

Pengembangan *Knowledge Management System* dengan Teknik *Information Retrieval*

<http://dx.doi.org/10.28932/jutisi.v7i1.3316>

Riwayat Artikel

Received: 21 Januari 2021 | Final Revision: 24 Februari 2021 | Accepted: 4 Maret 2021

Try Atmaja Linggan Jaya^{#1}, Mewati Ayub^{✉*2}

[#] Program Studi Magister Ilmu Komputer, Universitas Kristen Maranatha

Jl. Surya Sumantri No.65, Bandung

¹atmaja94@gmail.com

²mewati.ayub@it.maranatha.edu

Abstract — Data sets can be used as information to solve problems or share knowledge with others. In the implementation of the new system, many input errors, or obscurity of the workflow of the program, are experienced repeatedly by the same person or people in the same department. Besides that, for new employees, it takes time to adapt and solve the problem. To solve it, a system is needed to record problems and their solutions, or share knowledge, both for old and new employees as the First Aid. Knowledge Management System is expected to solve the problems as a tool to find the information needed. The knowledge consists of collected data which contains errors, causes, solutions, business flow, user authorization, and other data. The data from a collection of tickets, personal messages or e-mail, and user knowledge will be entered into the database as the storage for the knowledge. In the input process, each word will be broken down based on the character "space", tokenizing, filtering, and Vector Space Model (VSM) and stored into the database. Users can search for information by entering keywords or sentences according to user needs. The input will be processed by tokenizing, filtering, and calculating the length using VSM. After getting the input length, the results will use the TF-IDF algorithm and cosine similarity to find the information. The system will display the results in a list form and show the details if the results of the list are selected.

Keywords— Cosine Similarity; Information Retrieval; Knowledge Management System; TF-IDF; Vector Space Model.

I. PENDAHULUAN

Pengimplementasian sistem baru membuat karyawan menyesuaikan cara kerja, dan penggunaan sistem itu sendiri. Dalam pergantian sistem, hal yang pasti berubah adalah proses bisnis perusahaan dari awal proses (*SO – Sales Order*) hingga pembayaran pajak (setelah *billing*). Karena tahap penyesuaian, terdapat banyak *error input* data, atau *error* karena alur proses yang salah yang berulang – ulang setiap harinya. Karena bagian produksi bekerja selama 24

jam dan kantor bekerja 8 jam per harinya, hambatan lain yang dialami yaitu *helpdesk* diluar jam kerja kantor atau hari libur (Sabtu dan Minggu) sehingga menghambat kerja bagian produksi.

Maka dari itu, untuk membantu mengatasi kebutuhan *user*, dibutuhkan pertolongan pertama untuk *user* sebagai pengganti bagian *helpdesk* dalam pemecahan masalah yang dialami *user*.

Aplikasi yang akan dibangun berupa *Knowledge Management System* sebagai pertolongan pertama untuk mengatasi kebutuhan *user*. *Knowledge management system* adalah informasi, pengetahuan, ide, keterampilan, dan pengalaman yang diperoleh, dipertukarkan dan dibagikan diantara orang – orang, organisasi dan lembaga [3]. Pengetahuan individu dibagikan keseluruhan organisasi dikarenakan pengetahuan tidak boleh hilang jika karyawan tersebut meninggalkan organisasi [3].

KMS dapat didefinisikan sebagai perangkat yang dapat digunakan untuk menyimpan, menyebarkan, berkolaborasi, mengidentifikasi sumber pengetahuan untuk mendukung pembuatan, penangkapan, berbagi, pengambilan dan penggunaan pengetahuan untuk meningkatkan akses ke sumber informasi dan pengetahuan oleh individu, organisasi, dan negara secara keseluruhan [3].

Selain hal tersebut, peranan *KMS* diharapkan dapat membantu *helpdesk* dalam menyelesaikan masalah baik secara *flow*, atau *problem solving*. Dengan dibuatnya suatu tempat untuk menyimpan semua informasi baik *flow*, *error message*, atau informasi lain, diharapkan dapat meringankan perusahaan dalam kebutuhan *helpdesk* sehingga orang yang ada dapat bekerja secara maksimal untuk kepentingan lain.

Dalam suatu organisasi *KMS* seperti suatu aset. Efektifitas *KMS* bergantung pada pemanfaatan orang, proses, dan system yang ada. Pengetahuan berfokus pada pembuatan pengetahuan untuk eksplorasi dan eksploitasi pengetahuan. Pengetahuan dapat dihasilkan salah satunya

melalui pemecahan masalah bersama yang dikumpulkan untuk dibagikan kepada banyak orang [3].

