

**RANCANG BANGUN PENCARIAN BERBASIS SEMANTIC UNTUK
DATA KEPEGAWAIAN MENGGUNAKAN METODE *COSINE*
*SIMILARITY***

SKRIPSI



**UIN SUNAN AMPEL
S U R A B A Y A**

Disusun Oleh:

**RIFKY FIRDAUS ALFALAKHI
H06216021**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL AMPEL
SURABAYA
2022**

PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini,

NAMA : Rifqi Firdaus Alfalaksi

NIM : H06216021

PROGRAM STUDI : Sistem Informasi

ANGKATAN : 2016

Menyatakan bahwa saya tidak melakukan plagiat dalam penulisan skripsi yang berjudul "*RANCANG BANGUN PENCARIAN BERBASIS SEMANTIC UNTUK DATA KEPEGAWAIAN MENGGUNAKAN METODE COSINE SIMILARITY*".

Apabila suatu saat nanti terbukti saya melakukan tindakan plagiat, maka saya bersedia menerima sanksi yang telah ditetapkan.

Demikian pernyataan keaslian ini saya buat dengan sebenar-benarnya.

Surabaya, 13 Januari 2023



Rifky Firdaus Alfalaksi

H06216021

LEMBAR PERSETUJUAN PEMBIMBING

Skripsi oleh

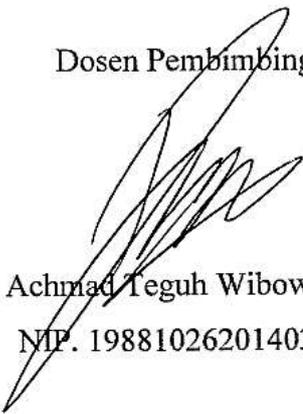
NAMA : RIFKY FIRDAUS ALFALAKHI
NIM : H06216021
JUDUL : RANCANG BANGUN PENCARIAN BERBASIS
SEMANTIC UNTUK DATA KEPEGAWAIAN
MENGUNAKAN METODE *COSINE SIMILARITY*

Ini telah diperiksa dan disetujui untuk diujikan.

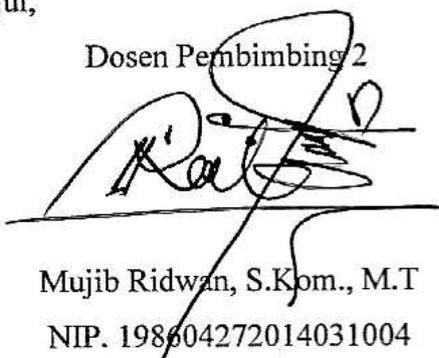
Surabaya, 13 Januari 2023

Menyetujui,

Dosen Pembimbing 1


Achmad Teguh Wibowo, MT
NIP. 198810262014031003

Dosen Pembimbing 2


Mujib Ridwan, S.Kom., M.T
NIP. 198604272014031004

PENGESAHAN TIM PENGUJI

Skripsi Rifky Firdaus Alfalaksi ini telah dipertahankan didepan tim penguji
skripsi di Surabaya, 13 Januari 2023

Mengesahkan,
Dewan Penguji

Penguji 1



Moch Yasin, S.Kom., M. Kom, M.B.A.

NIP. 198808302014031001

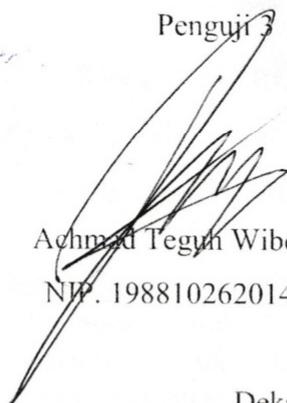
Penguji 2



Andhy Permadi, M.Kom

NIP. 198110142014031002

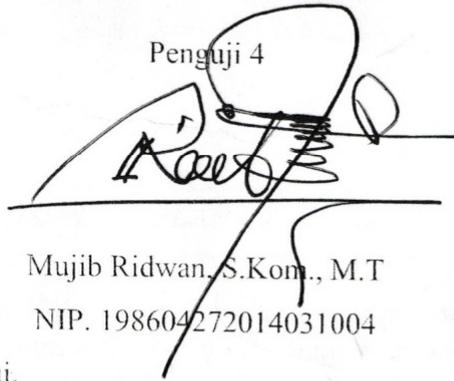
Penguji 3



Achmad Teguh Wibowo, MT

NIP. 198810262014031003

Penguji 4



Mujib Ridwan, S.Kom., M.T

NIP. 198604272014031004

Mengetahui,

Dekan Fakultas Sains dan Teknologi

Sunan Ampel Surabaya




Saepul Hamdani, M.Pd

NIP. 196507312000031002



UNIVERSITAS ISLAM NEGERI
SUNAN AMPEL SURABAYA

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
PERPUSTAKAAN

Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300
E-Mail: perpus@uinsby.ac.id

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama : RIFKY FIRDAUS ALFALAKHI
NIM : H06216021
Fakultas/Jurusan : FAKULTAS SAINS DAN TEKNOLOGI/SISTEM INFORMASI
E-mail address : firdausrifky670@gmail.com

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan UIN Sunan Ampel Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :

Skripsi Tesis Desertasi Lain-lain (.....)

yang berjudul :

RANCANG BANGUN PENCARIAN

BERBASIS SEMANTIC UNTUK DATA

KEPEGAWAIAN MENGGUNAKAN METODE *COSINE SIMILARITY*

berserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Eksklusif ini Perpustakaan UIN Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain secara *fulltext* untuk kepentingan akademis tanpa perlu meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan atau penerbit yang bersangkutan.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN Sunan Ampel Surabaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Surabaya, 13 Januari 2023
Penulis

(Rifky Firdaus Alfalaksi)

ABSTRAK

RANCANG BANGUN PENCARIAN BERBASIS *SEMANTIC* UNTUK DATA KEPEGAWAIAN MENGGUNAKAN METODE *COSINE SIMILARITY*

PT. Kecap Kenari merupakan salah satu perusahaan pemasok distribusi brand Kecap Kenari. Karyawan pada PT. Kecap Kenari memiliki tugas, latar belakang pendidikan, dan keahlian yang berbeda. Peneliti melakukan observasi pada PT. Kecap Kenari, ditemukan bahwa pencarian data karyawan masih bersifat manual berdasarkan *keyword* saja. Untuk mengatasi hal tersebut dibutuhkan fitur yang memungkinkan dalam pencarian data selain berdasarkan *keyword* juga berdasarkan persamaan makna semantik *keyword*. Sehingga hal ini dapat mendukung pencarian data dengan lingkup pencarian yang lebih lengkap. Beberapa tahapan utama dalam pencarian data pada penelitian ini adalah tahap *preprocessing text*, tahap pencarian makna semantik *keyword*, dan tahap pencocokan *string*. Pada tahap *preprocessing text* memanfaatkan *library PHP Sastrawi* untuk menormalisasikan teks, tahap pencarian makna semantik menggunakan *library PHP Satria/Synonim Antonym*, sedangkan pencarian data pegawai menggunakan metode *Cosine Similarity* sebagai teknik pencocokan *string* untuk mengetahui tingkat kemiripan antara data pegawai dengan *keyword* dan makna semantik *keyword* yang digunakan. Hasil dari penelitian ini, bahwa sistem berhasil normalisasi teks, menemukan makna semantik *keyword*, dan menemukan tingkat kemiripan data yang dicari dengan *cosine similarity*, serta penyajian data yang diurutkan dari tingkat kemiripan tertinggi, sehingga dapat memudahkan pengguna dalam membaca *output system* terkait pencarian data pegawai.

Kata Kunci: Pencarian Data Semantik, *Cosine Similarity*.

UIN SUNAN AMPEL
S U R A B A Y A

ABSTRACT
SEMANTIC-BASED SEARCH DESIGN FOR EMPLOYMENT DATA USING
THE COSINE SIMILARITY METHOD

PT. Kenari Ketchup is one of the suppliers and distributors of the Kenari Ketchup brand. Employees at PT. Kecap Kenari has different tasks, educational backgrounds, and expertise. Researchers made observations at PT. Kecap Kenari, it was found that the search for employee data was still manual based on keywords only. To overcome this, a feature is needed that allows data search besides being based on keywords, also based on the semantic similarities/meanings of keywords. So that it can support data search with a more complete search scope. Some of the main stages in searching for data in this study are the text preprocessing stage, the search for keyword semantic meanings, and the string matching stage. At the preprocessing stage the text utilizes the PHP Literary library to normalize the text, the semantic meaning search stage uses the PHP Satria/Synonym Antonym library, while the employee data search uses the Cosine Similarity method as a string matching technique to determine the level of similarity between employee data and keywords and the semantic meaning of the keywords used . The results of this study, that the system succeeded in normalizing the text, finding the semantic meaning of the keywords, and finding the level of similarity of the data sought with cosine similarity, as well as presenting data sorted from the highest level of similarity, so as to make it easier for users to read system output related to searching employee data.

Keywords: *Semantic Data Search, Cosine Similarity.*

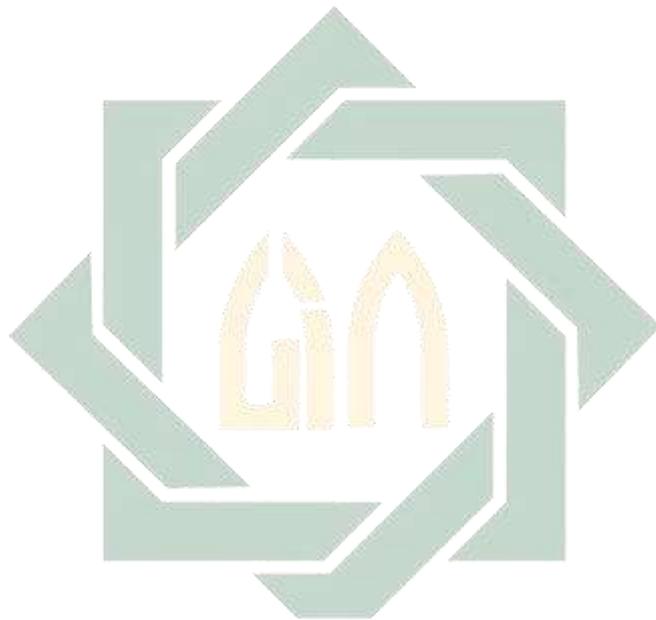
UIN SUNAN AMPEL
S U R A B A Y A

DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN.....	ii
LEMBAR PERSETUJUAN PEMBIMBING	iii
PENGESAHAN TIM PENGUJI	iv
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS	i
MOTTO	i
HALAMAN PERSEMBAHAN	ii
KATA PENGANTAR	iii
ABSTRAK	iv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	2
1.4 Batasan Masalah.....	3
1.5 Manfaat Penelitian.....	3
BAB II KAJIAN PUSTAKA.....	4
2.1 Tinjauan Penelitian Terdahulu	4
2.2 Pengertian Rancang Bangun	7
2.3 Pengertian Sistem	7
2.4 Pengertian Aplikasi	7
2.5 Pengertian Aplikasi Web	8
2.6 Sistem Pencarian	8
2.7 Semantik	9
2.8 <i>Web Semantic</i>	9
2.9 <i>Interoperability</i>	9
2.10 <i>Text Preprocessing</i>	9
2.11 Sastrawi	10
2.12 <i>Algoritma Cosine Similarity</i>	10
2.13 <i>Data Flow Diagram</i>	11
2.14 <i>Entity Relational Diagram (ERD)</i>	12

2.15	Flowchart (Diagram Alur).....	13
2.16	<i>Use Case Diagram</i>	13
2.17	PHP/ <i>Hypertext Preprocessor</i>	13
2.18	<i>Database</i>	14
2.19	MySQL.....	15
2.20	<i>Rapid Application Development (RAD)</i>	17
2.21	<i>Pseudocode</i>	18
2.22	<i>Codeigniter</i>	18
2.23	Pengertian MVC (<i>Model – View - Controller</i>).....	20
2.24	<i>Bootstrap</i>	22
2.25	Integrasi Keilmuan	22
BAB III METODOLOGI PENELITIAN.....		26
3.1	Objek Penelitian	26
3.2	Sumber Data	26
3.3	Instrumen Penelitian.....	26
3.4	Desain Penelitian.....	28
BAB IV HASIL DAN PEMBAHASAN		31
4.1.	<i>Requirement Planning</i>	31
4.1.1	<i>Data Flow Diagram</i>	31
4.1.2	<i>Entity Relationship Diagram</i>	36
4.1.3	<i>Flowchart Sistem</i>	37
4.1.4	<i>Use Case Diagram</i>	38
4.1.5	<i>Use Case Description</i>	39
4.1.6	Spesifikasi <i>Software dan Hardware</i>	40
4.2.	Pembangunan Aplikasi.....	50
4.2.1	Implementasi <i>Preprocessing Data</i>	50
4.2.2	Implementasi Pencarian Makna Semantik.....	51
4.2.3	Implementasi Pencocokan <i>String (Cossine Similarity)</i>	52
4.3.	Implementasi <i>Screenshot</i> Aplikasi	54
4.2.4	Implementasi Tampilan <i>User Interface</i>	54
4.2.5	Implementasi Tampilan <i>Admin Interface / Backend</i>	55
4.4.	Skenario Pengujian Sistem.....	59

4.5. Pembahasan	65
BAB V PENUTUP.....	66
5.1 Kesimpulan.....	66
5.2 Saran.....	66
DAFTAR PUSTAKA	v
LAMPIRAN.....	ix



UIN SUNAN AMPEL
S U R A B A Y A

DAFTAR TABEL

Tabel 2 1. Tinjauan Penelitian Terdahulu	4
Tabel 4 1. Tabel array nilai pencocokan string	52
Tabel 4 2. Pengujian fungsional sistem.....	60
Tabel 4 3. Requirements Traceability Matrix	62



UIN SUNAN AMPEL
S U R A B A Y A

DAFTAR GAMBAR

Gambar 2. 1. RAD Design	17
Gambar 2. 2 Framework Codeigniter.....	20
Gambar 2. 3. Konsep MVC	21
Gambar 3. 1 Struktur Organisasi.....	26
Gambar 3. 2 Metodologi Penelitian	28
Gambar 4. 1 Flowchart Sistem.....	37
Gambar 4. 2 Diagram konteks	31
Gambar 4. 3 DFD level 0	32
Gambar 4. 4 DFD level 1 dari proses 2.0.....	33
Gambar 4. 5 DFD level 1 dari proses 3.0.....	34
Gambar 4. 6 DFD level 2 dari proses 3.1, 3.2, 3.3, 3.4.....	35
Gambar 4. 7 ERD diagram sistem.....	36
Gambar 4. 8 Use Case Diagram sistem.....	38
Gambar 4. 9 Tampilan User (halaman pencarian)	54
Gambar 4. 10 Preprocessing data keyword.....	50
Gambar 4. 11 Hasil pencarian dan tingkat cosine similarity	53
Gambar 4. 12 Halaman login admin	55
Gambar 4. 13 Halaman submenu divisi	56
Gambar 4. 14 Tambah data divisi	56
Gambar 4. 15 Ubah data divisi.....	56
Gambar 4. 16 Hapus data divisi	57
Gambar 4. 17 Halaman submenu jabatan	57
Gambar 4. 18 Tambah data jabatan.....	57
Gambar 4. 19 Ubah data jabatan	58
Gambar 4. 20 Hapus data jabatan	58
Gambar 4. 21 Halaman submenu pegawai.....	58
Gambar 4. 22 Tambah data pegawai.....	59
Gambar 4. 23 Ubah data pegawai	59
Gambar 4. 24 Hapus data pegawai.....	59

BAB I PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, teknologi informasi memiliki peranan penting dan masif dalam pekerjaan manusia. Teknologi informasi secara umum dibedakan menjadi dua yakni perangkat lunak (pemroses informasi) dan perangkat keras (perangkat *input* dan *output*). Perangkat keras dapat diartikan sebagai seluruh perangkat fisik komputer, baik sebagai input maupun sebagai output informasi, sedangkan perangkat lunak adalah program komputer sebagai pemroses/ pengolah data dari hasil input yang akan ditampilkan pada perangkat output (Abdillah et al., 2020).

Berkaitan dengan perangkat lunak sekarang ini, semakin banyak data yang dimasukkan dalam database dapat menjadi problem tersendiri, yakni sulitnya dalam pencarian data terkait (Firman Nurdiyansyah, Samsul Arifin, 2018). Data yang semakin banyak dapat menyulitkan dalam pencarian jika belum terdapat pengelompokan atau kategori data didalamnya, sehingga memerlukan metode lain dalam pencarian data terkait. Salah satu metode dalam software yang dapat membantu dalam pencarian informasi terkait adalah Web Semantik (Arifin & Prasetyo, 2021).

Web Semantik merupakan sebuah teknologi dimana mesin komputer dapat memahami input yang dimasukkan (Pratama & Ropianto, 2020). Salah satu metode untuk pencarian data adalah *cosine similarity*. Algoritma *Cosine Similarity* adalah algoritma yang digunakan untuk menghitung kemiripan berdasarkan *vector space similarity measure* atau derajat kemiripan antara dua objek (Prasetya & Et.all, 2019).

PT. Kecap Kenari merupakan salah satu perusahaan pemasok distribusi brand Kecap Kenari. Karyawan pada PT. Kecap Kenari memiliki tugas, latar belakang pendidikan, dan keahlian yang berbeda. Peneliti melakukan observasi ke PT. Kecap Kenari, ditemukan bahwa pencarian data karyawan masih bersifat manual dengan pencarian Excel pada satu *worksheet* data pegawai berdasarkan *keyword* saja, sehingga lingkup pencarian terbatas sesuai *keyword* yang digunakan tanpa melihat makna semantik *keyword*. Untuk mengatasi hal tersebut dibutuhkan fitur atau

aplikasi yang memungkinkan dalam pencarian data selain dari *keyword* juga berdasarkan persamaan/ makna semantik *keyword* yang dimasukkan oleh pengguna. Penentuan makna semantik dari *keyword* dihasilkan dari interaksi dengan sistem lain yang memiliki *database* persamaan kata sesuai KBBI (*Interoperability*), sehingga hal ini dapat mendukung pencarian data dengan lingkup pencarian yang lebih lengkap.

Penelitian pernah dilakukan oleh Rais Hafiyyan tahun 2019, dengan judul penerapan metode *SWFilter System* pada fungsi pencarian data. Hasil pengujian menunjukkan bahwa metode *SWFilter* dapat menjadi salah satu solusi pada permasalahan pencarian semantik di web serta memiliki akurasi yang cukup baik dalam hasil pencarian (Hafiyyan et al., 2019).

Dari uraian di atas, masalah yang muncul adalah belum adanya pencarian data semantik pada perusahaan yang memungkinkan pengguna menemukan informasi terkait dengan lingkup lebih luas daripada sekedar *keyword* yang dimasukkan. Diharapkan teknologi *web semantic* ini dapat menjadi solusi dan membawa kemudahan dalam pencarian informasi. Objek penelitian ini adalah sistem yang dirancang khusus untuk pencarian data karyawan perusahaan.

1.2 Rumusan Masalah

Berdasarkan latar belakang, dapat dirumuskan masalah sebagai berikut:

1. Bagaimana sistem dapat menghasilkan informasi dari *keyword* berupa persamaan kata *keyword* sesuai dengan makna *semantic keyword*?
2. Bagaimana sistem dapat menghasilkan nilai pencocokan *string* berbasis semantik pada data pegawai menggunakan metode *cosine similarity*?

1.3 Tujuan Penelitian

Dari rumusan masalah, maka tujuan penelitian ini adalah sebagai berikut:

1. Mengetahui informasi dengan cakupan lebih luas yakni dari *keyword* serta dari persamaan kata *keyword* sesuai dengan maknanya.
2. Mengetahui nilai pencocokan *string* berbasis semantik dengan metode *cosine similarity*.

1.4 Batasan Masalah

Pada penulisan tugas akhir ini akan membatasi topik pembahasan hanya pada:

1. Sistem informasi *web semantic* mencakup tentang data pegawai, nama, jenis_kelamin, jabatan, departemen, divisi, latar belakang pendidikan, dan pengalaman pekerjaan yang terdapat di PT. Kecap Kenari.
2. Sistem informasi *web semantic* pencarian data pegawai menggunakan metode *cosine similarity*.
3. Pembuatan sistem berpusat pada penerapan *keyword semantic search*.
4. Data penyimpanan data pegawai tersimpan di MySQL.

1.5 Manfaat Penelitian

Penelitian ini diharap mampu memberi manfaat, baik sisi akademis maupun sisi aplikatif yaitu:

1.5.1 Manfaat Akademis.

1. Menambah wawasan tentang proses implementasi *web search Semantic Interoperability* pada “Rancang Bangun Pencarian Berbasis Semantic untuk Data Kepegawaian Menggunakan Metode *Cosine Similarity*”.
2. Mengembangkan hasil dari penelitian sebelumnya untuk menemukan informasi berdasarkan makna/persamaan *input search keyword*.

1.5.2 Manfaat Praktis.

1. Membantu pihak perusahaan agar dapat menemukan informasi seputar pegawai yang berkaitan dengan *input keyword* untuk keperluan pihak perusahaan khususnya divisi *Human Resource Development*.
2. Membantu pihak perusahaan mendapatkan informasi alternatif yang memiliki persamaan kata dengan *keyword* yang dimasukkan.

