E-ISSN: 2715-2731

Analisis dan Desain *Software* Jejaring Kata untuk Menghitung Kemiripan antar Kata Menggunakan *Graph Database*

Muhammad Rizal Musthofa[1,*], Bintang Miftaqul Huda[2], Muhammad Zaim Maulana[3], Muhammad Ainul Yaqin[4]

Jurusan Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim, Indonesia

[1]15650048@student.uin-malang.ac.id; [2]19650093@student.uin-malang.ac.id; [3]19650058@student.uin-malang.ac.id; [4]yaqinov@ti.uin-malang.ac.id
* corresponding author

INFO ARTIKEL

Sejarah Artikel
Diterima: 29 Mei 2021
Direvisi: 18 Juni 2021
Diterbitkan: 30 Agustus 2021

Kata Kunci
Jejaring kata
Graph database
Metode *Path Similarity*
Waterfall Model

ABSTRAK

Jejaring kata atau *wordnet* merupakan representasi hubungan semantik berupa sinonim antar kata. Saat ini penelitian terkait *wordnet* berbahasa Indonesia masih begitu minim. Adapun *wordnet* berbahasa Indonesia yang ada, belum menggunakan *graph database* sehingga belum bisa menampilkan kemiripan antar kata secara visual. Oleh karena itu, masalah yang dirumuskan pada penelitian ini yaitu cara menentukan tingkat kemiripan antar kata dalam jejaring kata berbahasa indonesia. Penelitian ini bertujuan untuk mendesain *software wordnet* berbahasa Indonesia untuk menentukan tingkat kemiripan antar kata berbahasa Indonesia. Dengan *wordnet* ini akan memudahkan proses pencarian nilai similaritas pada suatu kata. Metode yang ada dalam pengembangan aplikasi *wordnet* menggunakan metode *waterfall model* yang terdiri dari beberapa tahap *requirement*, desain, implementasi, verifikasi, dan *maintenance*. Aplikasi ini terdiri atas struktur dataset kata berbahasa Indonesia berbentuk *graph database* Neo4J yang digunakan dalam pencarian kemiripan antar dua kata. Proses pencarian nilai similaritas dalam aplikasi ini menggunakan metode *path similarity* sehingga diperolah nilai kemiripan dari dua kata yang diinputkan ke dalam sistem. Adapun penelitian ini hanya menggunakan dataset kata berbahasa Indonesia dan memiliki jumlah dataset yang masih terbatas. Penelitian ini berkontribusi dalam pengembangan *software* jejaring kata berbahasa Indonesia menggunakan *graph database*. Hasil penelitian yang didapat berupa desain *software* jejaring kata untuk menghitung kamiripan kata menggunakan *graph database*. Dengan adanya pengembangan *software* tersebut, mampu menyelesaikan permasalahan seputar perhitungan kemiripan kata.

PENDAHULUAN

Jejaring kata adalah sebuah *database* leksikal yang memiliki hubungan semantik dengan menggolongkan kata kerja (*verb*), kata keterangan (*adverb*), kata sifat (*adjective*), dan kata benda (*noun*) menjadi set sinonim yang dikenal dengan istilah *synset*. Hubungan semantik ini berfungsi untuk mengatur basis pengetahuan leksikal dari mulai dari hipernim sampai sinonim kata. Set sinonim kata (*synset*) diatur menjadi suatu indra, memberikan sinonim dari setiap kata, dan juga menyediakan struktur seperti pohon hirarki untuk masing-masing istilah [1].

Wordnet atau jejaring kata dalam Bahasa Indonesia pada saat ini yakni sinonimkata.com. *Wordnet* sinonimkata.com merepresentasikan hubungan semantik berupa sinonim antar kata. Sinonimkata.com merupakan kamus tesaurus yang menyediakan dataset Bahasa Indonesia

yang bisa dipergunakan dalam pencarian kemiripan kata [2]. Wordnet sinonimkata.com direpresentasikan dalam bentuk *graph*.

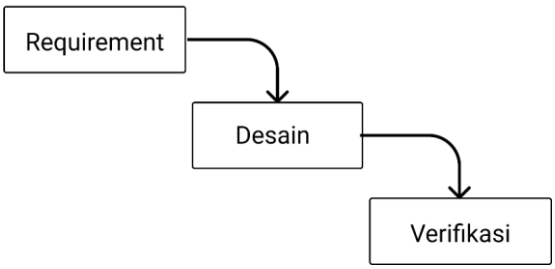
Salah satu metode penyimpanan data kata dalam *wordnet* yakni dengan menggunakan *graph database*. *Graph database* merupakan penyimpanan data berbentuk diagram atau grafik yang struktur datanya terdiri dari *edge* dan *vertices*. Graf sendiri adalah pasangan terurut G = (V, E) yang tersusun dari himpunan V atau simpul bersama-sama dengan set E tepi atau garis [3]. *Graph database* digunakan untuk mencari keterkaitan antar kata dalam suatu jejaring kata dengan cara megukur panjang jalur antara satu kata dengan kata yang lain. Penggunaan *graph database* lebih dinamis dan mampu mengetahui secara visual bentuk sinonim kata. Kelebihan model graf dalam representasi hubungan antar objek atau *node* mulai diadopsi ke dalam basis data. Selain itu *graph database* sangat baik untuk pengelolaan basis data yang tidak terstruktur. *Query* dari *graph database* lebih sederhana. Representasi dari *graph database* dengan sekala besar, diambil sinonim yang menghasilkan *graph* yang lebih spesifik atau skala kecil [4]. *Graph database* lebih dinamis, hal itu disebabkan karena pada *relational database* akan membutuhkan *query* yang lebih kompleks, sehingga ketika akan direlasikan, perlu *query* satu persatu. Karakteristik *graph database* ini sesuai dengan karakteristik dataset kata pada jejaring kata yang kompleks, sehingga lebih mudah untuk diimplementasikan untuk membangun *software* jejaring kata [5]. Pada penelitian ini, digunakan Neo4j untuk membantu menyimpan dataset dalam bentuk *graph*.

