_frasa += 1
```

Gambar 3.4 Pseudocode Tahap Ekstraksi Frasa Topik (**P3**)

klaster, seperti pada baris ke-1. Untuk setiap dokumen pada klaster, daftar indeks aktif diperlukan yang mencatat seluruh indeks awal dari frasa dengan ukuran frasa tertentu dan jumlah frekuensi yang lebih dari nilai parameter *support* yang ditentukan, seperti pada baris ke-10 s.d 11. Daftar indeks aktif tersebut akan digunakan untuk mencari frasa yang bersebelahan pada dokumen, seperti pada baris ke-14 s.d ke-18. Dokumen artikel ilmiah yang tidak memiliki frasa dengan ukuran panjang frasa tertentu tidak akan lagi digunakan untuk mencari frasa dengan ukuran yang lebih panjang, seperti pada baris ke-20 s.d ke-21. Daftar frasa yang telah ditemukan akan dipecah lagi menjadi daftar kata penyusunnya, seperti pada baris ke-26 s.d ke-29. Daftar kata penyusun frasa tersebut akan menjadi

Tabel 3.5 Contoh Keluaran Tahap Ekstraksi Frasa

ID Klaster	**Daftar Frasa**
4	['test length', 'pseudorandom testing', 'required test', 'circular self', 'self test', 'test path', 'testing module', 'module placement', 'test set', 'random pattern', 'pattern testable', 'proposed procedure', 'test points', 'probe points', 'test sequence', 'random vector', 'sequential circuits', 'sequential circuit', 'fault coverage'.....']
	Daftar Kata
	['test', 'length', 'pseudorandom', 'testing', 'required', 'circular', 'self', 'path', 'module', 'placement', 'set', 'random', 'pattern', 'testable', 'proposed', 'procedure', 'points', 'probe', 'sequence', 'vector', 'sequential', 'circuits',…...']

keluaran untuk menjadi masukkan pada tahap konstruksi graf klaster selanjutnya. Keluaran dari tahap ini adalah daftar frasa dan daftar kata untuk setiap klaster, seperti pada Tabel 3.5. Pada tahap konstruksi graf klaster daftar kata tersebut akan digunakan sebagai vertex dari graf klaster.

3.3.3.2 Konstruksi Graf Representasi Klaster (P4 pada Gambar 3.3)

```
def konstruksiGrafKlaster(masukkan=(daftarKata_klaster,w2v_model)) :
    #INISIALISASI GRAF KLASTER
    graph = Graph()

    #TAMBAHKAN VERTEX KE GRAF KLASTER
    graph.tambah_vertex(daftarKata_klaster)

    #TAMBAHKAN EDGE KE GRAF KLASTER
    kombinasi_vertex = kombinasi(graph.vertex())
    for pasangan_vertex in kombinasi_vertex :
        similaritas = w2v_model.hitung_similaritas(pasangan_vertex)
        if similaritas > 0.5 :
            graph.tambah_edge(pasangan_vertex, weight=similaritas)

    return graph
```

Gambar 3.5 Pseudocode Tahap Konstruksi Graf Klaster (**P4**)

Tujuan tahap konstruksi graf representasi klaster adalah untuk membuat graf yang dapat memetakan hubungan antar kata di dalam sebuah klaster artikel ilmiah. Alur pseudocode tahap konstruksi graf dapat dilihat pada Gambar 3.5. Masukkan pada tahap konstruksi graf adalah daftar kata keluaran tahap ekstraksi frasa seperti pada Tabel 3.5 dan model *Word2Vec* yang menyimpan vektor kata seperti pada Tabel 3.6. Pembuatan vektor kata dengan model *Word2Vec* diimplementasi menggunakan pustaka gensim pada pemrograman *python*3.6.

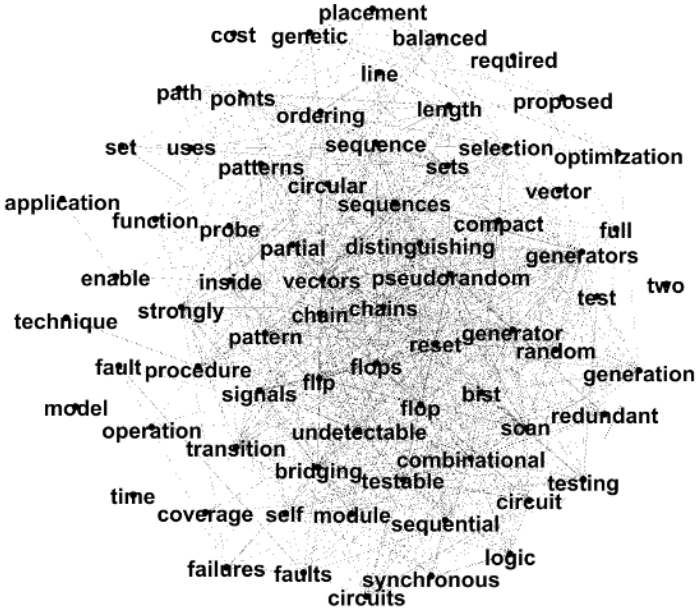
Struktur data graf menggunakan pustaka *networkx* pada *python3.6*. Seluruh kata pada daftar kata akan dijadikan sebagai *vertex* pada graf klaster. *Edge* pada graf klaster dibentuk dengan mencari pasangan kata yang memiliki similaritas *Word2Vec* lebih dari

Tabel 3.6 Contoh Model Kata Word2Vec

Kata	Vektor Kata (dimensi = 300)
circuits	[-0.00039145 0.00316087 0.00768429 -0.00942764 -0.0075033 0.00480119 0.00288294 0.00070033 0.00022734 0.00061888 0.0028147 -0.00451374 ….]
scan	[-0.1351647 0.24729827 0.5253942 0.05551716 0.06554069 0.3630426 0.09835636 -0.37275463 0.43852735 -0.16854073 -0.08030003 0.47590494 ….]

Tabel 3.7 Contoh Hasil Graf Klaster dalam Bentuk CSV

Vertex Sumber (kata)	Vertex Target (kata)	Tipe Edge	Bobot Edge
circuits	Scan	Undirected	0.630
circuits	sequential	Undirected	0.534
circuits	testable	Undirected	0.759
testable	circular	Undirected	0.597
testable	scan	Undirected	0.834
testable	sequential	Undirected	0.675
pseudorandom	ordering	Undirected	0.526
pseudorandom	points	Undirected	0.516
pseudorandom	generator	Undirected	0.947



Gambar 3.6 Visualisasi Graf Klaster

0.5 seperti pada baris ke-9 s.d ke-13 pada pseudocode. Graf klaster yang telah dibuat disimpan dalam format *Comma-seperated Value* (CSV) seperti pada Tabel 3.7. Visualisasi graf klaster yang telah dibuat seperti pada Gambar 3.6 dilakukan dengan aplikasi *Gephi* menggunakan file CSV yang telah dihasilkan.

3.3.3.3 Penggabungan Klaster dengan Pengukuran similaritas Maximum Common Subgraph (P5 pada Gambar 3.3)

Tahap ini bertujuan untuk menggabungkan klaster – klaster yang memiliki kemiripan topik. Pseudocode tahap penggabungan klaster dapat dilihat pada []. Masukkan

dari tahap ini adalah graf – graf representasi klaster. Pertama – tama matriks jarak antar klaster dibuat dengan pengukuran jarak menggunakan MCS (*Maximum Common Subgraph*) seperti pada baris. Pembuatan matriks jarak antar klaster dilakukan seperti pada pseudocode di Gambar 3.7 pada baris ke-4 s.d ke-10. Contoh keluaran hasil matriks jarak antar klaster dapat dilihat pada Tabel 3.8.

```
1  def penggabunganKlaster(masukkan=(daftarGraf, threshold)) :
2      n = jumlah(daftarGraf)
3
4      #PEMBUATAN MATRIKS JARAK ANTAR KLASTER
5      matriks_jarak = matriks(n, n)
6      for i in range(n-1) :
7          for j in range(i+1, n) :
8              graf_x = daftarGraf[i]
9              graf_y = daftarGraf[j]
10             matriks_jarak[i][j] = jarak_graf(graf_x, graf_y)
11
12     #PENGGABUNGAN KLASTER SECARA HIRARKIKAL
13     while(min(matriks_jarak) < threshold) :
14         baris, kolom = min(matriks_jarak)
15         gabung_klaster(baris, kolom)
16         perbaharui_jarak(matriks_jarak)
```

Gambar 3.7 Pseudocode Tahap Penggabungan Klater (**P5.1)**

Pada pseudocode penggabungan klaster di Gambar 3.7 terlihat bahwa matriks jarak antar klaster diisi oleh nilai yang diambil dari modul `jarak_graf()` yang menghitung jarak antar klaster menggunakan model graf. Pseudocode perhitungan jarak klaster berbasis graf dapat dilihat pada Gambar 3.8. Namun, pertama – tama graf *Maximum Common Subgraph* (MCS) harus dibuat menggunakan pasangan graf seperti pada baris ke-3. Lalu jarak antar klaster dapat dihitung berdasarkan jumlah *vertex* dan *edge* yang terdapat pada graf MCS antara kedua graf, seperti pada baris ke-22.

Sebelum perhitungan jarak klaster berbasis graf dapat dilakukan, pembuatan graf MCS perlu dilakukan antara pasangan graf. Pseudocode pembuatan graf MCS pada Gambar 3.9 menunjukkan bahwa pasangan graf, `graf_x` dan `graf_y` diperlukan untuk menjadi masukkan. Pertama – tama daftar *vertex* yang muncul pada kedua graf disimpan dan dijadikan sebagai *vertex* pada graf MCS, seperti pada baris ke-3 s.d ke-5. Setelah itu diperiksa apakah terdapat *edge* yang sama antara `graf_x` dan `graf_y` untuk dijadikan sebagai *edge* pada graf MCS, seperti pada baris ke-8 s.d ke-17. Sama seperti graf klaster, graf MCS dapat disimpan dalam format CSV seperti Tabel 3.7 dan di-visualisasikan menggunakan aplikasi *Gephi* seperti pada Gambar 3.6.

```
def jarak_graf(masukkan=(graf_x, graf_y)) :
    #KONTRUKSI GRAF MCS
    grafMCS = konstruksiGrafMCS(graf_x, graf_y)

    #PERHITUNGAN JUMLAH VERTEX & EDGE PADA GRAF MCS
    jmlvertex_grafMCS = jumlah(grafMCS.vertex())
    jmledge_grafMCS = jumlah(grafMCS.edge())

    #PERHITUNGAN JUMLAH VERTEX & EDGE PADA GRAF X
    jmlvertex_graf_x = jumlah(graf_x.vertex())
    jmledge_graf_x = jumlah(graf_x.edge())

    #PERHITUNGAN JUMLAH VERTEX & EDGE PADA GRAF Y
    jmlvertex_graf_y = jumlah(graf_y.vertex())
    jmledge_graf_y = jumlah(graf_y.edge())

    #MENCARI JUMLAH MAX VERTEX & EDGE
    jmlvertex_max = max(jmlvertex_graf_x, jmlvertex_graf_y)
    jmledge_max = max(jmledge_graf_x, jmledge_graf_y)

    #PERHITUNGAN JARAK KLASTER BERBASIS GRAF
    jarak = 1-( alpha*(float(jmlvertex_grafMCS /jmlvertex_max))
            + (1-alpha)*(float(jmledge_grafMCS /jmledge_max)) )
    return jarak
```

Gambar 3.8 Pseudocode Perhittungan Jarak Klaster Berbasis Graf (**P5.2**)

```
def konstruksiGrafMCS(masukkan=(graf_x, graf_y)) :
    #CARI VERTEX YANG SAMA ANTARA GRAF X DAN GRAF Y
    vertex_umum = [vertex yang sama antara graf_x & graf_y]
    grafMCS = graph()
    grafMCS.tambah_vertex(vertex_umum)