BAB II KAJIAN PUSTAKA

2.1 Tinjauan Penelitian Terdahulu

Sebagai acuan penelitian dan korelasinya dengan penelitian terkait web semantik. Berikut ini beberapa penelitian sebelumnya yang terkait dengan Semantik Web:

Tabel 2 1. Tinjauan Penelitian Terdahulu

Penelitian	Hasil	Korelasi Penelitian
Pencarian Lowongan Pekerjaan Berbasis Agen Berdasarkan Profil Pencari Kerja dengan Pendekatan Semantic Web Service (Susyanto & Mustofa, 2016)	Penelitian ini telah berhasil menghasilkan prototipe aplikasi yang dapat menghasilkan saran lowongan pekerjaan berdasarkan profil pencari kerja	Sama-sama mencari rekomendasi, namun pada penelitian ini memfokuskan pada rekomendasi pegawai berdasarkan keyword dan adanya pengembangan <i>interoperability</i> dengan sistem aplikasi lain
Penerapan Semantic Web Untuk Pencarian Data Akademik Pada Institusi Perguruan Tinggi (Nisa, 2020)	Konversi <i>query natural language</i> ke dalam bentuk SPARQL menggunakan <i>algoritma Booyer Moore</i> , dengan data uji 20 soal dan 19 soal menghasilkan data yang sesuai, tingkat kelulusan <i>Passage</i> adalah 95%.	Juga melakukan metode konversi untuk pencarian keyword dengan query pada database, yang mana keyword merupakan hasil konversi dari sistem lain dengan memanfaatkan daftar persamaan kata sesuai KBBI (<i>interoperability</i>) dalam mencari makna semantik.
Perancangan Web Semantik Aplikasi Pencarian	Bahwa teknologi web semantic dapat diterapkan	Juga menggunakan pengujian fungsionalitas

<p>Tumbuhan Obat Menggunakan Protege 4.3.0 (Jusuf & Azimah, 2016)</p>	<p>Sebagai teknologi pencarian. Pengujian aplikasi dilakukan dengan melihat fungsionalitas sistem dari pengguna web. Dari 10 pengguna menunjukkan hasil kuisioner yang diperoleh, memiliki nilai rata-rata 3 yang artinya fungsionalitas aplikasi Web berjalan baik.</p>	<p>pengguna, namun dengan tambahan perbandingan data cakupan informasi dari pencarian <i>keyword</i> masukkan saja dibandingkan dengan pencarian dari keyword masukan dan makna semantik <i>keyword</i>.</p>
<p>Implementasi Semantik Web untuk Pencarian Koleksi Perpustakaan Universitas Muria Kudus (Izza et al., 2022)</p>	<p>Menghasilkan aplikasi Pencarian dengan Teknologi <i>Semantic Web Based</i> menggunakan Bahasa pemrograman PHP.</p>	<p>Juga menggunakan Bahasa pemrograman PHP namun dengan interaksi dengan aplikasi lain (<i>interoperability</i>) dalam menemukan makna <i>semantic</i> inputan <i>keyword</i>.</p>
<p>Penerapan metode Semantic Web Filtering System (SWFilter) pada fungsi pencarian data : Studi Kasus pada websitetelkomtesthouse.co.id (Hafiyyan et al., 2019)</p>	<p>Menghasilkan pencarian dengan system SWFilter (<i>Semantic Web Filter</i>), yakni Setiap frase yang dimasukkan pengguna akan ditangani oleh pengindeksan kata kunci semantic. Selanjutnya dengan proses <i>filtering</i> untuk menyaring <i>keyword</i> dengan <i>web service</i></p>	<p>Juga menggunakan model <i>system filtering</i> pada dataset yang ditampilkan serta rancangan sistem IR (<i>Indexing Retrieval</i>) atau system temu balik informasi yang relevan dengan kebutuhan pengguna.</p>

Berdasarkan tinjauan penelitian terdahulu pada Tabel 2.1, terdapat beberapa hasil yang kemudian akan dikembangkan lebih lanjut yakni dari segi pembangunan aplikasi, penulis melihat penelitian (Susyanto & Mustofa, 2016) berjudul “Pencarian Lowongan Pekerjaan Berbasis Agen Berdasarkan Profil Pencari Kerja dengan Pendekatan Semantic Web Service”, berhasil menghasilkan prototipe aplikasi yang dapat menghasilkan saran lowongan pekerjaan berdasarkan profil

pencari kerja. Dan pada penelitian (Izza et al., 2022) berjudul “Implementasi Teknologi Semantik Web untuk Pencarian Koleksi Perpustakaan Universitas Muria Kudus”, menghasilkan aplikasi Pencarian dengan Teknologi *Semantic Web* menggunakan bahasa pemrograman PHP. Dengan melihat korelasi terhadap penelitian ini yakni guna menampilkan data subjek pegawai perusahaan terkait serta pembuatan aplikasi berbasis PHP, akan dicoba untuk membuat integrasi atau interaksi dengan sistem aplikasi lain dalam menemukan makna semantik dari *keyword* yang diberikan (*interoperability*).

Kemudian dari segi pemrosesan data, pada penelitian (Nisa, 2020) berjudul “Penerapan *Semantic Web* Untuk Pencarian Data Akademik Pada Institusi Perguruan Tinggi”, mengkonversi query bahasa alami menjadi bentuk SPARQL menggunakan algoritma Booyer Moore. Sedangkan dalam penelitian ini melakukan metode konversi *keyword* dengan *query* pada *database*, yang mana *keyword* merupakan hasil konversi dari sistem lain dengan pengembangan memanfaatkan daftar persamaan kata sesuai KBBI (*interoperability*) dalam mencari makna semantik.

Selanjutnya dari segi penerapan atau output yang dihasilkan, mengambil rujukan pada penelitian (Hafiyyan et al., 2019) berjudul “Penerapan metode *Semantic Web Filtering System (SWFilter)* pada fungsi pencarian data: Studi Kasus pada websitetelkomtesthouse.co.id”, menghasilkan pencarian data dengan sistem *SWFilter (Semantic Web Filter)*, yakni setiap kalimat yang diinputkan pengguna akan dilanjutkan pada *proses indexing keyword* dengan rancangan penyajian output informasi sistem IR (*Indexing Retrieval*) atau sistem temu balik. Sedangkan pada penelitian ini, akan mencoba menemukan informasi terkait baik dari *keyword* kalimat maupun makna semantik tiap *keyword* dalam kalimat menggunakan model *system filtering* pada data ditampilkan serta perancangan sistem IR (*Indexing Retrieval*) atau sistem temu balik informasi yang relevan dengan *keyword* serta makna *keyword* masukan.

Serta dari segi pengujian sistem, merujuk pada penelitian (Jusuf & Azimah, 2016) berjudul “Perancangan Web Semantik Aplikasi Pencarian Tumbuhan Obat Menggunakan Protege 4.3.0”, menghasilkan pengujian dengan model pengujian fungsionalitas aplikasi oleh 10 orang pengguna dengan pengisian kuesioner.

Sedangkan pada penelitian ini akan dicoba selain menggunakan pengujian fungsionalitas pengguna, akan ditambah dengan perbandingan data cakupan informasi dari hasil pencarian *keyword* masukkan saja dibandingkan dengan pencarian dari *keyword* masukan dan makna semantik *keyword*.

2.2 Pengertian Rancang Bangun

Dalam jurnalnya (Nurhayati et al., 2018) bahwa “Rancang bangun adalah kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada. Serta dapat dipahami juga bahwa Rancang Bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi”. Dari pengertian diatas Penulis menyimpulkan bahwa Rancang Bangun adalah suatu tahapan lengkap dalam menganalisis, membuat, dan mengembangkan aplikasi sesuai dengan fungsi yang diharapkan.

2.3 Pengertian Sistem

Dalam penelitian (Jembarnata, 2011), “bahwa sistem adalah istilah yang digunakan untuk menggambarkan suatu kombinasi elemen-elemen terhubung yang bekerja sama untuk mencapai tujuan. Sebuah sistem dapat memiliki banyak subsistem dan sub-subsistem. Subsistem ini melengkapi sistem dan saling berhubungan serta harus bekerja sama dengan subsistem lain untuk membuat sebuah sistem yang lengkap.

Dapat dipahami juga bahwa sistem adalah sekelompok komponen-komponen yang saling berhubungan dan bekerja sama untuk mencapai suatu tujuan dengan menerima masukan (input) dan menghasilkan keluaran (output) pada sebuah proses transformasi (perubahan) tertentu. Input terkait dengan elemen-elemen yang memasuki sistem untuk selanjutnya di proses. Sedangkan, proses berkaitan dengan proses transformasi elemen-elemen menjadi output. Lalu, Output berperan dalam memindahkan elemen-elemen yang dihasilkan dari proses transformasi menuju suatu tujuan”.

2.4 Pengertian Aplikasi

Dalam penelitian (Jembarnata, 2011), “Kata Aplikasi berasal dari bahasa inggris, yaitu *To applicate* yang artinya menerapkan atau terapan. Namun

pengertian mengenai aplikasi secara umum adalah suatu paket program yang sudah jadi dan dapat digunakan. Sedangkan arti Aplikasi adalah program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu.

Komputer itu sendiri kaitannya dengan aplikasi yang terdiri dari beberapa unit fungsional untuk mencapai tujuan pelaksanaan pengolahan data yaitu:

1. Bagian yang membaca data (Input data atau input unit)
2. Bagian yang mengolah data (Control processing unit)
3. Bagian yang mengeluarkan hasil pengolahan data (Output data)”.

2.5 Pengertian Aplikasi Web

Dalam penelitian (Jembarnata, 2011), “bahwa aplikasi web (bahasa Inggris: web application atau sering disingkat webapp) adalah suatu aplikasi yang diakses menggunakan browser melalui suatu jaringan seperti Internet atau intranet. Ia juga merupakan suatu aplikasi perangkat lunak komputer yang dikodekan dalam bahasa yang didukung penjelajah web (seperti HTML, JavaScript, AJAX, Java, dll) dan bergantung pada penjelajah tersebut untuk menampilkan aplikasi.

Aplikasi web menjadi populer karena kemudahan akses bagi kliennya yakni menggunakan browser. Kemampuan untuk memperbaiki dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstalasi perangkat lunak pada kemungkinan ribuan komputer klien merupakan alasan kunci popularitasnya. Aplikasi web yang umum misalnya webmail, toko retail online, lelang online, wiki, papan diskusi, weblog, serta MMORPG”.

2.6 Sistem Pencarian

Sistem pencarian adalah system komputasi yang diperuntukan agar dapat menghasilkan informasi yang hendak dicari. Menurut Basmalah (Basmalah Wicaksono et al., 2016), “Dengan adanya sistem ini, proses pencarian informasi dapat menjadi lebih mudah yakni dengan memproses *input keyword* untuk mendapatkan informasi yang relevan dalam suatu aplikasi. Misalnya kita ingin mencari sebuah informasi tentang sebuah produk barang di sistem e-commerce untuk cara cepat kita memasukkan kata kunci pada menu pencarian, jika kata kunci yang dimasukkan tidak ada, maka produk tersebut tidak ada pada daftar sistem. Perilaku pencarian informasi juga dapat dipahami sebagai aktivitas yang

mengidentifikasi dan mendefinisikan pesan untuk memenuhi kebutuhan informasi kognitif’.

2.7 Semantik

Kata semantik secara bahasa berasal dari Yunani yakni kata *semantikos* yang berarti memberi tanda. Menurut Ayuwandira (Ayuwandira et al., 2021), “Dengan kata lain, semantik adalah ilmu yang mempelajari makna dalam kata, kode, dan jenis representasi yang lain. Dalam linguistik, semantik adalah sub bidang yang mempelajari tentang makna, seperti yang melekat di tingkat kata, frasa, kalimat, dan teks. Tujuan secara umum dari pencarian informasi adalah mencari informasi yang mendukung dalam menyelesaikan tugas maupun kepentingan tertentu ketika seseorang tidak memiliki pengetahuan yang cukup atau membutuhkan informasi tambahan”.

2.8 Web Semantic

Web Semantic mengacu pada teknik yang memungkinkan komputer untuk lebih memahami konten halaman web (Saputra et al., 2020). Semantic Web memiliki konten yang diekspresikan baik dalam bahasa alami yang dimengerti manusia, serta dalam bentuk perangkat lunak yang dapat dipahami, ditafsirkan, dan digunakan dalam indexing komputer. Berkat web semantik, berbagai perangkat lunak dapat menemukan keterkaitan berbagai informasi yang saling terintegrasi dengan lebih sederhana (Paramartha & Wiguna, 2020).

2.9 Interoperability

Interoperabilitas adalah suatu interaksi antar aplikasi melalui protokol yang disepakati. Menurut Fibrianto (Fibrianto et al., 2022) “Interoperabilitas merupakan hal yang kompleks dalam pertukaran data antar sistem dengan platform yang berbeda, seperti pada e-banking dan e-government. Tidak mungkin menyeragamkan format dan skema data di semua sistem, juga tidak dapat diseragamkan di seluruh sistem. Pendekatan yang diterima adalah menstandarisasi format data yang dipertukarkan memberikan jawaban yang benar”.

2.10 Text Preprocessing

Menurut (Priyanto & Ma’arif, 2018) dalam jurnalnya “pada tahap *text preprocessing* memiliki fungsi yakni sebagai tahap awal pengolahan teks sebelum diolah lebih lanjut. Data teks tidak terstruktur memiliki noise seperti tanda baca,

angka, imbuhan, karakter khusus dan lain sebagainya. Dalam tahapan ini, data teks dibersihkan hingga tersisa bentuk dasar saja yang kemudian dapat dianalisis lebih lanjut. Pada penelitian ini tahapan *text processing* menggunakan library sastrawi yang diintegrasikan dengan aplikasi berbasis web PHP terdiri dari 4 langkah, yaitu *Case folding*, *Stemming*, *Filtering*, dan *Tokenizing* sebagai berikut”.

a. *Case Folding*

Case folding merupakan proses untuk menghilangkan karakter-karakter selain huruf, seperti tanda baca, angka, dan menyeragamkan semua teks menjadi huruf besar atau sebaliknya (Rahman, 2017).

b. *Stemming*

Stemming adalah proses untuk menghilangkan imbuhan kata atau mengubah kata berimbuhan menjadi kata dasar (Rahman, 2017).

c. *Filtering*

Filtering atau *stopword removal* dapat dipahami sebagai proses penghilangan *stopwords* atau kata-kata yang tidak menggambarkan isi tulisan, sehingga dapat dibuang dengan pendekatan *bag-of-words*. Proses ini bertujuan untuk mengurangi jumlah kata.

d. *Tokenizing*

Tokenizing/ Tokenization merupakan proses untuk memisahkan setiap kata dalam kalimat (Rahman, 2017).

2.11 Sastrawi

Menurut (Kirana et al., 2019), “*Library Sastrawi* merupakan *library* yang mempunyai peran penting di dalam proses *Text Mining*, terutama di dalam pengolahan teks Bahasa Indonesia”. *Library Sastrawi* didalam penelitian ini digunakan pada tahap *Stemming* dan *Filtering*, dimana pada tahap tersebut berfungsi untuk menghapus kata-kata imbuhan atau menjadikan suatu kata ke bentuk kata dasar dan menghilangkan kata penghubung yang tidak mencerminkan isi masukan keyword.

2.12 Algoritma *Cosine Similarity*

Menurut (Nurdiana, 2016), “Algoritma *cosine similarity* merupakan alur atau algoritma untuk menghitung derajat kemiripan atau kemiripan antara dua objek. Perhitungan berdasarkan ukuran kesamaan ruang vektor”. Rumus untuk

menghitung tingkat kemiripan teks dokumen dengan query tertentu adalah sebagai berikut.

$$\text{cosSim}(d_i, d_j) = \frac{\sum_{i=1}^n (td_{ij} \times tq_{ik})}{\sqrt{\sum_{i=1}^n td_{ij}^2 \times \sum_{i=1}^n tq_{ik}^2}} \quad (1)$$

Keterangan :

cosSim(dj, dk) = tingkat kesamaan query dengan dokumen tertentu

tdij = kata ke-i dalam vektor untuk dokumen ke-j

tqik = kata ke-i dalam vektor untuk query ke-k

n = jumlah term yang unik dalam dataset

Dalam jurnal (Nugroho et al., 2021), “bahwa langkah-langkah perhitungan manual algoritma Cosine Similarity antara lain:

1. Langkah pertama terlebih dahulu kita tentukan *query* dari jawaban (A) dan query dari key jawaban (B) dan kemudian gabungan dari kedua query tersebut (*Queries*).
2. Dari ketiga query yang ada selanjutnya dilakukan penghapusan stoplist atau tanda baca yang tidak mempengaruhi penilaian, seperti tanda titik(.), tanda koma(,), tanda seru(!), dan tanda baca lainnya.
3. Menghilangkan stopword (kata-kata umum) yang lazim digunakan pada ketiga query tersebut kata-kata umum dalam suatu query, seperti "di", "dan", "tetapi", "jika", "namun", dan kata hubung lainnya.
4. Menghitung nilai term frequency dari query jawaban dan query key jawaban terhadap queries. e)Hitung nilai banyaknya file (N) atau *document frequency* (n) yang memiliki suatu term untuk tiap term dalam queries.
5. Menghitung nilai *inverse document frequency* dengan rumus $\log (N/n) + 1$.”.

2.13 Diagram Konteks

Menurut (Safwandi, 2021), “bahwa Diagram Konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram Konteks merupakan lever I tertinggi dari DFD yang menggambarkan seluruh Input ke sistem atau Output dari sistem. Dalam diagram konteks berisi gambaran umum (secara garis besar) sistem yang akan dibuat. Diagram konteks ini berisi siapa saja

yang memberi data (dan data apa saja) ke sistem, serta kepada siapa saja informasi (dan informasi apa saja) yang harus dihasilkan sistem”.

2.14 Data Flow Diagram

Menurut (Priandika & Wantoro, 2017) dalam jurnalnya “DFD merupakan gambaran pembagian sistem ke modul yang lebih kecil. Satu kelebihan dari DFD yaitu memudahkan pemakai yang kurang memahami bidang komputer untuk mengerti system yang akan dibangun”.

Menurut (Permana & Sumaryana, 2018) dalam jurnalnya “DFD digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misalnya lewat telpon, surat dan sebagainya) atau lingkungan fisik dimana data tersebut akan disimpan (misalnya file kartu, microfiche, hardisk, tape, diskette, dan lain sebagainya). DFD merupakan alat yang cukup populer, karena dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas. DFD juga merupakan dokumentasi dari sistem yang baik”.

2.15 Entity Relational Diagram (ERD)

Menurut (Fajri, 2016) “ERD merupakan model data yang menggunakan beberapa notasi untuk menggambarkan data dalam konteks entitas dan hubungan yang dideskripsikan oleh data tersebut”.

Menurut (Jembarnata, 2011) “Elemen-elemen ERD antara lain:

1. Entitas (*Entity*) Pada ERD, entity digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. *entity* diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu: orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya).
2. Hubungan (*Relationship*), *Relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya *Relationship* diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau kalimat pasif).
3. Atribut, Secara umum atribut adalah sifat atau karakteristik dari tiap entitas

maupun tiap *relationship*. Maksudnya, atribut adalah sesuatu yang menjelaskan apa yang sebenarnya yang dimaksud entity maupun *relationship*, sehingga sering dikatakan atribut adalah elemen dari setiap entity dan *relationship*.

4. Kardinalitas (*Cardinality*), Kardinalitas mendefinisikan jumlah kemunculan baik minimum maupun maksimum satu entitas yang dapat dihubungkan dengan kemunculan tunggal entitas lain”.

2.16 Flowchart (Diagram Alur)

Menurut (Jembarnata, 2011) “*Flowchart* (Diagram Alur) adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* juga merupakan sebuah diagram dengan simbol grafis yang menyatakan tipe operasi program yang berbeda. Sebagai representasi dari sebuah program, *flowchart* dapat menjadi alat bantu untuk memudahkan perancangan alur urutan logika suatu program, memudahkan pelacakan sumber kesalahan program, dan alat untuk menerangkan logika program”.

2.17 Use Case Diagram

Menurut (Djaelangara et al., 2015) bahwa “*Use case* adalah rangkaian/ uraian kelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Use case* digunakan untuk membentuk tingkah-laku benda/ *things* dalam sebuah model serta direalisasikan oleh sebuah collaboration. Umumnya *use case* digambarkan dengan sebuah *elips* dengan garis yang solid, biasanya mengandung nama. *Use case* gambar 1 menggambarkan *proses system* (kebutuhan sistem dari sudut pandang *user*)”.

2.18 PHP/ Hypertext Preprocessor

PHP adalah pemrograman berbasis web yang berekstensi file.php yang dapat diakses dengan *web server*. Menurut Hidayatullah (Hidayatullah, Priyanto, 2017). bahwa “PHP *Hypertext Preprocessor* adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*”.

Menurut Maimunah (Maimunah, David Ericson Manalu, 2017) “Pengertian Pemrograman PHP diambil dari singkatan *Hypertext Preprocessor* yaitu bahasa pemograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML*

embedded scripting). Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru *uptodate*. Semua script dieksekusi pada *server* dimana *script* tersebut dijalankan.

PHP termasuk dalam *open source product*, jadi pengguna dapat merubah *source code* dan mendistribusikannya secara bebas. PHP juga diedarkan secara gratis. Pengguna bisa mendapatkannya secara gratis. PHP juga dapat berjalan di berbagai *web server* semisal IIS, Apache, PWS, dan lain-lain. Adapun kelebihan-kelebihan dari PHP yaitu:

- a. PHP mudah dibuat dan kecepatan akses tinggi.
- b. PHP dapat berjalan dalam *web server* yang berbeda dan dalam sistem operasi yang berbeda pula. PHP dapat berjalan di sistem operasi UNIX, Windows 98, Windows NT dan Macintosh.
- c. PHP diterbitkan secara gratis.
- d. PHP juga dapat berjalan pada *web server* Microsoft Personal Web Server, Apache, IIS, Xitami dan sebagainya.
- e. PHP adalah termasuk bahasa yang *embedded* (bisa di tempel atau diletakan dalam tag HTML).
- f. PHP termasuk *server-side programming*".

2.19 Database

Basis data (*database*) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang akan disimpan. Basis data merupakan aspek yang sangat penting dalam sistem informasi karena berfungsi sebagai gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat mengorganisasi data, menghindari duplikasi data, menghindari hubungan antar data yang tidak jelas dan juga update yang rumit.

Proses memasukkan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang disebut dengan sistem manajemen basis data (*database management system / DBMS*). *DBMS* merupakan sistem perangkat lunak yang memungkinkan pengguna basis data (*database user*) untuk memelihara,

mengontrol dan mengakses data secara praktis dan efisien. Dengan kata lain, semua akses ke basis data akan ditangani oleh *DBMS*. *DBMS* ini menjadi lapisan yang menghubungkan basis data dengan program aplikasi untuk memastikan bahwa basis data tetap terorganisasi secara konsisten dan dapat diakses dengan mudah.

Menurut (Yanto, 2016), basis data merupakan “kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi), untuk memenuhi berbagai kebutuhan”. Sedangkan menurut (Lubis, 2016), basis data adalah “tempat berkumpulnya data yang saling berhubungan dalam suatu wadah (organisasi/ perusahaan) bertujuan agar dapat mempermudah dan mempercepat untuk pemanggilan atau pemanfaatan kembali data tersebut”.

2.20 MySQL

MySQL adalah salah satu jenis database *server*. MySQL termasuk jenis *RDBMS (Relational Database Management System)*. MySQL adalah Sebuah program database *server* yang mampu menerima dan mengirimkan datanya sangat cepat, *multi user* serta menggunakan perintah dasar *SQL (Structured Query Language)*. *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* (bahasa Inggris: *database management system*) atau *DBMS* yang *multithread, multiuser*, dengan sekitar 6 juta instalasi diseluruh dunia. *MySQL*.