Adapun fakta penelitian terkait *wordnet* yang telah dilakukan peneliti sebelumnya, membahas mengenai hubungan semantik pada *WordNet*. *WordNet* pada penelitian tersebut merupakan *software* jejaring kata Bahasa Inggris yang didesain oleh Universitas Princeton [6]. Pada penelitian sebelumnya juga, peneliti ada yang membahas mengenai pegukuran kemiripan makna antar kalimat. Pada penelitian tersebut, pengukuran kemiripan menggunakan *software* WS4J dengan pendekatan *lin, wu palmer, dan path* [7]. Terdapat juga penelitian yang membahas mengenai pengukuran kemiripan makna kalimat berbahasa Indonesia [2]. Pada penelitian tersebut, kalimat yang digunakan berbahasa Indonesia. Namun hasil penelitian tersebut belum mengimplementasikan *graph database* Neo4J ke dalam bentuk *software*.

Dari fakta penelitian diatas, menunjukkan bahwa rata-rata *software* jejaring kata yang ada saat ini adalah *wordnet* berbahasa Inggris. Adapun *software* jejaring kata untuk menghitung kemiripan kata menggunakan *graph database* berbahasa Indonesia masih minim dikembangkan oleh peneliti. Hal ini dibuktikan dari adanya beberapa software jejaring kata berbahasa Inggris seperti WS4J (https://ws4jdemo.appspot.com/) dan WordNet Princenton (http://wordnetweb.princeton.edu). Kemudian dibuktikan juga *software* jejaring kata berbahasa Indonesia seperti WordNet Indonesia dan WordNet Bahasa Indonesia berbasis Linked Data [8][9].

Oleh karena itu, masalah yang dirumuskan pada penelitian ini yaitu bagaimana menentukan tingkat kemiripan antar kata dalam jejaring kata berbahasa Indonesia berdasarkan desain *software* yang dibuat. Untuk menentukan kemiripan antar kata pada jejaring kata diperlukan struktur data dan algoritma perhitungan kemiripan kata yang sesuai. Berdasarkan hal tersebut, digunakanlah *graph database* untuk mencari keterkaitan kata dalam jejaring kata dan mendapatkan panjang jalur setiap *node* nya. Panjang jalur yang dihasilkan akan menjadi komponen penting dalam perhitungan kemiripan kata. Penelitian ini ditujukan untuk mendesain dan menganalisis *software* jejaring kata untuk menghitung kemiripan kata menggunakan *graph database*. *Software* ini harapannya akan berguna untuk membantu menentukan tingkat kemiripan antar kata berbahasa Indonesia.

METODE



Gambar 1. Prosedur penelitian

Gambar 1 menunjukkan proses pengembangan suatu sistem perangkat lunak menggunakan model *waterfall*. Model tersebut menyediakan ancangan dari alur pengembangan perangkat lunak dalam konteks sekuensial (*sequential linear*). Alur pengembangan pada penelitian ini terurut mulai dari *requirement*, desain, sampai verifikasi desain perangkat lunak [10].

Data Penelitian

Data penelitian berasal dari data primer (data awal) dan data sekunder. Data primer (data awal) yang digunakan dalam penelitian berasal dari sinonimkata.com. Sedangkan data sekunder menggunakan metode *path* sebagai metode untuk meyelesaikan permasalahan penelitian.

Analisis *Requirement*

Pada saat mendeskripsikan suatu sistem terasa sulit jika tanpa sebuah analisis kebutuhan *software*. Hal ini bisa saja mengakibatkan kebutuhan sistem yang dibuat tidak bisa menyelesaikan permasalahan. Oleh karena itu, diperlukan suatu analisis *requirement* supaya dapat dipahami perangkat lunak atau *software* seperti apa yang diperlukan oleh pengguna. Analisis kebutuhan *software* digunakan untuk mengungkapkan keperluan dan batasan yang ada pada produk perangkat lunak atau *software* yang berkontribusi pada pemecahan beberapa masalah dunia nyata [11]. Analisis kebutuhan dibagi atas dua hal, yakni kebutuhan non-fungsional, dan kebutuhan fungsional.

Fungsional *Requirement*

Fungsional *requirement* adalah suatu keperluan bagaimana suatu sistem memproses *input* dan *output*. Fungsional *requirement* menggambarkaan fungsi yang dijalankan oleh perangkat lunak. Fungsional *requirement* sebagai salah satu rangkaian uji coba terbatas untuk menvalidasi perilaku sebuah *software* [11].