    #PENCARIAN EDGE YANG SAMA ANTARA GRAF X DAN GRAF Y
    for vertex_i in grafMCS.vertex() :
        for vertex_j in tetangga(vertex_i, graf_x) :
            if graf_y.punya(vertex_i,vertex_j) :
                #PERHITUNGAN BOBOT EDGE GRAF MCS
                bobot_graf_x = graf_x[vertex_i][vertex_j]['weight']
                bobot_graf_y = graf_y[vertex_i][vertex_j]['weight']
                bobot_min = min(bobot_graf_x, bobot_graf_y)
                bobot_max = max(bobot_graf_x, bobot_graf_y)
                bobot_MCS = (bobot_min / bobot_max)
                grafMCS.tambah_edge(vertex_i, vertex_j, weight=bobot_MCS)
```

Gambar 3.9 Pseudocode Pembuatan Graf MCS (**P5.3**)

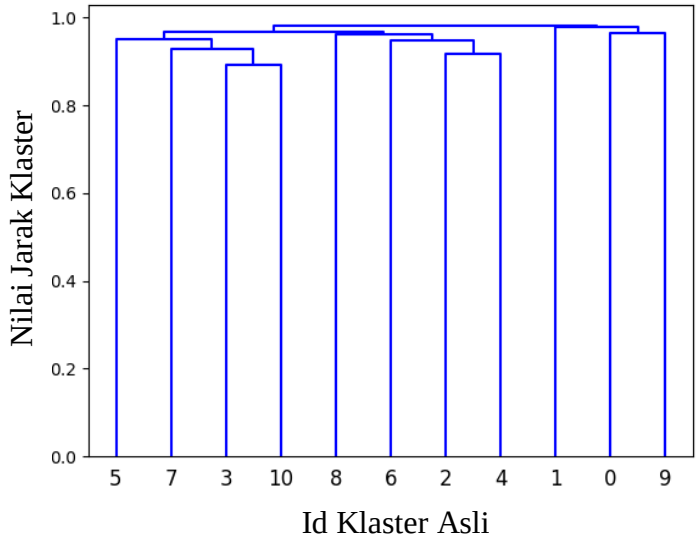
Penggabungan klaster hierarkikal secara agglomeratif dilakukan menggunakan matriks jarak antar klaster yang telah dibuat, seperti pada baris ke-13 s.d ke-16 di pseudocode pada Gambar 3.7. Penggabungan secara hierarkikal terus dilakukan selama jarak terendah antar klaster tidak melebihi nilai *threshold* yang telah ditentukan seperti yang tertera pada kondisi di baris ke-13. Visualisasi penggabungan secara hierarkikal, menggunakan pustaka *matplotlib* pada *python3.6*, dapat dilihat pada

Gambar 3.10. Hasil penggabungan klaster disimpan dalam bentuk CSV seperti pada Tabel 3.9, dimana kolom pertama merepresentasikan *id* klaster baru hasil penggabungan klaster, dan kolom kedua merepresentasikan *id* klaster asli sebelum

Tabel 3.9 Perubahan Id Klaster setelah Penggabungan Klaster

ID Klaster Baru	1	1	1	2	3	3	4	5	6	7	8
ID Klaster Asli	4	8	11	6	3	5	7	9	1	10	2

dilakukan penggabungan klaster. Pasca-pemrosesan dilakukan memnggunakan hasil penggabungan klaster di Tabel 3.9 untuk membentuk file simpanan dengan format seperti di Tabel 3.4, dimana kolom *id* klaster diperbaharui seperti hasil penggabungan klaster.



Gambar 3.10 Visualisasi Penggabungan Klaster Secara Hierarkikal

3.3.4 . Pelabelan Klaster (P6 pada Gambar 3.3)

Pada tahap ini graf representasi klaster dibuat dari klaster hasil penggabungan sebagai masukkan pada tahap ekstraksi frasa kunci menggunakan *TopicRank*. *TopicRank* memberikan skor pada *node* secara iteratif berdasarkan struktur graf dan distribusi topik yang didapatkan dengan metode *Latent Dirchlet Allocation* (LDA). Kata – kata dengan nilai tertinggi akan dicocokan dengan daftar frasa klaster artikel ilmiah. Jika frasa tersusun atas kata – kata yang memiliki nilai *TopicRank* tertinggi, maka frasa tersebut dapat merepresentasikan label klaster artikel ilmiah. Detil metode *TopicRank* dijelaskan pada subbab 2.6.

```
def pelabelanKlaster(masukkan=(klasterGabungan, klasterAsli)) :
    #MENYATUKAN SELURUH TEKS DOKUMEN DI KLASTER MENJADI SATU TEKS
    teksKlasterGabungan = string()
    for dokumen in klasterGabungan :
        teksKlasterGabungan += dokumen.teks

    teksKlasterAsli = string()
    for dokumen in klasterAsli :
        teksKlasterAsli += dokumen.teks

    #EKSTRAKSI FRASA KUNCI KLASTER
    keyphrase_klasterGabungan = ekstraksiKeyphrhase(teksKlasterGabungan)
    keyphrase_klasterAsli = ekstraksiKeyphrhase(teksKlasterAsli)

    #PERHITUNGAN KOHERENSI LABEL KLASTER
    koherensi_klasterGabungan = koherensiTopik(keyphrase_klasterGabungan)
    koherensi_klasterAsli = koherensiTopik(keyphrase_klasterAsli)