Menurut (Kadir, 2014) “MySQL merupakan software yang tergolong *database server* dan bersifat *open source*”. Sedangkan menurut Saputra dan Agustin (dalam Hinsa dan Ahmad Ishag 2016:65) “MySQL bukan termasuk bahasa pemrograman. MySQL merupakan salah satu database populer dan mendunia. MySQL bekerja menggunakan *SQL Language (Structure Query Language)*. Itu dapat diartikan bahwa MySQL merupakan standar penggunaan *database* di sunia untuk pengolahan data”. Dari definisi para ahli di atas, dapat disimpulkan bahwa MySQL adalah tipe data relasional, kode sumber terbuka, sehingga banyak digunakan di dunia.

MySQL memiliki perbedaan mendasar dengan RDMS yang lain. Sebagai contoh perbedaannya dengan *SQL SERVER* sendiri menurut (Hostinger, 2023) “bahwa ada beberapa perbedaan antara lain:

- 1 Biaya

Perbedaan yang paling jelas antara SQL Server dan MySQL tentu adalah biayanya. Microsoft mengharuskan user membeli lisensi untuk mengakses fitur lengkap SQL Server. Sebaliknya, MySQL bersifat open-source dengan lisensi General Public License sehingga bebas untuk digunakan.

2 Kompatibilitas Default

SQL Server utamanya ditujukan bagi developer yang menggunakan .NET sebagai bahasa pengembangannya. Sementara itu, MySQL bisa digunakan dengan bahasa apa saja, terutama PHP. MySQL kompatibel dengan semua sistem operasi utama, meskipun biasanya diasosiasikan dengan Linux. Di sisi lain, SQL Server dulu hanya bisa berjalan di Windows, sebelum Microsoft meluncurkan dukungan untuk Linux dan Mac di tahun 2016.

3 Sintaksis

Contoh perbedaan pada query untuk memanggil data dengan jumlah tertentu (limit). Pada SQL SERVER menggunakan SELECT TOP 3 WITH TIES, sedangkan pada MySQL cukup dengan LIMIT 3.

4 Mesin Penyimpanan

SQL Server menggunakan mesin penyimpanan tunggal yang dikembangkan oleh Microsoft. Sebaliknya, MySQL memberikan kebebasan kepada developer untuk memilih mesin apa pun untuk berbagai tabel berbeda, yang didasarkan pada kecepatan, keandalan, atau dimensi lainnya. Dua mesin penyimpanan MySQL yang paling populer adalah InnoDB dan MyISAM.

5 Metode Filtering

User, tabel, dan baris MySQL bisa difilter dengan beberapa cara. Namun, MySQL mengharuskan user menjalankan beberapa query pada tiap database secara terpisah. Ini berbeda dengan filtering SQL Server yang dilakukan database per database, dan didasarkan pada baris.

SQL Server menyederhanakan keseluruhan prosesnya sehingga developer bisa memfilter baris tanpa perlu mempertimbangkan jumlah database yang dihosting di server.

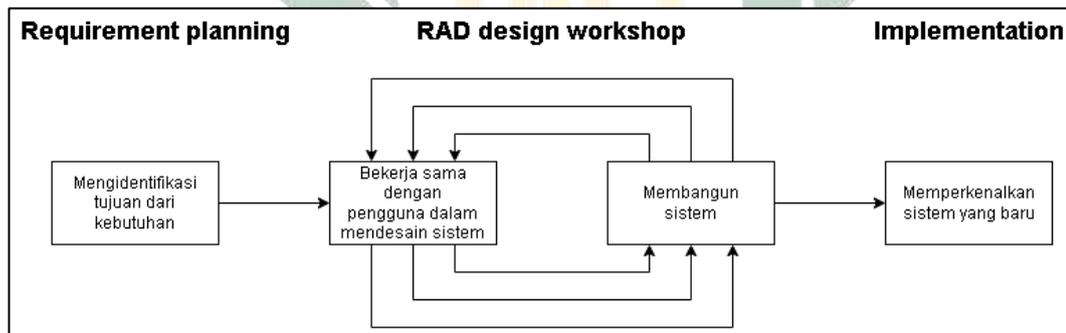
6 Kesimpulan

Prinsipnya, kalau Anda akan mengembangkan aplikasi ringan atau menengah dan menggunakan PHP dalam sebagian besar prosesnya, pilihlah MySQL.

Sebaliknya, kalau Anda tertarik mengembangkan aplikasi perusahaan berskala besar yang aman dan tangguh, SQL Server bisa menjadi pilihan yang sesuai”.

2.21 Rapid Application Development (RAD)

Metode pengembangan sistem dalam penelitian ini menggunakan *Rapid Application Development (RAD)*. Menurut (Jembarnata, 2011) “RAD adalah pendekatan object-oriented pada pengembangan sistem yang meliputi metode pengembangan serta perangkat lunak. Masuk akal jika mendiskusikan RAD dan prototyping pada bahasan yang sama, karena kedua konsep tersebut mendekati. Keduanya memiliki tujuan yang sama yaitu mempersingkat waktu yang biasanya dibutuhkan dalam SDLC tradisional pada saat desain dan implementasi pada sistem informasi. Pada akhirnya, baik RAD dan *prototyping* berusaha memenuhi kebutuhan bisnis yang berubah cepat secara lebih baik. Sekali anda mempelajari konsep *prototyping*, maka lebih mudah untuk memahami maksud dari RAD, yang dapat dianggap sebagai implementasi spesifik dari *prototyping*”. Tahapan RAD sendiri dapat dilihat pada Gambar 2.1 berikut:



Gambar 2. 1. RAD Design

Sumber: Jembarnata (2011). (diolah kembali)

Menurut (Jembarnata, 2011) “bahwa ada tiga tahapan dalam RAD yang melibatkan pengguna dan analisis yaitu:

1. Tahapan Requirement Planning, ditahap ini pengguna dan analisis bertemu untuk menentukan tujuan dari aplikasi atau sistem, juga untuk menentukan informasi yang dibutuhkan berdasarkan tujuan-tujuan tersebut. Pada tahapan ini dibutuhkan keterlibatan yang intens dari kedua belah pihak.
2. Tahapan RAD Design Workshop, ditahap ini pengguna merespon model prototipe dan analisis menyempurnakan modul yang sudah dirancang agar

sesuai dengan respon dari pengguna.

3. Tahapan Implementation, ditahap ini seorang analis mengimplementasikan model prototipe yang telah disetujui pengguna pada sistem yang dibangun”.

2.22 Pseudocode

Menurut (Handayani & Anggriani, 2015) bahwa “*Pseudocode* adalah deskripsi dari algoritma pemrograman komputer yang menggunakan struktur sederhana dari beberapa bahasa pemrograman, tetapi bahasa tersebut hanya ditujukan agar dapat mudah dibaca manusia. Perbedaannya terletak pada cara penyampaiannya, *pseudocode* menggunakan kata-kata untuk menjelaskan suatu algoritma, sedangkan *flowchart* menggunakan gambar. Tujuan penggunaan utama dari *pseudocode* sendiri adalah untuk memudahkan manusia dalam memahami prinsip-prinsip dari suatu algoritma”.

2.23 Codeigniter

Menurut (Sahi, 2020) dalam jurnalnya membahas terkait *codeigniter* bahwa “*CodeIgniter* adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis PHP dibanding jika menulis semua kode program dari awal.

CodeIgniter pertama kali dibuat oleh Rick Ellis, CEO Ellislab, Inc. (<http://ellislab.com>), sebuah perusahaan yang memproduksi CMS (*Content Management System*) yang cukup handal, yaitu Expression Engine (<http://www.expressionengine.com>). Saat ini, *CodeIgniter* dikembangkan dan dikembangkan oleh *Expression Engine Development Team*.

Adapun beberapa keuntungan menggunakan *CodeIgniter*, diantaranya:

1. Gratis

CodeIgniter berlisensi di bawah *Apache/BSD open source*.

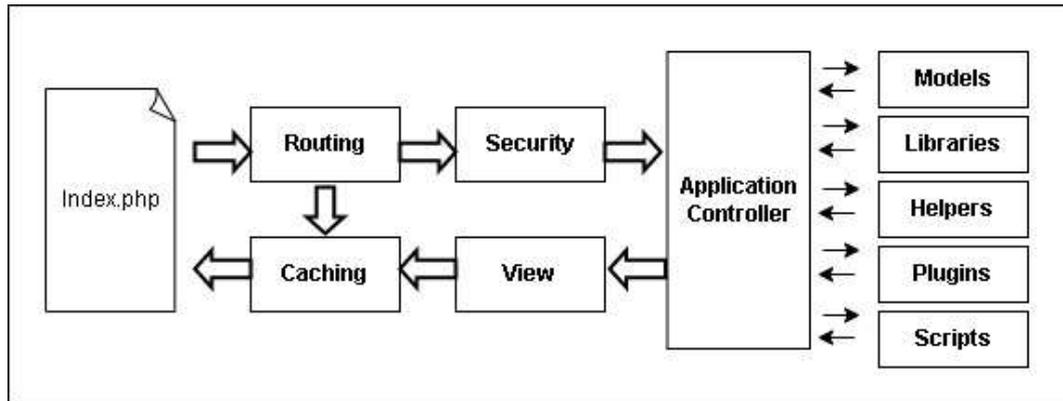
2. Ditulis Menggunakan PHP 4

Kode program *CodeIgniter* dibuat dengan menggunakan PHP 4. Hal ini disebabkan karena rilis awal *CodeIgniter* ini telah dibuat saat menggunakan versi bahasa PHP 4.

3. Berukuran Kecil

CodeIgniter yang kecil merupakan keunggulan tersendiri. Dibanding dengan *framework* lain yang berukuran besar.

4. Menggunakan Konsep MVC
CodeIgniter menggunakan konsep MVC yang memungkinkan pemisahan *layer application-logic* dan *presentation*
 5. URL yang Sederhana
Secara *default*, URL yang dihasilkan *CodeIgniter* sangat bersih dan *Search Engine Friendly* (SEF).
 6. Memiliki Paket *Library* yang Lengkap
CodeIgniter mempunyai *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis web, misalnya mengakses database, mengirim email, memvalidasi *form*, menangani *session* dan sebagainya.
 7. *Extensible*
Sistem dapat dikembangkan dengan mudah menggunakan *plugin* dan *helper*, atau dengan menggunakan *hooks*.
 8. Tidak Memerlukan *Template Engine*
Meskipun *CodeIgniter* dilengkapi dengan *template parser* sederhana yang dapat digunakan, tetapi hal ini tidak mengharuskan kita untuk menggunakannya.
 9. Dokumentasi Lengkap dan Jelas
Dari sekian banyak *framework*, *CodeIgniter* adalah satu-satunya *framework* dengan dokumentasi yang lengkap dan jelas.
 10. Komunitas
Komunitas *CodeIgniter* saat ini berkembang pesat. Salah satu komunitasnya bisa dilihat di (<http://codeigniter.com/forum/>).
- Proses aliran data aplikasi pada sistem dapat diilustrasikan seperti terlihat pada Gambar 2.



Gambar 2. 2 Framework Codeigniter

Sumber: (Sahi, 2020)

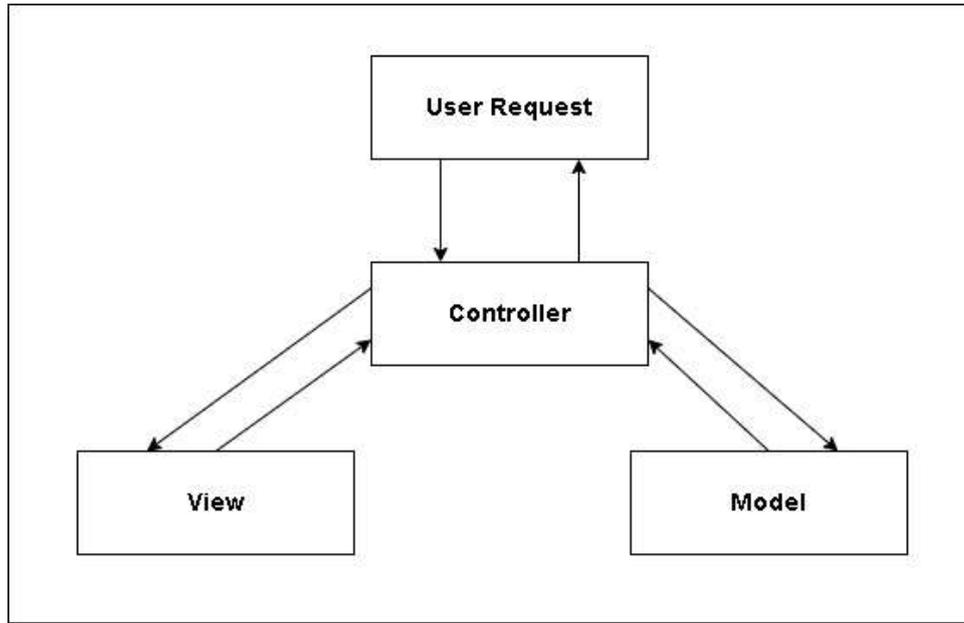
Berdasarkan Gambar 2.2, menunjukkan bagaimana proses aliran data aplikasi sistem yang dimulai dari *index*, *routing*, *controller*, *model*, *library*, *helper*, kembali ke *controller*, dan, terakhir *view*. Penjelasan secara rinci dari tiap proses adalah sebagai berikut.

- a. *Index.php* berfungsi sebagai *front controller*, menginisialisasi *base resource* untuk menjalankan *CodeIgniter*.
- b. *Router* memeriksa *HTTP request* untuk menentukan apa yang harus dilakukan dengannya.
- c. Jika *Cache* aktif, maka hasilnya akan langsung dikirimkan ke browser dengan mengabaikan aliran data normal.
- d. *Security*. Sebelum *Controller* dimuat, *HTTP request* dan data yang dikirimkan user akan difilter untuk keamanan.
- e. *Controller* memuat *model*, *core libraries*, *plugins*, *helpers*, dan semua *resource* yang diperlukan untuk memproses request.
- f. Akhirnya *View* yang dihasilkan akan dikirimkan ke browser. Jika *Cache* aktif, maka *View* akan disimpan sebagai *Cache* dahulu, sehingga pada *request* berikutnya langsung dapat ditampilkan.”

2.24 Pengertian MVC (*Model – View - Controller*)

Dalam jurnalnya (Sahi, 2020) membahas terkait konsep MVC bahwa “*CodeIgniter* adalah *framework* PHP yang dibuat berdasarkan kaidah *Model-View-Controller*. Dengan MVC, maka memungkinkan pemisahan antara *layer application-logic* dan *presentation*. Sehingga, dalam sebuah pengembangan web, seorang *programmer* bisa berkonsentrasi pada *core-system*, sedangkan web

designer bisa berkonsentrasi pada tampilan web. Menariknya, *script* PHP, *query* MySQL, *Javascript* dan CSS bisa saling terpisah, tidak dibuat dalam satu *script* berukuran besar yang membutuhkan *resource* besar pula untuk mengesekusinya. Adapun alur program aplikasi berbasis *framework* *Codeigniter* dapat dilihat pada Gambar 3.



Gambar 2. 3. Konsep MVC

Sumber: (Sahi, 2020)

Gambar 3 diatas menerangkan bahwa ketika datang sebuah *user request*, maka akan ditangani oleh *controller*, kemudian *controller* akan memanggil model jika memang diperlukan operasi *database*. Hasil dari *query* oleh model kemudian akan dikembalikan ke *controller*. Selanjutnya *controller* akan memanggil *view* yang tepat dan mengkombinasikannya dengan hasil *query* model. Hasil akhir dari operasi ini akan ditampilkan dibrowser, Dalam konteks *CodeIgniter* dan aplikasi berbasis web, maka penerapan konsep MVC mengakibatkan kode program dapat dibagi menjadi tiga kategori, yaitu:

1. Model Kode program (berupa OOP *class*) yang digunakan untuk memanipulasi *database*.
2. *View* Berupa *template html/xml* atau php untuk menampilkan data pada *browser*
3. *Controller* Kode program (berupa OOP *class*) yang digunakan untuk mengontrol aliran aplikasi (sebagai pengontrol *model* dan *View*)”.

2.25 Bootstrap

Dalam jurnal (Christian et al., 2018) memuat pengertian bahwa “*Bootstrap* adalah paket aplikasi siap pakai untuk membuat *front-end* sebuah *website*. Bisa dikatakan, *bootstrap* adalah *template* desain web dengan fitur *plus*. *Bootstrap* diciptakan untuk mempermudah proses desain web bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman. Cukup bermodalkan pengetahuan dasar mengenai HTML dan CSS, anda pun siap menggunakan *bootstrap*”.

2.26 Integrasi Keilmuan

Integrasi keilmuan merupakan paparan konsep penelitian dalam sudut pandang Islam. Bagian ini membahas hubungan atau korelasi antara penelitian dengan firman Allah Swt dalam kitab suci Al-Quran. Beberapa korelasi yang dimaksud adalah sebagai berikut.

1. Rancang Bangun

Korelasi yang pertama pada penelitian ini berkaitan dengan rancang bangun aplikasi yang bertujuan untuk menemukan informasi pegawai bagi pihak manajemen. Pelaksanaan manajemen data yang baik merupakan hasil dari bentuk kesatuan dan integrasi data, seperti dalam Q.S Ali Imran ayat 103 yang menerangkan pentingnya bentuk kesatuan yang utuh atau saling terintegrasi sebagai berikut.

وَاعْتَصِمُوا بِحَبْلِ اللَّهِ جَمِيعًا وَلَا تَفَرَّقُوا وَاذْكُرُوا
نِعْمَتَ اللَّهِ عَلَيْكُمْ إِذْ كُنْتُمْ أَعْدَاءً فَأَلَّفَ بَيْنَ قُلُوبِكُمْ
فَأَصْبَحْتُمْ بِنِعْمَتِهِ إِخْوَانًا وَكُنْتُمْ عَلَى شَفَا حُفْرَةٍ مِّنَ
النَّارِ فَأَنْقَذَكُمْ مِنْهَا كَذَلِكَ يُبَيِّنُ اللَّهُ لَكُمْ آيَاتِهِ لَعَلَّكُمْ
تَهْتَدُونَ

Artinya:

“Berpegang teguhlah kamu semuanya pada tali (agama) Allah, janganlah bercerai berai, dan ingatlah nikmat Allah kepadamu ketika kamu dahulu bermusuhan, lalu Allah mempersatukan hatimu sehingga dengan karunia-Nya kamu menjadi bersaudara. (Ingatlah pula ketika itu) kamu berada di tepi jurang neraka, lalu Allah menyelamatkan kamu dari sana. Demikianlah Allah menerangkan ayat-ayat-Nya kepadamu agar kamu mendapat petunjuk.”

Tafsir Q.S Ali Imran ayat 103 menurut Al Muyassar yakni “Dan berpegang teguhlah kalian semua kepada kitab suci Tuhan kalian dan petunjuk Nabi kalian, dan jangan melakukan hal-hal yang menyerat kalian kepada perpecahan. Dan ingatlah anugerah besar dari Allah yang tercurah pada kalian, tatkala kalian di masa dahulu wahai kaum mukminin, sebelum Islam, saling bermusuhan. Kemudian Allah menyatukan hati kalian di atas cinta kepadaNya dan cinta kepada RasulNya, dan meletakkan pada hati kalian rasa saling mencintai kepada sesama kalian, sehingga kalian dengan karunia Allah menjadi orang-orang bersaudara yang saling mencintai. Padahal dahulu kalian sudah berada di tepi jurang Neraka Jahannam, lalu Allah memberi kalian hidayah kepada Islam dan menyelamatkan kalian dari Neraka. Dan sebagaimana Allah sudah menjelaskan kepada kalian simbol-simbol iman yang benar, demikian pula Dia telah menjelaskan kepada kalian segala yang mendatangkan kemaslahatan bagi kalian, agar kalian mendapatkan hidayah menuju jalan yang lurus dan menapakinya, sehingga kalian pun tidak tersesat darinya (Hikmat Basyir, dkk, 2016:186- 187)”.

Dari ayat dan tafsir diatas terkait dengan rancang bangun aplikasi, bahwa untuk mendatangkan kemaslahatan dan mendapatkan hidayah menuju jalan yang lurus (kebenaran suatu informasi) adalah dengan berpegang teguh pada kitab suci dan bersatu atau tidak berpecah belah (data perusahaan yang saling terintegrasi untuk menemukan informasi terkait yang valid).

2. Pencarian Data

Korelasi selanjutnya pada penelitian ini terkait dengan pencarian data, yang dekat kaitannya dengan pencarian kebenaran atau jalan yang lurus, seperti yang terdapat dalam firman Q.S An-Nur ayat 46 berbunyi:

لَقَدْ أَنْزَلْنَا آيَاتٍ مُّبِينَاتٍ وَاللَّهُ يَهْدِي مَنْ يَشَاءُ إِلَى صِرَاطٍ مُسْتَقِيمٍ

Artinya:

“Sungguh, Kami telah menurunkan ayat-ayat yang memberi penjelasan. Dan Allah memberi petunjuk siapa yang Dia kehendaki ke jalan yang lurus”.

Dari Q.S An-Nur ayat 46, berikut beberapa tafsir sebagai penjelasan tentang keterkaitan antara ayat Al-Qur’an dengan penelitian ini, diambil dari (Tafsir As-Sa’di, hlm. 601-602), menafsirkan dari Q.S An-Nur ayat 46 “Dan Allah menunjuki siapa yang dikehendaki-Nya”, Artinya, barang siapa yang sungguh-sungguh pada kebaikan dan jujur dalam pencariannya akan kebenaran akan dituntun “di jalan yang benar”, yaitu jalan yang jelas dan ringkas, dan mengarah kepada Allah Swt ke surga-Nya, yang berisi pemahaman ilmu kebenaran. Jadi penjelasan yang sempurna berlaku untuk semua kecuali bimbingan yang Dia inginkan. Itu adalah hadiah dari Tuhannya.

Dari ayat dan tafsir diatas menjelaskan bahwa setiap insan yang berusaha dalam mencari kebenaran maka akan dihasilkan, ditampilkan, atau ditunjukkan kepada jalan yang lurus, yaitu hasil atau jalan yang jelas, ringkas, yang mengandung kebenaran. Sehingga dari ayat diatas mengenai jalan kebenaran dapat menjadi landasan integrasi keilmuan yang dilakukan dalam penelitian ini.

3. Tingkat Kecocokan/Kemiripan Data dengan *Cosine Similarity*

Korelasi yang terakhir pada penelitian ini terkait dengan pencocokan data dengan metode *cosine similarity*. Pencocokan data antara makna semantik *keyword* dengan database pegawai berusaha untuk tidak mencampuradukkan informasi yang diperoleh dengan informasi lain yang tidak masuk dalam kriteria pencarian, seperti dalam Firman Allah Swt dalam Q.S Al-Baqarah ayat 42 yang berbunyi:

وَلَا تَلْبِسُوا الْحَقَّ بِالْبَاطِلِ وَتَكْتُمُوا الْحَقَّ وَأَنْتُمْ تَعْلَمُونَ

Artinya:

“Janganlah kamu campur adukkan kebenaran dengan kebatilan dan (jangan pula) kamu sembunyikan kebenaran, sedangkan kamu mengetahui(-nya).”