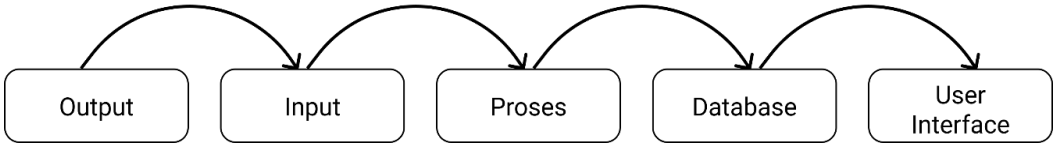
Non-fungsional Requirement

Non-fungsional *requirement* merupakan suatu kebutuhan yang menitikberatkan pada pembatasan solusi yang dimiliki suatu perangkat lunak [11]. Kebutuhan non-fungsional membatasi layanan atau fungsi yang ditawarkan oleh sistem. Batasan tersebut diantaranya *usability*, *portability*, *reliability*, dan *supportability*.

Fitur *Software*

Setelah *requirement* berhasil dideskripsikan, selanjutnya mendeskripsikan fitur *software* jejaring kata. Fitur merupakan fungsi khusus yang disediakan oleh sistem. Terdapat dua fitur yang akan disediakan oleh *software* jejaring kata ini, yaitu fitur *input* otomatis dan fitur hitung kemiripan kata.

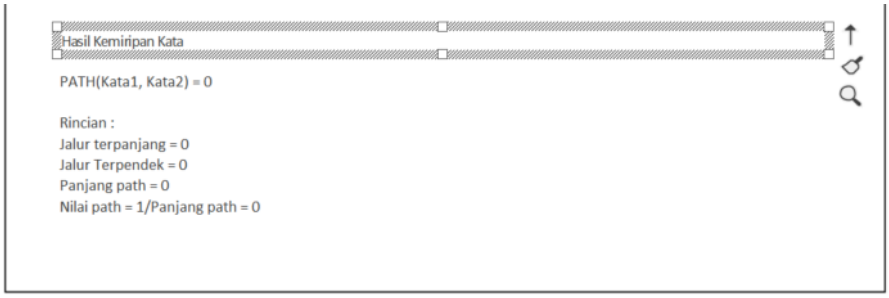
Desain *Software*



Gambar 2. Alur desain *software*

Gambar 2 menunjukkan alur desain suatu sistem perangkat lunak. Alur desain pada penelitian ini terurut mulai dari desain *output*, *input*, proses, *database*, dan *user interface*.

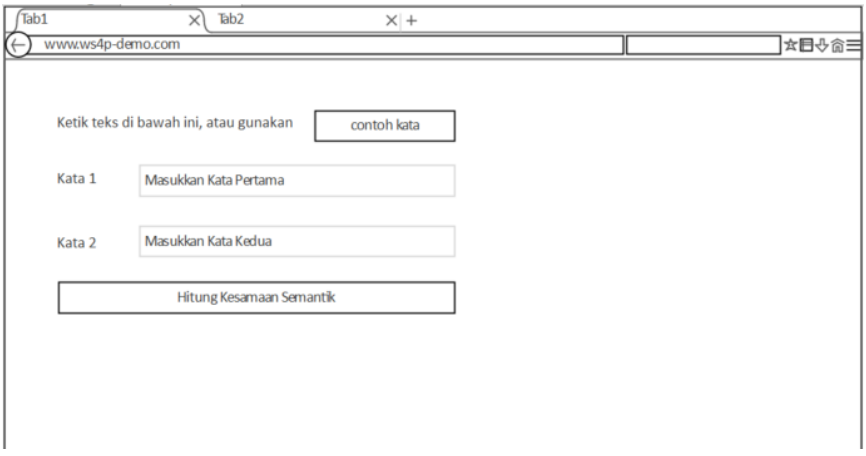
Desain *Output*



Gambar 3. Desain *output*

Gambar 3 menunjukkan tampilan desain *output* yang akan dihasilkan oleh *software* jejaring kata. Desain *output* adalah tampilan yang menunjukkan hasil masukkan yang telah dilakukan sebelumnya. Pada penelitian ini, *output* dari *software* jejaring kata menghasilkan suatu nilai kemiripan kata Bahasa Indonesia.