    #KELUARAN HASIL FRASA KUNCI KLASTER & KOHERENSI TOPIK LABEL KLASTER
    #DALAM FORMAT CSV
    keluaran_dalam_CSV(keyphrase_klasterGabungan, keyphrase_klasterAsli,
                       koherensi_klasterGabungan, koherensi_klasterAsli)
```

Gambar 3.11 Pseudocode Tahap Pelabelan Klaster (**P6**)

Pseudocode tahap pelabelan klaster dapat dilihat pada Gambar 3.11 dimana kumpulan artikel ilmiah klaster gabungan (hasil dari penggabungan klaster) dan klaster asli menjadi masukkan. Pertama – tama seluruh teks artikel ilmiah pada klaster digabung menjadi satu teks seperti pada baris ke-2 s.d ke-5. Setelah itu, identifikasi frasa kunci dilakukan dengan menggunakan pustaka *pke* (*Python Keyphrase Extraction*) pada *python2.7*. Lalu frasa kunci yang dihasilkan dievaluasi menggunakan koherensi topik dengan pustaka *palmetto* pada *python2.7*. Hasil frasa kunci klaster dan koherensi topik label klaster disimpan dalam format CSV seperti pada Tabel 6.1 (lampiran).

3.4 Evaluasi

3.3.1. Skenario Evaluasi

Tahapan selanjutnya setelah implementasi usulan metode dilakukan adalah tahap pengujian. Tahap pengujian bertujuan untuk mengevaluasi performa usulan metode dan kesesuaian terhadap tujuan penelitian. Performa usulan metode akan diukur berdasarkan kualitas label frasa klaster menggunakan koherensi topik. Pada tahap pengujian dilakukan beberapa skenario pengujian yaitu :

1. Membandingkan kualitas label klaster pada penggunaan model graf relasi *Word2Vec* di dataset 1 dan dataset 2 dengan menggunakan *threshold* yang berbeda.

2. Membandingkan kualitas label klaster pada penggunaan model graf relasi *Co-occurence* di dataset 1 dan dataset 2 dengan menggunakan *threshold* yang berbeda.
3. Membandingkan kualitas label klaster menggunakan graf representasi klaster yang memiliki perbedaan ukuran.

BAB 4
UJI COBA & ANALISIS HASIL

4.1 Lingkungan Uji Coba

Implementasi dan uji coba pada penelitian ini dilakukan pada perangkat keras dengan spesifikasi seperti berikut :

a. Sistem Operasi Windows 10 64-bit
b. *Processor* Intel i5-4200U CPU @ 1.60GHz (4 CPUs)
c. Kapasistas RAM 8GB
d. Kapasitas *harddisk* 500GB

Perangkat lunak pendukung dalam implementasi usulan metode pada penelitian ini adalah seperti berikut :

a. Python 2.7 dan Python 3.6
b. Basis Data *MySQL* 5.7

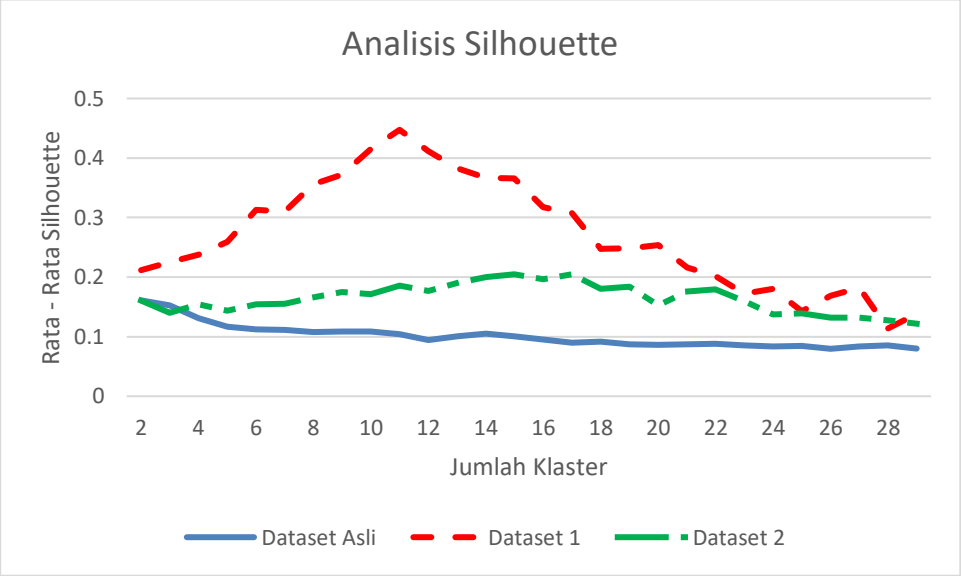
4.2 Analisis Dataset

Dataset asli merupakan kumpulan artikel ilmiah yang bersumber dari *Aminer* jurnal *IEEE Transaction of Computers* beserta artikel ilmiah kutipannya yang memiliki total jumlah ±12.000 artikel ilmiah. Observasi awal dilakukan pada dataset asli dengan melakukan analisis *silhouette* untuk menentukan jumlah klaster optimal. Hasil observasi awal pada dataset asli dapat dilihat pada Gambar 4.1 yang menunjukkan dataset asli tidak memiliki jumlah klaster optimal. Hal tersebut disebabkan oleh banyaknya data – data artikel ilmiah yang tidak memiliki kemiripan oleh klaster manapun (ditandai dengan nilai *silhouette* yang rendah) atau biasa disebut data derau (*outlier*). Pada dataset asli dengan jumlah klaster 11 terdapat 51.16% data artikel ilmiah yang memiliki nilai silhouette kurang dari 0.1. Oleh karena itu dataset asli dibersihkan dari data derau dengan 2 cara :

1. Memilih 50 data artikel ilmiah dengan nilai *silhouette* tertinggi pada tiap klaster dari hasil klasterisasi dataset asli dengan jumlah klaster 11. Hasil pemilihan data tersebut berjumlah 550 data artikel ilmiah dan seterusnya disebut sebagai **dataset 1**.

2. Memilih data artikel ilmiah dengan nilai *silhouette* lebih dari 0.1 dari hasil klasterisasi dataset asli dengan jumlah klaster 17. Hasil pemilihan data tersebut berjumlah ±5500 artikel ilmiah dan seterusnya disebut sebagai **dataset 2**.

Hasil analisis *silhouette* pada dataset 1 dan 2 pada Gambar 4.1 menunjukkan bahwa dataset 1 memiliki rata – rata nilai *silhouette* yang lebih tinggi dibandingkan dataset 2, sehingga dataset 1 bersifat homogen dan dataset 2 bersifat heterogen. Gambar 4.1 juga menunjukkan bahwa jumlah klaster optimal pada dataset 1 dan dataset 2 adalah 11 dan 15 berturut – turut.



Gambar 4.1. Analisis *Silhouette* Dataset Asli, Dataset 1, dan Dataset 2

4.3 Skenario Uji Coba

Pengujian usulan metode dilakukan dengan 2 skenario seperti pada

Tabel 4.1. Pada skenario 1 pembuatan model graf klaster dengan relasi kata dibangun berdasarkan *Word2Vec* dilakukan, sedangkan pada skenario 2 relasi kata pada model graf klaster dibangun menggunakan *Co-occurrence*. Pada setiap skenario dataset 1 dan dataset 2 akan digunakan, masing – masing menggunakan *threshold* penggabungan klaster yang berbeda. *Threshold* penggabungan klaster ditentukan dengan mencari jarak antara klaster – klaster yang ada. Lalu seluruh jarak antar klaster tersebut diurutkan dari nilai yang terkecil sampai terbesar. Nilai persentil ke-5% dan ke-10% terendah dipilih dari seluruh jarak antar klaster yang telah diurutkan tersebut untuk *threshold* penggabungan klaster. Oleh karena itu, skenario 1 dan 2 bertujuan untuk mengetahui

pengaruh penggunaan relasi kata dan penentuan *threshold* yang berbeda terhadap penggabungan klaster berbasis graf menggunakan *Maximum Common Subgraph* (MCS).

Skenario 3 menguji usulan metode menggunakan parameter *support* pada metode *Frequent Phrase Mining* (FPM) di tahap ekstraksi frasa kandidat. Parameter *support* FPM membatasi jumlah kata yang digunakan untuk pembuatan graf klaster, sehingga berpengaruh pada ukuran graf klaster yang dihasilkan. Semakin kecil nilai parameter *support* FPM maka semakin banyak kata yang digunakan untuk pembuatan graf klaster sehingga semakin besar ukuran graf klaster yang dihasilkan. Sehingga skenario 3 bertujuan untuk mengetahui pengaruh ukuran graf terhadap penggabungan klaster berbasis graf menggunakan *Maximum Common Subgraph*.

Tabel 4.1 Skenario Pengujian Usulan Metode

Skenario	**Dataset**	**Parameter**	**Nilai**
1	1	Relasi Kata	*Word2Vec*
		Threshold Persentil (%)	5, 10
	2	Relasi Kata	*Word2Vec*
		Threshold Persentil (%)	5, 10
2	1	Relasi Kata	*Co-occurrence*
		Threshold Persentil (%)	5, 10
	2	Relasi Kata	*Co-occurrence*
		Threshold Persentil (%)	5, 10
3	1	Relasi Kata	*Word2Vec*
		Support FPM	3, 2
	2	Relasi Kata	*Word2Vec*
		Support FPM	3, 2

4.4 Hasil Uji Coba

4.4.1 Skenario 1

Pada usulan metode tahap penggabungan klaster, klaster digabung secara hirarkikal berdasarkan nilai batas (*threshold*) yang ditentukan. Skenario 1 membandingkan hasil pelabelan klaster pada dataset 1 dan dataset 2 menggunakan nilai *threshold* yang berbeda dengan tujuan untuk mengetahui pengaruh *threshold* pada hasil

pelabelan klaster. Nilai *threshold* penggabungan klaster pada skenario 1 ditentukan berdasarkan persentil 10% dan 5% dari seluruh jarak antar klaster. Pada skenario ini parameter *support* pada tahap ekstraksi frasa menggunakan metode *Frequent Phrase Mining* (FPM) adalah 3.

Tabel 4.2. Hasil Rata - Rata Koherensi Topik Skenario 1

Percobaan	Dataset	Threshold Persentil (%)	Klaster Asli		Klaster Gabungan	
			Jumlah Klaster	Rata - Rata Koherensi Topik	Jumlah Klaster	Rata - Rata Koherensi Topik
1	1	10	11	0.5011	8	**0.5061**
2		5	11	0.5011	9	**0.5180**
3	2	10	15	**0.4513**	9	0.4397
4		5	15	**0.4513**	11	0.4501

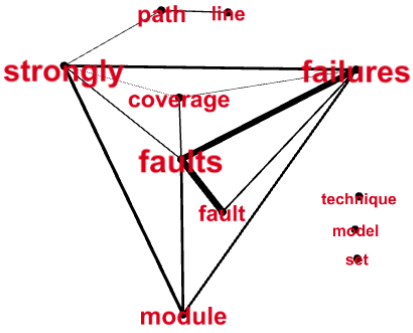
a. Analisa Hasil Penggabungan dan Pelabelan Klaster Dataset 1 Skenario 1

Hasil skenario 1 pada Tabel 4.2 menunjukan bahwa percobaan 1 dengan menggunakan dataset 1 dan *threshold* pada persentil 10% menghasilkan rata – rata koherensi topik klaster gabungan lebih kecil dibandingkan dengan *threshold* pada persentil 5%. Tabel 6.1 memperlihatkan hasil penggabungan dan pelabelan klaster pada percobaan 1 dimana klaster gabungan KG-1 merupakan klaster hasil penggabungan atas klaster asli KA-4, KA-8, dan KA-11. Hasil penggabungan dan pelabelan klaster pada klaster gabungan KG-1 di percobaan 1 dapat dilihat pada Tabel 4.3. Graf *Maximum*

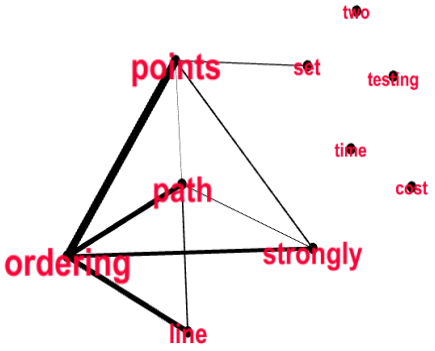
Tabel 4.3 Contoh Hasil Penggabungan dan Pelabelan Klaster Percobaan 1 Skenario 1

Klaster Gabungan			Klaster Asli			
ID Klaster	Label Klaster	Koherensi Topik	ID Klaster	Label Klaster	Koherensi Topik	Koherensi Topik Gabungan
KG-1	• test generation algorithm • test generation time • new fault model • test application time • system fault diagnosis	0.5302	KA-4	test pattern generation, test generator circuit, test generation method, test generation time, test generation algorithm	0.4687	0.4890
			KA-8	polynomial time algorithm, n algorithm, n log n, time algorithm, best such algorithm	0.5654	
			KA-11	system fault diagnosis, new fault model, fault injection techniques, transient fault tolerance, rtl fault model	0.4329	

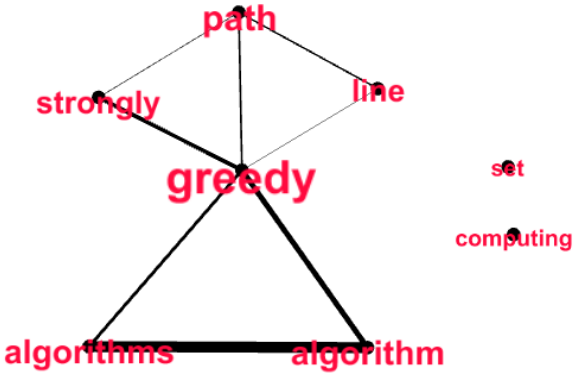
Common Subgraph (MCS) yang merepresentasikan similartas antar klaster asli penyusun klaster gabungan KG-1 ditujukkan pada Gambar 4.2, Gambar 4.3, dan Gambar 4.4.



Gambar 4.3. Graf MCS Klaster Asli KA-4 dan KA-11 pada Dataset 1 Skenario 1



Gambar 4.2. Graf MCS Klaster Asli KA-4 dan KA-8 pada Dataset 1



Gambar 4.4. Graf MCS Klaster Asli KA-8 dan KA-11 pada Dataset 1 Skenario 1

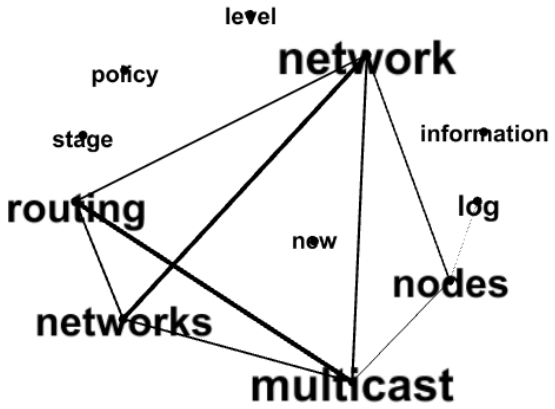
Gambar 4.3 memperlihatkan graf MCS antara klaster asli KA-4 dan KA-11. Pada graf MCS tersebut relasi antara kata [‘*fault*’] dan [‘*coverage*’] menunjukkan bahwa klaster asli KA-4 dan KA-11 memiliki kesamaan pada topik analisa kesalahan sistem. Sementara graf MCS antara klaster asli KA-4 dan KA-8 yang ditunjukkan oleh Gambar 4.2 sulit diinterpretasi ke dalam sebuah konteks dimana terdapat sebuah relasi terbentuk antara kata [‘*ordering*’] dan [‘*path*’]. Seperti pada Tabel 6.5 penggunaan kata [‘*ordering*’] dan [‘*path*’] digunakan dalam konteks yang berbeda pada artikel ilmiah di klaster KA-4 dan KA-8.

Rendahnya kualitas konteks graf MCS juga terjadi pada graf MCS antara klaster asli KA-8 dan KA-11 dimana terdapat relasi antara kata [‘*greedy*’] dan [‘*algorithm*’] yang

Tabel 4.4 Contoh Hasil Penggabungan dan Pelabelan Klaster Percobaan 2 Skenario 1

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-1	• new fault model • system fault diagnosis • high fault coverage • test pattern generation • complete fault coverage	0.5786	KA-4	test pattern generation, test generator circuit, test generation method, test generation time, test generation algorithm	0.4687	0.4508
			KA-11	system fault diagnosis, new fault model, fault injection techniques, transient fault tolerance, rtl fault model	0.4329	

menunjukkan bahwa konteks similaritas klaster adalah algoritma *greedy*. Namun pada Tabel 6.6 terlihat bahwa jumlah artikel ilmiah yang mengandung konteks algoritma *greedy* pada masing – masing klaster hanya 1 artikel ilmiah. Berdasarkan pengamatan tersebut dapat disimpulkan bahwa klaster asli KA-8 kurang tepat jika digabung dengan dengan klaster asli KA-4 dan KA-11. Konteks similaritas yang tidak sesuai disebabkan oleh penggunaan model graf *Word2Vec* yang mengabaikan frekuensi kata.



Gambar 4.5. Graf MCS Klaster Asli KA-3 dan KA-5 pada Dataset 1 Skenario 1

Penggabungan lain pada percobaan 1 terjadi antara klaster asli KA-3 dan KA-5. Graf MCS antara klaster tersebut pada Gambar 4.5 menunjukkan bahwa terdapat relasi antara kata [‘*network*’] dan [‘*multicast*’] yang menunjukkan bahwa similaritas kedua klaster tersebut terdapat pada konteks jaringan. Penggabungan klaster asli KA-3 dan KA-5 menghasilkan koherensi topik yang lebih baik jika dibandingkan dengan koherensi topik sebelum penggabungan.

Pada percobaan 2 usulan metode diimplementasi pada dataset dengan menggunakan *threshold* penggabungan klaster pada persentil 5%. Hasil penggabungan

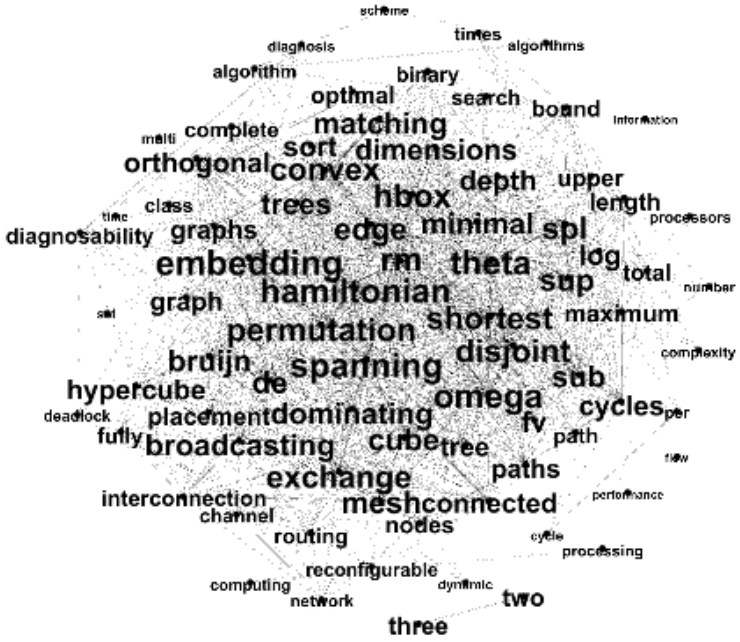
dan pelabelan klaster pada percobaan 2 dapat dilihat pada Tabel 6.2. Hasil penggabungan dan pelabelan klaster pada klaster gabungan KG-1 di Tabel 4.4 menunjukkan bahwa klaster gabungan KG-1 hanya tersusun atas klaster asli KA-4 dan KA-11 saja. Hilangnya klaster asli KA-8 sebagai klaster penyusun KG-1 disebabkan karena jumlah *vertex* dan *edge* pada graf MCS yang berhubungan dengan KA-8 memiliki lebih sedikit dibandingkan dengan graf MCS lainnya. Hal ini menyebabkan terjadinya peningkatan koherensi topik label klaster KG-1.

b. Analisa Hasil Penggabungan dan Pelabelan Klaster Dataset 2 Skenario 1