Diharapkan dengan adanya *Knowledge Management System*, kebutuhan *user* dalam mengatasi *error* dapat terpenuhi. Selain untuk mengatasi *error*, *Knowledge Management System* juga digunakan untuk mencatat *error* apa saja yang pernah terjadi, sehingga *user* dapat mengatasi *error* tersebut jika terjadi kembali di kemudian hari.

II. TINJAUAN PUSTAKA

A. Knowledge Management System

Pengetahuan terbagi menjadi dua bagian, yaitu pengetahuan eksplisit dan implisit. Pengetahuan eksplisit merupakan pengetahuan yang telah ada baik dalam bentuk teks, diagram, spesifikasi produk, dsb yang dapat ditemukan dalam laporan, dokumen, dan manual yang dikumpulkan dan disimpan sebagai dasar pengetahuan.

Pengetahuan implisit merupakan pengetahuan yang tidak berwujud dibandingkan dengan pengetahuan eksplisit. Pengetahuan tacit adalah dimensi dari pengetahuan implisit yang meliputi hubungan, norma, dan nilai – nilai yang tidak dapat diartikulasikan dan sulit untuk dirinci, disalin, atau didistribusikan [4].

Pengetahuan merupakan aset yang berharga untuk meningkatkan suatu organisasi atau perusahaan. Pengetahuan dibedakan menjadi 2, yaitu:

1) *Tacit*: Pengetahuan yang tidak mudah untuk didokumentasikan, bersifat sangat berharga atau membutuhkan penerapan yang tepat, yang didapatkan berdasarkan pengalaman dan bersifat personal.

2) *Explicit*: Pengetahuan yang bersifat sistematis dan dapat didokumentasikan. Berasal dari penerapan tacit dan ditulis untuk dapat dipelajari atau diterapkan oleh orang lain [5].

B. Information Retrieval

Information retrieval digunakan sebagai pencarian informasi sesuai yang dibutuhkan oleh pengguna. Berikut tahapan dalam *information retrieval*:

1) *Tokenizing*: Proses *tokenizing* adalah proses untuk memecah suatu kalimat menjadi kata berdasarkan suatu karakter tertentu. Proses *tokenizing* sangat penting peranannya karena sebagai dasar penggunaan algoritma *TF-IDF* dimana dalam proses *TF* dilakukan pengelompokan dan perhitungan kata. Karakter biasa dijadikan pemisah yaitu spasi, titik (.), koma (,), *semicolon* (;), dan titik dua (:). [6].

2) *Filtering*: Merupakan proses pembuangan kata – kata yang tidak memiliki arti. Biasanya kata – kata tersebut merupakan kata sambung dalam kalimat (dan, atau, yang). Kata yang tidak termasuk dengan kata tersebut akan dilanjutkan pada proses selanjutnya. Dari hasil *tokenizing*,

akan dibuang kata – kata yang tidak memiliki arti atau kata penghubung [7].

3) *Vector Space Model (VSM)*: *VSM* merupakan model aljabar yang digunakan untuk pengambilan informasi melalui kata – kata yang dimiliki setiap kalimat / dokumen [8]. *VSM* pertama kali ditemukan pada tahun 1979 oleh Salton et al. Terdapat 3 kelas pada *VSM*, yaitu didasarkan oleh *term-document*, *word-context*, *pair-pattern matrix*. *VSM term-document* merupakan yang paling banyak digunakan untuk mencari persamaan antar dokumen [9]. Setelah didapat nilai *TF-IDF* tersebut, dilakukan perhitungan similarity measure dari *query* dengan setiap kalimat / dokumen yang ada dengan rumus.

$$|q| = \sqrt{\sum_{j=1}^t (W_{iq})^2} \quad (1)$$

Nilai $|q|$ yang merupakan jarak *query*, dan W_{iq} adalah bobot *query* dokumen ke-I, maka jarak *query* dihitung untuk mendapatkan jarak *query* dari bobot *query* dokumen yang terambil. Setelah mendapatkan jarak *query*, dilakukan perhitungan jarak dokumen dengan menggunakan

$$|d_j| = \sqrt{\sum_{i=1}^t (W_{ij})^2} \quad (2)$$

Dengan didapatkannya jarak dokumen, dilakukan perhitungan *similarity query* dokumen (*inner product*) dengan menggunakan

$$Sim(q, d_j) = \sum_{i=1}^t W_{iq} \cdot W_{ij} \quad (3)$$

Setelah mendapatkan bobot dari dokumen, dilakukan perhitungan *cosine similarity* menggunakan persamaan

$$Sim(q, d_j) = \frac{q \cdot d_j}{|q| \cdot |d_j|} = \frac{\sum_{i=1}^t W_{iq} \cdot W_{ij}}{\sqrt{\sum_{j=1}^t (W_{iq})^2} \cdot \sqrt{\sum_{i=1}^t (W_{ij})^2}} \quad (4)$$

Perhitungan similaritas akan menghasilkan bobot dokumen dari 0 hingga 1 atau menghasilkan bobot lebih besar dibandingkan dengan nilai yang dihasilkan dari *inner product* [10].