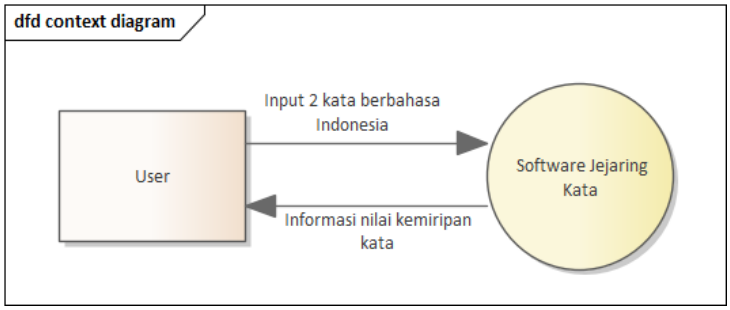
Desain *Input*



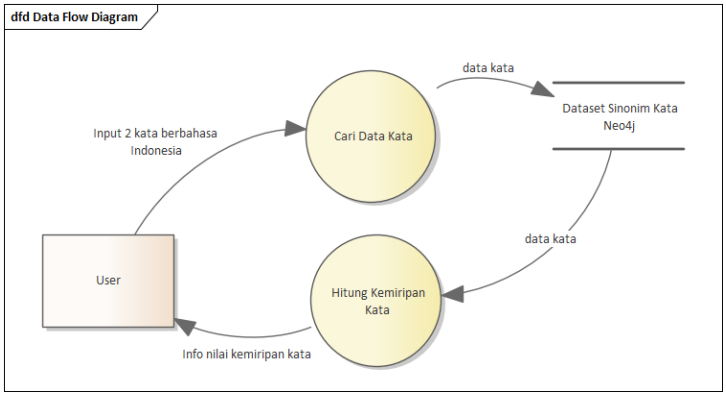
Gambar 4. Desain *input*

Gambar 4 menunjukkan desain *input software* jejaring kata. Desain *input* adalah tampilan yang digunakan untuk memasukkan data ke dalam *database*. Pada penelitian ini, *software* jejaring kata menyediakan dua *form* untuk penginputan kata berbahasa Indonesia. Dua kata yang telah di inputkan nantinya akan dihitung kemiripan katanya.

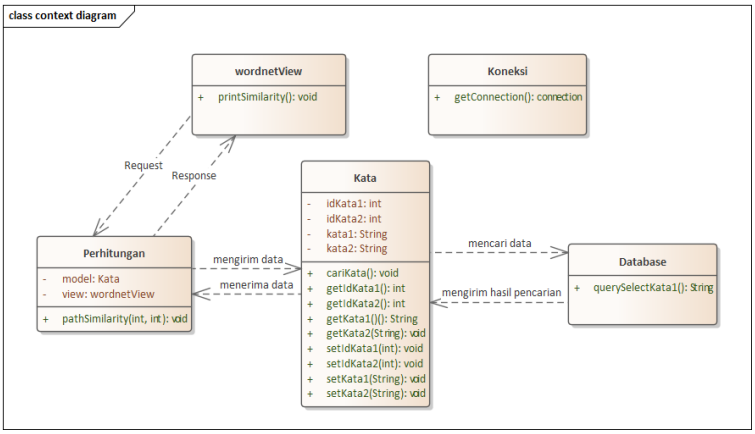
Desain Proses



Gambar 5. Diagram Konteks



Gambar 6. *Data Flow Diagram* (DFD)



Gambar 7. *Class Diagram*

Desain proses merupakan tahapan yang dilakukan oleh perangkat lunak untuk mencapai tujuannya. Pada penelitian ini, desain proses pada *software* jejaring kata dijabarkan dalam tiga diagram. Diagram pertama (Gambar 5) menggambarkan sistem secara garis besar dalam bentuk diagram konteks (DAD). Diagram kedua (Gambar 6) menggambarkan fungsi logika dari sebuah *software* dalam bentuk *Data Flow Diagram* (DFD). Diagram ketiga (Gambar 7) menggambarkan set kelas dan objek yang memiliki hubungan timbal balik dalam suatu *class diagram* [11].

Desain *Database*

Desain *database software* jejaring kata menjelaskan konsep *graph* yang digunakan. *Database* manajemen dalam *software* jejaring kata ini menggunakan Neo4j untuk menyimpan data dalam bentuk *graph*. Konsep *graph* yang digunakan dibagi menjadi dua bagian, yakni bagian relasi dan *node* [12].

1. Bagian Node

Bagian ini menjelaskan *node* di dalam *graph database* Neo4j. Dalam *software* ini, *node* mempresentasikan kata kerja dan kata benda yang telah dimasukkan dari sinonimkata.com. Di dalam *node* terdapat sebuah label sebagai tipe dari *node* tersebut. Keseluruhan label yang digunakan pada *software* jejaring kata ditunjukkan pada Tabel 1.

Tabel 1. Label dalam Graph Database Software Jejaring Kata

Label	Keterangan	Sampel Data Kata
Kt_Kerja	Label ini merujuk pada dataset kata Bahasa Indonesia yang tergolong dalam kata kerja.	Membagi, membuat, mengisi, mendata, aktivasi, merancang.
Kt_Benda	Label ini merujuk pada dataset kata Bahasa Indonesia yang tergolong dalam kata benda.	Setoran, santri, pondok, koperasi, masyarakat, quran.

Selain label, *node* juga mempunyai sebuah properti yang mendeskripsikan keterangan tambahan pada *node*. Seluruh *node* pada dataset kata memiliki properti yang sama yaitu *name*. Properti ini merujuk pada nama dari sebuah *node* yang diambil dari dataset sinonimkata.com.

2. Bagian Relasi

Bagian ini menjelaskan hubungan antar *node* di dalam *graph database* Neo4j. Relasi ini mempresentasikan hubungan kata pertama dengan kata lainnya dalam suatu tipe relasi. Dalam dataset yang dibuat, tipe relasi yang digunakan yaitu sinonim. Tipe relasi ini mempresentasikan hubungan kemiripan antar kata dalam *graph database* Neo4j.

Desain *User Interface*

User interface menentukan tampilan perangkat lunak yang akan disajikan kepada pengguna. Desain *user interface* atau antarmuka pengguna merupakan bagian krusial dari proses desain *software*. Desain ini mencakup interaksi yang dilakukan pengguna terhadap sistem, sekaligus informasi yang ditampilkan sistem kepada pengguna [12]. Desain *user interface* perlu memastikan bahwa interaksi antar mesin dan manusia menyajikan pengoperasian yang efisien dan efektif. Informasi utama yang disajikan sistem ini berupa nilai kemiripan kata beserta rincian hasil perhitungan kemiripan kata.