Hasil penggabungan dan pelabelan klaster percobaan 3 pada Tabel 4.5 menunjukkan bahwa klaster gabungan KG-5 tersusun atas klaster asli KA-2 dan KA-11. Penggabungan klaster pada klaster gabungan KG-5 dilakukan berdasarkan similaritas graf yang dihitung dari kesamaan struktur graf representasi klaster KA-2 dan graf representasi klaster KA-11. Kesamaan struktur antara kedua graf tersebut disebut sebagai graf *Maximum Common Subgraph* (MCS). Pada perhitungan similaritas antar klaster berbasis graf, graf MCS merepresentasikan konteks kemiripan antara dua klaster. Graf MCS antara klaster KA-2 dan KA-11 dapat dilihat pada Gambar 4.6. Tabel daftar relasi antar kata pada graf MCS antara KA-2 dan KA-11 dapat dilihat di Tabel 7.1. Penjelasan lebih detil mengenai alur pembuatan graf MCS dapat dilihat pada subbab 3.3.3.

Tabel 4.5 Contoh Hasil Penggabungan dan Pelabelan Klaster Percobaan 3 Skenario 1

Klaster Gabungan			**Klaster Asli**			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-5	• network flow algorithm • optimal routing algorithm • network algorithm • linear time algorithm • polynomial time algorithm	0.4997	**KA-2**	fault-tolerant interconnection networks, fault-tolerant interconnection network, binary hypercube networks, efficient interconnection networks, arbitrary interconnection networks	0.4869	0.5184
			KA-11	linear time algorithm, polynomial time algorithm, time algorithm, n algorithm, n log n	0.5500	



Gambar 4.6. Graf MCS Klaster Asli KA-2 dan KA-11 pada Dataset 2 Skenario 1

Koherensi topik label klaster KG-5 lebih rendah daripada rata – rata koherensi topik klaster penyusunnya. Graf MCS antara KA-2 dan KA-11 pada Gambar 4.6 terdapat kata notasi seperti [‘*rm*’], [‘*theta*’], [‘*log*’], [‘*de*’], dan [‘*fv*’]. Kata – kata tersebut seharusnya tidak menjadi representasi similaritas antar klaster karena tidak memiliki konteks. Hal ini disebabkan karena parameter *support* FPM yang tidak sesuai dengan jumlah dokumen artikel ilmiah pada dataset 2 yang besar. Sehingga kata – kata notasi tersebut tidak tersaring pada tahap ekstraksi frasa untuk pembentukan graf klaster. Pada graf MCS antara KA-2 dan KA-11 juga terlihat bahwa kata – kata seperti [‘*permutation*’], [‘*hamiltonian*’], dan [‘*spanning*’] memiliki bobot yang tinggi. Namun sebenarnya kata – kata tersebut memiliki persentase kemunculan yang rendah pada artikel ilmiah di klaster KA-2 dan KA-11. Kata [‘*permutation*’] dalam klaster KA-2 dan KA-11 muncul pada 13.9% dan 3.94% artikel ilmiah. Kata [‘*hamiltonian*’] dalam klaster KA-2 dan KA-11 muncul pada 2.7% dan 1.64% artikel ilmiah. Kata [‘*spanning*’] dalam klaster KA-2 dan KA-11 muncul pada 3.47% dan 2.96% artikel ilmiah. Sehingga kata – kata tersebut kurang tepat untuk merepresentasikan konteks similaritas antar klaster.

Hasil penggabungan dan pelabelan klaster pada percobaan 4 juga menunjukkan permasalahan yang serupa, seperti pada Tabel 4.6. Pada percobaan 4 klaster gabungan KG-4 terbentuk dari klaster asli KA-6 dan KA-14. Similaritas antar kedua graf tersebut

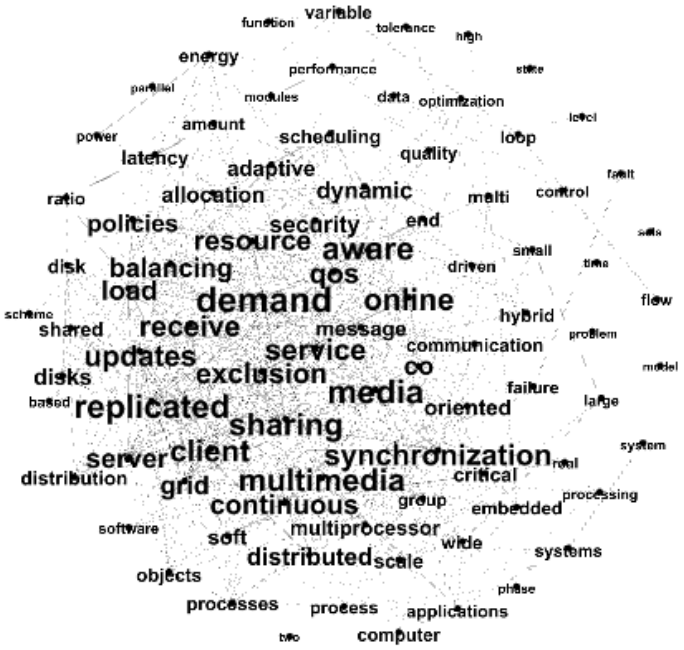
Tabel 4.6 Contoh Hasil Penggabungan dan Pelabelan Klaster Percobaan 4 Skenario 1

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-4	• real-time task systems • real-time control system • distributed real-time systems • real-time distributed systems • real-time system design	0.4471	KA-6	real-time task scheduling, real-time task systems, real-time scheduling algorithm, real-time task model, task scheduling problem	0.4779	0.4929
			KA-14	data management system, data storage system, data management systems, data storage systems, file system performance	0.5079	

direpresentasikan oleh kesamaan struktur graf representasi klaster yang disebut oleh graf *Maximum Common Subgraph* (MCS). Tabel relasi antar kata pada graf MCS antar klaster asli KA-6 dan KA-14 terdapat pada Tabel 7.2. Penjelasan lebih detil mengenai alur pembuatan graf MCS dapat dilihat pada subbab 3.3.3. Graf MCS antara kalster KA-6 dan KA-14 seperti di Gambar 4.7, menunjukkan rendahnya kualitas konteks yang merepresentasikan similaritas antar klaster jika diinterpretasi secara intuitif. Berdasarkan pengamatan yang telah dilakukan pada percobaan 3 dan 4, pembobotan relasi antar kata yang kurang tepat dapat menyebabkan representasi konteks similaritas klaster yang tidak sesuai. Seperti pada hasil skenario 1, hal ini disebabkan karena pembuatan model graf klaster menggunakan relasi antar kata *Word2Vec* yang tidak memperhitungkan frekuensi.

c. Kesimpulan Analisa Hasil Skenario 1

Hasil skenario 1 menunjukkan bahwa pada dataset 1 rata – rata koherensi topik label klaster gabungan lebih baik dari rata – rata koherensi topik klaster asli penyusunnya, dengan pemilihan *threshold* optimal untuk tahap penggabungan klaster pada persentil 5% seluruh jarak antar klaster. Sedangkan untuk dataset 2 pada pemilihan *threshold* penggabungan klaster 10% dan 5% tetap menghasilkan rata – rata koherensi topik label klaster gabungan yang lebih rendah daripada rata – rata koherensi topik label klaster asli penyusunnya. Hal ini dapat disebabkan oleh dataset 2 yang memiliki lebih banyak data bersifat derau daripada dataset 1. Hasil skenario 1 juga menunjukkan bahwa penggunaan *Word2Vec* pada pembuatan model graf klaster yang mengabaikan frekuensi kata dapat menghasilkan graf MCS dengan konteks similaritas yang tidak sesuai.



Gambar 4.7. Graf MCS Klaster Asli KA-6 dan KA-14 pada Dataset 2 Skenario 1

Pada hasil skenario 1 terlihat bahwa dengan menggunakan pesentil yang lebih rendah pada penggabungan klaster maka jumlah klaster gabungan akan semakin banyak. Hal ini disebabkan karena persentil yang rendah pada penentuan *threshold* penggabungan klaster membatasi jumlah klaster yang akan digabung. Berkurangnya jumlah klaster yang digabung menyebabkan jumlah klaster akhir (setelah penggabungan) akan mendekati jumlah klaster asli.

Tabel 4.7. Hasil Rata – Rata Koherensi Topik Skenario 2

Percobaan	**Dataset**	**Threshold Persentil (%)**	**Klaster Asli**		**Klaster Gabungan**	
			Jumlah Klaster	**Rata - Rata Koherensi Topik**	**Jumlah Klaster**	**Rata - Rata Koherensi Topik**
1	1	10	11	0.5011	8	**0.5061**
2		5	11	0.5011	9	**0.5180**
3	2	10	15	0.4513	11	**0.4541**
4		5	15	0.4513	12	**0.4595**

4.4.2 Skenario 2

Konteks similaritas klaster pada dataset 2 di skenario 1 tidak terepresentasikan dengan baik pada graf *Maximum Common Subgraph* (MCS) karena penggunaan relasi *Word2Vec* tidak memperhitungkan frekuensi kata. Pada skenario 2 hasil penggunaan relasi *co-occurrence* dalam pembentukan model graf klaster dievaluasi. Hasil rata – rata koherensi topik label klaster gabungan skenario 2 dapat dilihat pada Tabel 4.7.

a. Analisa Hasil Penggabungan dan Pelabelan Klaster Dataset 1 Skenario 2

Hasil skenario 2 di Tabel 4.7 menunjukkan bahwa pada dataset 1 tidak terjadi perubahan pada nilai rata – rata koherensi topik pada *threshold* persentil 10% maupun 5% jika dibandingkan dengan hasil skenario 1. Nilai koherensi topik yang sama pada dataset 1 disebabkan karena tidak adanya perubahan hasil penggabungan klaster dari skenario 1, seperti yang dapat dilihat pada Tabel 6.1 dan Tabel 6.2. Namun terdapat

Gambar 4.8. Graf MCS Klaster Asli KA-3 dan KA-5 pada Dataset 1 Skenario 2

perubahan pada graf MCS yang dihasilkan. Graf MCS dihasilkan pada proses perhitungan similaritas klaster, dimana kesamaan struktur graf representasi klaster diidentifikasi. Salah satu contoh graf MCS yang dihasilkan pada dataset 1 di skenario 2 adalah graf MCS antara klaster asli KA-4 dan KA-11 di Gambar 4.8. Jika dibandingkan dengan graf MCS

Tabel 4.8 Relasi Kata Pada Graf MCS Klaster Asli KA-3 dan KA-5 Dataset 1 Skenario 2

Vertex Sumber (Kata)	Vertex Target (Kata)	Tipe Relasi Kata	Bobot Edge
network	nodes	Undirected	1
multicast	new	Undirected	2

Tabel 4.10 Contoh Hasil Penggabungan dan Pelabelan Klaster Percobaan 3 Skenario 2

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-1	• data cache performance • memory system performance • data cache memory • cache memory system • data cache access	**0.4560**	KA-7	network time protocol, network performance requirements, interconnection network performance, network performance constraints, overall network performance	0.3321	0.4236
			KA-14	data management system, data storage system, data management systems, data storage systems, file system performance	0.5079	
			KA-15	data cache memory, data cache performance, cache memory system, cache memory design, instruction cache performance	0.4308	

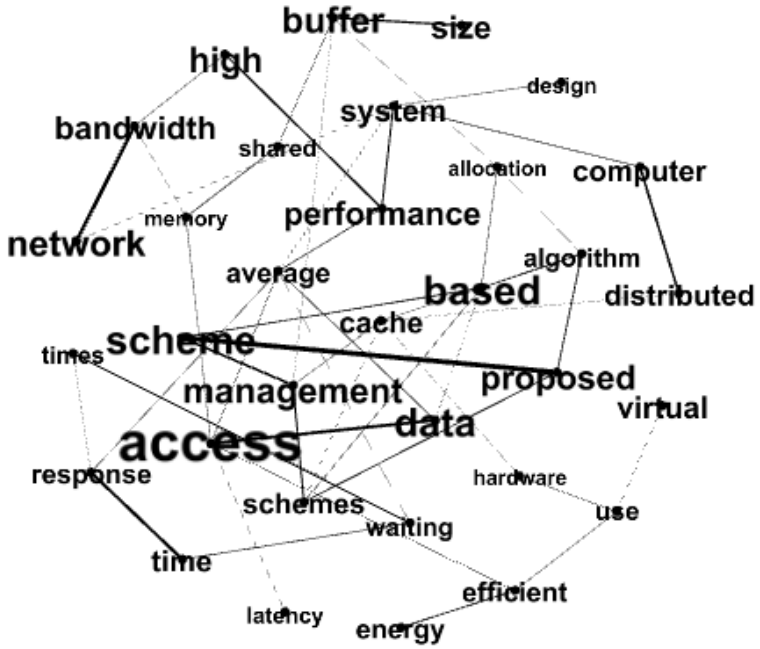
Tabel 4.9 Contoh Hasil Penggabungan dan Pelabelan Klaster Percobaan 4 Skenario 2

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-1	• data cache performance • data cache memory • memory system performance • instruction cache performance • cache memory system	0.4671	KA-7	network time protocol, network performance requirements, interconnection network performance, network performance constraints, overall network performance	0.3321	0.3814
			KA-15	data cache memory, data cache performance, cache memory system, cache memory design, instruction cache performance	0.4308	

pada skenario 1, graf MCS antara klaster asli KA-4 dan KA-11 memiliki relasi kata yang lebih sedikit. Daftar relasi kata yang terbentuk pada graf MCS antara klaster asli KA-4 dan KA-11 terdapat pada Tabel 4.8. Penjelasan lebih detil mengenai alur pembuatan graf MCS dapat dilihat pada subbab 3.3.3.

b. Analisa Hasil Penggabungan dan Pelabelan Klaster Dataset 2 Skenario 2

Pada dataset 2 terjadi peningkatan rata – rata koherensi topik label klaster gabungan dibandingkan dengan rata – rata koherensi topik label klaster pada skenario 1, dengan *threshold* pada persentil 5% menghasilkan hasil terbaik. Hasil penggabungan dan pelabelan klaster pada dataset 2 di Tabel 4.10 dan Tabel 4.9 menunjukkan bahwa

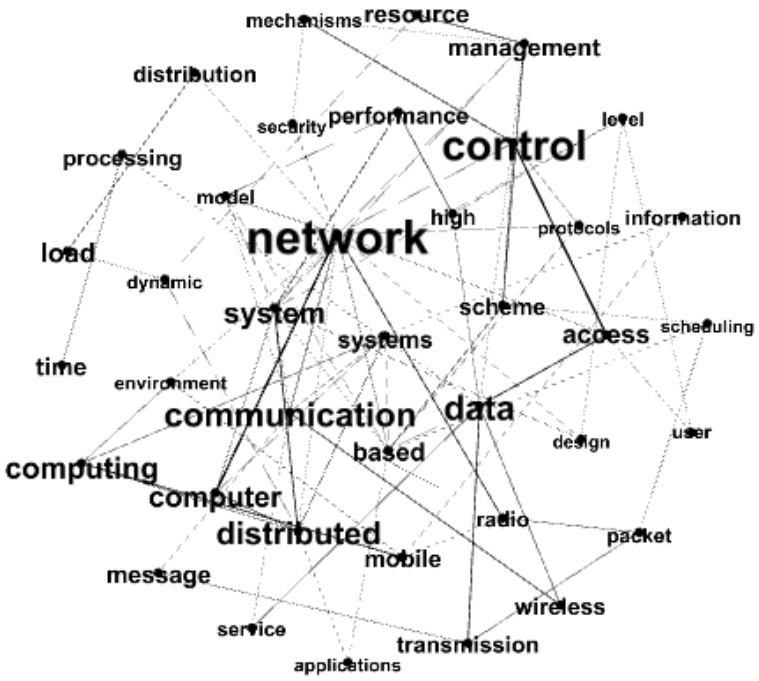


Gambar 4.9. Graf MCS Klaster Asli KA-7 dan KA-15 pada Dataset 2 Skenario 2

hilangnya klaster KA-14 sebagai klaster penyusun klaster gabungan KG-1 menyebabkan peningkatan rata – rata koherensi topik label klaster gabungan.

Pada *threshold* persentil 10%, klaster gabungan KG-1 tersusun atas KA-7, KA-14, dan KA-15. Similaritas antar klaster tersebut diukur berdasarkan kesamaan struktur antara graf representasi klaster. Penjelasan lebih detil tentang pembuatan graf MCS terdapat pada subbab 3.3. Pada dataset 2 skenario 2 graf MCS yang dihasilkan berukuran sangat besar, sehingga untuk kepentingan visualisasi graf MCS di-*filter* dengan hanya menampilkan *vertex* yang memiliki relasi lebih dari 10 kata. Graf MCS yang telah di-*filter* antara klaster – klaster penyusun KG-1 dapat dilihat pada Gambar 4.9, Gambar 4.10, dan Gambar 4.11, tabel daftar relasi graf – graf MCS tersebut dapat dilihat pada Tabel 7.3, Tabel 7.4, dan Tabel 7.5.

Pada graf MCS antar klaster KA-14 dan KA-15 dapat dilihat jumlah *vertex* dan *edges* yang paling sedikit di antara graf MCS lainnya dan bobot *edge* yang hanya terpusat antara kata [‘*file*’], [‘*system*’], dan [’*operating*’]. Sehingga pada *threshold* persentil 5% klaster gabungan KG-1 hanya tersusun atas klaster KA-14 dan KA-15 saja. Graf MCS antara klaster KA-7 dan KA-15 pada Gambar 4.9 menunjukkan bahwa konteks similaritas antar klaster tersebut adalah pengaturan *cache* memori pada jaringan komputer. Konteks



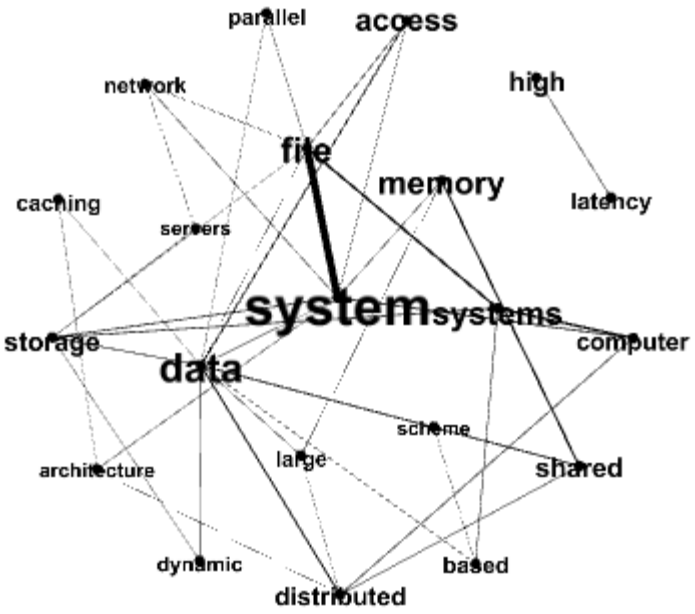
Gambar 4.10. Graf MCS Klaster Asli KA-7 dan KA-14 pada Dataset 2 Skenario 2

similaritas tersebut konsisten dengan konten artikel ilmiah yang terdapat pada kedua klaster seperti dilihat di Tabel 6.9.

c. Kesimpulan Hasil Analisa Skenario 2

Hasil skenario 2 menunjukkan bahwa, jika dibandingkan dengan model graf relasi *Word2Vec*, penggunaan model graf relasi *co-occurrence* memberikan hasil yang sama pada dataset 1 dan hasil yang lebih baik pada dataset 2. Hal tersebut terjadi karena dataset 1 memiliki klaster yang homogen sehingga pengaruh penggunaan model graf *co-occurrence* lebih terlihat pada dataset 2 yang memiliki klaster yang heterogen. Namun berdasarkan pengamatan pada graf MCS yang dihasilkan, graf MCS model graf relasi *co-occurrence* memiliki relasi kata yang lebih sedikit dibandingkan dengan penggunaan model graf relasi *Word2Vec*. Sehingga representasi relasi antar kata pada model graf *co-occurrence* lebih sedikit daripada model graf *Word2Vec*.

Seperti pada skenario 1, jumlah klaster setelah penggabungan akan semakin besar jika *threshold* persentil untuk penggabungan klaster semakin kecil. Hal ini disebabkan karena pengecilan persentil pada *threshold* penggabungan klaster menyebabkan jumlah klaster asli yang digabung semakin sedikit. Sehingga jumlah klaster setelah penggabungan akan mendekati jumlah klaster asli.



Gambar 4.11. Graf MCS Klaster Asli KA-14 dan KA-15 pada Dataset 2 Skenario 2

4.4.3 Skenario 3