4) *TF-IDF*: *TF-IDF* merupakan algoritma untuk memeriksa relevansi kata – kata. *TF-IDF* berasal dari 2 kata, yaitu *TF* (*Term Frequency*) untuk mengetahui berapa banyak suatu syarat muncul dalam suatu dokumen [11].

$$TF = \text{jumlah suatu kata yang muncul}$$

IDF (*Inverse Document Frequency*) pemberian bobot untuk setiap kata yang telah dihitung pada proses *TF*. Pemberian bobot dilakukan dengan cara memberi lebih rendah untuk kata-kata yang sering dan memberikan bobot lebih besar untuk kata-kata yang jarang [12].

$$IDF = \log_e TF \quad (5)$$

TF-IDF merupakan hasil perkalian dari *TF* dan *IDF* dengan rumus

$$TF - IDF = TF * IDF \quad (6)$$

Hasil dari *tokenizing* dan *filtering* akan dikelompokkan dan dihitung jumlah perkata pada proses *TF* dengan rumus

$$Wdt = tfdt * IDFt \quad (7)$$

Keterangan:

- d : dokumen ke- d
- t : kata ke- t dari kata kunci
- W : bobot dokumen ke- d terhadap kata ke- t
- tf : banyaknya kata yang dicari pada sebuah dokumen
- IDF : *inversed document frequency (IDF)* didapatkan dari DF : $\log 2 (D/df)$
- D : total dokumen
- DF : banyak dokumen yang mengandung kata yang dicari

Setelah setiap dokumen diketahui bobotnya, dilakukan proses pengurutan dimana semakin besar nilai bobot, semakin besar tingkat kemiripan suatu dokumen terhadap kata kunci [13].

C. Time Series ARIMA

ARIMA yang merupakan suatu metode untuk peramalan berdasarkan data berupa deret waktu. Metode *ARIMA Box Jenkins* diperoleh dari $AR(p)$ dan $MA(q)$ sehingga dinamakan menjadi $ARIMA(p,q)$. Kebutuhan data *time series* umumnya terdapat data tidak *stationer* karena *mean* dan *varians*. Untuk mengatasi masalah pada *mean* dilakukan *differencing*, sedangkan untuk *varians* dilakukan transformasi *Box-Cox*. Model *time series* yang telah diolah tersebut dinamakan *Autoregressive Integrated Moving Average (ARIMA)*.

Model $AR(p)$ dan $MA(q)$ dapat ditulis sebagai berikut:

$$Y_t = \varphi_1 Y_{t-1} + \dots + \varphi_p Y_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (8)$$

Dan model $ARIMA(p,q)$ dapat ditulis menjadi :

$$\varphi_p(B)Y_t = \theta_q(B)a_t \quad (9)$$

Dengan

$$Y = Y_t - \mu \quad (10)$$

$$\varphi_p(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \dots - \varphi_p B^p \quad (11)$$

$$\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q \quad (12)$$

Beberapa konsep yang terdapat pada ARIMA:

1) *Konsep Stationer*: merupakan konsep dimana dalam proses pengolahan data yang terdapat perubahan kecenderungan dalam *mean* atau *varians*. Konsep tersebut dapat dilihat berdasarkan grafik yang sejajar dengan sumbu waktu.

2) *Konsep Differencing*: Konsep yang berfungsi untuk mengatasi masalah dalam pengolahan data khususnya proses yang tidak *stationer* dalam *mean* dengan mengurangi suatu pengamatan dengan pengamatan sebelumnya.

3) *Konsep Transformasi Cox-Box*: Konsep yang berfungsi untuk mengatasi masalah dalam pengolahan data khususnya proses yang tidak *stationer* dalam *varians* [15].

D. Penelitian Terkait

Literatur yang digunakan dalam penelitian ini salah satunya diambil dari penelitian Adi Ryansyah dan Sri Andayani [10] untuk implementasi *information retrieval* dalam mengukur kesamaan dokumen. Dalam penelitian

tersebut, telah dilakukan pengukuran dengan algoritma *TF-IDF* untuk pengelompokan dokumen yang sama.