Perhitungan Kemiripan Kata

Pada bagian ini menjabarkan mengenai implementasi metode *semantic similarity* untuk menghitung kemiripan kata. Metode perhitungan kemiripan kata pada penelitian ini menggunakan pendekatan *path*. Metode *path* dibuat untuk bekerja pada struktur hirarki. Metode ini dihitung melalui persamaan berikut [13] :

$$Sim(C1, C2) = 2 \times Max(C1, C2) - SP \qquad (1)$$

Persamaan (1) adalah persamaan yang digunakan untuk mengakumulasi nilai dari kemiripan kata dengan memanfaatkan pendekatan *path*. Fungsi *Max()* merupakan rute maksimum antara node C1 dan C2 pada pengelompokkan kata. SP merupakan rute terpendek yang merelasikan C1 dan C2. Contoh perhitungan menggunakan *path* :
Kata 1 : mengisi
Kata 2 : mendata

Dalam pencarian kata ‘mengisi’ dan ‘mendata’ memiliki jalur maksimal 5 dan jalur minimal 2. Perhitungan kemiripan kata tersebut ditunjukkan pada Tabel 2.

Tabel 2. Nilai kemiripan kata

	Nilai	
	mengisi (C1)	mendata (C2)
Max(C1,C2)	5	
SP	2	
Sim(C1,C2)	$2 \times 5 - 2 = 8$	
Path(C1,C2)	1/8 = 0.125	

Skenario Eksperimen

Skenario eksperimen dilakukan untuk menganalisis kesesuaian antara desain perangkat lunak dengan permasalahan. Tujuan yang akan dicapai dari eksperimen ini adalah untuk menjawab persoalan yang ada pada penelitian ini. Berikut adalah skenario eksperimen yang dilakukan: (1) Skenario Pertama adalah enganalisis kesesuaian antara desain, *requirement*, dan permasalahan yang ada. (2) Skenario Kedua yakni mengukur kemiripan kata menggunakan metode *path*. Dari skenario eksperimen tersebut, penelitian ini memberikan sebuah kontribusi dalam pengembangan *software* jejaring kata berbahasa Indonesia menggunakan *graph database*. Kontribusi yang diberikan yaitu menjelaskan proses perhitungan kemiripan kata pada software jejaring kata menggunakan graph database. Hal itu dikarenakan pada penelitian sebelumnya belum mengimplementasikan proses perhitungan kemiripan kata [14].

Rancangan Pengujian *Software* (Verifikasi)

Rancangan pengujian *software* menjabarkan metode yang digunakan untuk memverifikasi desain perangkat lunak. Dalam perancangan pengujian ini, akan diuji fungsionalitas dari *software* jejaring kata. Pengujian fungsionalitas *software* mencakup bagian *input* kata, dan visualisasi hasil. Berikut skenario perancangan pengujian *software* yang ditunjukkan pada Tabel 3.

Tabel 3. Skenario rancangan pengujian fungsionalitas *software* jejaring kata

Bagian	**Cara Pengujian**	**Hasil yang Diharapkan**
***Input* kata**	Memasukkan dua kata berbahasa Indonesia ke dalam *form input* kata.	*Software* memvalidasi kata yang dimasukkan.
Visualisasi hasil	Menekan tombol ‘Hitung Kemiripan kata’ setelah memasukkan kata.	*Software* menampilkan hasil berupa nilai kemiripan kata menggunakan metode *path*.

Sedangkan pemetaan verifikasi kesesuaian rumusan masalah, *requirement*, dan desain ditunjukkan pada Tabel 4.

Tabel 4. Verifikasi rumusan masalah, *requirement*, dan desain

Rumusan Masalah	***Requirement***	**Desain**
Bagaimana menentukan tingkat kemiripan antar kata dalam jejaring kata berbahasa Indonesia	Menyediakan *form* untuk menginputkan dua kata berbahasa Indonesia untuk dihitung kemiripan katanya.	Terdapat *form input* kata yang digunakan untuk memasukkan dua kata berbahasa Indonesia.
	Software harus dapat melakukan perhitungan kemiripan kata berbahasa Indonesia.	Terdapat tombol ‘Hitung Kemiripan kata’ untuk mengukur tingkat kemiripan kata menggunakan metode *path*.

HASIL DAN PEMBAHASAN

Hasil Analisis *Requirement*

Fungsional *Requirement*

Pada bagian ini telah dihasilkan beberapa kebutuhan fungsional berdasarkan permasalahan yang ada. Kebutuhan fungsional ini nantinya akan dikembangkan menjadi sebuah fitur. Kebutuhan fungsional dari *software* jejaring kata dijabarkan pada Tabel 5 seperti berikut:

Tabel 5. Fungsional *requirement software* jejaring kata

No.	Kebutuhan Fungsional
1	*Software* harus dapat melakukan perhitungan kemiripan kata berbahasa Indonesia.
2	Menyediakan *form* untuk memasukkan dua kata berbahasa Indonesia untuk dihitung kemiripan katanya.
3	Dataset yang digunakan untuk perhitungan kemiripan kata berasal dari sinonimkata.com.
4	*Software* harus dapat memvisualisasikan hasil perhitungan kemiripan kata.