Pada skenario 3 hasil usulan metode dibandingkan dengan nilai parameter *support* pada metode *Frequent Phrase Mining* (FPM) di tahap ekstraksi kandidat frasa. Parameter tersebut mengatur ukuran graf klaster yang dihasilkan dengan membatasi jumlah kata yang menjadi *vertex* pada pembuatan graf klaster. Sehingga skenario 3 bertujuan untuk mengamati pengaruh ukuran graf terhadap koherensi topik label klaster.

Tabel 4.11 Hasil Skenario 3

Pengujian	**Data set**	**FPM *support***	**Rata-Rata Ukuran Graf Klaster Asli**		**Rata-Rata Ukuran Graf MCS**		**Jumlah *Vertex* Gmcs (terkecil, terbesar)**	**Rata-Rata Koherensi Topik Label**	
			Jumlah *vertex*	**Jumlah *edge***	**Jumlah *vertex***	**Jumlah *edge***		**Klaster Asli**	**Klaster Gabungan**
1	1	3	72	1.123	5,29 % (4 *vertex*)	0,15 % (2 *edge*)	(0 , 11)	0.501	**0.506**
2		2	203	7.137	12,65 % (26 vertex)	0,60 % (43 *edge*)	(8 , 45)	0.501	**0.519**
3	2	3	379	34.305	18,89 % (72 vertex)	1,40 % (479 *edge*)	(21 , 211)	0,449	**0,452**
4		2	422	40.051	33,88% (143 vertex)	3,91% (1.565 *edge*)	(26, 247)	**0,449**	0,441

Hasil skenario 3 dapat dilihat pada Tabel 4.11. Pada tabel tersebut terlihat bahwa graf MCS dalam Pengujian 1 sangat kecil dengan rata – rata persentase ukuran graf MCS

hanya 5,29% untuk jumlah *vertex* dan 0,15% untuk jumlah *edge*. Artinya rata – rata jumlah *vertex* graf MCS hanya 4 *vertex* dan rata – rata jumlah *edge* graf MCS hanya 2 *edge*. Kecilnya ukuran graf MCS menyebabkan jumlah informasi semantik yang digunakan semakin sedikit. Informasi semantik tersebut tersimpan dalam relasi kata pada graf MCS. Oleh karena itu, pada Pengujian 2 ukuran graf klaster asli diperbesar dengan mengatur parameter *support* FPM menjadi 2. Penurunan parameter *support* FPM menyebabkan banyaknya kata yang akan digunakan dalam pembuatan graf klaster semakin tinggi. Peningkatan ukuran graf klaster menyebabkan peningkatan persentase ukuran graf MCS yang dihasilkan.

Perbandingan ukuran graf pada Tabel 4.11 menunjukkan bahwa penurunan nilai *support* FPM meningkatkan persentase ukuran graf MCS. Peningkatan persentase ukuran graf MCS juga diikuti dengan peningkatan koherensi topik label klaster gabungan pada dataset 1, tercermin pada hasil pengujian 1 dan 2. Namun pada dataset 2 peningkatan persentase ukuran graf MCS menyebabkan penurunan koherensi topik label klaster gabungan, tercermin pada hasil pengujian 3 dan 4. Sehingga, koherensi topik label klaster akan menurun jika ukuran graf yang dihasilkan terlalu besar, seperti pada dataset 2.

BAB 5
PENUTUP

Berdasarkan pengujian pada metode pelabelan klaster artikel ilmiah yang telah diusulkan, dapat ditarik beberapa kesimpulan dan saran penelitian yang akan dilakukan selanjutnya.

5.1 Kesimpulan

Penelitian ini mengusulkan metode pelabelan klaster artikel ilmiah dengan penggabungan klaster berdasarkan similaritas graf. Perhitungan similaritas graf menggunakan model graf sebagai representasi teks. Hasil pengujian usulan metode menunjukkan bahwa model graf dengan pendekatan *co-occurrence* menghasilkan koherensi topik label klaster yang lebih baik dibandingkan dengan pendekatan *Word2Vec*. Namun, model graf dengan pendekatan *Word2Vec* mampu mengekstrak relasi kata dengan kuantitas yang lebih banyak daripada dengan pendekatan *co-occurrence*.

Perhitungan similaritas klaster menggunakan graf *Maximum Common Subgraph* (MCS) dapat mengidentifikasi klaster artikel ilmiah yang memiliki kemiripan topik. Hal ini ditunjukkan dengan hasil koherensi topik label klaster gabungan yang lebih tinggi dibandingkan dengan label klaster asli. Perbandingan konteks graf MCS dengan konteks artikel ilmiah pada klaster juga dilakukan untuk memastikan konsistensi konteks similaritas klaster. Namun, identifikasi klaster artikel ilmiah yang memiliki kemiripan topik dipengaruhi oleh penentuan *threshold* penggabungan dan pendekatan dalam menentukan relasi kata. Penentuan *threshold* dengan menggunakan persentil ke-5% terhadap seluruh jarak antar klaster menghasilkan hasil yang lebih baik secara kuantitatif dan kualitatif dibandingkan dengan persentil ke-10%.

Pengujian dilakukan pada 2 dataset yang memiliki perbedaan karakteristik. Dataset 1 bersifat homogen ditandai dengan rata – rata nilai *silhouette* yang tinggi dibandingkan dataset 2. Koherensi topik label klaster gabungan terbaik dihasilkan oleh dataset 1, meskipun dataset 2 memiliki jumlah klaster yang lebih besar daripada dataset 1. Hal ini menunjukkan bahwa homogenitas dan tingkat derau dataset juga mempengaruhi hasil pelabelan dan penggabungan klaster.

Pada seluruh skenario pengujian terlihat bahwa semakin kecil *threshold* persentil yang digunakan malah jumlah klaster setelah penggabungan akan semakin

banyak (namun tidak melebihi jumlah klaster sebelum penggabungan). Hal ini disebabkan karena penggunaan *threshold* yang lebih kecil (persentil rendah) maka jumlah klaster yang digabung semakin sedikit, sehingga jumlah akhir klaster setelah proses penggabungan akan lebih banyak (mendekati jumlah klaster sebelum proses penggabungan).

5.2 Saran

Pada metode penggabungan klaster berbasis graf yang diusulkan, penggunaan model graf relasi *Word2Vec* tidak memperhatikan frekuensi kata sehingga dapat menghasilkan konteks similaritas klaster yang kurang tepat. Namun penggunaan model graf relasi *co-occurrence* menghasilkan informasi semantik, direpresentasikan oleh relasi kata, yang lebih sedikit. Sehingga pada penelitian selanjutnya akan dikembangkan pembobotan relasi kata pada model graf yang menggabungkan informasi semantik dari *Word2Vec* dan frekuensi kata dari *co-occurence*.

DAFTAR PUSTAKA

[1] Z. Li, J. Li, Y. Liao, S. Wen, and J. Tang, "Labeling clusters from both linguistic and statistical perspectives: A hybrid approach," *Knowledge-Based Syst.*, vol. 76, pp. 219–227, 2015.

[2] D. Carmel, H. Roitman, and N. Zwerdling, "Enhancing cluster labeling using wikipedia," *Proc. 32nd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '09*, no. January 2016, pp. 139–146, 2009.

[3] P. Lopez and L. Romary, "HUMB : Automatic Key Term Extraction from Scientific Articles in GROBID," *Proc. 5th Int. Work. Semant. Eval.*, no. July, pp. 248–251, 2010.

[4] P. Lopez and L. Romary, "GRISP : A Massive Multilingual Terminological Database for Scientific and Technical Domains," *Knowl. Creat. Diffus. Util.*, pp. 2269–2276, 2010.

[5] A. El-Kishky, Y. Song, C. Wang, C. Voss, and J. Han, "Scalable Topical Phrase Mining from Text Corpora," *Proc. VLDB Endow.*, vol. 8, no. 3, pp. 305–316, 2014.

[6] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," *J. Comput. Sci. Technol.*, vol. 15, no. 6, pp. 487–499, 1994.

[7] C. Aalla and V. Pudi, "Mining Research Problems from Scientific Literature," *2016 IEEE Int. Conf. Data Sci. Adv. Anal.*, pp. 351–360, 2016.

[8] K. S. Hasan and V. Ng, "Automatic Keyphrase Extraction: A Survey of the State of the Art," *Proc. 52nd Annu. Meet. Assoc. Comput. Linguist. (Volume 1 Long Pap.*, pp. 1262–1273, 2014.

[9] L. H. Suadaa and A. Purwarianti, "Combination of Latent Dirichlet Allocation (LDA) and Term Frequency-Inverse Cluster Frequency (TFxICF) in Indonesian text clustering with labeling," *2016 4th Int. Conf. Inf. Commun. Technol. ICoICT 2016*, vol. 4, no. c, 2016.

[10] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," *Proc. EMNLP*, vol. 85, pp. 404–411, 2004.

[11] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," *World Wide Web Internet Web Inf. Syst.*, vol. 54, no. 1999–66, pp. 1–17, 1998.

[12] X. Wan and J. Xiao, “CollabRank: towards a collaborative approach to single-document keyphrase extraction,” *Proc. 22nd Int. Conf. Comput. Linguist. Coling 2008*, no. August, pp. 969–976, 2008.

[13] Z. Liu, W. Huang, Y. Zheng, and M. Sun, “Automatic Keyphrase Extraction via Topic Decomposition,” *Comput. Linguist.*, no. October, pp. 366–376, 2010.

[14] L. Sterckx, T. Demeester, J. Deleu, and C. Develder, “Topical Word Importance for Fast Keyphrase Extraction,” *Proc. 24th Int. Conf. World Wide Web - WWW ’15 Companion*, no. 2, pp. 121–122, 2015.

[15] Q. Mei, X. Shen, and C. Zhai, “Automatic labeling of multinomial topic models,” *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD ’07*, p. 490, 2007.

[16] N. Y. Saiyad, H. B. Prajapati, and V. K. Dabhi, “A Survey of Document Clustering using Semantic Approach,” *Int. Conf. Electr. Electron. Optim. Tech.*, vol. 6, no. 4, pp. 2555–2562, 2016.

[17] L. Xiong, “Survey on text clustering algorithm,” *2011 IEEE 2nd Int. Conf. Softw. Eng. Serv. Sci.*, no. 4, pp. 901–904, 2011.

[18] A. Krauza, “Extension of fuzzy Gustafson-Kessel algorithm based on adaptive cluster merging,” *2015 IEEE MIT Undergrad. Res. Technol. Conf. URTC 2015*, pp. 0–3, 2016.

[19] F. De Morsier, D. Tuia, M. Borgeaud, V. Gass, and J. P. Thiran, “Cluster validity measure and merging system for hierarchical clustering considering outliers,” *Pattern Recognit.*, vol. 48, no. 4, pp. 1474–1485, 2015.

[20] I. Czarnowski, P. Jędrzejowicz, and I. Member, “Consensus-based Cluster Merging for the Prototype Selection,” in *2013 IEEE International Conference on Cybernetics*, 2013.

[21] S. Sonawane and P. Kulkarni, “Graph based Representation and Analysis of Text Document: A Survey of Techniques,” *Int. J. Comput. Appl.*, vol. 96, no. 19, pp. 1–8, 2014.

[22] F. Role and M. Nadif, “Beyond cluster labeling: Semantic interpretation of clusters’ contents using a graph representation,” *Knowledge-Based Syst.*, vol. 56, pp. 141–155, 2014.

[23] A. Perer and B. Shneiderman, “Integrating Statistics and Visualization: Case

Studies of Gaining Clarity during Exploratory Data Analysis," *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, p. 265, 2008.

[24] C. Jin and Q. Bai, "Text Clustering Algorithm Based on the Graph Structures of Semantic Word Co- occurrence," *2016 Int. Conf. Inf. Syst. Artif. Intell.*, 2016.

[25] D. Arthur and S. Vassilvitskii, "K-Means++: the Advantages of Careful Seeding," *Proc. eighteenth Annu. ACM-SIAM Symp. Discret. algorithms*, vol. 8, pp. 1027–1025, 2007.

[26] J. De Knijff, F. Frasincar, and F. Hogenboom, "Data & Knowledge Engineering Domain taxonomy learning from text : The subsumption method versus hierarchical clustering," *Data Knowl. Eng.*, vol. 83, no. 0, pp. 54–69, 2013.

[27] S. K. Popat and M. Emmanuel, "Review and Comparative Study of Clustering Techniques," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 1, pp. 805–812, 2014.

[28] J. Jayabharathy, S. Kanmani, and A. A. Parveen, "Document clustering and topic discovery based on semantic similarity in scientific literature," *Commun. Softw. Networks (ICCSN), 2011 IEEE 3rd Int. Conf.*, pp. 425–429, 2011.

[29] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Proc. Int. Conf. Learn. Represent. (ICLR 2013)*, pp. 1–12, 2013.

[30] M. Röder, A. Both, and A. Hinneburg, "Exploring the Space of Topic Coherence Measures," *Proc. Eighth ACM Int. Conf. Web Search Data Min. - WSDM '15*, pp. 399–408, 2015.

[Halaman ini sengaja dikosongkan]

LAMPIRAN PENGGABUNGAN DAN PELABELAN KLASTER

a. Hasil Skenario 1

Tabel 6.1 Hasil Penggabungan dan Pelabelan Klaster pada Percobaan 1 dalam Skenario 1

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-1	• test generation algorithm • test generation time • new fault model • test application time • system fault diagnosis	0.5302	KA-4	test pattern generation, test generator circuit, test generation method, test generation time, test generation algorithm	0.4687	0.4890
			KA-8	polynomial time algorithm, n algorithm, n log n, time algorithm, best such algorithm	0.5654	
			KA-11	system fault diagnosis, new fault model, fault injection techniques, transient fault tolerance, rtl fault model	0.4329	
KG-2	• ternary logic functions • universal logic functions • multiple-valued logic functions • fuzzy logic functions • b-ternary logic functions	0.3844	KA-6	ternary logic functions, universal logic functions, multiple-valued logic functions, fuzzy logic functions, b-ternary logic functions	0.3844	0.3844
KG-3	• new multicast network • new interconnection network • wireless data networks • fault-tolerant interconnection network • multicast network	0.5737	KA-3	fault-tolerant interconnection networks, virtual interconnection networks, arbitrary interconnection networks, multistage interconnection networks, fault-tolerant interconnection network	0.6101	0.5503
			KA-5	wireless data networks, network performance, wireless data broadcast, network traffic, network weather service	0.4905	
KG-4	• instruction cache performance • data cache performance • cache memory designs • data cache access • instruction cache design	0.4939	KA-7	instruction cache performance, data cache performance, cache memory designs, data cache access, instruction cache design	0.4939	0.4939