Tafsir Q.S Al-Baqarah ayat 42 menurut Imam Ibnu Katsir dalam tafsirnya mengutip Abul Aliyah “bahwa berpesan kepada Ahli Kitab agar tidak mencampur yang hak dengan yang batil dan meminta mereka untuk menyampaikan nasihat yang benar kepada hamba-hamba Allah yang menjadi umat Muhammad SAW”.

Sehingga dari ayat dan tafsir diatas yang kaitannya dengan pencocokan data yakni bahwa untuk informasi yang telah diperoleh (makna semantik *keyword*), diharapkan sistem tidak salah dalam melakukan pencocokan data sehingga tidak terjadi sistem yang mencampuradukkan data pegawai yang tidak sesuai dengan makna semantik *keyword* yang telah diperoleh.

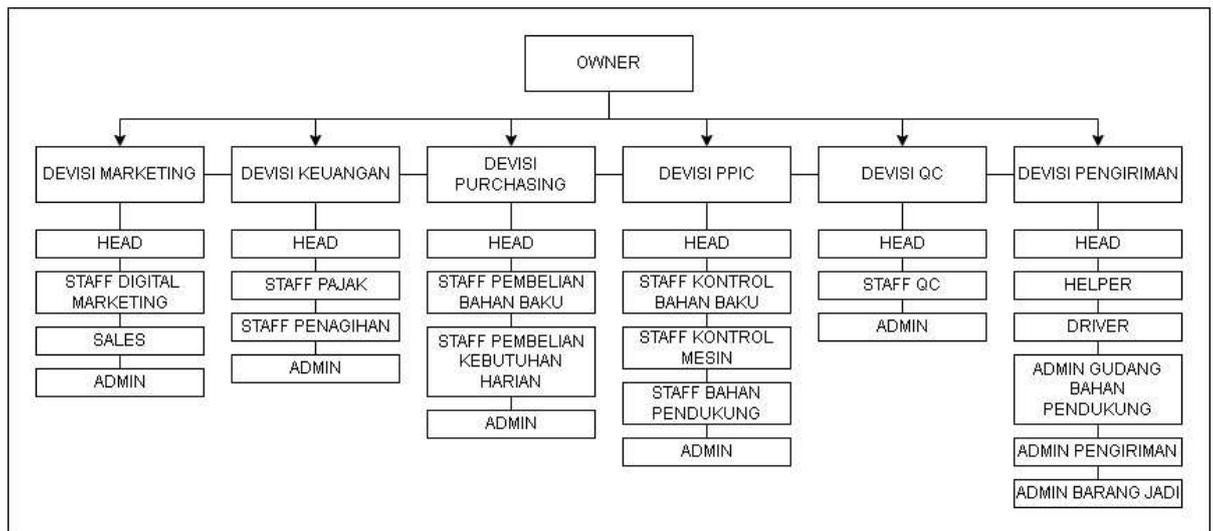


UIN SUNAN AMPEL
S U R A B A Y A

BAB III METODOLOGI PENELITIAN

3.1 Objek Penelitian

Penelitian dilakukan di Jl. Donokerto XI No.14 - 16, Kapasan, Kec. Simokerto, Kota SBY, Jawa Timur. Yang secara detail struktur organisasi dari objek penelitian adalah sebagai berikut.



Gambar 3. 1 Struktur Organisasi

Struktur Organisasi terdiri dari 6 divisi yang masing-masing memiliki staff terkait, dengan Owner perusahaan sebagai puncak pimpinan. Penelitian dilakukan pada bulan Agustus 2021 sampai dengan bulan Oktober 2022, bertujuan untuk menghasilkan pencarian data pegawai secara akurat dengan aplikasi pencarian pegawai berbasis *semantic* dengan metode *cosine similarity*.

3.2 Sumber Data

Sumber data menggunakan data pegawai yang diperoleh dari divisi *Human Resource Development* (HRD) PT. Kecap Kenari. Data pegawai berisikan beberapa atribut seperti nama, jenis kelamin, pendidikan terakhir, divisi, dan jabatan. Data yang digunakan dalam penelitian dapat dilihat pada Lampiran 1.

Data yang diperoleh berjumlah 99 data pegawai tetap dari total keseluruhan karyawan kurang lebih 400 karyawan tetap dan *outsourcing*. Hal ini dikarenakan kebijakan perusahaan terkait pemberian data pegawai *outsourcing*.

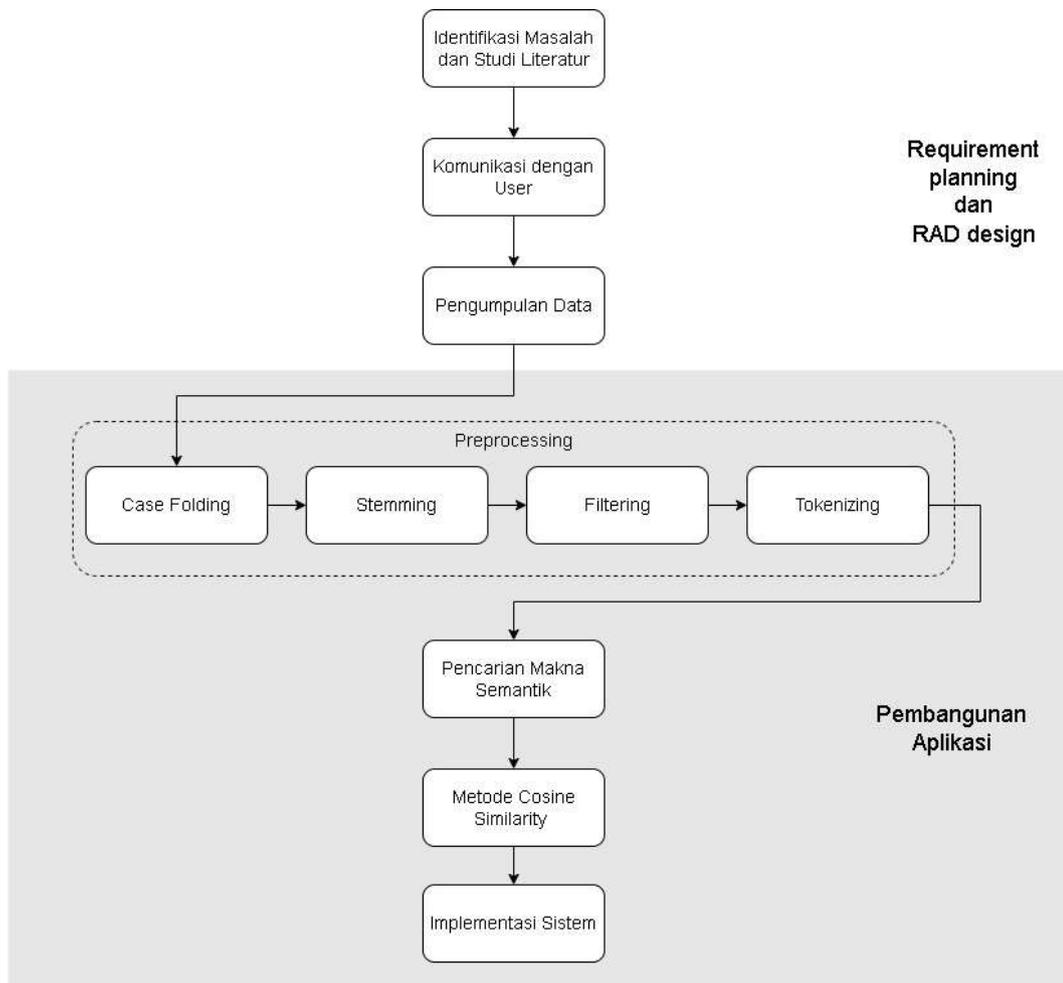
3.3 Instrumen Penelitian

Beberapa instrumen atau alat bantu dalam penelitian ini adalah sebagai berikut:

- 1 Microsoft Excel, digunakan untuk memetakan data sementara dari sumber data penelitian yang kemudian akan diimport ke dalam database.
- 2 Xampp, berperan sebagai *server local* untuk MySQL yang dapat disesuaikan setiap entitas data serta hubungan antar entitas data.
- 3 MySQL, sebagai RDMS (*Relational Database Management System*) dalam penelitian ini.
- 4 Codeigniter, sebagai framework untuk membangun *frontend* dan *backend* yang menghubungkan pengguna, *system interoperability*, dan database.
- 5 Sastrawi, sebagai modul *library* yang diintegrasikan dengan aplikasi pencarian data dalam melakukan *preprocessing* data inputan sebelum dicari makna semantiknya.
- 6 Satria/Synonym-Antonym-Indonesia, sebagai modul *library* yang diintegrasikan dengan aplikasi pencarian data dalam menemukan makna kata dari setiap *keyword* yang diberikan sesuai KBBI.
- 7 Mlwmw/php-cosine-similarity, sebagai *library* yang diintegrasikan dengan aplikasi pencarian data pegawai dalam menemukan nilai *cosine similarity* dengan membandingkan tingkat kemiripan atribut pegawai dalam *database* dengan atribut data target yang diharapkan sesuai *keyword* yang dihasilkan.

UIN SUNAN AMPEL
S U R A B A Y A

3.4 Desain Penelitian



Gambar 3. 2 Metodologi Penelitian

Berdasarkan metodologi penelitian pada Gambar 4 lebih lanjut dijelaskan secara sistematis sebagai berikut:

1 Identifikasi Masalah dan Studi Literatur

Tahapan pertama ini berisi proses mencari masalah yang dapat mendorong adanya penelitian guna mencari solusi untuk masalah tersebut. Serta secara tidak langsung dari proses mencari solusi untuk permasalahan, dilakukan pengumpulan informasi terkait solusi yang digunakan yakni *Web Semantic*. Tahapan ini menghasilkan topik penelitian, serta metode yang digunakan dalam penelitian.

2 Komunikasi dengan user

Pada tahap ini, penulis mendiskusikan dengan user mengenai kebutuhan sistem yang akan dibangun. Mulai dari kebutuhan data sampai alur proses

sistem serta progress apa saja yang telah dikembangkan dari respon user yang telah diterima pada waktu sebelumnya. Tahapan ini menghasilkan desain sistem yang akan dibangun yakni *flowchart* sistem, *Entity Relationship Design*, dan *Data Flow Diagram*.

3 Pengumpulan Data

Pada tahap ini, data diperoleh dari sumber data penelitian, dengan instrument penelitian *Microsoft Excel* yang digunakan untuk penyimpanan awal data perusahaan. Tahapan ini menghasilkan daftar pegawai perusahaan yang menjadi studi kasus penelitian serta ditentukannya ERD untuk rancangan database.

4 *Text preprocessing*

Pada tahap *text preprocessing* memiliki fungsi yakni sebagai tahap awal pengolahan teks sebelum diolah lebih lanjut. Data teks tidak terstruktur memiliki noise seperti tanda baca, angka, imbuhan, karakter khusus dan lain sebagainya. Dalam tahapan ini, data teks dibersihkan hingga tersisa bentuk dasar saja yang kemudian dapat dianalisis lebih lanjut. Tahapan ini menghasilkan *keyword* masukan yang telah disederhanakan. Pada penelitian ini tahapan *text processing* menggunakan library sastrawi yang diintegrasikan dengan aplikasi berbasis web PHP terdiri dari 4 langkah, yaitu *Case folding*, *Stemming*, *Filtering*, dan *Tokenizing* sebagai berikut.”

a. *Case folding*

Case folding merupakan proses untuk menghilangkan karakter-karakter selain huruf, seperti tanda baca, angka, dan menyeragamkan semua teks menjadi huruf besar atau sebaliknya.

b. *Stemming*

Stemming adalah proses untuk menghilangkan imbuhan kata atau mengubah kata berimbuhan menjadi kata dasar.

c. *Filtering*

Filtering atau *stopword removal* dapat dipahami sebagai proses penghilangan *stopwords* atau kata-kata yang tidak menggambarkan isi tulisan, sehingga dapat dibuang dengan pendekatan *bag-of-words*. Proses ini bertujuan untuk mengurangi jumlah kata.

d. *Tokenizing*

Tokenizing/ Tokenization merupakan proses untuk memisahkan setiap kata dalam kalimat.

5 Pencarian Makna Semantik

Pada tahap ini, dilakukan pendalaman modul *text preprocessing* dengan fokus makna semantik dari hasil *data preprocessing* menggunakan *library Satria/Synonym Antonym* kata yang diintegrasikan dengan aplikasi. Tahapan ini menghasilkan daftar makna semantik dari *keyword* yang dimasukkan.

6 *Cosine Similarity*

Algoritma *cosine similarity* adalah algoritma untuk menghitung derajat kemiripan atau kemiripan antara dua objek. Perhitungan berdasarkan ukuran kesamaan ruang vektor (Basmalah Wicaksono et al., 2016). Tahapan ini menghasilkan nilai tingkat kemiripan antara *keyword* dan makna semantik dengan daftar pegawai.

7 Implementasi Sistem

Pada tahap implementasi sistem, dilakukan implementasi tampilan *frontend* dan *backend* serta pengujian sistem dengan melihat fungsionalitas aplikasi. Baik dari pengelolaan data, menemukan makna semantic suatu *keyword*, dan menghasilkan prioritas rekomendasi pegawai berdasarkan tingkat kemiripan dengan metode *cosine similarity*. Tahapan ini menghasilkan daftar skenario dalam penggunaan sistem.

UIN SUNAN AMPEL
S U R A B A Y A

BAB IV HASIL DAN PEMBAHASAN

4.1. Requirement Planning

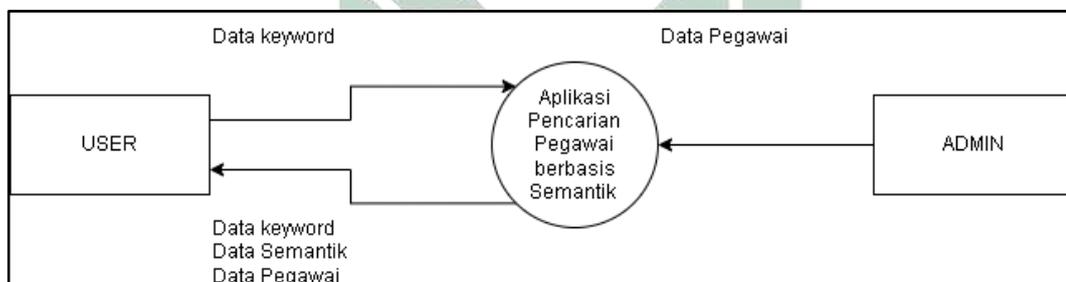
Pada bagian ini penulis melakukan pertemuan dengan user untuk mendiskusikan kebutuhan system yang hendak dibangun yaitu *Data Flow Diagram* sebagai rancangan aliran data, *Entity Relationship Diagram* sebagai rancangan database, *Flowchart System*, dan *Use Case Diagram* sebagai rancangan gambaran user, serta spesifikasi software dan hardware sistem dalam menggunakan aplikasi sebagai berikut:

4.1.1 Data Flow Diagram

Data Flow Diagram pada rancangan system pencarian data pegawai menjabarkan secara rancangan yang secara terstruktur terbagi menjadi 3 tingkat yaitu, diagram konteks, DFD level 0, dan DFD level 1. Penjelasan lebih detail adalah sebagai berikut:

1. Diagram Konteks / *Context Diagram*

Context diagram mencerminkan aliran data secara garis besar dari dan ke sistem pencarian data pegawai. Sebagai berikut:

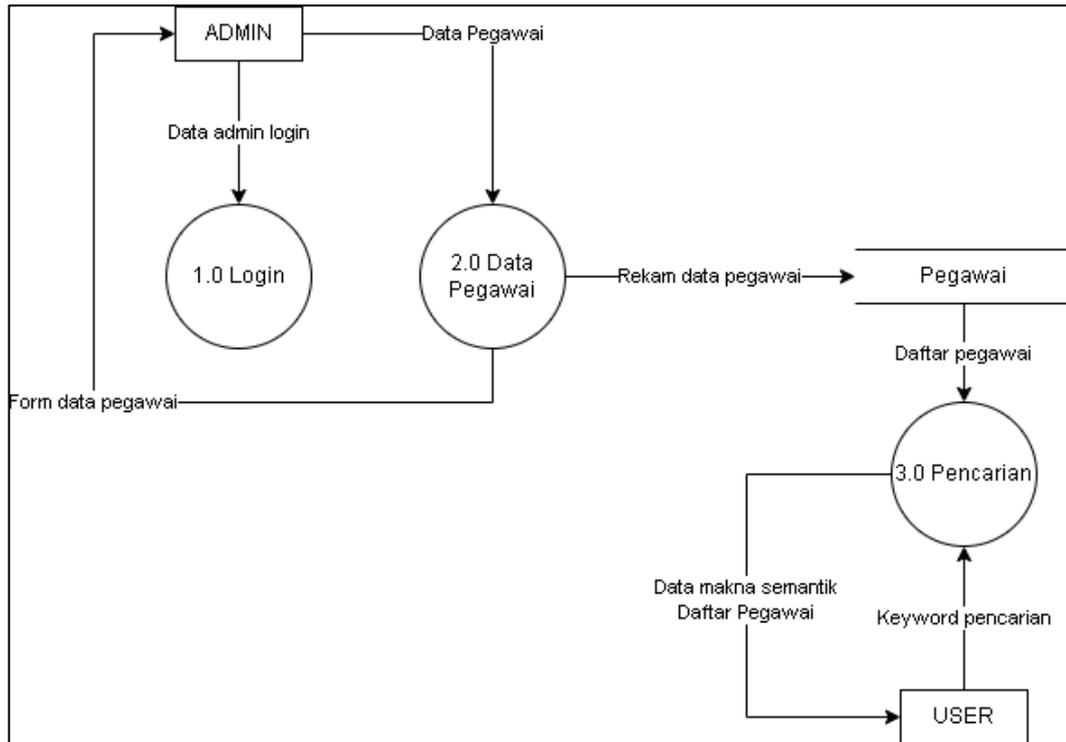


Gambar 4. 1 Diagram konteks

Dari Gambar 6, dapat diketahui terdapat aliran data dalam sistem yang dibagi sebagai berikut:

- a. Aliran data dari *user* ke sistem dan sebaliknya, seperti data *keyword*, data makna *semantic keyword*, dan data pegawai (nama, jabatan, divisi, dan lain sebagainya).
 - b. Aliran data dari admin ke sistem, yakni data pegawai (nama, jabatan, divisi, dan lain sebagainya).
2. *Data Flow Diagram* (DFD) level 0

Data Flow Diagram (DFD) Level 0 merupakan hasil pengembangan dari Context Diagram ke dalam komponen yang lebih detail. Diagram DFD level 0 dapat dilihat sebagai berikut:

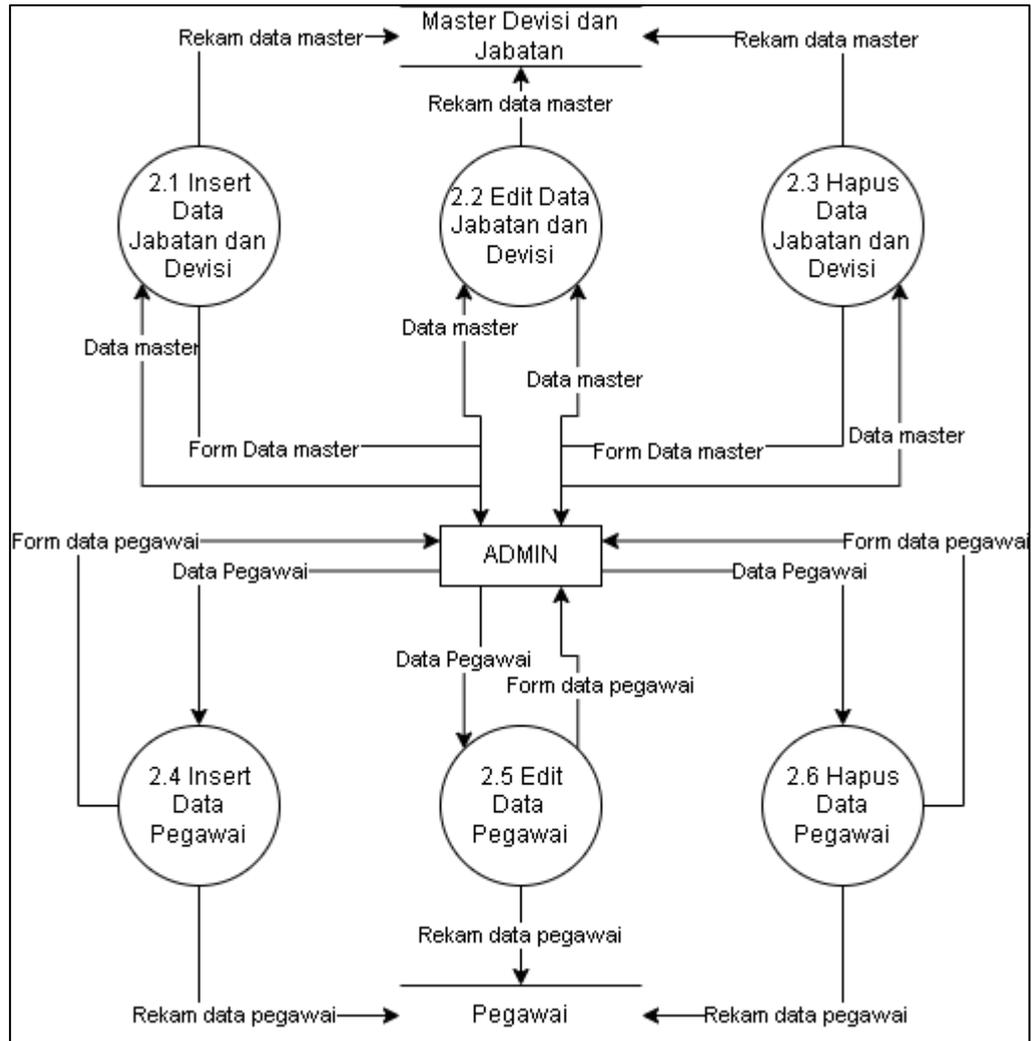


Gambar 4. 2 DFD level 0

Pada Gambar 7, diagram alir data / DFD level 0 secara berurutan dapat dipahami sebagai berikut:

- a. Data pegawai yang telah direkam akan masuk ke dalam *database* dan dapat digunakan dalam pencarian data pegawai berdasarkan *keyword* dan makna semantiknya.
 - b. *Keyword* yang telah diinputkan akan digunakan dalam pencarian data pegawai. Memanfaatkan *keyword* dan makna semantiknya menghasilkan daftar pegawai yang terkait dengan *keyword* dan makna *semantic keyword*.
3. *Data Flow Diagram* (DFD) level 1

Pada DFD Level 1 menggambarkan aliran data yang lebih terperinci dari input data pegawai dan pencarian data pegawai yang secara rinci adalah sebagai berikut.