Non-fungsional *Requirement*

Pada penelitian ini, non-fungsional *requirement* dari *software* jejaring kata dijabarkan pada Tabel 6 seperti berikut:

Tabel 6. Non-fungsional *requirement software* jejaring kata

Kategori	Kebutuhan Non-Fungsional
Usability	Sistem menyediakan *form* penginputan kata Bahasa Indonesia untuk menghitung kemiripan kata.
Portability	Sistem dibangun dalam sebuah aplikasi web.
Reliability	-Sistem dapat diakses dengan cepat. -Sistem memproses perhitungan kemiripan kata dengan cepat.
Supportability	Sistem mendukung penginputan kata berbahasa Indonesia.

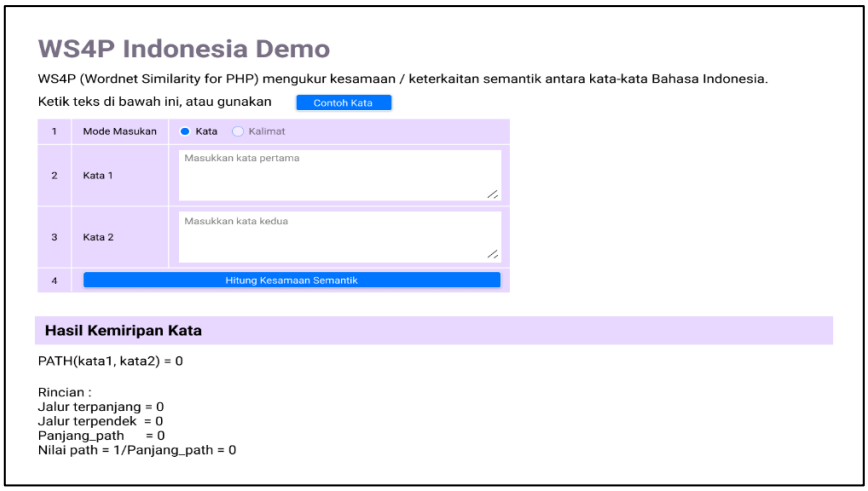
Fitur *Software*

Bagian ini merupakan pengembangan dari kebutuhan yang telah dirumuskan diatas. Terdapat tiga fitur yang disediakan oleh *software* jejariung kata, yakni *input* otomatis, *input* manual, dan hitung kesamaan semantik. Tabel 7 menjabarkan fitur yang disediakan oleh *software* jejaring kata.

Tabel 7. Fitur *software* jejaring kata

Fitur	Kemampuan
***Input* otomatis**	Memasukkan kata berbahasa Indonesia secara otomatis dengan menekan tombol “contoh kata”.
***Input* manual**	Memasukkan kata berbahasa Indonesia pada *form* yang telah disediakan.
Hitung kesamaan semantik	Menghitung kemiripan kata dari dua kata yang telah dimasukkan pada *form* dan menampilkan hasil visualnya.

Hasil Desain *User Interface*

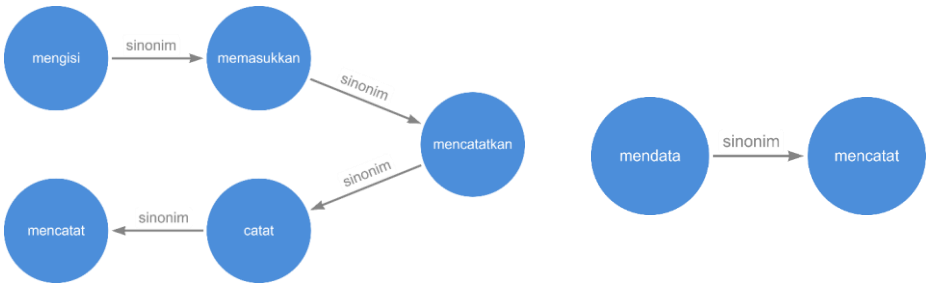


Gambar 8. *User interface software* jejaring kata

Pada bagian ini, telah di buat desain *user interface software* jejaring kata. Gambar 8 menunjukkan tampilan desain *user interface* dari *software* jejaring kata. Pada tampilan *user interface* tersebut terdapat bagian *input* dan *output* yang dihasilkan. Pada bagian *input* terdapat dua fitur yang disediakan, yakni *input* otomatis yang ditandai dengan adanya tombol contoh kata dan fitur manual yang ditandai dengan *form input* kata. Pada bagian ini disediakan fitur hitung kesamaan semantik yang digunakan untuk mencari nilai kemiripan kata menggunakan metode *path*. Pada bagian *output* ditampilkan hasil kemiripan kata berupa nilai *path similarity*.

Mengukur Kemiripan Kata

Pengukuran kemiripan kata menggunakan pendekatan *path*. Sebelum melakukan pengukuran kemiripan kata, dilakukan pencarian kata pada dataset sinonimkata.com yang telah dimasukkan pada Neo4J. Setelah data ditemukan, maka dicari panjang jalur terpanjang dan terpendeknya. Gambar 9 menunjukkan illustrasi panjang jalur kata yang dicari dalam bentuk graf.