KG-5	• periodic real-time tasks • task scheduling problem • periodic task systems • real-time task model • dynamic scheduling algorithm	0.4744	KA-9	periodic real-time tasks, task scheduling problem, periodic task systems, real-time task model, dynamic scheduling algorithm	0.4744	0.4744
KG-6	• system software • operating system design • computer-aided design system • system requirements engineering • system design guide	0.5025	KA-1	system software, operating system design, computer-aided design system, system requirements engineering, system design guide	0.5025	0.5025

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-7	• systolic modular multiplication • normal basis multiplication • n-bit modular multiplication • higher radix multiplication • finite field multiplication	0.5797	KA-10	systolic modular multiplication, normal basis multiplication, n-bit modular multiplication, higher radix multiplication, finite field multiplication	0.5797	0.5797
KG-8	• logic design technique • logic design techniques • computer-aided logic design • logic design • fuzzy logic circuits	0.5099	KA-2	logic design technique, logic design techniques, computer-aided logic design, logic design, fuzzy logic circuits	0.5099	0.5099
	Rata - rata Koherensi Topik	0.5061			0.5011	

Tabel 6.2 Hasil Penggabungan dan Pelabelan Klaster pada Percobaan 2 dalam Skenario 1

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-1	• test generation algorithm • test generation time • new fault model • test application time • system fault diagnosis	0.5786	KA-4	test pattern generation, test generator circuit, test generation method, test generation time, test generation algorithm	0.4687	0.4508
			KA-11	system fault diagnosis, new fault model, fault injection techniques, transient fault tolerance, rtl fault model	0.4329	
KG-2	• polynomial time algorithm • n algorithm • n log n • time algorithm • best such algorithm	0.5654	KA-8	polynomial time algorithm, n algorithm, n log n, time algorithm, best such algorithm	0.5654	0.5654
KG-3	• ternary logic functions • universal logic functions • multiple-valued logic functions • fuzzy logic functions • b-ternary logic functions	0.3844	KA-6	ternary logic functions, universal logic functions, multiple-valued logic functions, fuzzy logic functions, b-ternary logic functions	0.3844	0.3844
KG-4	• new multicast network • new interconnection network • wireless data networks • fault-tolerant interconnection network • multicast network	0.5737	KA-3	fault-tolerant interconnection networks, virtual interconnection networks, arbitrary interconnection networks, multistage interconnection networks, fault-tolerant interconnection network	0.6101	0.5503
			KA-5	wireless data networks, network performance, wireless data broadcast, network traffic, network weather service	0.4905	
KG-5	• instruction cache performance • data cache performance • cache memory designs • data cache access • instruction cache design	0.4939	KA-7	instruction cache performance, data cache performance, cache memory designs, data cache access, instruction cache design	0.4939	0.4939

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-6	• periodic real-time tasks • task scheduling problem • periodic task systems • real-time task model • dynamic scheduling algorithm	0.4744	KA-9	periodic real-time tasks, task scheduling problem, periodic task systems, real-time task model, dynamic scheduling algorithm	0.4744	0.4744
KG-7	• system software • operating system design • computer-aided design system • system requirements engineering • system design guide	0.5025	KA-1	system software, operating system design, computer-aided design system, system requirements engineering, system design guide	0.5025	0.5025
KG-8	• systolic modular multiplication • normal basis multiplication • n-bit modular multiplication • higher radix multiplication • finite field multiplication	0.5797	KA-10	systolic modular multiplication, normal basis multiplication, n-bit modular multiplication, higher radix multiplication, finite field multiplication	0.5797	0.5797
KG-9	• logic design technique • logic design techniques • computer-aided logic design • logic design • fuzzy logic circuits	0.5099	KA-2	logic design technique, logic design techniques, computer-aided logic design, logic design, fuzzy logic circuits	0.5099	0.5099
	Rata - rata Koherensi Topik	0.5180			0.5011	

Tabel 6.3 Hasil Penggabungan dan Pelabelan Klaster pada Percobaan 3 dalam Skenario 1

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-1	• system fault diagnosis • system fault tolerance • computer system design • hardware system design • system design techniques	0.4214	KA-4	system fault diagnosis, fault diagnosis techniques, new fault model, fault diagnosis algorithm, multiprocessor fault diagnosis	0.4165	0.4804
			KA-13	computer system design, digital system design, hardware system design, digital systems design, hardware systems design	0.5442	
KG-2	• memory system performance • data cache performance • cache memory system • data cache memory • file system performance	0.4597	KA-6	real-time task scheduling, real-time task systems, real-time scheduling algorithm, real-time task model, task scheduling problem	0.4779	0.4371
			KA-7	network time protocol, network performance requirements, interconnection network performance, network performance constraints, overall network performance	0.3321	
			KA-14	data management system, data storage system, data management systems, data storage systems, file system performance	0.5079	

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
			KA-15	data cache memory, data cache performance, cache memory system, cache memory design, instruction cache performance	0.4308	
KG-3	• new search algorithm • flow analysis algorithm • new exact algorithm • new heuristic algorithm • new data structure	0.4000	KA-8	new search algorithm, flow analysis algorithm, new exact algorithm, new heuristic algorithm, new data structure	0.4000	0.4000
KG-4	• parallel instruction execution • program memory performance • processor instruction execution • parallel processor performance • machine code instructions	0.3997	KA-1	parallel instruction execution, program memory performance, processor instruction execution, parallel processor performance, machine code instructions	0.3997	0.3997
KG-5	• network flow algorithm • optimal routing algorithm • network algorithm • linear time algorithm • polynomial time algorithm	0.4997	KA-2	fault-tolerant interconnection networks, fault-tolerant interconnection network, binary hypercube networks, efficient interconnection networks, arbitrary interconnection networks	0.4869	0.5184
			KA-11	linear time algorithm, polynomial time algorithm, time algorithm, n algorithm, n log n	0.5500	
KG-6	• single fault test • delay fault test • fault detection test • sequential test generation • sequential circuit test	0.3732	KA-3	fault detection test, single fault test, delay fault test, test set generation, test pattern generation	0.3660	0.4100
			KA-9	general boolean functions, arbitrary boolean functions, multiple-valued logic functions, other boolean functions, universal boolean functions	0.4541	
KG-7	• logic design level • fpga architecture design • threshold logic design • combinational logic design • logic design effort	0.4953	KA-10	logic design level, fpga architecture design, threshold logic design, combinational logic design, logic design effort	0.4953	0.4953
KG-8	• binary multiplication algorithm • modular multiplication algorithm • complex multiplication algorithm • montgomery multiplication algorithm • bit-serial multiplication algorithm	0.4576	KA-12	binary multiplication algorithm, modular multiplication algorithm, complex multiplication algorithm, montgomery multiplication algorithm, bit-serial multiplication algorithm	0.4576	0.4576
KG-9	• unidirectional error codes • error control codes • arithmetic error codes • error correcting codes • error codes	0.4511	KA-5	unidirectional error codes, error control codes, arithmetic error codes, error correcting codes, error codes	0.4511	0.4511
	Rata - rata Koherensi Topik	0.4397			0.4513	

Tabel 6.4 Hasil Penggabungan dan Pelabelan Klaster pada Percobaan 4 dalam Skenario 1

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-1	• system fault diagnosis • fault diagnosis techniques • new fault model • fault diagnosis algorithm • multiprocessor fault diagnosis	0.4165	KA-4	system fault diagnosis, fault diagnosis techniques, new fault model, fault diagnosis algorithm, multiprocessor fault diagnosis	0.4165	0.4165
KG-2	• computer system design • digital system design • hardware system design • digital systems design • hardware systems design	0.5442	KA-13	computer system design, digital system design, hardware system design, digital systems design, hardware systems design	0.5442	0.5442
KG-3	• data cache performance • data cache memory • memory system performance • instruction cache performance • cache memory system	0.4671	KA-7	network time protocol, network performance requirements, interconnection network performance, network performance constraints, overall network performance	0.3321	0.3814
			KA-15	data cache memory, data cache performance, cache memory system, cache memory design, instruction cache performance	0.4308	
KG-4	• real-time task systems • real-time control system • distributed real-time systems • real-time distributed systems • real-time system design	0.4471	KA-6	real-time task scheduling, real-time task systems, real-time scheduling algorithm, real-time task model, task scheduling problem	0.4779	0.4929
			KA-14	data management system, data storage system, data management systems, data storage systems, file system performance	0.5079	
KG-5	• new search algorithm • flow analysis algorithm • new exact algorithm • new heuristic algorithm • new data structure	0.4000	KA-8	new search algorithm, flow analysis algorithm, new exact algorithm, new heuristic algorithm, new data structure	0.4000	0.4000
KG-6	• parallel instruction execution • program memory performance • processor instruction execution • parallel processor performance • machine code instructions	0.3997	KA-1	parallel instruction execution, program memory performance, processor instruction execution, parallel processor performance, machine code instructions	0.3997	0.3997
KG-7	• network flow algorithm • optimal routing algorithm • network algorithm • linear time algorithm • polynomial time algorithm	0.4997	KA-2	fault-tolerant interconnection networks, fault-tolerant interconnection network, binary hypercube networks, efficient interconnection networks, arbitrary interconnection networks	0.4869	0.5184

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
			KA-11	linear time algorithm, polynomial time algorithm, time algorithm, n algorithm, n log n	0.5500	
KG-8	• single fault test • delay fault test • fault detection test • sequential test generation • sequential circuit test	0.3732	KA-3	fault detection test, single fault test, delay fault test, test set generation, test pattern generation	0.3660	0.4100
			KA-9	general boolean functions, arbitrary boolean functions, multiple-valued logic functions, other boolean functions, universal boolean functions	0.4541	
KG-9	• logic design level • fpga architecture design • threshold logic design • combinational logic design • logic design effort	0.4953	KA-10	logic design level, fpga architecture design, threshold logic design, combinational logic design, logic design effort	0.4953	0.4953
KG-10	• binary multiplication algorithm • modular multiplication algorithm • complex multiplication algorithm • montgomery multiplication algorithm • bit-serial multiplication algorithm	0.4576	KA-12	binary multiplication algorithm, modular multiplication algorithm, complex multiplication algorithm, montgomery multiplication algorithm, bit-serial multiplication algorithm	0.4576	0.4576
KG-11	• unidirectional error codes • error control codes • arithmetic error codes • error correcting codes • error codes	0.4511	KA-5	unidirectional error codes, error control codes, arithmetic error codes, error correcting codes, error codes	0.4511	0.4511
	Rata - rata Koherensi Topik	0.4501			0.4513	

Tabel 6.5 Contoh Artikel Ilmiah pada Klaster KA-4 dan KA-8 di Dataset 1 Skenario1

Klaster KA-4		**Klaster KA-8**	
Judul	**Abstrak**	**Judul**	**Abstrak**