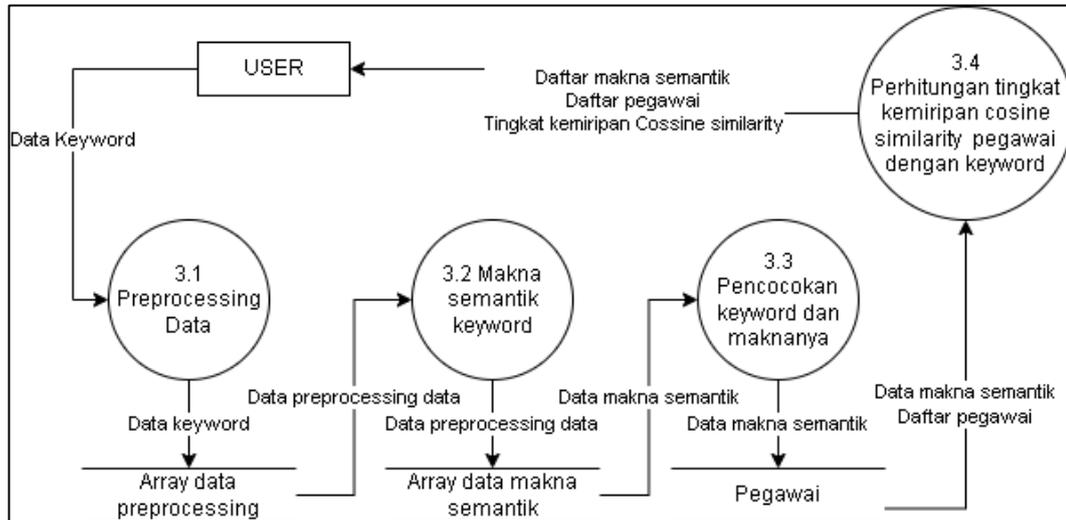


Gambar 4. 3 DFD level 1 dari proses 2.0

DFD level 1 pada proses Data Pegawai berdasarkan Gambar 8 secara rinci sebagai berikut:

- Proses 2.1, merupakan proses *input data* master divisi dan jabatan yang disimpan ke dalam penyimpanan data pegawai.
- Proses 2.2, merupakan proses *edit data* master divisi dan jabatan yang disimpan ke dalam penyimpanan data pegawai.
- Proses 2.3, merupakan proses hapus data master divisi dan jabatan yang disimpan ke dalam penyimpanan data pegawai.
- Proses 2.4, merupakan proses *input data* pegawai yang disimpan ke dalam penyimpanan data pegawai.
- Proses 2.5, merupakan proses *edit data* pegawai yang disimpan ke dalam penyimpanan data pegawai.

- f. Proses 2.6, merupakan proses hapus data pegawai yang disimpan ke dalam penyimpanan data pegawai.

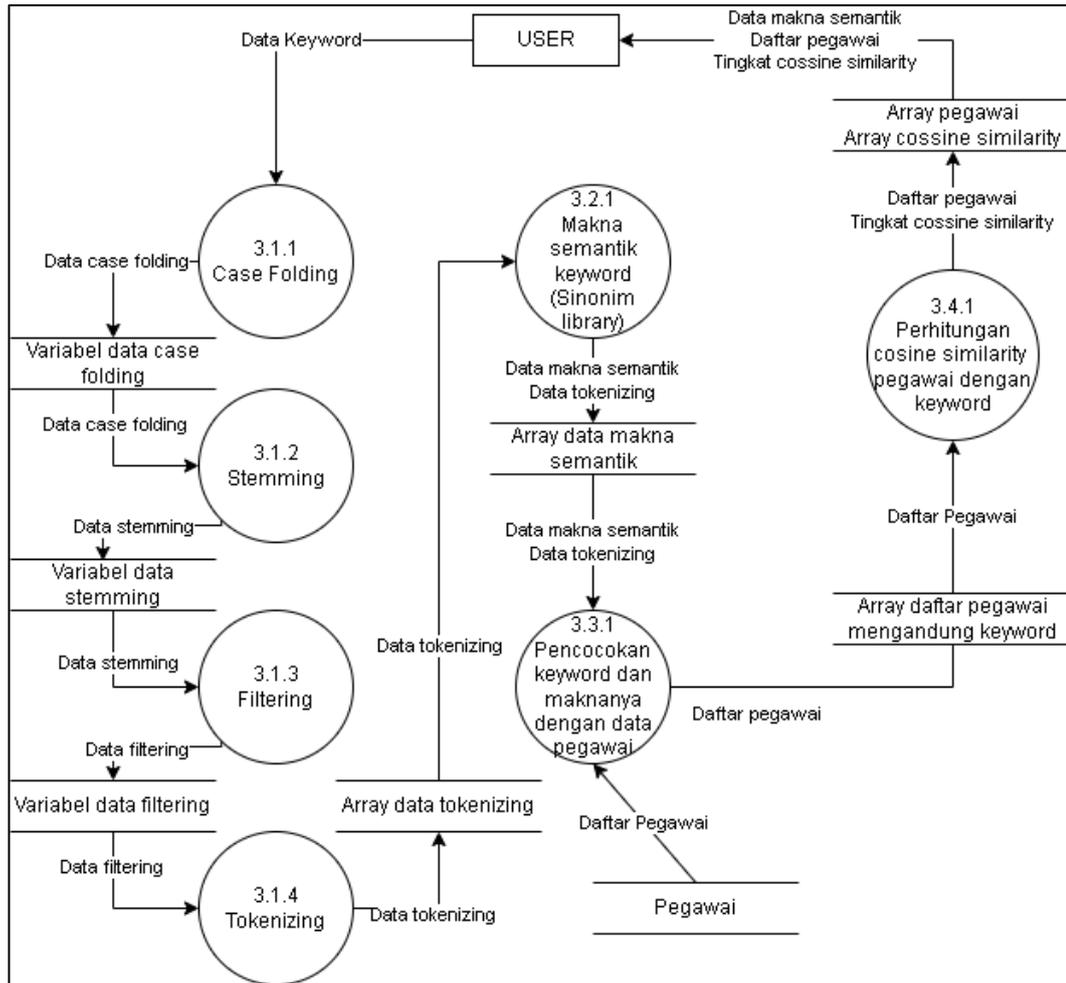


Gambar 4. 4 DFD level 1 dari proses 3.0

DFD level 1 pada proses pencarian data berdasarkan Gambar 9 secara rinci sebagai berikut:

- a. Proses 3.1, merupakan proses *preprocessing* data *keyword* yang disimpan sementara ke dalam penyimpanan *data array preprocessing*.
 - b. Proses 3.2, merupakan proses menemukan makna semantik dari *keyword* yang disimpan sementara ke dalam penyimpanan *data array* makna semantik.
 - c. Proses 3.3, merupakan proses mencari daftar pegawai yang terkait dengan *keyword* dan makna semantiknya.
 - d. Proses 3.4, merupakan proses menemukan tingkat kemiripan *cosine similarity* antara daftar pegawai yang mengandung salah satu atau lebih *keyword*.
4. *Data Flow Diagram* (DFD) level 2

Pada DFD Level 2 menggambarkan aliran data yang lebih terperinci dari pencarian data pegawai yang secara rinci adalah sebagai berikut.



Gambar 4. 5 DFD level 2 dari proses 3.1, 3.2, 3.3, 3.4

DFD level 2 pada proses Pencarian data berdasarkan Gambar 10 secara rinci sebagai berikut:

- Proses 3.1.1, merupakan proses *case folding (preprocessing)* dari *data keyword* yang diinputkan kemudian dihilangkan karakter special dalam *string* dan menyimpan sementara ke dalam *variable data case folding*.
- Proses 3.1.2, merupakan proses *stemming (preprocessing)* dari *data case folding* yang diperoleh kemudian dihilangkan imbuhan kata dalam *string* dan menyimpan sementara ke dalam *variable data stemming*.
- Proses 3.1.3, merupakan proses *filtering (preprocessing)* dari *data stemming* yang diperoleh kemudian dihilangkan kata penghubung seperti (dan, atau, tetapi, yang, daripada, dan sebagainya) dalam *string* dan menyimpan sementara ke dalam *variable data filtering*.
- Proses 3.1.4, merupakan proses *tokenizing (preprocessing)* dari *data filtering*

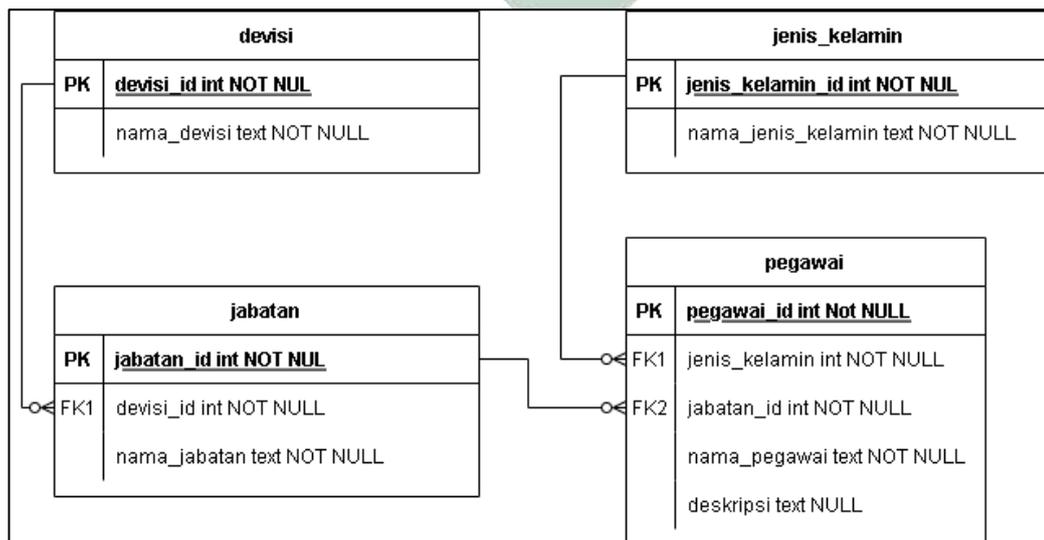
yang diperoleh kemudian dipisahkan tiap kata dalam string dan menyimpan sementara ke dalam *variable data array tokenizing*. Dari *data tokenizing* ini kemudian akan diproses lebih lanjut dalam mencari makna semantik tiap kata yang dihasilkan.

- e. Proses 3.2.1, merupakan proses menemukan makna semantic dari *data array tokenizing* yang dihasilkan dengan memanfaatkan *library PHP Satria Synonym* teks bahasa Indonesia. Data makna semantik yang dihasilkan kemudian disimpan sementara ke dalam penyimpanan data *array* makna semantik.
- f. Proses 3.3.1, merupakan proses menemukan daftar pegawai yang terkait dengan *keyword* dan *data array* makna semantik yang diperoleh sebelumnya.
- g. Proses 3.4.1, merupakan proses menemukan tingkat kemiripan dengan *cosine similarity* antara pegawai dengan *keyword/* makna semantik.

Dari hasil pencarian ini pengguna dapat mengetahui daftar pegawai yang mengandung kata kunci/ *keyword* dan makna semantik serta tingkat kemiripan antara *atribut* data pegawai dengan keseluruhan kata kunci dan makna semantik yang dihasilkan.

4.1.2 Entity Relationship Diagram

Berdasarkan rancangan *Data Flow Diagram* maka struktur *ERD database* dapat disusun sebagai berikut:

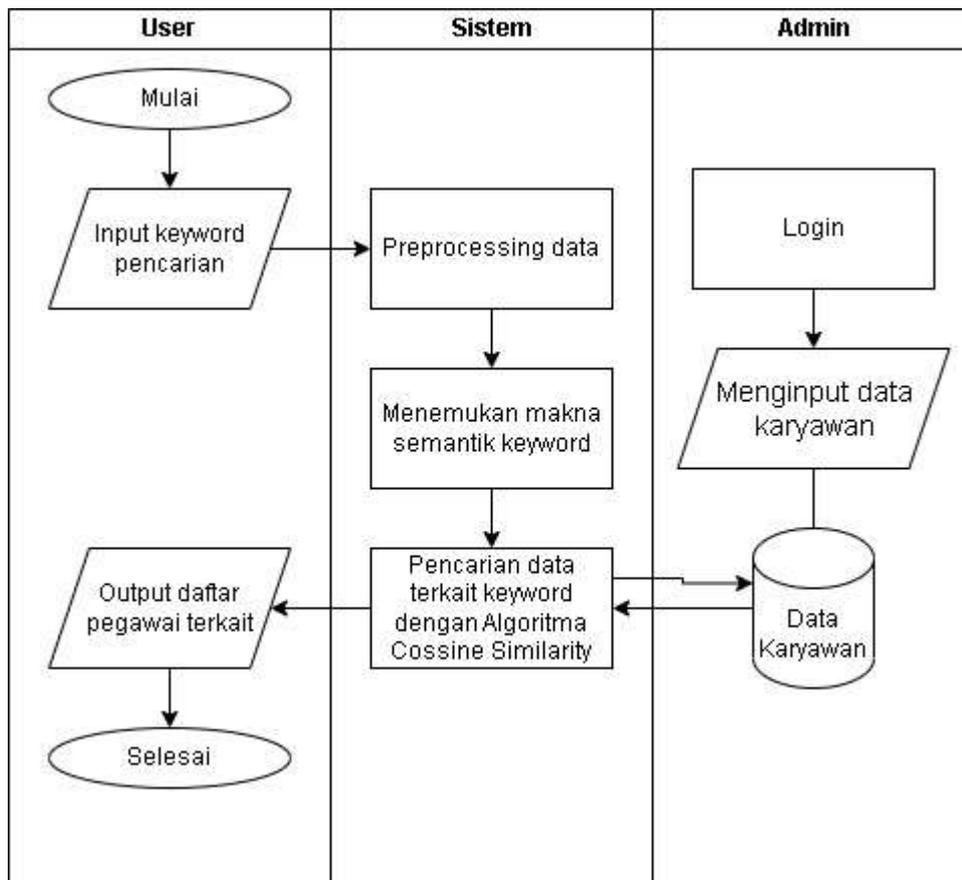


Gambar 4. 6 ERD diagram sistem

Dari rancangan ERD Gambar 11, menunjukkan gambaran dari *database system* yang akan dibangun. Berfokus pada data pegawai yang berisi identitas pegawai dan relasi dengan atribut pendukung data pegawai seperti jenis kelamin, divisi, dan jabatan pegawai yang menjadi *foreign key* pada *table* pegawai. Database yang digunakan dalam penelitian ini menggunakan MySQL.

4.1.3 Flowchart Sistem

Berdasarkan rancangan *Data Flow Diagram* serta ERD yang diperoleh maka dihasilkan *flowchart* sistem sebagai berikut:



Gambar 4. 7 Flowchart Sistem

Mengenai *flowchart system* yang hendak dibangun adalah berikut:

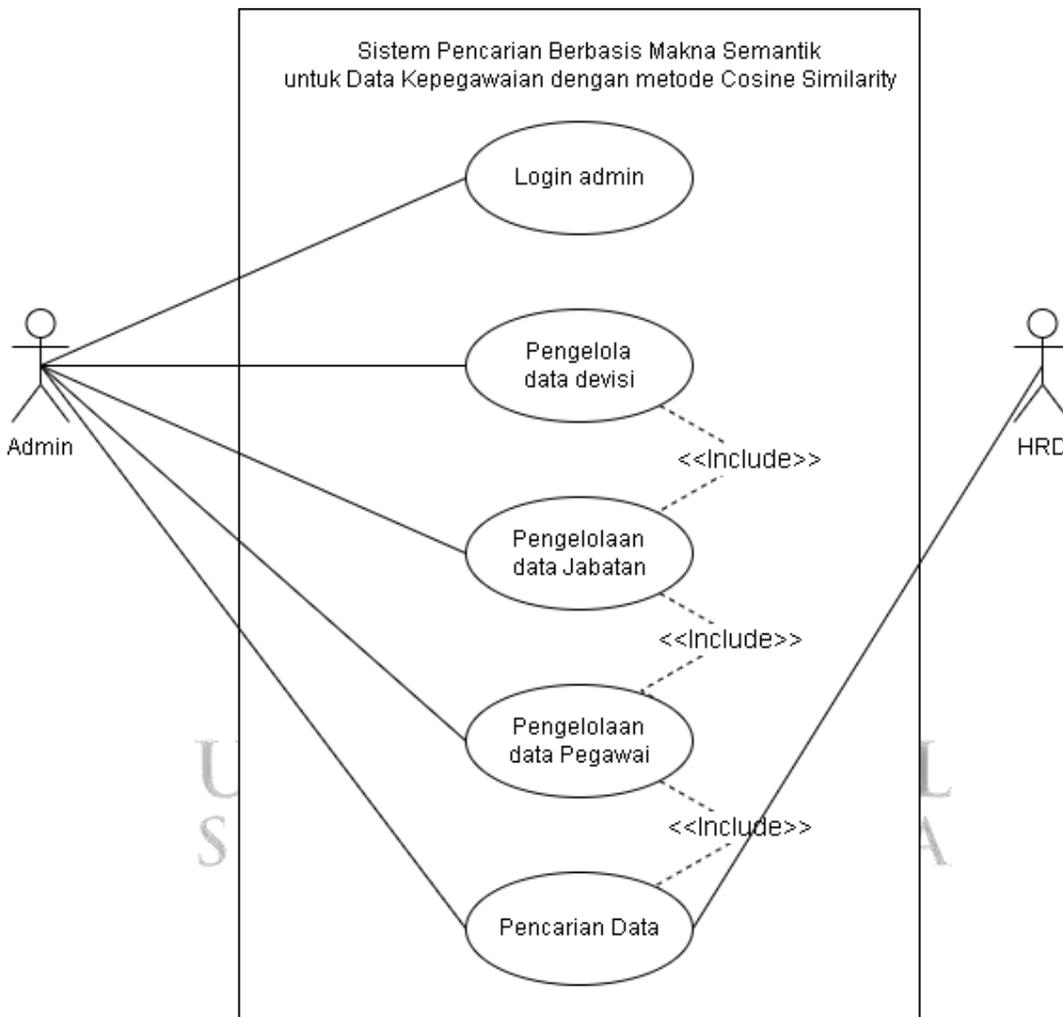
1. Analisis dari sisi User, berisi tahapan penggunaan aplikasi pencarian data pegawai yakni dimulai dengan menginputkan *keyword* pencarian dan mendapatkan *output* daftar pegawai terkait.
2. Analisis dari sisi *system*, berisi tahapan transformasi *keyword* oleh *user* yang secara berurutan dimulai dengan melakukan *preprocessing* data *keyword*, mencari makna semantik *keyword*, dan melakukan pencarian data pegawai

berdasarkan *keyword* dan makna semantiknya dengan algoritma *cosine similarity*.

3. Analisis dari sisi Admin, berisi tahapan kesediaan data pegawai dalam database pencarian yakni admin melakukan login dan menginputkan data pegawai dalam aplikasi sehingga dapat digunakan oleh mesin pencari.

4.1.4 Use Case Diagram

Dari jabaran *flowchart* dan DFD, maka gambaran dari fitur pengguna dapat dilihat pada *Use Case Diagram* berikut.



Gambar 4. 8 Use Case Diagram sistem

Dari *use case diagram* dapat dilihat ada dua peran yaitu admin dan *user* biasa. Admin dapat mengelola dan menggunakan pencarian data sedangkan *user* biasa hanya bisa melakukan pencarian data. Yang bertindak sebagai admin dalam sistem adalah pihak Kepala HRD sedangkan yang bertindak sebagai user adalah staff HRD

ataupun Kepala Divisi lain yang hendak mencari daftar pegawai internal yang memiliki kompetensi sesuai kebutuhan divisi.

4.1.5 Use Case Description

Dari *Use Case Diagram*, maka penjelasan fitur lebih rinci dari pengguna aplikasi ini adalah sebagai berikut

1. Fitur Admin

Pada fitur admin sistem pencarian pegawai berdasarkan makna semantik dengan metode *cosine similarity* adalah berkaitan dengan pengelolaan data pegawai. Baik dari master divisi, jabatan, serta data pegawai itu sendiri. Beberapa fitur yang dapat dilakukan oleh admin adalah sebagai berikut.

a. Login Admin

Fitur ini bertujuan sebagai pintu masuk admin untuk menuju halaman backend yang berisi semua menu untuk pengelolaan data.

b. Pengelolaan Data Divisi

Fitur ini bertujuan untuk pembuatan master data divisi pada suatu perusahaan.

c. Pengelolaan Data Jabatan

Fitur ini bertujuan untuk pembuatan master data divisi pada suatu perusahaan.

d. Pengelolaan Data Pegawai

Fitur ini bertujuan untuk memasukkan data pegawai yang bekerja diperusahaan, termasuk menentukan divisi dan jabatan pada perusahaan sesuai dengan data master divisi dan jabatan yang telah dibuat sebelumnya.

e. Pencarian Data Pegawai

Fitur ini berisi sistem pencarian pegawai berdasarkan makna semantik dengan metode *cosine similarity* adalah berkaitan dengan pencarian data pegawai dari *keyword* yang dimasukkan.

2. Fitur User

Pada fitur *user* dari sistem pencarian pegawai berdasarkan makna semantik dengan metode *cosine similarity* adalah berkaitan dengan pencarian data pegawai dari *keyword* yang dimasukkan. Yang mana fitur pada *user* adalah hal yang menjadi

pembahasan utama dari penelitian ini, yaitu dapat mengolah *keyword* hingga menjadi *dataset* pencarian sesuai makna semantiknya dan hasil dari pencarian dapat menunjukkan tingkat kemiripan *cosine similarity*.

4.1.6 Pseudocode

Secara singkat algoritma pemrograman komputer dideskripsikan sebagai berikut.

1. Tahap *Preprocessing Text*

a. *Case folding*

Case folding merupakan tahap untuk menghilangkan karakter unik dari *string*. Dari tahap *Case folding* menunjukkan bahwa karakter unik pada kalimat telah dihilangkan menjadi “mencari pegawai yang ahli dibidang listrik dan mekanik mesin”. *Pseudocode* dari tahap *case folding* sebagai berikut.

<i>Case folding</i>
<p><i>Declaration:</i></p> <p><i>bad_chars: list of useless characters</i></p>
<p><i>Algorithm:</i></p> <pre> FOR bad_chars in keyword do IF keyword contain bad_chars{ Remove bad_chars, from keyword } ENDFOR RETURN string_to_lower(keyword) </pre>

b. *Stemming*

Stemming merupakan tahap yang bertujuan untuk mengubah kata yang memiliki imbuhan kata menjadi kata dasar saja. Dari tahap *stemming* menunjukkan bahwa imbuhan kata pada kata “mencari” menjadi “cari”. Sehingga menghasilkan kalimat “cari pegawai yang ahli bidang listrik dan mekanik mesin”. *Pseudocode* dari tahap *stemming* sebagai berikut.