Pada penelitian tersebut, terdapat suatu masalah berupa dokumen yang memiliki banyak versi sehingga dalam penggunaan dokumen tersebut tidak diketahui dokumen mana yang terbaru. Terjadinya banyak versi tersebut dikarenakan penamaan versi yang berbeda atau melakukan penyalinan dokumen.

Untuk mengurangi masalah tersebut, perlu dilakukan pengecekan tiap dokumen serta mengelompokkan dokumen yang sama agar tidak terlalu banyak versi dokumen yang ada. Jika melakukan hal tersebut, tentu akan membuang waktu dalam pengecekan dokumen dengan jumlah yang banyak. Dengan menggunakan algoritma *TF-IDF*, dapat dilakukan perbandingan dokumen apakah suatu dokumen mirip dengan dokumen lain, dan juga dapat dilakukan pengelompokan dokumen jika dokumen tersebut dirasa mirip.

Dengan melakukan perhitungan kata dan pemberian bobot untuk setiap dokumen serta menghitung persamaan sudut setiap dokumen menggunakan *cosine similarity*, maka akan diketahui kelompok dokumen tersebut [15].

Literatur lain yang digunakan yaitu hasil penelitian A. A. Maarif [13], dalam penelitian tersebut, algoritma *TF-IDF* digunakan sebagai metode pencarian informasi karya ilmiah. Kendala pencarian informasi merupakan inti dari penelitian tersebut. Bagaimana untuk mencari informasi dari data yang banyak khususnya karya ilmiah. Bagi manusia tentu mudah untuk pencarian informasi yang diinginkan dengan membandingkan beberapa informasi lain, tetapi masalah akan timbul jika jumlah pembanding sangat banyak. Tentu kemampuan manusia akan terbatas dengan jumlah pembanding informasi yang sangat banyak dan akan memakan waktu yang lama dalam pencariannya.

Algoritma *TF-IDF* digunakan untuk pencarian informasi dari banyak data dengan membandingkan *keyword* yang dicari dengan judul informasi yang tersedia.

Tahap pertama yang akan dilakukan yaitu pemecahan kata berdasarkan karakter tertentu. Lalu akan dilakukan *filtering* dari kata – kata tersebut sebagai proses pengelompokan kata. Setelah itu dilakukan perhitungan *TF-IDF* untuk mengetahui bobot setiap dokumen sebagai pembanding kemiripan [7].

Literatur lain yang dijadikan acuan adalah hasil penelitian B. Herwijayanti, D. Eka Ratnawati, dan L. Muflikhah [18], yang menggunakan algoritma *TF-IDF* untuk pengelompokan berita dengan studi kasus *kompas.com*.

Latar belakang dari penelitian tersebut dikarenakan berita yang dimasukan dapat dilakukan oleh siapa saja dan jumlahnya banyak, serta *human error* menyebabkan suatu berita tidak sesuai dengan kategorinya, misalkan berita yang seharusnya berkategori *news* dimasukan pada kategori *sport*.

Dengan menggunakan algoritma *TF-IDF*, dilakukan perbandingan judul berita yang akan dimasukan dengan judul berita yang sudah ada sehingga dapat dilakukan

pengkategorian berita untuk meminimalisir kesalahan pengisian kategori berita [17].

Dari beberapa kajian tersebut, dapat dijadikan sebagai acuan dalam penelitian karena terdapat kemiripan tujuan penelitian yaitu perbandingan suatu kalimat dengan kalimat lain, dan pengolahan datanya berupa pencarian informasi berdasarkan suatu kata kunci. Dengan anggapan suatu informasi merupakan *error message* dan suatu dokumen adalah *problem solving*, maka dapat dilakukan perhitungan bobot untuk dilakukan pencarian informasi dalam *Knowledge Management System* menggunakan algoritma *TF-IDF*.

III. METODOLOGI PENELITIAN

Bagian ini menjelaskan secara rinci dan lengkap tentang metodologi / desain penelitian berdasarkan hipotesis – hipotesis pada penelitian.

A. Hipotesis

Dengan dilakukan implementasi *Knowledge Management System*, diperkirakan beban *helpdesk* akan berkurang sebanyak 40%, atau dapat dikatakan rata – rata tiket dengan kategori SAP yang masuk per-harinya kurang lebih sebanyak 8 tiket dalam *problem solving* atau pencarian informasi lain, seseorang tidak perlu menunggu respon dari departemen IT dan diperkirakan akan menemukannya dalam waktu kurang dari 3 menit.