The Ballast Methodology for Structured Partial Scan Design	An efficient partial scan technique called Ballast (balanced structure scant test) is presented. Scan **path** storage elements (SPSEs) are selected such that the remainder of the circuit has certain desirable testability properties. A complete test set is obtained using combinatorial automatic test pattern generation (ATPG). Some SPSEs may need to be provided with a HOLD mode; their number is minimized by **ordering** the registers in the scan path and formatting the test patterns appropriately.	Fast Approximation Algorithms on Maxcut, k-Coloring, and k-Color **Ordering** for VLSI Applications	..... Along the line, we first propose a linear-time approximation algorithm on maxcut and two closely related problems: k-coloring and maximal k-color **ordering** problem. The k-coloring is a generalization of the maxcut and the maximal k-color **ordering** is a generalization of the k-coloring. For a graph G with e edges and n vertices .....
Single-Clock Partial Scan	..... In this article, we lift this assumption and address the problems of test generation, scan flip-flop selection and **ordering** of scan registers for partial scan designs that use the system clock for the scan operation. An existing test generation algorithm is modified to incorporate the scan-shifting concept for such design	Rectilinear shortest paths with rectangular barriers	We address ourselves to an instance of the Shortest **Path** problem with obstacles where a shortest path in the Manhattan (or L1) distance is sought between two points (source and destination) and the obstacles are n disjoint rectangles with sides parallel to the coordinate axes .....
A method of test generation for fault location in combinational logic	The **Path** Generating Method is a simple procedure to obtain, from a directed graph, an irredundant set of **paths** that is sufficient to detect and isolate all distinguishable failures. It was developed as a tool for diagnostic generation at the system level, e.g., to test data **paths** and register loading and to test a sequence of transfer instructions	A Note on the Complexity of Dijkstra's Algorithm for Graphs with Weighted Vertices	Let ${\cal G}(V,\ E)$ be a directed graph in which each vertex has a nonnegative weight. The cost of a **path** between two vertices in $\cal G$ is the sum of the weights of the vertices on that **path.** In this paper, we show that, for such graphs, the time complexity of Dijkstra's algorithm, implemented with a binary heap, is ${\cal O}(\|E\| + \|V\|\ \log\ \|V\|).$

Tabel 6.6 Contoh Artikel Ilmiah pada Klaster KA-8 dan KA-11 di Dataset 1 Skenario1

Klaster KA-8		**Klaster KA-11**	
Judul	**Abstrak**	**Judul**	**Abstrak**
Parallelism and **greedy algorithms**	A number of **greedy algorithms** are examined and are shown to be probably inherently sequential. **Greedy algorithms** are presented for finding a maximal path, for finding a maximal set of disjoint paths in a layered dag, and for finding the largest induced subgraph of a graph that has all vertices of degree at least k. It is shown that for all of these **algorithms,** the problem of determining if a given node is in the solution set of the **algorithm** is P-complete. This means that it is unlikely that these sequential **algorithms** can be sped up significantly using parallelism.	**Greedy** Diagnosis as the Basis of an Intermittent-Fault/ Transient-Upset Tolerant System Design	.... Designs for such systems, which exploit a new so-called **greedy** diagnosis theory, are developed. Using **greedy** diagnosis, assessments on the condition of a unit (intermittent-fault case) or the integrity of data (transient-upset case) can be made on the basis of syndromes formed from comparisons of the results of jobs performed by pairs of units. **Greedy** diagnosis avoids the requirement that for such syndromes to be useful, they must be interpretable from a permanent-fault/continuous-upset perspective.

b. Hasil Skenario 2

Tabel 6.7 Hasil Penggabungan dan Pelabelan Klaster pada Percobaan 3 dalam Skenario 2

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-1	• data cache performance • memory system performance • data cache memory • cache memory system • data cache access	**0.4560**	KA-7	network time protocol, network performance requirements, interconnection network performance, network performance constraints, overall network performance	0.3321	0.4236
			KA-14	data management system, data storage system, data management systems, data storage systems, file system performance	0.5079	
			KA-15	data cache memory, data cache performance, cache memory system, cache memory design, instruction cache performance	0.4308	
KG-2	• real-time scheduling algorithm • linear time algorithm • polynomial time algorithm • processing time algorithm • optimal real-time algorithm	0.4693	KA-6	real-time task scheduling, real-time task systems, real-time scheduling algorithm, real-time task model, task scheduling problem	0.4779	**0.5140**
			KA-11	linear time algorithm, polynomial time algorithm, time algorithm, n algorithm, n log n	0.5500	
KG-3	• fault-tolerant interconnection networks • fault-tolerant interconnection network • binary hypercube networks • efficient interconnection networks • arbitrary interconnection networks	0.4869	KA-2	fault-tolerant interconnection networks, fault-tolerant interconnection network, binary hypercube networks, efficient interconnection networks, arbitrary interconnection networks	0.4869	0.4869
KG-4	• new search algorithm • flow analysis algorithm • new exact algorithm • new heuristic algorithm • new data structure	0.4000	KA-8	new search algorithm, flow analysis algorithm, new exact algorithm, new heuristic algorithm, new data structure	0.4000	0.4000
KG-5	• binary multiplication algorithm • modular multiplication algorithm • complex multiplication algorithm • montgomery multiplication algorithm • bit-serial multiplication algorithm	0.4576	KA-12	binary multiplication algorithm, modular multiplication algorithm, complex multiplication algorithm, montgomery multiplication algorithm, bit-serial multiplication algorithm	0.4576	0.4576
KG-6	• fault detection test • single fault test • delay fault test • fault test verification • test generation system	0.3810	KA-3	fault detection test, single fault test, delay fault test, test set generation, test pattern generation	0.3660	**0.3912**
			KA-4	system fault diagnosis, fault diagnosis techniques, new fault model, fault diagnosis algorithm, multiprocessor fault diagnosis	0.4165	

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-7	• logic design level • fpga architecture design • threshold logic design • combinational logic design • logic design effort	0.4953	KA-10	logic design level, fpga architecture design, threshold logic design, combinational logic design, logic design effort	0.4953	0.4953
KG-8	• general boolean functions • arbitrary boolean functions • multiple-valued logic functions • other boolean functions • universal boolean functions	0.4541	KA-9	general boolean functions, arbitrary boolean functions, multiple-valued logic functions, other boolean functions, universal boolean functions	0.4541	0.4541
KG-9	• parallel instruction execution • program memory performance • processor instruction execution • parallel processor performance • machine code instructions	0.3997	KA-1	parallel instruction execution, program memory performance, processor instruction execution, parallel processor performance, machine code instructions	0.3997	0.3997
KG-10	• computer system design • digital system design • hardware system design • digital systems design • hardware systems design	0.5442	KA-13	computer system design, digital system design, hardware system design, digital systems design, hardware systems design	0.5442	0.5442
KG-11	• unidirectional error codes • error control codes • arithmetic error codes • error correcting codes • error codes	0.4511	KA-5	unidirectional error codes, error control codes, arithmetic error codes, error correcting codes, error codes	0.4511	0.4511
	Rata - rata Koherensi Topik	0.4541			0.4513	

Tabel 6.8 Hasil Penggabungan dan Pelabelan Klaster pada Percobaan 4 dalam Skenario 2

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-1	• data cache performance • data cache memory • memory system performance • instruction cache performance • cache memory system	0.4671	KA-7	network time protocol, network performance requirements, interconnection network performance, network performance constraints, overall network performance	0.3321	0.3814
			KA-15	data cache memory, data cache performance, cache memory system, cache memory design, instruction cache performance	0.4308	
KG-2	• data management system • data storage system • data management systems • data storage systems • file system performance	0.5079	KA-14	data management system, data storage system, data management systems, data storage systems, file system performance	0.5079	0.5079
KG-3	• real-time scheduling algorithm • linear time algorithm • polynomial time algorithm • processing time algorithm • optimal real-time algorithm	0.4693	KA-6	real-time task scheduling, real-time task systems, real-time scheduling algorithm, real-time task model, task scheduling problem	0.4779	0.5140
			KA-11	linear time algorithm, polynomial time algorithm, time algorithm, n algorithm, n log n	0.5500	
KG-4	• fault-tolerant interconnection networks • fault-tolerant interconnection network • binary hypercube networks • efficient interconnection networks • arbitrary interconnection networks	0.4869	KA-2	fault-tolerant interconnection networks, fault-tolerant interconnection network, binary hypercube networks, efficient interconnection networks, arbitrary interconnection networks	0.4869	0.4869
KG-5	• new search algorithm • flow analysis algorithm • new exact algorithm • new heuristic algorithm • new data structure	0.4000	KA-8	new search algorithm, flow analysis algorithm, new exact algorithm, new heuristic algorithm, new data structure	0.4000	0.4000
KG-6	• binary multiplication algorithm • modular multiplication algorithm • complex multiplication algorithm • montgomery multiplication algorithm • bit-serial multiplication algorithm	0.4576	KA-12	binary multiplication algorithm, modular multiplication algorithm, complex multiplication algorithm, montgomery multiplication algorithm, bit-serial multiplication algorithm	0.4576	0.4576
KG-7	• fault detection test • single fault test • delay fault test • fault test verification • test generation system	0.3810	KA-3	fault detection test, single fault test, delay fault test, test set generation, test pattern generation	0.3660	0.3912
			KA-4	system fault diagnosis, fault diagnosis techniques, new fault model, fault diagnosis algorithm, multiprocessor fault diagnosis	0.4165	

Klaster Gabungan			Klaster Asli			
ID Klaster	**Label Klaster**	**Koherensi Topik**	**ID Klaster**	**Label Klaster**	**Koherensi Topik**	**Koherensi Topik Gabungan**
KG-8	• logic design level • fpga architecture design • threshold logic design • combinational logic design • logic design effort	0.4953	KA-10	logic design level, fpga architecture design, threshold logic design, combinational logic design, logic design effort	0.4953	0.4953
KG-9	• general boolean functions • arbitrary boolean functions • multiple-valued logic functions • other boolean functions • universal boolean functions	0.4541	KA-9	general boolean functions, arbitrary boolean functions, multiple-valued logic functions, other boolean functions, universal boolean functions	0.4541	0.4541
KG-10	• parallel instruction execution • program memory performance • processor instruction execution • parallel processor performance • machine code instructions	0.3997	KA-1	parallel instruction execution, program memory performance, processor instruction execution, parallel processor performance, machine code instructions	0.3997	0.3997
KG-11	• computer system design • digital system design • hardware system design • digital systems design • hardware systems design	0.5442	KA-13	computer system design, digital system design, hardware system design, digital systems design, hardware systems design	0.5442	0.5442
KG-12	• unidirectional error codes • error control codes • arithmetic error codes • error correcting codes • error codes	0.4511	KA-5	unidirectional error codes, error control codes, arithmetic error codes, error correcting codes, error codes	0.4511	0.4511
	Rata - rata Koherensi Topik	0.4595			0.4513	

Tabel 6.9 Contoh Artikel Ilmiah pada Klaster KA-7 dan KA-15 di Dataset 2 Skenario2

Klaster KA-7		Klaster KA-15	
Judul	Abstrak	Judul	Abstrak