<i>Stemming</i>

<p><i>Declaration:</i></p> <p><i>library_stemming: library stemming function</i></p>
<p><i>Algorithm:</i></p> <p><i>CALL library_stemming</i></p> <p><i>RETURN library_stemming->stem(keyword)</i></p>

c. *Filtering*

Filtering merupakan tahap yang bertujuan untuk menghilangkan kata penghubung atau kata yang dianggap tidak penting. Sehingga dari tahap ini kalimat berubah menjadi “cari pegawai ahli bidang listrik mekanik mesin”.

Pseudocode dari tahap *filtering* sebagai berikut.

<p><i>Filtering</i></p>
<p><i>Declaration:</i></p> <p><i>library_filtering: library filtering function</i></p>
<p><i>Algorithm:</i></p> <p><i>CALL Library_filtering</i></p> <p><i>RETURN library_filtering->remove(keyword)</i></p>

d. *Tokenizing*

Tokenizing merupakan tahap yang bertujuan untuk memisahkan tiap kata dalam kalimat yang mana tiap kata akan dicari makna semantiknya. Sehingga dari tahap ini menghasilkan *keyword* masing-masing “cari, pegawai, ahli, bidang, listrik, mekanik, mesin,”. *Pseudocode* dari tahap *tokenizing* adalah sebagai berikut.

<p><i>Tokenizing</i></p>
<p><i>Declaration:</i></p> <p><i>library_tokenizing: library tokenizing function</i></p>
<p><i>Algoritma:</i></p> <p><i>CALL Library_tokenizing</i></p> <p><i>RETURN library_tokenizing->tokenize(keyword)</i></p>

2. Implementasi Pencarian Makna Semantik

Pada tahap ini aplikasi web yang dibangun memanfaatkan *library Satria/SynonymAntonym* untuk menemukan makna semantik dari tiap kata yang dihasilkan oleh tahap *preprocessing* data. *Pseudocode* dari tahap pencarian makna semantik sebagai berikut.

<i>Pencarian Makna Semantik</i>
<i>Declaration:</i> <i>array_tokenizing: array from tokenizing</i> <i>library_dictionary: dictionary Satria/synonym Bahasa Indonesia</i> <i>makna_semantik: array result translation semantic variable from array tokenize</i> <i>kumpulan_semantik: array all result from \$makna_semantik</i>
<i>Algorithm:</i> <i>CALL library_dictionary</i> <i>Foreach array_tokenizing as tokenize{</i> <i> makna_semantik = \$dictionary->word(tokenize)->synonym();</i> <i> Foreach makna_semantik as semantic{</i> <i> Array_push(kumpulan_semantik, with semantic);</i> <i> }</i> <i>}</i> <i>RETURN kumpulan_semantik</i>

Prinsip dari *pseudocode* Pencarian Makna Semantik adalah menemukan seluruh makna semantik yang terkandung dalam setiap *keyword* dari hasil *preprocessing* data. Dengan memanfaatkan *library Satria/ SynonymAntonym*, data hasil proses *tokenizing* dilakukan *looping* setiap katanya dan dihasilkan satu atau lebih persamaan katanya. Yang kemudian setiap daftar persamaan kata akan dikumpulkan dan digabungkan pada satu variabel (*kumpulan_semantik*).

3. Implementasi Pencocokan String (*Cossine Similarity*)

Pada tahap ini aplikasi web mencari daftar pegawai yang memiliki kandungan data seperti salah satu daftar makna semantik yang dihasilkan. *Pseudocode* tahap pencocokan *string* sebagai berikut.

Pencocokan String Cosine Similarity

Declaration:

kumpulan_semantik: array all result from *makna_semantik*

library_cosine_similarity: library for calculate cosine similarity

daftar_pegawai: list of pegawai

contain_pegawai: array of list pegawai which contain with *kumpulan_semantik*

sorting_pegawai: array sorting pegawai descending by cosine value

Algorithm:

```
CALL library_cosine_similarity
//Looping data pegawai untuk menemukan kecocokan string dengan
data pegawai
Foreach daftar_pegawai as pegawai{
  Foreach kumpulan_semantik as semantik{
    If string semantik contain in this pegawai->nama OR
    string semantik contain in this pegawai->divisi OR
    string semantik contain in this pegawai->jabatan{
      get this pegawai into array_push(contain_pegawai, this
pegawai);
    }
  }
}
//Looping data pegawai memiliki kecocokan untuk menemukan nilai
cosine similarity
Foreach contain_pegawai as contain{
  nilai_cosine_pegawai = cosine_similarity-
>calculate(keyword_target, contain['pegawai']);
}
sorting_pegawai = usort(nilai_cosine_similarity);
//Looping pegawai untuk output user
```

```

Foreach sorting_pegawai as pegawai{
    RETURN pegawai[Divisi] . echo pegawai[Jabatan];
    RETURN pegawai>Nama pegawai];
    RETURN pegawai[cosine similarity];
}

```

4. Implementasi Tampilan Admin Interface / Backend

Pada tahap ini merupakan tahap implementasi halaman backend sebagai halaman admin untuk mengelola data pegawai yang hendak dimasukkan dalam sistem. Dengan tampilan sebagai berikut:

a. Login Admin

Sebelum masuk dalam pengelolaan data pegawai, perlu dilakukan login sebagai admin terlebih dahulu. Admin memasukkan *username* dan *password* untuk mengelola data aplikasi yang kemudian akan disimpan dalam *database system*. *Pseudocode* untuk proses *login* adalah sebagai berikut.

Login admin

Declaration:

username: variable username admin

password: variable password admin

Algorithm:

1. *form login admin*

Input data username dan password

Klik Sign in

2. *Proses decision checking data admin*

If \$username and \$password == true{

Redirect -> View admin page

}else{

Redirect -> View form login admin

}

Pada proses *login admin* akan dilihat apakah benar akun yang digunakan betul sebagai admin yang memiliki otoritas untuk mengatur atau mengelola *database system* terkait pegawai perusahaan.

b. CRUD (*Create, Read, Update, Delete*) Data Divisi

Halaman ini berfungsi untuk mengelola data master divisi. Pada halaman submenu divisi, Admin dapat mengelola data master divisi baik menambah, mengubah, dan menghapus data. *Pseudocode* dari proses CRUD data divisi sebagai berikut.

<i>CRUD (Create, Read, Update, Delete) Data Divisi</i>
<i>Declaration:</i> <i>divisi: variable nama divisi</i>
<i>Algorithm:</i> <i>1. Create Divisi</i> <i>Klik tambah (form tambah divisi)</i> <i>Input data divisi</i> <i>Klik Save</i> <i>If(count(\$divisi)>0){ //cek jika data divisi sudah ada</i> <i>Data not saved to database</i> <i>Redirect view page divisi -> flashmessage('Gagal, Data sudah ada!');</i> <i>}else{</i> <i> Data saved to database</i> <i>Redirect view page divisi -> flashmessage('Data saved!');</i> <i>}</i> <i>2. Update Divisi</i> <i>Klik ubah (form ubah divisi)</i> <i>Input data divisi</i> <i>Klik Save</i> <i>If(count(\$divisi)>0){ //cek jika data divisi sudah ada</i> <i>Data not saved to database</i>

```

Redirect view page divisi -> flashmessage('Gagal, Data sudah
ada!');

}else{

  Data saved to database

Redirect view page divisi -> flashmessage('Data saved!');

}

3. Delete data

Klik delete (form alert hapus data)

Klik OK

If(klik OK){

Data delete from database

Redirect view page divisi -> flashmessage('Data deleted!');

}else{

Data not delete from database

Redirect view page divisi

}

```

Pengelolaan data divisi berkaitan dengan data jabatan yang mana data divisi berperan sebagai data induk, dan jabatan berperan sebagai subdatanya.

c. *CRUD (Create, Read, Update, Delete) Data Jabatan*

Halaman ini berfungsi untuk mengelola data master jabatan. Pada halaman *backend* menu kepegawaian submenu jabatan, *Admin* dapat mengelola data master jabatan baik menambah, mengubah, dan menghapus data. *Pseudocode* untuk proses *CRUD* data jabatan sebagai berikut.

<i>CRUD (Create, Read, Update, Delete) Data Jabatan</i>
<p><i>Declaration:</i></p> <p><i>\$divisi: variable nama divisi</i></p> <p><i>\$jabatan: variable nama jabatan</i></p>
<p><i>Algorithm:</i></p> <p>1. <i>Create jabatan</i></p> <p><i>Klik tambah (form tambah jabatan)</i></p>

```

Input data divisi
Input data jabatan

Klik Save

If(count($jabatan)>0){ //cek jika data jabatan sudah ada
Data not saved to database

Redirect view page jabatan -> flashmessage('Gagal, Data sudah
ada!');

}else{

  Data saved to database
Redirect view page jabatan -> flashmessage('Data saved!');
}

2. Update Jabatan
Klik ubah (form ubah jabatan)
Input data divisi
Input data jabatan
Klik Save

If(count($jabatan)>0){ //cek jika data jabatan sudah ada
Data not saved to database

Redirect view page jabatan -> flashmessage('Gagal, Data sudah
ada!');

}else{
  Data saved to database
Redirect view page jabatan -> flashmessage('Data saved!');
}

3. Delete data
Klik delete (form alert hapus data)
Klik OK

If(klik OK){

Data delete from database

```

```

Redirect view page jabatan -> flashmessage('Data deleted!');
}else{
Data not delete from database
Redirect view page jabatan
}

```

Pengelolaan data jabatan berkaitan dengan data pegawai yang mana data pegawai berperan sebagai data induk, dan jabatan berperan sebagai subdatanya.

d. CRUD (Create, Read, Update, Delete) Data Pegawai

Halaman ini berfungsi untuk mengelola data pegawai. Pada halaman *backend* menu kepegawaian submenu pegawai, *Admin* dapat mengelola data pegawai baik menambah, mengubah, dan menghapus data. *Pseudocode* dalam pengelolaan data pegawai sebagai berikut.

CRUD (Create, Read, Update, Delete) Data Pegawai
<p>Declaration:</p> <pre> \$nama_pegawai: variable nama pegawai \$jenis_kelamin: variable jenis kelamin pegawai \$jabatan: variable nama jabatan </pre>
<p>Algorithm:</p> <p>1. Create pegawai</p> <p>Klik tambah (form tambah pegawai)</p> <p>Input nama pegawai</p> <p>Input jenis kelamin</p> <p>Input jabatan pegawai</p> <p>Klik Save</p> <p>Data saved to database</p> <p>Redirect view page pegawai -> flashmessage('Data saved!');</p> <p>2. Update Pegawai</p> <p>Klik ubah (form ubah pegawai)</p> <p>Input nama pegawai</p>

```

Input jenis kelamin
Input jabatan pegawai
Klik Save
Data saved to database
Redirect view page pegawai -> flashmessage('Data saved!');

3. Delete data
Klik delete (form alert hapus data)
Klik OK
If(klik OK){
Data delete from database
Redirect view page pegawai -> flashmessage('Data deleted!');
}else{
Data not delete from database
Redirect view page pegawai
}

```

Pengelolaan data pegawai berkaitan dengan pencarian data yang mana data pegawai berperan sebagai database yang akan menentukan tingkat *cosine similarity* dengan *keyword* yang diinputkan dalam pencarian pegawai.

4.1.7 Spesifikasi *Software* dan *Hardware*

Dalam membangun aplikasi pencarian data pegawai berbasis web, penulis menggunakan spesifikasi perangkat sebagai berikut:

1. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan antara lain:

- a. Xampp-windows-x64-7.2.33-0-VC15-installer
- b. Sublime Text versi 3.2.2
- c. PHP versi 7.2.33
- d. MYSQL versi 5.0.12
- e. Chrome versi 106.0.5249.119 – 64bit

2. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan adalah laptop dengan spesifikasi antara lain:

- a. Windows 10 home 64bit

- b. RAM 4 Gb
- c. Hardisk 500Gb
- d. *Processor* Intel Core I3

4.2. Pembangunan Aplikasi

Setelah mengetahui kebutuhan *user*, selanjutnya pada tahap ini berisi proses pembangunan sistem atau implementasi sistem yaitu menentukan spesifikasi *software* dan *hardware* sistem, dan implementasi sistem (*coding*) sesuai dengan kebutuhan user.

4.2.1 Implementasi *Preprocessing Data*

Pada tahap ini dibutuhkan *script* program yang berfungsi untuk melakukan *preprocessing* data sebagai bentuk persiapan data sebelum dicari makna semantiknya. Dengan memanfaatkan *library Sastrawi* yang diintegrasikan dengan aplikasi web berbasis PHP dihasilkan data dengan tiap tahap dari *preprocessing* sebagai berikut:



Gambar 4. 9 *Preprocessing data keyword*

Dari gambar diatas menunjukkan bahwa implementasi tahap *preprocessing* adalah sebagai berikut:

- e. *Keyword*

Keyword yang dimasukkan berupa kalimat “mencari pegawai yang ahli dibidang listrik dan mekanik mesin###%!?@”, yang mengandung beberapa karakter unik serta kata hubung dan imbuhan kata.

f. *Case folding*

Case folding merupakan tahap untuk menghilangkan karakter unik dari *string*. Dari tahap *Case folding* menunjukkan bahwa karakter unik pada kalimat telah dihilangkan menjadi “mencari pegawai yang ahli dibidang listrik dan mekanik mesin”.

g. *Stemming*

Stemming merupakan tahap yang bertujuan untuk mengubah kata yang memiliki imbuhan kata menjadi kata dasar saja. Dari tahap *stemming* menunjukkan bahwa imbuhan kata pada kata “mencari” menjadi “cari”. Sehingga menghasilkan kalimat “cari pegawai yang ahli bidang listrik dan mekanik mesin”.

h. *Filtering*

Filtering merupakan tahap yang bertujuan untuk menghilangkan kata penghubung atau kata yang dianggap tidak penting. Sehingga dari tahap ini kalimat berubah menjadi “cari pegawai ahli bidang listrik mekanik mesin”.

i. *Tokenizing*

Tokenizing merupakan tahap yang bertujuan untuk memisahkan tiap kata dalam kalimat yang mana tiap kata akan dicari makna semantiknya. Sehingga dari tahap ini menghasilkan *keyword* masing-masing “cari, pegawai, ahli, bidang, listrik, mekanik, mesin”.

4.2.2 Implementasi Pencarian Makna Semantik

Pada tahap ini aplikasi web yang dibangun memanfaatkan *library Satria/SynonymAntonym* untuk menemukan makna semantik dari tiap kata yang dihasilkan oleh tahap *preprocessing* data.

Hasil makna semantik dari *keyword* adalah sebagai berikut:

Daftar keyword dan makna semantic
cari, pegawai, ahli, bidang, listrik, mekanik, mesin, gagar, geledah, kerekau, kerossek, mencekau, raba, fungsionaris, karyawan, personel, sida-sida, tenaga

kerja, andal, bernas, cakap, campin, hebat, jago, jauhari, jempolan, kawakan, kompeten, lihai, mahir, mampu, mumpuni, pandai, terampil, ulung, unggul, berida, berilmu, berpengalaman, berpengetahuan, cendekiawan, cerdik pandai, empu, johar, juara, master, pakar, pendeta, piawai, profesional, sarjana, spesialis, teknikus, tukang, ulama, anggota, bani, kaum, peserta, wakil, warga, darah daging, famili, kadim, keluarga, kerabat, keturunan, marga, sanak, saudara, aspek, bagian, datar, disiplin, dunia, faktor, jurusan, keahlian, kelompok, lapangan, latar, lebar, lingkungan, lopak, papak, papar, permukaan, petak, piring, segi, sektor, sisi, sudut pandangan, tempat, vak, elektrik, setrum, ahli mesin, insinyur, operator, teknikus, teknisi, alat, alat perkakas, instrumen, jentera, motor, pesawat,

Dari hasil pencarian makna semantik menghasilkan beberapa *keyword* yang merupakan gabungan dari inputan *field keyword* dengan tiap kata hasil *preprocessing* yang telah dihasilkan makna semantiknya. *Keyword* asli ditunjukkan dengan kata yang bergaris bawah, sedangkan yang tanpa garis bawah merupakan daftar makna semantik tiap kata setelah dilakukan *preprocessing* data.

4.2.3 Implementasi Pencocokan *String* (*Cossine Similarity*)

Pada tahap ini aplikasi web mencari daftar pegawai yang memiliki kandungan data seperti salah satu daftar makna semantik yang dihasilkan. Hasil dari pencocokan *string* dengan data pegawai menghasilkan matriks *array* sebagai berikut.

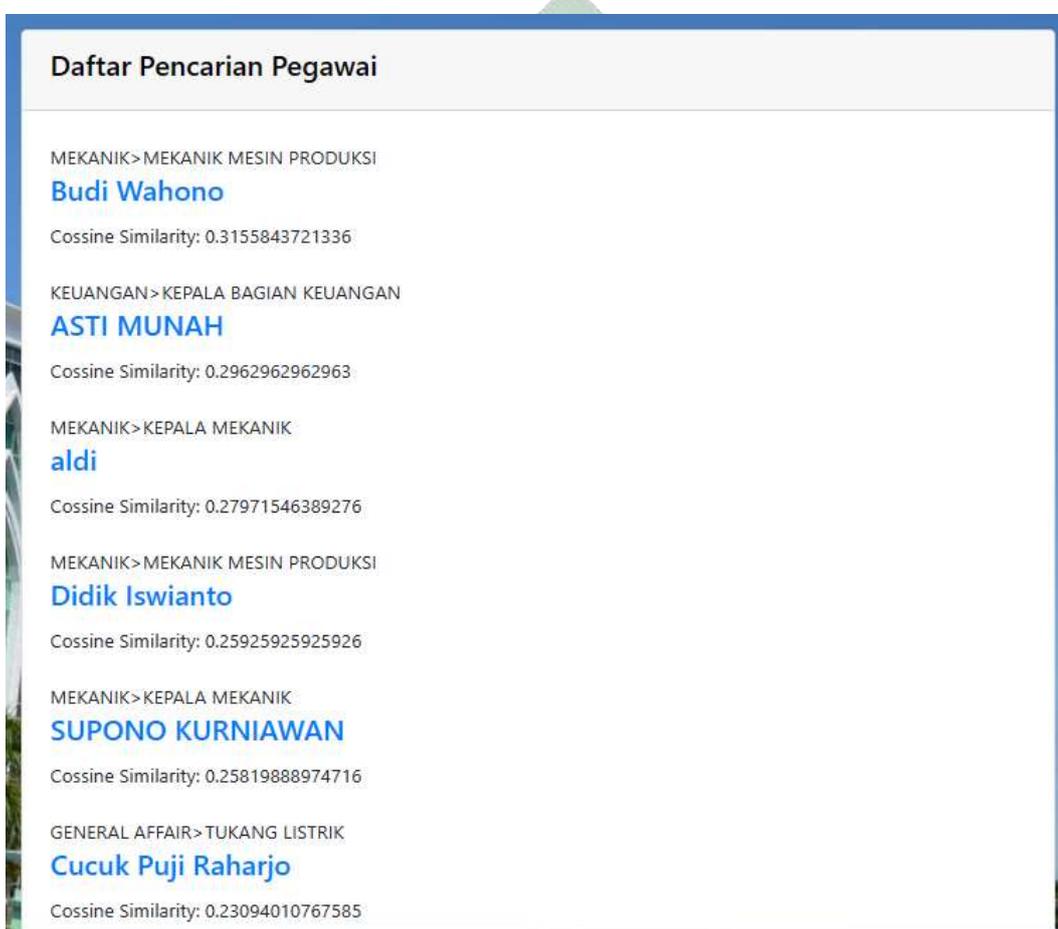
Tabel 4 1. Tabel *array* nilai pencocokan *string*

	cari	pegawai	ahli	bidang	listrik	mekanik	mesin
Pegawai target	1	1	1	1	1	1	1
Pegawai 4						1	
Pegawai 5						1	
Pegawai 6						1	
Pegawai 7						1	
Pegawai 12					1		
Pegawai 15					1	1	
Pegawai 16					1	1	
Pegawai 17					1	1	
Pegawai 18					1	1	

Pegawai 19					1	1	
------------	--	--	--	--	---	---	--

Matriks yang dihasilkan pada *table* diatas mengambil 7 kata kunci (cari, pegawai, ahli, bidang, listrik, mekanik, mesin) dari input *keyword* yang telah melalui *preprocessing* data. Matriks ini hanya contoh karena kata kunci yang dihasilkan berjumlah 108 kata (beserta makna semantik).

Tiap pegawai akan dicari kecocokan dengan kata kuncinya dan akan diberi nilai 1 jika kata kunci terkandung pada data pegawai. Yang mana perhitungan *cosine similarity* akan mencari tingkat kemiripan antara pegawai target yang memiliki kecocokan pada semua kata kunci dengan pegawai dalam database. Sehingga akan dihasilkan tingkat kecocokan *cosine similarity* sebagai berikut.



Gambar 4. 10 Hasil pencarian dan tingkat *cosine similarity*

Tingkat kemiripan *cosine similarity* ditampilkan pada hasil pencarian, yang merupakan hasil perbandingan dari target pegawai yang dicari (memiliki kecocokan pada semua kata kunci) dengan pegawai dalam *database*. Perhitungan tingkat kemiripan data pegawai dengan *keyword* yang dihasilkan menggunakan metode

cosine similarity diurutkan dari tingkat kemiripan tertinggi atau secara *descending*. Hal ini agar pengguna dapat dengan mudah dalam menemukan rekomendasi pegawai yang cocok dengan *keyword*.