B. Analisis Penelitian

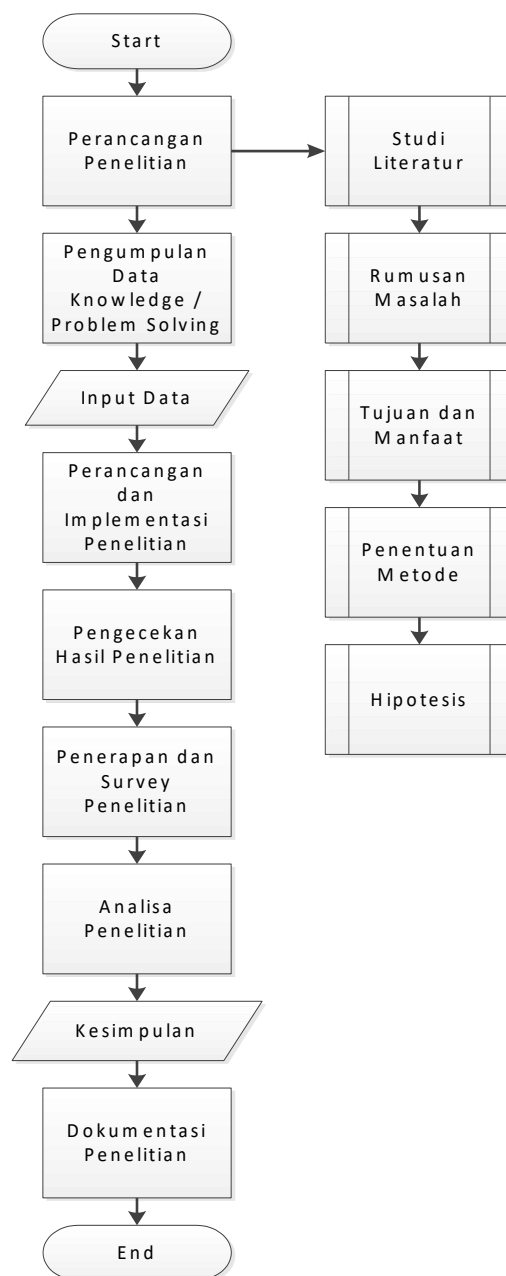
Dalam analisis penelitian, akan dijabarkan mengenai tahapan yang dilakukan dalam penelitian dari perancangan hingga dokumentasi penelitian. Pada penelitian ini, akan dibuatkan *Knowledge Management System (KMS)* yang digunakan oleh kurang lebih 1000 *user* yang terdiri dari *user* biasa (*Non IT*) dan *user admin (IT)* yang telah digunakan sejak bulan Juni 2020.

Dalam analisis penelitian, dilakukan rancangan penelitian terlebih dahulu untuk menentukan studi literatur, rumusan masalah, tujuan dan manfaat, penentuan metode, serta hipotesis penelitian. Setelah pembuatan rancangan, dilakukan pengumpulan data berupa pengetahuan / *problem solving* yang akan dibagikan pada *KMS*. Lalu dilakukan perancangan dan implementasi penelitian sesuai dengan rancangan yang dibuat dan setelahnya akan dilakukan pengecekan apakah seluruh proses sudah berjalan dengan baik atau belum, dan sudah memenuhi kebutuhan atau belum.

Dalam perancangan dan implementasi, dilakukan tahapan pembuatan *KMS* dimulai dari analisis kebutuhan, perancangan *UI*, perancangan *database*, pengumpulan dan *input data*. Pengumpulan data dilakukan dengan menyebarkan *template* untuk semua karyawan (berisikan judul / *error*, deskripsi / *reason*, dan isi pengetahuan / *solving*). Setelah data terkumpul, maka dilanjutkan dengan perancangan sistem (*Tokenizing, Filtering, VSM, TF-IDF*),

pengecekan sistem sebelum digunakan, dan penerapan sistem.

Setelah semua sesuai dengan yang diharapkan, dan hasil penelitian diterapkan kepada *user*, akan dilakukan survei terhadap kepuasan *user* dalam menggunakan hasil penelitian. Hasil dari survei tersebut akan dianalisis untuk menghasilkan kesimpulan penelitian yang dapat berguna untuk hipotesis yang telah ditentukan sebelumnya dan dilakukan dokumentasi terhadap penelitian.



Gambar 1. Desain penelitian

C. Perancangan dan Implementasi Penelitian

Pada perancangan dan implementasi penelitian, akan dijelaskan mengenai tahapan dalam perancangan sistem dan implementasi penelitian yang dilakukan pada sebuah perusahaan *textile* di Bandung.