Analysis and Comparison of **Cache** Coherence Protocols for a Packet-Switched Multiprocessor	Analytical models are developed for seven existing **cache** protocols, namely, Write-Once, Write-Through, Synapse, Berkeley, Illinois, Firefly, and Dragon. The protocols are implemented on a multiprocessor with a packet-switched shared bus. The models are based on queuing **networks** that consist of both open and closed classes of customers ….	Efficient use of memory **bandwidth** to improve **network** processor throughput	We consider the efficiency of packet **buffers** used in packet switches built using **network** processors (NPs). Packet **buffers** are typically implemented using DRAM, which provides plentiful buffering at a reasonable cost. The problem we address is that a typical NP workload may be unable to utilize the peak DRAM **bandwidth**. Since the **bandwidth** of the packet **buffer** is often the bottleneck in the performance of a shared-memory packet switch, inefficient use of available DRAM **bandwidth** further reduces the packet throughput. Specialized hardware-based **schemes** that alleviate the DRAM **bandwith** problem ….

Klaster KA-7		Klaster KA-15	
Judul	Abstrak	Judul	Abstrak
Analysis of directory based **cache** coherence **schemes** with multistage **networks**	Designing efficient **cache** coherence **schemes** for shared memory multiprocessors has attracted much attention of the researchers in the area. Snoopy **cache** protocols have been designed for bus based multiprocessors. However, the snoopy protocols are not applicable to general interconnection **networks**. On the other hand, the directory based **cache** protocols adapt very well to any kind of interconnection network such as a Multistage Network. Since different protocols have different cost overheads, and may give different performance, the protocol to be used must be wisely selected …	Comparative Modeling and Evaluation of CC-NUMA and COMA on Hierarchical Ring Architectures	Parallel computing performance on scalable shared-memory architectures is affected by the structure of the interconnection **networks** linking processors to memory modules and on the efficiency of the memory/**cache management systems**. **Cache** Coherence Nonuniform Memory **Access** (CC-NUMA) and **Cache** Only Memory **Access** (COMA) are two effective memory **systems**, and the hierarchical ring structure is an efficient interconnection **network** in hardware. This paper focuses on comparative performance modeling and evaluation of CC-NUMA and COMA on a hierarchical ring shared-memory architecture. Analytical models for the two memory **systems** for comparative evaluation are presented ….

LAMPIRAN DAFTAR RELASI KATA PADA GRAF MAXIMUM COMMON SUBGRAPH

Tabel 7.1 50 Relasi Kata pada Graf MCS antara KA-2 dan KA-11 pada Dataset 2 di Percobaan 3 Skenario 1

Vertex Sumber (Kata)	Vertex Target (Kata)	Tipe Relasi Kata	Bobot Edge
exchange	network	Undirected	0.6932
exchange	log	Undirected	0.638
exchange	interconnection	Undirected	0.7595
exchange	graphs	Undirected	0.7176
exchange	deadlock	Undirected	0.5165
exchange	routing	Undirected	0.7477
exchange	connected	Undirected	0.837
exchange	cycles	Undirected	0.5655
exchange	sup	Undirected	0.6937
exchange	de	Undirected	0.888
exchange	bruijn	Undirected	0.9183
exchange	paths	Undirected	0.7142
exchange	mesh	Undirected	0.8219
exchange	spanning	Undirected	0.8926
exchange	tree	Undirected	0.7592
exchange	cube	Undirected	0.9006
exchange	hypercube	Undirected	0.8296
exchange	shortest	Undirected	0.744
exchange	path	Undirected	0.522
exchange	graph	Undirected	0.6506
exchange	class	Undirected	0.5916
exchange	sub	Undirected	0.627
exchange	permutation	Undirected	0.8958
exchange	dominating	Undirected	0.7401
exchange	nodes	Undirected	0.7567
exchange	omega	Undirected	0.8743
exchange	minimal	Undirected	0.6605
exchange	dimensions	Undirected	0.7248
exchange	embedding	Undirected	0.8842
exchange	trees	Undirected	0.7882
exchange	maximum	Undirected	0.5139
exchange	broadcasting	Undirected	0.9324

Vertex Sumber (Kata)	Vertex Target (Kata)	Tipe Relasi Kata	Bobot Edge
exchange	fully	Undirected	0.5955
exchange	hamiltonian	Undirected	0.8262
exchange	theta	Undirected	0.7386
exchange	orthogonal	Undirected	0.7182
exchange	convex	Undirected	0.6638
exchange	spl	Undirected	0.7168
exchange	edge	Undirected	0.743
exchange	disjoint	Undirected	0.7687
exchange	sort	Undirected	0.6244
exchange	depth	Undirected	0.6171
exchange	fv	Undirected	0.7059
exchange	channel	Undirected	0.7125
exchange	diagnosability	Undirected	0.5313
exchange	hbox	Undirected	0.8103
exchange	rm	Undirected	0.8022
exchange	matching	Undirected	0.6205
exchange	placement	Undirected	0.5964

Tabel 7.2 50 Relasi Kata pada Graf MCS antara KA-6 dan KA-14 pada Dataset 2 di Percobaan 3 Skenario 1

Vertex Sumber (Kata)	Vertex Target (Kata)	Tipe Relasi Kata	Bobot Edge
systems	system	Undirected	0.75
systems	distributed	Undirected	0.5675
systems	computer	Undirected	0.5242
systems	scale	Undirected	0.5026
systems	applications	Undirected	0.5507
distributed	allocation	Undirected	0.51
distributed	computer	Undirected	0.5345
distributed	resource	Undirected	0.5159
distributed	sharing	Undirected	0.6047
distributed	exclusion	Undirected	0.7549
distributed	oriented	Undirected	0.5972
distributed	shared	Undirected	0.6443
distributed	communication	Undirected	0.5513
distributed	processes	Undirected	0.5752
distributed	replicated	Undirected	0.7543
distributed	co	Undirected	0.5218
distributed	balancing	Undirected	0.5841
distributed	online	Undirected	0.6707
distributed	adaptive	Undirected	0.5418
distributed	multiprocessor	Undirected	0.6925
distributed	synchronization	Undirected	0.6063
distributed	continuous	Undirected	0.5975
distributed	media	Undirected	0.5031
distributed	demand	Undirected	0.51
distributed	multimedia	Undirected	0.5665
distributed	grid	Undirected	0.6844
time	online	Undirected	0.5
real	sharing	Undirected	0.5907
real	critical	Undirected	0.5881
real	service	Undirected	0.507
real	online	Undirected	0.528
real	continuous	Undirected	0.71
real	multimedia	Undirected	0.564
allocation	scheduling	Undirected	0.8142
allocation	resource	Undirected	0.7314
allocation	load	Undirected	0.7055

Vertex Sumber (Kata)	Vertex Target (Kata)	Tipe Relasi Kata	Bobot Edge
allocation	sharing	Undirected	0.71
allocation	exclusion	Undirected	0.6241
allocation	dynamic	Undirected	0.8155
allocation	policies	Undirected	0.6731
allocation	shared	Undirected	0.5577
allocation	replicated	Undirected	0.557
allocation	balancing	Undirected	0.8261
allocation	online	Undirected	0.6445
allocation	adaptive	Undirected	0.5779
allocation	multiprocessor	Undirected	0.5011
allocation	demand	Undirected	0.5317
allocation	grid	Undirected	0.5196
allocation	aware	Undirected	0.6182

Tabel 7.3 50 Relasi Kata pada Graf MCS antara KA-7 dan KA-15 pada Dataset 2 di Percobaan 3 Skenario 2

Vertex Sumber (Kata)	Vertex Target (Kata)	Tipe Relasi Kata	Bobot Edge
data	average	Undirected	3
data	access	Undirected	9
data	replication	Undirected	1
data	based	Undirected	1
data	compression	Undirected	2
data	multiple	Undirected	2
data	spatial	Undirected	2
data	streams	Undirected	2
data	forwarding	Undirected	2
data	traffic	Undirected	1
protocols	cache	Undirected	2
protocols	based	Undirected	2
network	traffic	Undirected	8
network	bandwidth	Undirected	10
network	processors	Undirected	1
network	system	Undirected	1
network	hierarchical	Undirected	2
large	distributed	Undirected	2
large	performance	Undirected	1
distributed	computer	Undirected	7
distributed	architectures	Undirected	1
distributed	cache	Undirected	1
protocol	based	Undirected	5
protocol	management	Undirected	1
level	high	Undirected	2
control	access	Undirected	15
control	scheme	Undirected	3
information	sharing	Undirected	3
allocation	based	Undirected	2
allocation	methods	Undirected	1
allocation	algorithm	Undirected	1
allocation	buffer	Undirected	1

Vertex Sumber (Kata)	Vertex Target (Kata)	Tipe Relasi Kata	Bobot Edge
algorithm	based	Undirected	4
algorithm	proposed	Undirected	4
algorithm	distribution	Undirected	2
virtual	channels	Undirected	14
virtual	support	Undirected	2
virtual	use	Undirected	1
ring	hierarchical	Undirected	3
access	average	Undirected	2
access	multiple	Undirected	15
access	vector	Undirected	2
access	efficient	Undirected	2
access	latency	Undirected	1
access	memory	Undirected	2
access	bus	Undirected	1
average	performance	Undirected	3
average	response	Undirected	2
average	cost	Undirected	1

Tabel 7.4 50 Relasi Kata pada Graf MCS antara KA-7 dan KA-14 pada Dataset 2 di Percobaan 3 Skenario 2

Vertex Sumber (Kata)	Vertex Target (Kata)	Tipe Relasi Kata	Bobot Edge
data	replicated	Undirected	3
data	high	Undirected	3
data	transmission	Undirected	8
data	transfer	Undirected	2
data	service	Undirected	4
data	access	Undirected	9
data	based	Undirected	1
data	management	Undirected	1
data	processing	Undirected	1
data	wireless	Undirected	5
data	traffic	Undirected	1
protocols	control	Undirected	2
protocols	based	Undirected	2
protocols	network	Undirected	2
network	model	Undirected	3
network	communication	Undirected	5
network	access	Undirected	2
network	radio	Undirected	8
network	computer	Undirected	15
network	area	Undirected	7
network	congestion	Undirected	6
network	latency	Undirected	2
network	architecture	Undirected	10
network	system	Undirected	1
network	security	Undirected	2
network	environment	Undirected	1
network	design	Undirected	1
network	distributed	Undirected	1
network	management	Undirected	1
network	based	Undirected	3
network	control	Undirected	2
network	distribution	Undirected	1

Vertex Sumber (Kata)	Vertex Target (Kata)	Tipe Relasi Kata	Bobot Edge
large	distributed	Undirected	2
large	client	Undirected	1
distributed	computer	Undirected	7
distributed	system	Undirected	9
distributed	dynamic	Undirected	2
distributed	computing	Undirected	4
distributed	asynchronous	Undirected	1
distributed	systems	Undirected	4
distributed	support	Undirected	1
distributed	applications	Undirected	1
protocol	based	Undirected	5
protocol	communication	Undirected	5
level	high	Undirected	2
level	user	Undirected	1
level	system	Undirected	1
level	design	Undirected	1
flow	control	Undirected	34

Tabel 7.5 50 Relasi Kata pada Graf MCS antara KA-14 dan KA-15 pada Dataset 2 di Percobaan 3 Skenario 2

Vertex Sumber (Kata)	Vertex Target (Kata)	Tipe Relasi Kata	Bobot Edge
control	scheme	Undirected	1
control	access	Undirected	4
system	operating	Undirected	52
system	file	Undirected	45
system	storage	Undirected	6
system	support	Undirected	7
system	memory	Undirected	3
system	design	Undirected	8
system	architecture	Undirected	3
system	computer	Undirected	7
system	access	Undirected	2
system	performance	Undirected	4
system	data	Undirected	1
system	network	Undirected	3
main	memory	Undirected	10
main	storage	Undirected	1
memory	large	Undirected	2
memory	management	Undirected	9
memory	shared	Undirected	11
memory	operations	Undirected	1
systems	computer	Undirected	13
systems	based	Undirected	4
systems	operating	Undirected	13
systems	file	Undirected	13
systems	storage	Undirected	6
systems	software	Undirected	3
time	latency	Undirected	3
time	access	Undirected	1
data	sets	Undirected	3
data	large	Undirected	3
data	parallel	Undirected	1
data	sharing	Undirected	4

Vertex Sumber (Kata)	Vertex Target (Kata)	Tipe Relasi Kata	Bobot Edge
data	shared	Undirected	8
data	distributed	Undirected	9
data	remote	Undirected	1
data	based	Undirected	2
data	access	Undirected	7
data	dynamic	Undirected	3
data	storage	Undirected	5
data	caching	Undirected	1
data	traffic	Undirected	1
data	file	Undirected	1
file	access	Undirected	4
file	storage	Undirected	1
file	network	Undirected	2
file	parallel	Undirected	2
write	operations	Undirected	1
storage	servers	Undirected	3
storage	dynamic	Undirected	2

BIODATA PENULIS

Penulis dilahirkan di Surabaya, 1 Maret 1994, merupakan anak kedua dari 3 bersaudara. Penulis telah menempuh pendidikan formal yaitu TK Sumbangsih (1999-2001), SD Taruna Bakti (2001-2006), SMP Taruna Bakti (2006-2009), SMA Negeri 3 Bandung (2009-2011), dan mahasiswa S1 Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Selama masa kuliah, penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Computer (HMTC). Diantaranya adalah menjadi staff departemen Media Informasi Himpunan Mahasiswa Teknik Computer ITS 2012-2013. Penulis juga aktif dalam kegiatan kepanitiaan Schematics. Diantaranya penulis pernah menjadi staff Hubungan Masyarakat Schematics 2012 dan staff Kesertariatan Schematics 2013. Selama kuliah di teknik informatika ITS, penulis mengambil bidang minat Komputasi Cerdas Visi (KCV). Komunikasi dengan penulis dapat melalui email: nurilhamadhi@gmail.com

[Halaman ini sengaja dikosongkan]