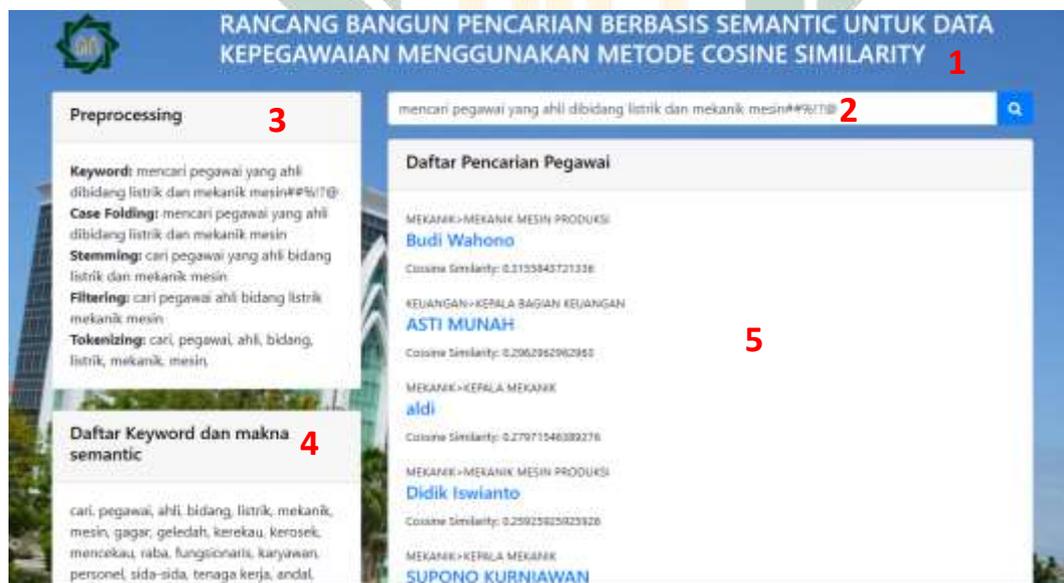
Hasil dari pencocokan *string* menunjukkan bahwa data pegawai yang teridentifikasi identik dengan kata mekanik, dan listrik. Hal ini dikarenakan atribut atau kelengkapan data pegawai masih berupa divisi, dan jabatan pegawai. Jika kelengkapan data mencakup data pegawai yang lebih luas seperti profil pegawai dari segi riwayat hidup, riwayat pekerjaan, hobi, keahlian, dan deskripsi keahlian maka hasil pencarian akan menghasilkan rekomendasi pegawai yang lebih lengkap.

4.3. Implementasi *Screenshot* Aplikasi

Bagian ini menjelaskan bagian-bagian interface sistem baik dari *frontend* maupun *backend*, beserta penjelasan tiap bagian.

4.2.4 Implementasi Tampilan *User Interface*

Pada tahap ini merupakan tahap implementasi aplikasi untuk pencarian pegawai dari sisi *user*. Dengan tampilan *user* sebagai berikut:



Gambar 4. 11 Tampilan *User* (halaman pencarian)

Tampilan antarmuka / *user interface* tersusun atas beberapa bagian yang secara urut yaitu:

1. *Top header* berisi Judul program / Aplikasi.
2. *Header Body* berisi *field input text* untuk *input keyword*.
3. *Sidebar* kiri-atas berisi tahap *preprocessing* data yang menampung hasil

preprocessing input keyword.

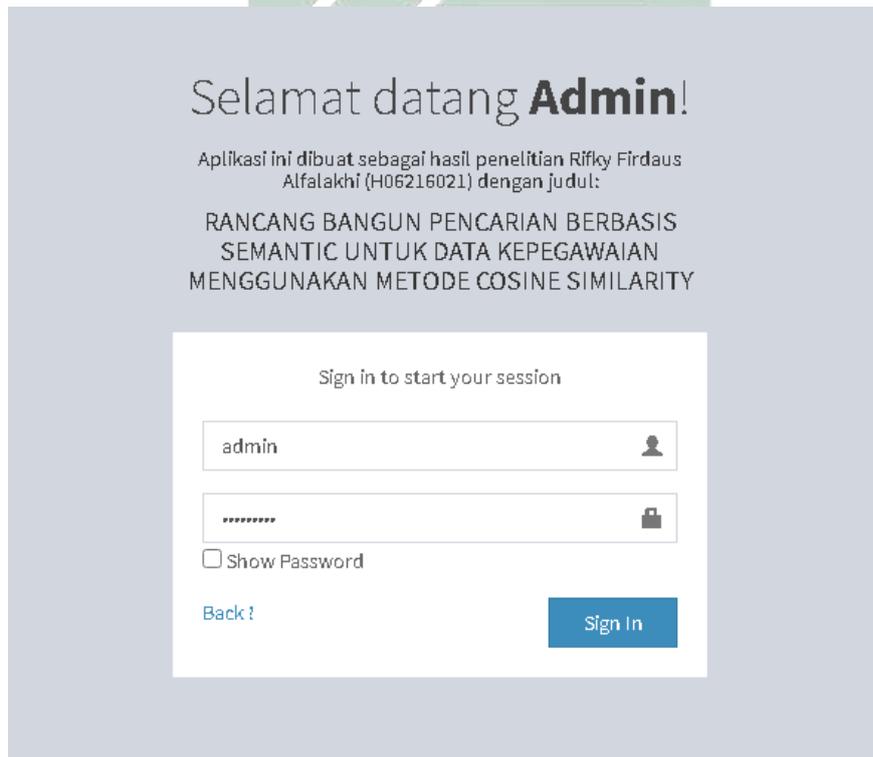
4. *Sidebar* kiri-bawah berisi data makna semantik yang merupakan hasil dari pencarian makna semantic dari hasil *preprocessing keyword.*
5. *Body* berisi daftar pegawai yang terkait dengan daftar *keyword* hasil *preprocessing* dan makna semantik yang dihasilkan.

4.2.5 Implementasi Tampilan *Admin Interface / Backend*

Pada tahap ini merupakan tahap implementasi halaman backend sebagai halaman admin untuk mengelola data pegawai yang hendak dimasukkan dalam sistem. Dengan tampilan sebagai berikut:

a. Login Admin

Sebelum masuk dalam pengelolaan data pegawai, perlu dilakukan login sebagai admin terlebih dahulu. Tampilan login admin sebagai berikut.

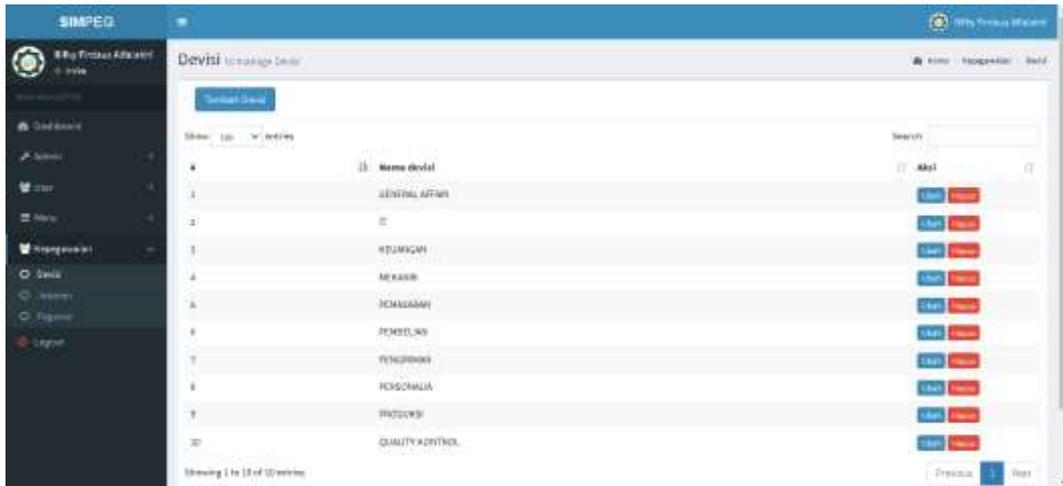


Gambar 4. 12 Halaman login admin

Admin memasukkan *username* dan *password* untuk mengelola data aplikasi yang kemudian akan disimpan dalam *database system*.

b. CRUD (*Create, Read, Update, Delete*) Data Divisi

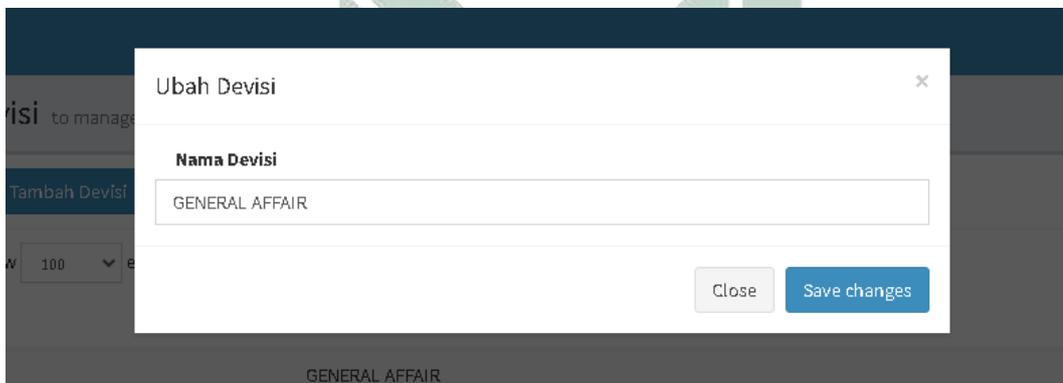
Halaman ini berfungsi untuk mengelola data master divisi dengan tampilan sebagai berikut.



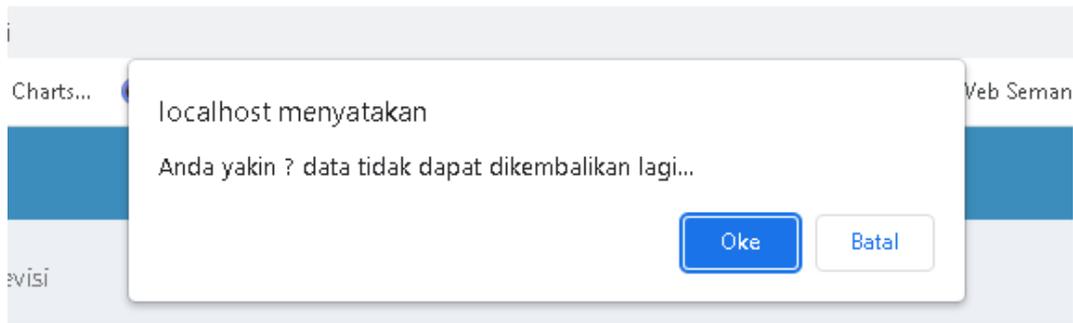
Gambar 4. 13 Halaman submenu divisi



Gambar 4. 14 Tambah data divisi



Gambar 4. 15 Ubah data divisi

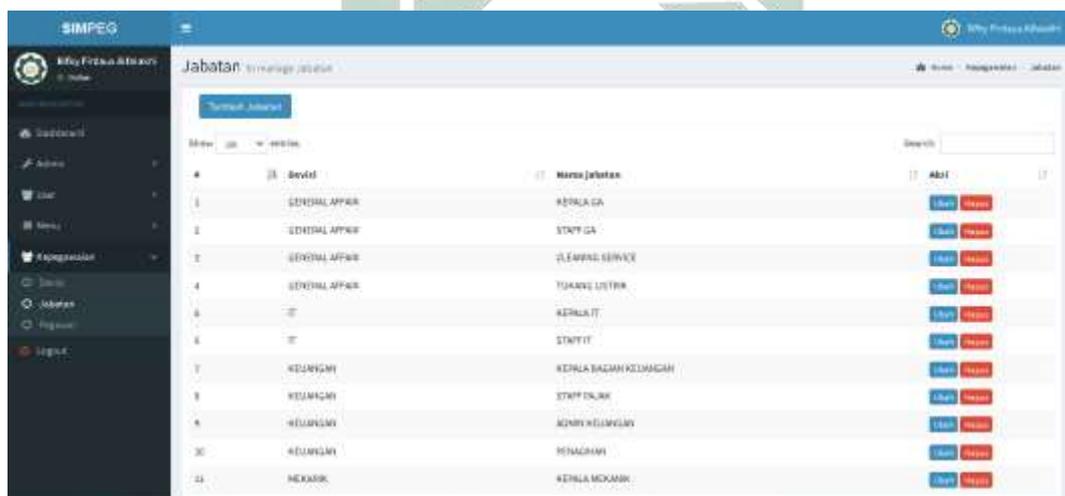


Gambar 4. 16 Hapus data divisi

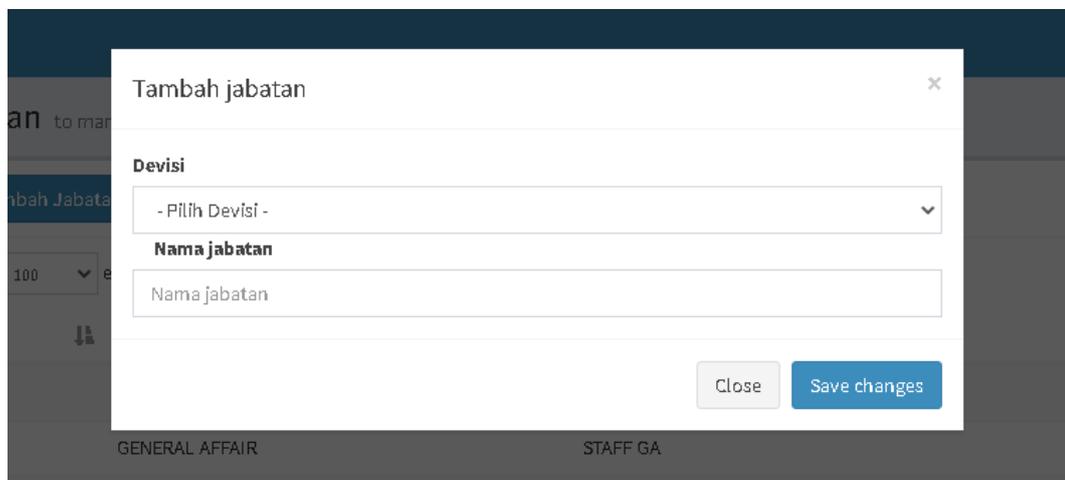
Pada halaman *backend* menu kepegawaian submenu divisi, Admin dapat mengelola data master divisi baik menambah, mengubah, dan menghapus data.

c. CRUD (*Create, Read, Update, Delete*) Data Jabatan

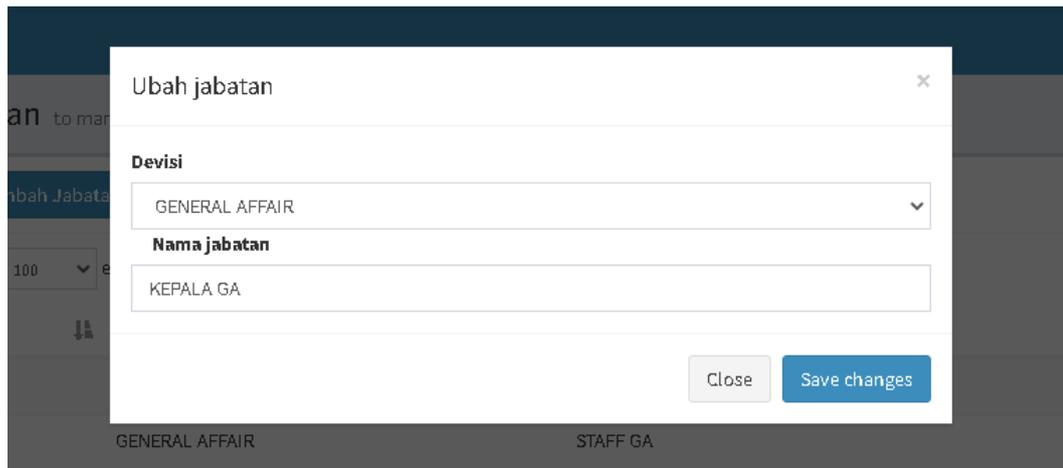
Halaman ini berfungsi untuk mengelola data master jabatan dengan tampilan sebagai berikut.



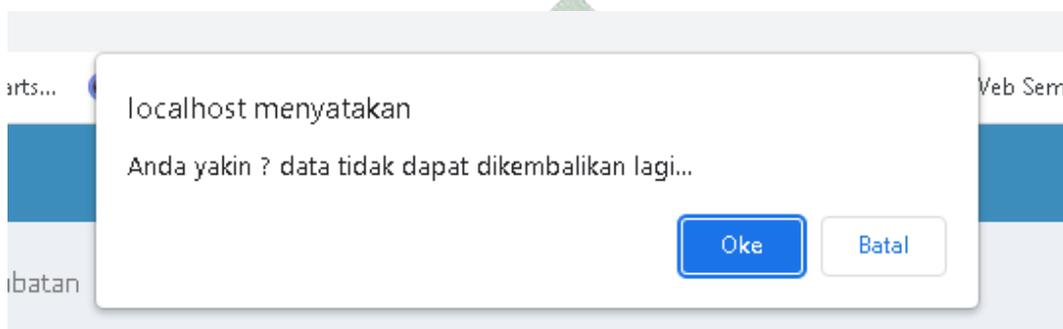
Gambar 4. 17 Halaman submenu jabatan



Gambar 4. 18 Tambah data jabatan



Gambar 4. 19 Ubah data jabatan

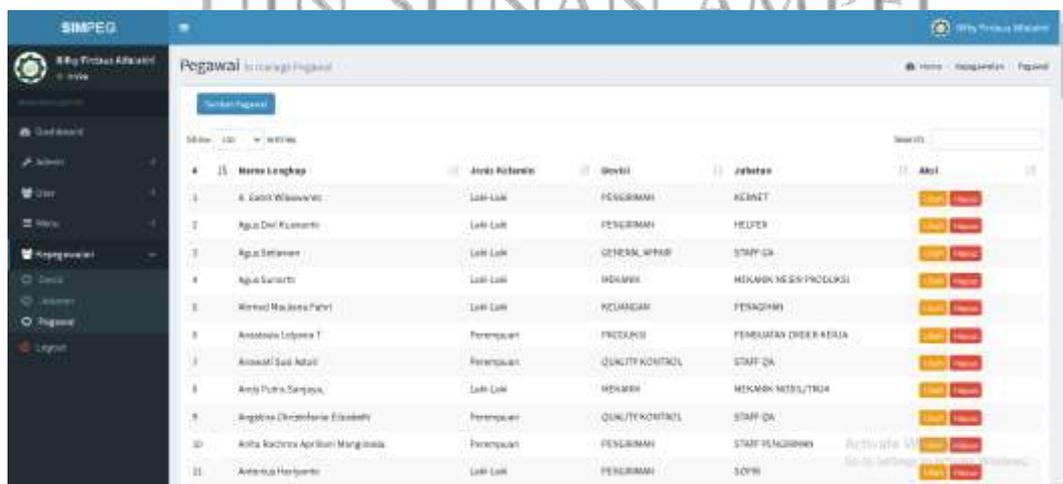


Gambar 4. 20 Hapus data jabatan

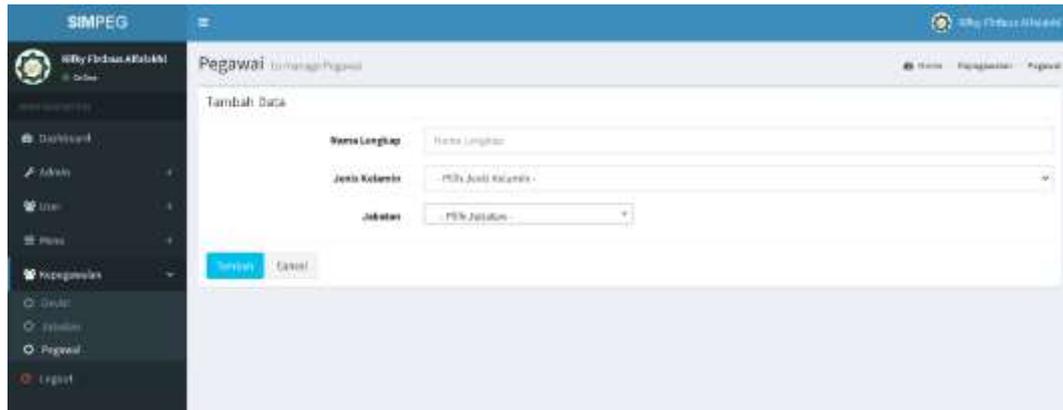
Pada halaman *backend* menu kepegawaian submenu jabatan, *Admin* dapat mengelola data master jabatan baik menambah, mengubah, dan menghapus data.

d. CRUD (*Create, Read, Update, Delete*) Data Pegawai

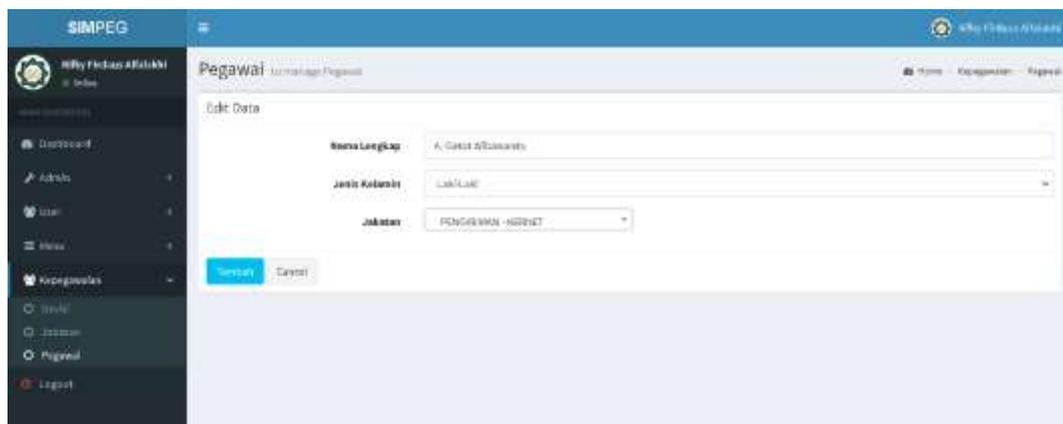
Halaman ini berfungsi untuk mengelola data pegawai dengan tampilan sebagai berikut.



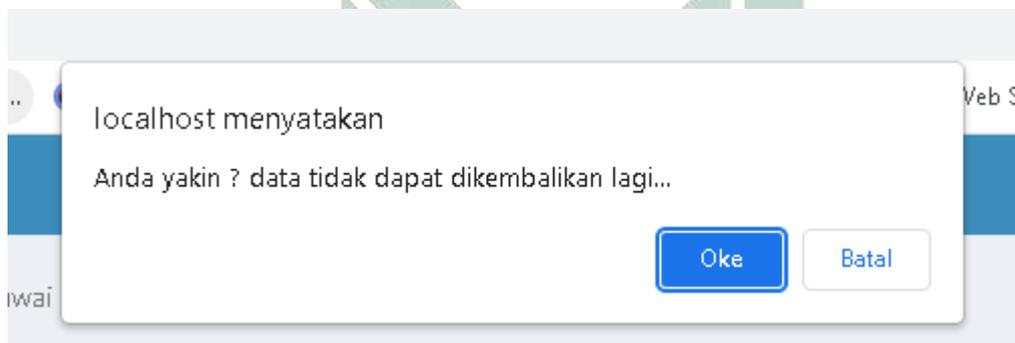
Gambar 4. 21 Halaman submenu pegawai



Gambar 4. 22 Tambah data pegawai



Gambar 4. 23 Ubah data pegawai



Gambar 4. 24 Hapus data pegawai

Pada halaman *backend* menu kepegawaian submenu pegawai, *Admin* dapat mengelola data pegawai baik menambah, mengubah, dan menghapus data.