Gambar 9. Graf kata ‘mengisi’ dan ‘mendata’

Pada pencarian kata ‘mengisi’ dan ‘mendata’, jalur maksimum bernilai 5 dan jalur minimum bernilai 2. Selanjutnya dua kata tersebut dihitung kemiripan katanya menggunakan persamaan 1 diatas.

Tabel 8. Perhitungan kemiripan kata

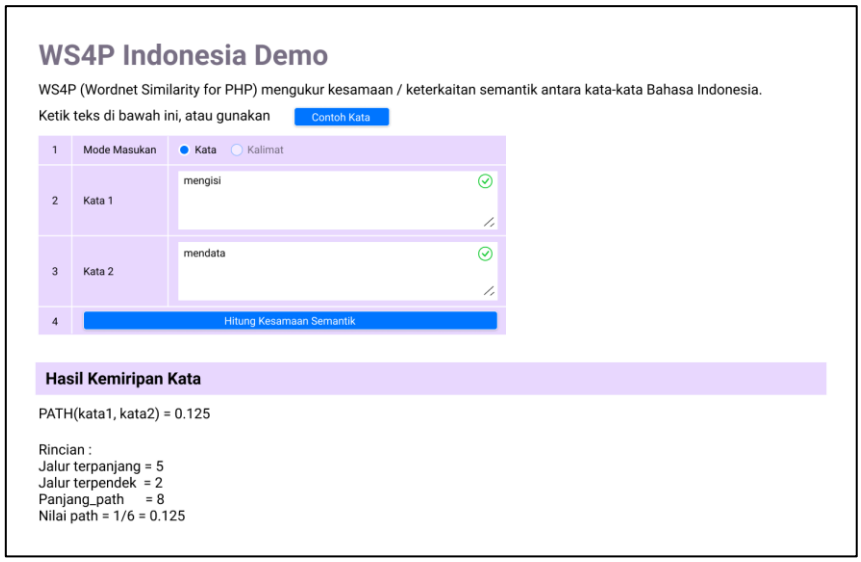
	Nilai	
	Mengisi (C1)	Mendata (C2)
Max(C1,C2)	5	
SP	2	

Sim(C1,C2)	$2 \times 5 - 2 = 8$
Path(C1,C2)	1/8 = 0.125

Setelah nilai kemiripan kata dihasilkan pada Tabel 8, maka akan ditampilkan pada *output software* seperti pada Gambar 11. Tampilan *output* yang dihasilkan berupa nilai *path similarity* dari kata yang telah dimasukkan.

Hasil Eksperimen (Hasil Analisis)

Data hasil eksperimen *software* jejaring kata menunjukkan status kesesuaian dari beberapa kasus yang diuji. Data ini didapatkan berdasarkan rancangan pengujian fungsional (verifikasi) dari perangkat lunak dan pengukuran kemiripan kata. Berikut visualisasi hasil eksperimen *software* jejaring kata :



Gambar 11. Visualisasi hasil eksperimen *software* jejaring kata

Gambar 11 menunjukkan visualisasi hasil eksperimen sebagai bukti kesesuaian antara permasalahan terhadap desain sistem yang dibuat. Visualisasi hasil tersebut menunjukkan kesesuaian terhadap fungsionalitas *software* jejaring kata.

Tabel 9. Hasil eksperimen

Bagian	Eksperimen	Hasil	Status
***Input* kata**	Memasukkan dua kata berbahasa Indonesia ke dalam *form input* kata.	*Software* berhasil memvalidasi kata yang dimasukkan.	Sesuai
Visualisasi hasil	Menekan tombol 'Hitung Kemiripan kata' setelah memasukkan kata.	*Software* berhasil menampilkan hasil berupa nilai kemiripan kata menggunakan metode *path*. Nilai kemiripan kata ditunjukkan pada Tabel 8.	Sesuai

Tabel 9 menunjukkan hasil fungsionalitas bahwa seluruh eksperimen berhasil dilakukan dan sesuai dengan harapan.

Pembahasan Hasil Eksperimen

Penjelasan hasil eksperimen *software* jejaring kata berfokus pada seluruh bagian yang diuji. Hal ini bertujuan untuk memberikan keterangan tambahan terkait kesesuaian antara desain perangkat lunak terhadap permasalahan. Hasil dari pengujian pada seluruh bagian ditunjukkan pada Tabel 9. Eksperimen pengukuran kemiripan kata menggunakan metode *path* telah ditunjukkan pada Tabel 8.

Tabel 10. Hasil Verifikasi rumusan masalah, *requirement*, dan desain

Rumusan Masalah	***Requirement***	**Desain**	**Status**
Bagaimana menentukan tingkat kemiripan antar kata dalam jejaring kata berbahasa Indonesia	Menyediakan *form* untuk menginputkan dua kata berbahasa Indonesia untuk dihitung kemiripan katanya.	Terdapat *form input* kata yang digunakan untuk memasukkan dua kata berbahasa Indonesia.	Sesuai
	Software harus dapat melakukan perhitungan kemiripan kata berbahasa Indonesia.	Terdapat tombol 'Hitung Kemiripan kata' untuk mengukur tingkat kemiripan kata menggunakan metode *path*.	Sesuai

Tabel 10 menunjukkan hasil verifikasi rumusan masalah, requirement, dan desain software jejaring kata. Hasil verifikasi tersebut diambil berdasarkan hasil eksperimen pada Tabel 9 dan visualisasi hasil eksperimen pada Gambar 11. Berdasarkan tabel tersebut dapat dievaluasi bahwa semua *requirement* yang dirumuskan sesuai dengan desain *software* yang dibuat dan mampu menyelesaikan permasalahan.