1) *VSM*: Pada implementasi penelitian, *VSM* dilakukan pada saat kontribusi pengetahuan. Proses *VSM* melewati proses *tokenizing*, *filtering*, barulah *VSM*. Terdapat kontribusi pengetahuan berupa “Otorisasi *Plan Maintenance*”. Maka kalimat tersebut akan dipecah berdasarkan karakter “Spasi” dan menghasilkan kata “otorisasi”, “*plan*”, dan “*maintenance*”. Setelah itu akan dilakukan pengecekan

apakah kata tersebut sudah pernah ada atau belum, jika belum maka akan ditambahkan kata kunci baru pada “*tbWords*” dan dihitung *length*-nya, dan jika sudah ada, maka akan ditambahkan jumlahnya dan dihitung *length*-nya.

2) *Tokenizing*: Terdapat 3 informasi berupa:

Kata 1: Untuk menambahkan otorisasi *equipment list*

Kata 2: Status *Equipment DCTV*

Kata 3: Otorisasi *Plant Maintenance*

Kata kunci yang akan dicari adalah “Otorisasi *equipment list*”. Maka, proses *tokenizing* berdasarkan karakter spasi akan ditunjukkan pada tabel I.

TABEL I
HASIL *TOKENIZING*

<i>Keywords</i>	otorisasi	equipment	list		
Kata 1	Untuk	menambahkan	otorisasi	equipment	list
Kata 2	Status	Equipment	DCTV		
Kata 3	Otorisasi	Plant	Maintenance		

3) *Filtering*: Merupakan proses pembuangan kata – kata yang tidak memiliki arti. Biasanya kata – kata tersebut merupakan kata sambung dalam kalimat (dan, atau, yang, dsb). Kata yang tidak termasuk dengan kata tersebut akan dilanjutkan pada proses selanjutnya.

Dari hasil *tokenizing*, akan dibuang kata – kata yang tidak memiliki arti atau kata penghubung. Maka proses *filtering* seperti tabel II.

TABEL II
HASIL *FILTERING*

<i>Keywords</i>	otorisasi	equipment	list		
Kata 1	Untuk	menambahkan	otorisasi	equipment	list
Kata 2	Status	Equipment	DCTV		
Kata 3	Otorisasi	Plant	Maintenance		

Kata “Untuk” yang diberi warna merah pada Kata 1 akan dibuang, sehingga Kata 1 hanya memiliki kata “Menambahkan”, “Otorisasi”, “Equipment”, “List” saja.

4) *TF-IDF*: Berikut merupakan proses pengolahan data dengan menggunakan algoritma *TF-IDF* dimana pada tabel III akan dilakukan perhitungan jumlah kata dari setiap kata yang muncul (*TF*). Sebelum melakukan perhitungan jumlah kata, dilakukan proses persamaan format kata (*Uppercase* atau *Lowercase*).

TABEL III
PROSES PERHITUNGAN KATA

	otorisasi	equipment	list	menambahkan	status	dctv	plant	maintenance
Kata 1	1	1	1	1	0	0	0	0
Kata 2	0	1	0	0	1	1	0	0
Kata 3	1	0	0	0	0	0	1	1

Pada tabel IV dilakukan perhitungan IDF menggunakan rumus (5), dan didapatkan hasil seperti berikut.

TABEL IV
PROSES PERHITUNGAN IDF

otorisasi	2	$\log_2 \frac{3}{2}$	0.585
equipment	2	$\log_2 \frac{3}{2}$	0.585
list	1	$\log_2 \frac{3}{1}$	1.585
menambahkan	1	$\log_2 \frac{3}{1}$	1.585
status	1	$\log_2 \frac{3}{1}$	1.585
dctv	1	$\log_2 \frac{3}{1}$	1.585
plant	1	$\log_2 \frac{3}{1}$	1.585
maintenance	1	$\log_2 \frac{3}{1}$	1.585

Pada tabel V dilakukan perhitungan menggunakan rumus TF-IDF (6), maka didapatkan hasil seperti berikut.

TABEL V
PROSES PERHITUNGAN TF-IDF

	otorisasi	equipment	list	menambahkan	status	dctv	plant	maintenance
Kata 1	0.585	0.585	1.585	1.585	0	0	0	0
Kata 2	0	0.585	0	0	1.585	1.585	0	0
Kata 3	0.585	0	0	0	0	0	1.585	1.585

Pada tabel VI dilakukan perhitungan *weight keyword d* dengan menggunakan rumus (7).

TABEL VI
PROSES PERHITUNGAN TF-IDF UNTUK KEYWORD

	otorisasi	equipment	list	menambahkan	status	dctv	plant	maintenance
Keyword	0.145	0.145	0.396	0	0	0	0	0

Pada tabel VII dilakukan perhitungan Length setiap kalimat menggunakan VSM dengan rumus (2).