4.4. Skenario Pengujian Sistem

Setelah mengimplementasikan kebutuhan *user* dan respon yang diberikan terkait kebutuhan sistem telah sesuai maka selanjutnya masuk pada tahap pengujian data. Pengujian data dilakukan dengan *blackbox testing*, yang akan menguji tiap komponen pada aplikasi. Hasil dari pengujian blackbox adalah sebagai berikut.

Tabel 4 2. Pengujian fungsional sistem

No	Nama	Use Case	Hasil yang diharapkan	Hasil Pengujian
1	Proses login	Login sebagai admin	Masuk ke halaman pegawai	OK
2	Input data divisi	Login sebagai admin	Data master divisi bertambah	OK
3	Edit data divisi	Login sebagai admin	Data master divisi berubah	OK
4	Hapus data divisi	Login sebagai admin	Data master divisi terhapus	OK
5	Input data jabatan	Login sebagai admin	Data master jabatan bertambah	OK
6	Edit data jabatan	Login sebagai admin	Data master jabatan berubah	OK
7	Hapus data jabatan	Login sebagai admin	Data master jabatan terhapus	OK
8	Input data pegawai	Login sebagai admin	Data pegawai bertambah	OK
9	Edit data pegawai	Login sebagai admin	Data pegawai berubah	OK
10	Hapus data pegawai	Login sebagai admin	Data pegawai terhapus	OK
11	Input data pencarian	Fitur User (tanpa login)	<i>Keyword</i> berhasil masuk untuk kemudian diolah dalam <i>preprocessing</i> data	OK
12	<i>Case folding</i> pencarian	Fitur User (tanpa login)	<i>Keyword</i> yang dimasukkan dapat dihilangkan karakter	OK

			uniknya (selain angka dan huruf).	
13	Stemming data pencarian	Fitur User (tanpa login)	<i>Keyword</i> hasil <i>case folding</i> dapat dijadikan dalam bentuk kata dasar (dihilangkan kata imbuhan jika ada).	OK
14	Filtering data pencarian	Fitur User (tanpa login)	<i>Keyword</i> hasil <i>stemming</i> dapat disaring dengan menghilangkan kata penghubung atau kata yang tidak memiliki makna.	OK
15	Tokenizing data pencarian	Fitur User (tanpa login)	<i>Keyword</i> hasil <i>filtering</i> dapat dipecah atau dipisah tiap kata nya dalam bentuk <i>array</i> .	OK
16	Pencarian makna semantic	Fitur User (tanpa login)	Tiap <i>keyword</i> hasil <i>preprocessing</i> dapat ditemukan makna atau sinonim katanya.	OK
17	Pencarian pegawai	Fitur User (tanpa login)	Daftar pegawai ditampilkan sesuai <i>keyword</i> yang telah diproses	OK
18	Perhitungan tingkat cosine similarity	Fitur User (tanpa login)	Setiap daftar pegawai ditampilkan nilai <i>cosine similarity</i> berdasarkan perhitungan matriks	OK

			<i>array keyword</i> dalam setiap pegawai yang terindikasi memiliki kecocokan dengan <i>keyword</i>	
--	--	--	---	--

Berdasarkan hasil pengujian, diketahui bahwa sistem dapat digunakan dengan baik. Mulai dari manajemen data pegawai sampai mengolah data *keyword* untuk ditemukan makna semantic dan pencarian data pegawai dengan perhitungan tingkat *cosine similarity*.

4.5. Requirements Traceability Matrix

Requirements Traceability Matrix (RTM) Sebagai tambahan terkait kebutuhan pengujian dengan keterkaitan dokumen atau system yang dibangun. ketelurusan dokumen terkait pengujian sistem secara runtut adalah sebagai berikut.

Tabel 4.3 *Requirements Traceability Matrix*

Nama Projek	Sistem Pencarian Pegawai Berbasis Semantik dengan Cosine Similarity
Deskripsi Projek	Pencarian Pegawai berdasarkan persamaan makna keyword dengan tingkat kemiripan (<i>cosine similarity</i>)

No	Kebutuhan Utama	Sub Kebutuhan	Kategori	Deskripsi	Relasi Sistem	Use Case	Dokumen DFD Desain	Hasil yang diharapkan
1	Backend Sistem	Proses login	Dibutuhkan	Proses verifikasi akun untuk akses halaman backend	Asosiasi	Login sebagai admin	Gambar 4.3 DFD level 0	Masuk ke halaman pegawai
2	Backend Sistem	Input data divisi	Dibutuhkan	Proses manage (insert) data divisi karyawan	Komposisi	Login sebagai admin	Gambar 4.4 DFD level 1 dari proses 2.0	Data master divisi bertambah
3	Backend Sistem	Edit data divisi	Dibutuhkan	Proses manage (update) data divisi karyawan	Komposisi	Login sebagai admin	Gambar 4.4 DFD level 1 dari proses 2.0	Data master divisi berubah
4	Backend Sistem	Hapus data divisi	Dibutuhkan	Proses manage (delete) data divisi karyawan	Komposisi	Login sebagai admin	Gambar 4.4 DFD level 1 dari proses 2.0	Data master divisi terhapus

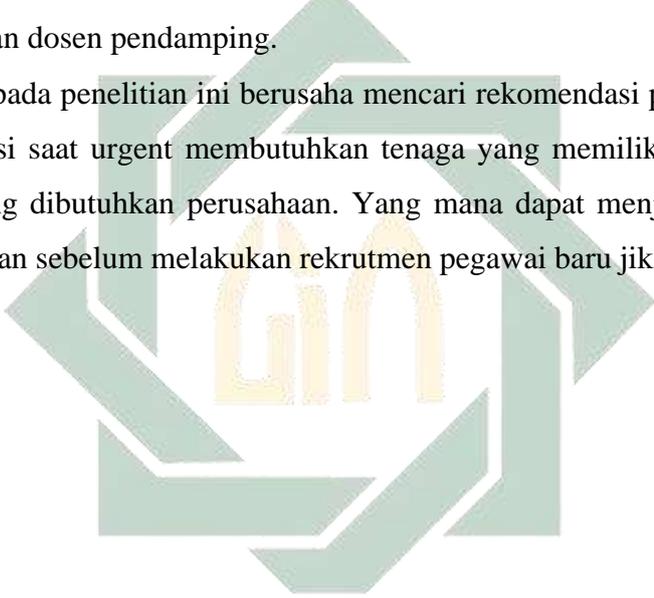
5	Backend Sistem	Input data jabatan	Dibutuhkan	Proses manage (insert) data jabatan karyawan	Komposisi	Login sebagai admin	Gambar 4.4 DFD level 1 dari proses 2.0	Data master jabatan bertambah
6	Backend Sistem	Edit data jabatan	Dibutuhkan	Proses manage (update) data jabatan karyawan	Komposisi	Login sebagai admin	Gambar 4.4 DFD level 1 dari proses 2.0	Data master jabatan berubah
7	Backend Sistem	Hapus data jabatan	Dibutuhkan	Proses manage (delete) data jabatan karyawan	Komposisi	Login sebagai admin	Gambar 4.4 DFD level 1 dari proses 2.0	Data master jabatan terhapus
8	Backend Sistem	Input data pegawai	Dibutuhkan	Proses manage (insert) data karyawan	Komposisi	Login sebagai admin	Gambar 4.4 DFD level 1 dari proses 2.0	Data pegawai bertambah
9	Backend Sistem	Edit data pegawai	Dibutuhkan	Proses manage (update) data karyawan	Komposisi	Login sebagai admin	Gambar 4.4 DFD level 1 dari proses 2.0	Data pegawai berubah
10	Backend Sistem	Hapus data pegawai	Dibutuhkan	Proses manage (delete) data karyawan	Komposisi	Login sebagai admin	Gambar 4.4 DFD level 1 dari proses 2.0	Data pegawai terhapus
11	Frontend Sistem	Input data pencarian	Dibutuhkan	Proses input keyword pencarian data	Asosiasi	Fitur User (tanpa login)	Gambar 4.6 DFD level 2 dari proses 3.1, 3.2, 3.3, 3.4	<i>Keyword</i> berhasil masuk untuk kemudian diolah dalam <i>preprocessing</i> data
12	Frontend Sistem	<i>Case folding</i> pencarian	Diharapkan	Proses text preprocessing keyword pencarian data (tahap case folding)	Asosiasi	Fitur User (tanpa login)	Gambar 4.6 DFD level 2 dari proses 3.1, 3.2, 3.3, 3.4	<i>Keyword</i> yang dimasukkan dapat dihilangkan karakter uniknya (selain angka dan huruf).
13	Frontend Sistem	Stemming data pencarian	Diharapkan	Proses text preprocessing keyword pencarian data (tahap stemming)	Asosiasi	Fitur User (tanpa login)	Gambar 4.6 DFD level 2 dari proses 3.1, 3.2, 3.3, 3.4	<i>Keyword</i> hasil <i>case folding</i> dapat dijadikan dalam bentuk kata dasar (dihilangkan

								kata imbuhan jika ada).
14	Frontend Sistem	Filtering data pencarian	Diharapkan	Proses text preprocessing keyword pencarian data (tahap filtering)	Asosiasi	Fitur User (tanpa login)	Gambar 4.6 DFD level 2 dari proses 3.1, 3.2, 3.3, 3.4	Keyword hasil stemming dapat disaring dengan menghilangkan kata penghubung atau kata yang tidak memiliki makna.
15	Frontend Sistem	Tokenizing data pencarian	Diharapkan	Proses text preprocessing keyword pencarian data (tahap tokenizing)	Asosiasi	Fitur User (tanpa login)	Gambar 4.6 DFD level 2 dari proses 3.1, 3.2, 3.3, 3.4	Keyword hasil filtering dapat dipecah atau dipisah tiap kata nya dalam bentuk array.
16	Frontend Sistem	Pencarian makna semantic	Diharapkan	Proses pencarian makna semantik keyword	Asosiasi	Fitur User (tanpa login)	Gambar 4.6 DFD level 2 dari proses 3.1, 3.2, 3.3, 3.4	Tiap keyword hasil preprocessing dapat ditemukan makna atau sinonim katanya.
17	Frontend Sistem	Pencarian pegawai	Diharapkan	Proses pencarian daftar pegawai dari kecocokan dengan keyword	Asosiasi	Fitur User (tanpa login)	Gambar 4.6 DFD level 2 dari proses 3.1, 3.2, 3.3, 3.4	Daftar pegawai ditampilkan sesuai keyword yang telah diproses
18	Frontend Sistem	Perhitungan tingkat cosine similarity	Diharapkan	Proses penentuan tingkat kemiripan dengan cosine similarity	Asosiasi	Fitur User (tanpa login)	Gambar 4.6 DFD level 2 dari proses 3.1, 3.2, 3.3, 3.4	Setiap daftar pegawai ditampilkan nilai cosine similarity berdasarkan perhitungan matriks array keyword dalam setiap pegawai yang terindikasi memiliki kecocokan dengan keyword

4.6. Pembahasan

Berdasarkan hasil yang telah dijabarkan menunjukkan bahwa pencarian data pegawai berdasarkan *keyword* dan makna semantik dengan metode *cosine similarity* dapat digunakan untuk menemukan rekomendasi pegawai terkait. Hal ini sejalan dengan penelitian (Atikah Azhari, 2022) tentang pencarian rekomendasi dosen pendamping bagi mahasiswa berdasarkan bidang penelitian. Yang mana dari hasil penelitian tersebut ditemukan bahwa metode *cosine similarity* dapat diterapkan untuk menemukan rekomendasi dosen pendamping bagi mahasiswa yang membutuhkan dosen pendamping.

Sedangkan pada penelitian ini berusaha mencari rekomendasi pegawai yang dapat menjadi opsi saat urgent membutuhkan tenaga yang memiliki kompetensi sesuai bidang yang dibutuhkan perusahaan. Yang mana dapat menjadi alternatif sementara pekerjaan sebelum melakukan rekrutmen pegawai baru jika dibutuhkan.



UIN SUNAN AMPEL
S U R A B A Y A

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan rumusan masalah serta hasil analisis dan perancangan Sistem Pencarian Berbasis *Semantic* Untuk Data Kepegawaian Menggunakan Metode *Cosine Similarity*, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Sistem dapat menghasilkan informasi berupa makna semantik dari *input keyword* yang digunakan untuk mencari data pegawai terkait. Informasi makna semantik dihasilkan melalui beberapa tahap proses yang dimulai dari tahap *preprocessing* (*case folding, stemming, filtering, dan tokenizing*) yang memanfaatkan *library PHP Sastrawi*, dan mentranslasikan *keyword* hasil *preprocessing* ke daftar sinonim dalam bahasa Indonesia dengan memanfaatkan *library PHP Satria/Synonym Antonym*.
2. Sistem dapat menemukan nilai perhitungan tingkat kemiripan antara *keyword* serta makna semantik dengan atribut data pegawai (nama pegawai, divisi, dan jabatan) menggunakan metode *cosine similarity*. Nilai tingkat kemiripan diperoleh dari pencocokan string dari setiap kata *keyword* dengan atribut data pegawai menggunakan *cosine similarity* dengan memanfaatkan *library PHP Mlwmw/php-cosine-similarity* yang sebelumnya telah dilakukan *preprocessing text* untuk mendapatkan *keyword* dan makna semantik *keyword*.

Sehingga dapat diambil kesimpulan bahwa pencarian pegawai berdasarkan makna semantik *keyword* dengan menggunakan metode *cosine similarity* dapat digunakan untuk menemukan rekomendasi pegawai sesuai bidang terkait.

5.2 Saran

Berdasarkan penelitian yang dilakukan, atribut data yang digunakan dalam penelitian ini hanya seputar nama pegawai, divisi, dan jabatan. Maka hanya seputar data tersebut yang dilakukan pencocokan *string* menggunakan metode *cosine similarity*. Sehingga dapat menjadi opsi untuk penelitian selanjutnya dalam implementasi metode *cosine similarity* menggunakan atribut data yang lebih luas seperti profil, hobi, latar belakang pendidikan, riwayat pekerjaan, dan lain

sebagainya, sehingga sistem pencarian dapat menghasilkan rekomendasi hasil yang dapat melihat kompetensi individu dari lingkup yang lebih luas.



UIN SUNAN AMPEL
S U R A B A Y A

DAFTAR PUSTAKA

- Abdillah, G. Y., Andryana, S., & Iskandar, A. (2020). Augmented Reality Sebagai Media Pembelajaran Perangkat Keras Komputer Dengan Fast Corner Dan Natural Feature Tracking. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 5(2), 79. <https://doi.org/10.29100/jipi.v5i2.1767>
- Arifin, N. Y., & Prasetyo, E. (2021). Perancangan sistem web semantik database dokumen qa. *Engineering and Technology International ...*, 3(1), 46–54. <https://www.mand-ycmm.org/index.php/eatij/article/view/65>
- Ayuwandira, A. H., Rambitan, S., & ... (2021). *Diksi dan Gaya Bahasa dalam Lirik Lagu Kelompok Musik “Kraftklub.”*
<https://ejournal.unsrat.ac.id/index.php/jefs/article/view/36553%0Ahttps://ejournal.unsrat.ac.id/index.php/jefs/article/download/36553/33993>
- Basmalah Wicaksono, V., Saptono, R., & Widya Sihwi, S. (2016). Analisis Perbandingan Metode Vector Space Model dan Weighted Tree Similarity dengan Cosine Similarity pada kasus Pencarian Informasi Pedoman Pengobatan Dasar di Puskesmas. *Jurnal Teknologi & Informasi ITSmart*, 4(2), 73. <https://doi.org/10.20961/its.v4i2.1768>
- Christian, A., Hesinto, S., & Agustina. (2018). Rancang Bangun Website Sekolah Dengan Menggunakan Framework Bootstrap. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 7(1), 22–27.
- Djaelangkara, R. T., Rizal Sengkey, ST., M., & Oktavian A. LAntang, ST, M. (2015). Perancangan Sistem Informasi Akademik Sekolah Berbasis Web Studi Kasus Sekolah Menengah Atas Kristen 1 Tomohon. *Comptes Rendus Des Seances de La Societe de Biologie et de Ses Filiales*, 4(3), 86–94.
- Fajri, R. M. (2016). Rancang Bangun Sistem Informasi Tracer Study Berbasis Web Studi Kasus Fakultas Ilmu Komputer Universitas Indo Global Mandiri. *Jurnal Ilmiah Informatika Global*, 7(1), 1–9.
- Fibrianto, E., Hidayat, T., Darmawan, W. B., & Djuyandi, Y. (2022). *DALAM*

MEWUJUDKAN PENGAMANAN WILAYAH PERBATASAN NEGARA (Studi Kasus di Laut Natuna Utara) dapat terwujud sesuai dengan apa yang diharapkan TNI sendiri . Interoperabilitas pemantauan adalah wilayah dimana Utara Terkait ketegasan Negara atas wilayah Laut N. 1(2), 119–128.
<https://doi.org/10.24198/aliansi.v1i2.39582>

Firman Nurdiyansyah, Samsul Arifin, F. M. (2018). Clustering algorithm untuk pengelompokan pelanggan dalam bidang usaha server reload. *Seminar Nasional Sistem Informasi*, 1043–1047.

Hafiyyan, R., Kusumo, D. S., & Sardi, I. L. (2019). Penerapan Metode Semantic Web Filtering System (SWFilter) pada Fungsi Pencarian Data: Studi Kasus pada Website telkomtesthouse. co. id. *EProceedings ...*, 6(1), 2227–2244.
<https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/8666>

Handayani, L., & Anggriani, D. (2015). Perbandingan Model Chen Dan Model Lee Pada Metode Fuzzy Time Series Untuk Prediksi Harga Emas. *Pseudocode*, 2(1), 28–36. <https://doi.org/10.33369/pseudocode.2.1.28-36>

Hostinger. (2023). *Perbedaan SQL dan MySQL: Penjelasan Mudah & Lengkap*. 04-01-2023. <https://www.hostinger.co.id/tutorial/perbedaan-sql-dan-mysql#:~:text=SQL dan MySQL memang sering,MySQL%2C padanannya adalah SQL Server.>

Izza, M. A., Jazuli, A., & Nurkamid, M. (2022). Implementasi Teknologi Semantik Web Untuk Pencarian Koleksi Perpustakaan Universitas Muria Kudus. *Jurnal Dialektika Informatika (Detika)*, 2(2), 56–62.
<https://doi.org/10.24176/detika.v2i2.7884>

Jembarnata, gemar ahmad. (2011). *Rancang Bangun Aplikasi Trouble Ticket Management Berbasis Web Dengan Menggunakan Expert System (Studi Kasus : Bppt) Management Berbasis Web Dengan Menggunakan*.

Jusuf, H., & Azimah, A. (2016). Perancangan Web Semantik Aplikasi Pencarian Tumbuhan Obat Menggunakan Protégé 4.3.0. *Seminar Nasional APTIKOM*,

861–863.

- Kirana, M. C., Perkasa, N. P., Lubis, M. Z., & Fani, M. (2019). Visualisasi Kualitas Penyebaran Informasi Gempa Bumi di Indonesia Menggunakan Twitter. *Journal of Applied Informatics and Computing*, 3(1), 23–32. <https://doi.org/10.30871/jaic.v0i0.1246>
- Nisa, G. A. (2020). *Penerapan Semantik Web untuk Pencarian Data Akademik pada Institusi Perguruan Tinggi (Studi Kasus: Fasilkom-TI USU)* [Universitas Sumatera Utara]. <http://repositori.usu.ac.id/handle/123456789/26521>
- Nugroho, F. A., Septian, F., Pungkastyo, D. A., & Riyanto, J. (2021). Penerapan Algoritma Cosine Similarity untuk Deteksi Kesamaan Konten pada Sistem Informasi Penelitian dan Pengabdian Kepada Masyarakat. *Jurnal Informatika Universitas Pamulang*, 5(4), 529. <https://doi.org/10.32493/informatika.v5i4.7126>
- Nurhayati, A. N., Josi, A., & Hutagalung, N. A. (2018). Penjualan. *Jurnal Teknologi Dan Informasi*, 7(2), 13–23.
- Paramartha, D., & Wiguna, A. (2020). Perancangan Website Cerdas Pemilihan Kampus dengan Semantic Web dan Grabbing Data. *TIERS Information Technology*, 1(1), 12–23.
- Permana, I. S., & Sumaryana, Y. (2018). SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT KULIT DENGAN METODE FORWARD CHAINING. *JUMANTAKA : Jurnal Informatika*, 1(4), 143–155.
- Prasetya, E. D., & Et.all. (2019). Pengembangan Sistem Aplikasi Pencarian Dosen Pembimbing Skripsi dengan Teknologi Web Semantik (Studi Kasus: Fakultas Ilmu Komputer Universitas Brawijaya). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer E-ISSN*, 2548(9), 964X.
- Pratama, A. L., & Ropianto, M. (2020). Perancangan Database Dosen Berbasis Semantic Web. *Academia.Edu*.

https://www.academia.edu/download/63977890/Jurnal_Semantic_Web20200720-5979-bhzx3u.pdf

- Priandika, A. T., & Wantoro, A. (2017). Sistem Pendukung Keputusan Penerimaan Calon Siswa Baru pada SMK SMTI Bandar Lampung dengan Menggunakan Metode Simple Additive Weighting (SAW). *Explore: Jurnal Sistem Informasi Dan Telematika*, 8(2). <https://doi.org/10.36448/jsit.v8i2.955>
- Rahman, A. (2017). Online News Classification Using Multinomial Naive Bayes. *Itsmart*, 6(1), 32–38. <https://doi.org/10.1177/1096348015584441>
- Safwandi. (2021). Analisis Perancangan Sistem Informasi Sekolah Menengah Kejuruan 1 Gandapura Dengan Model Diagram Konteks Dan Data Flow Diagram. *Jurnal Teknologi Terapan and Sains*, 2(2), 1–5.
- Sahi, A. (2020). Aplikasi Test Potensi Akademik Seleksi Saringan Masuk Lp3I Berbasis Web Online Menggunakan Framework Codeigniter. *Tematik*, 7(1), 120–129. <https://doi.org/10.38204/tematik.v7i1.386>
- Saputra, R., Prasetyo, E., Rukmana, I., & Ropianto, M. (2020). Database Arsip Dokumen Web UIS. *Academia.Edu*. https://www.academia.edu/download/63934812/1710128262194_Sugie_Armei_Eka_Saputra.pdf
- Susyanto, T., & Mustofa, K. (2016). Pencarian Lowongan Pekerjaan Berbasis Agen Berdasarkan Profil Pencari Kerja dengan Pendekatan Semantic Web Service. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 10(1), 103. <https://doi.org/10.22146/ijccs.12734>