KESIMPULAN

Berdasarkan hasil penelitian, penggunaan *graph database* pada *software* jejaring kata untuk menghitung kemiripan antar kata dapat dilakukan dengan menggunakan metode *path*. Metode *path* yang diimplementasikan diambil dari panjang jarak antar *node* pada *graph database* Neo4j. Node tersebut mempresentasikan kata berbahasa indonesia yang dicari. Hasil perhitungan kemiripan kata divisualisasikan dalam bentuk desain *software* jejaring kata. Pada analisis dan desain *software* jejaring kata diperlukan tahapan terstruktur. Tahapan yang dilakukan berupa analisis *requirement*, desain *software*, dan verifikasi. Pada tahap analisis *requirement* perlu dirumuskan kebutuhan fungsional dan non-fungsional *software* jejaring kata. Pada tahap desain *software* diperlukan beberapa tahapan seperti desain *output*, *input*, proses, *database*, dan *user interface*. Pada tahap verifikasi digunakan untuk memvalidasi antara permasalahan, requirement, dan desain. Dari tahapan tersebut didapatkan hasil bahwa desain software jejaring kata yang telah dikembangkan sudah sesuai dengan *requirement* dan mampu menyelesaikan permasalahan.

REFERENSI

[1] T. Wei, Y. Lu, H. Chang, Q. Zhou, and X. Bao, "A semantic approach for text clustering using WordNet and lexical chains," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 2264–2275, 2015, doi: 10.1016/j.eswa.2014.10.023.

[2] Y. Caterina, M. A. Yaqin, and S. Zaman, "Pengukuran Kemiripan Makna Kalimat Dalam Bahasa Indonesia Menggunakan Metode Path," *Fountain Informatics J.*, vol. 2, no. 1, pp. 1–6, 2016.

[3] E. G. Caldarola, A. Picariello, and A. M. Rinaldi, "Big graph-based data visualization experiences: The WordNet case study," *IC3K 2015 - Proc. 7th Int. Jt. Conf. Knowl. Discov. Knowl. Eng. Knowl.*

Manag., vol. 1, no. November, pp. 104–115, 2015, doi: 10.5220/0005632201040115.

[4] M. Syauqi Hanif Ardani, M. A. Yaqin, and M. H. Suhartono, "Implementasi Graph Database Untuk Menentukan Rute Perjalanan Transportasi Umum," 2019.

[5] W. Wahyudi and F. Akbar, "Ekstraksi Basis Pengetahuan Ke Dalam Basisdata Graf Menggunakan Graf Property," *J. Nas. Teknol. dan Sist. Inf.*, vol. 5, no. 1, pp. 41–48, 2019, doi: 10.25077/teknosi.v5i1.2019.41-48.

[6] W. Li, T. Wang, J. Cao, and C. Tao, *A Visual Semantic Relations Detecting Method Based on Wordnet*, vol. 294 LNCIST. 2019.

[7] M. A. Yaqin and G. U. Abriani, "Analisis implementasi Metode Semantic Similarity untuk Pengukuran Kemiripan Makna antar Kalimat," *J. Comput. Sci. Appl. Informatics*, vol. 1, no. 2, p. 15, 2019.

[8] D. Zamzami, D. Puspitasari, J. T. Informasi, P. Studi, T. Informatika, and P. N. Malang, "Aplikasi Wordnet Indonesia Berdasarkan Kamus Thesaurus Bahasa Indonesia Menggunakan."

[9] H. Hendrik and A. B. Cahyono, "Model WordNet Bahasa Indonesia berbasis Linked Data," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 6, no. 1, pp. 8–14, 2017, doi: 10.22146/jnteti.v6i1.288.

[10] H. Larasati and S. Masripah, "Analisa Dan Perancangan Sistem Informasi Pembelian GRC Dengan Metode Waterfall," *J. Pilar Nusa Mandiri*, vol. 13, no. 2, pp. 193–198, 2017.

[11] "Pierre Bourque, Richard E. Fairley - Guide to the Software Engineering Body of Knowledge (SWEBOK(r))_ Version 3.0-IEEE Computer Society Press (2014).pdf." .

[12] H. P. Ludyanto, *Rancang Bangun Aplikasi User Dependency Tool untuk Database MS SQL Server Berbasis Graph Neo4j*. 2018.

[13] T. Slimani, "Description and Evaluation of Semantic Similarity Measures Approaches," *Int. J. Comput. Appl.*, vol. 80, no. 10, pp. 25–33, 2013, doi: 10.5120/13897-1851.

[14] S. M. Pamungkas, M. A. Yaqin, K. Z. Matondang, A. N. Anggraini, and A. C. Fauzan, "Analisis dan Perancangan Software WordNet Bahasa Indonesia dengan Graph Database," *Ilk. J. Comput. Sci. Appl. Informatics*, vol. 2, no. 2, pp. 198–209, 2020, doi: 10.28926/ilkomnika.v2i2.52.