TABEL VII
PROSES PERHITUNGAN LENGTH UNTUK SETIAP KALIMAT

Length Kata 1	2.389
Length Kata 2	2.317
Length Kata 3	2.317
Length Keyword	0.447

Pada tabel VIII dilakukan perhitungan *CosSin Similarity* dengan menggunakan rumus (4).

TABEL VIII
PROSES PERHITUNGAN LENGTH UNTUK SETIAP KALIMAT

Similarity	
Cosine Kata 1	0.748
Cosine Kata 2	0.083
Cosine Kata 3	0.083

Dari perhitungan tabel I hingga tabel VIII, maka didapatkan bahwa kata yang mendekati dengan *keyword* “otorisasi *equipment list*” adalah Kata 1 “untuk menambahkan otorisasi *equipment list*” dan algoritma *TF-IDF* dapat digunakan untuk pencarian informasi atau *error message*.

5) *Pengembangan Sistem*: Dalam pembuatan sistem, dilakukan beberapa tahapan hingga sistem digunakan oleh

user. Setelah implementasi sistem, dilakukan pengembangan terus menerus yang disesuaikan dengan kebutuhan sistem dan user. Berikut merupakan tahapan dan pengembangan yang telah dilakukan yang dapat dilihat pada tabel IX.

TABEL IX
PROSES PERHITUNGAN *LENGTH* UNTUK SETIAP KALIMAT

Bulan	Pengembangan	Keterangan
April 2020	Analisis Kebutuhan User	Untuk mengetahui kebutuhan <i>user</i> terhadap solusi yang akan ditawarkan.
April 2020	Perancangan Sistem	Dilakukan rancangan sistem sebagai analisis sebelum pembuatan sistem yang disesuaikan dengan kebutuhan user.
Mei 2020	Pembuatan <i>database</i>	Pembuatan <i>database</i> sesuai dengan analisis sebelumnya dan kebutuhan sistem
Mei 2020	Implementasi Sistem	Pembuatan sistem sesuai dengan perancangan yang telah dibuat.
Juni 2020	Uji coba sistem	Dilakukan uji coba sistem yang telah dibuat, apakah sudah sesuai atau belum.
Juni 2020	Pengecekan Sistem	Dilakukan pengecekan sistem sebelum digunakan oleh <i>user</i> .
Juli 2020	Implementasi Sistem (<i>Go Live</i>)	Dilakukan implementasi sistem kepada <i>user</i> .
Oktober 2020	Perubahan <i>code</i> PHP <i>native</i> menjadi PDO	Agar <i>code</i> lebih rapih dan mudah untuk digunakan.
November 2020	Penambahan <i>table</i> <i>tbLogSearch</i> pada <i>database</i>	Untuk menyimpan <i>historical</i> pencarian <i>user</i> .
Desember 2020	Penambahan <i>table</i> <i>tbWords</i> pada <i>database</i>	Untuk menyimpan hasil perhitungan <i>VSM</i> pada saat <i>input</i> .
Desember 2020	Perubahan <i>code</i> <i>Contribution</i> dan <i>Search Knowledge</i>	Perubahan <i>code</i> dimana proses <i>tokenizing</i> , <i>filtering</i> , <i>TF</i> dan perhitungan <i>length</i> data yang sudah ada dilakukan pada proses <i>input</i> sebagai penerapan <i>VSM</i> dan akan dimasukkan kedalam <i>tbWords</i> . untuk proses <i>search knowledge</i> dilakukan proses <i>tokenizing</i> , <i>filtering</i> , <i>TF</i> dan perhitungan <i>length</i> untuk <i>input</i> -an saja yang akan digabungkan dengan data dari <i>table</i>

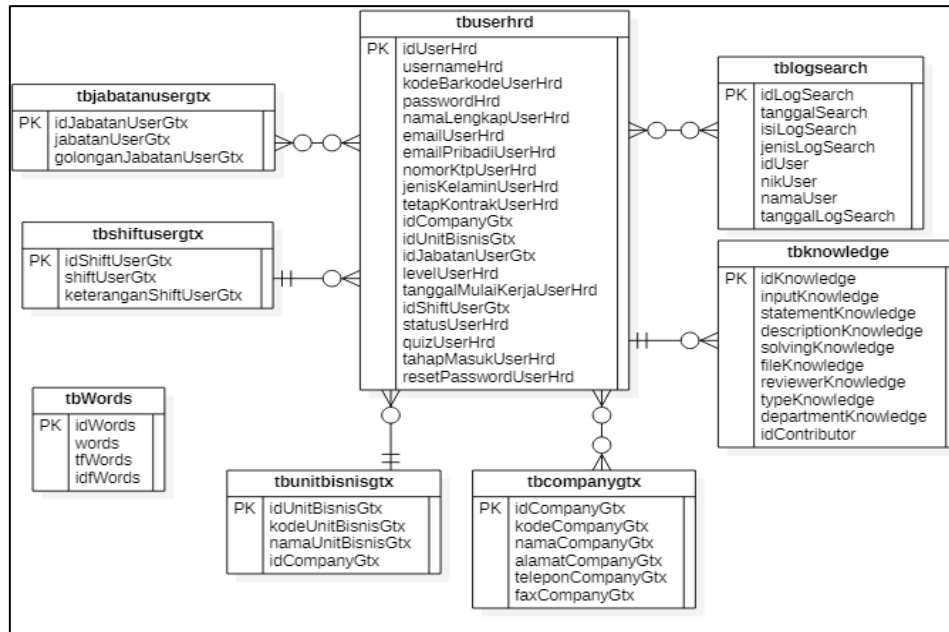
Bulan	Pengembangan	Keterangan
		<i>tbWords</i> .

IV. PERANCANGAN DAN IMPLEMENTASI

Dalam perancangan dan implementasi penelitian, akan dibahas mengenai tahapan – tahapan yang dilakukan dalam aplikasi yang dibuat dari proses *tokenizing* hingga pemberian kandidat solusi. Dalam proses temu balik informasi, tidak dilakukan proses *index* secara berkala saat ada penambahan pengetahuan dan dilakukan pada saat pencarian solusi, dikarenakan data yang dijadikan acuan pencarian cenderung pendek, sehingga dirasa tidak diperlukan untuk proses *index* tersebut. Oleh karena itu, pada perancangan penelitian, *TF-IDF* digunakan untuk proses *index* setiap kata yang akan menghasilkan kata, jumlah kata, dan *length* setiap kata pada proses kontribusi, dan *VSM* digunakan untuk menentukan *similarity* untuk setiap kalimat pada proses pencarian yang akan mengambil data hasil *TF-IDF* per kata, dan menghitung *cosine similarity* untuk menentukan solusi yang akan ditampilkan

A. Entity Relationship Diagram (ERD)

Gambar 2 merupakan rancangan *database* yang akan digunakan. Terdapat 8 *table* yang berisikan 3 *table* utama pada sistem *KMS* dan 5 *table* tambahan yang diambil dari sistem lain. 3 *table* utama tersebut yaitu “*tbLogSearch*” yang berisikan riwayat pencarian *user*, “*tbKnowledge*” yang berisikan kumpulan informasi / pengetahuan, dan “*tbWords*” yang berisikan kata yang telah dipecah, dan dihitung jumlah serta *length*-nya yang digunakan sebagai konsep dari *VSM*.



Gambar 2. Entity Relationship Diagram (ERD)

Tabel X merupakan penjelasan *detail* mengenai tabel yang dipakai untuk kebutuhan KMS.

TABEL X
TABEL KETERANGAN DETAIL ERD

Tabel	Deskripsi / Struktur	Tujuan
tbUserHrd	Merupakan tabel yang berisikan informasi <i>user</i>	Digunakan untuk login dan melihat informasi <i>user</i>
tbCompanyGtx	Merupakan tabel yang berisikan daftar perusahaan	Digunakan sebagai <i>foreign key</i> pada tabel tbUserHrd untuk menunjukan <i>user</i> tersebut berada pada perusahaan apa
tbUnitBisnisGtx	Merupakan tabel yang berisikan daftar departemen	Digunakan sebagai <i>foreign key</i> pada tabel tbUserHrd untuk menunjukan <i>user</i> tersebut berada pada departemen apa
tbJabatanUserGtx	Merupakan tabel yang berisikan daftar jabatan	Digunakan sebagai <i>foreign key</i> pada tabel tbUserHrd untuk menunjukan <i>user</i> tersebut memiliki

Tabel	Deskripsi / Struktur	Tujuan
tbShiftUserGtx	Merupakan tabel yang berisikan daftar <i>shift</i>	Digunakan sebagai <i>foreign key</i> pada tabel tbUserHrd untuk menunjukan <i>user</i> tersebut berada pada <i>shift</i> apa
tbLogSearch	Merupakan tabel yang berisikan <i>historical</i> pencarian <i>user</i> pengguna KMS	Digunakan sebagai analisis pencarian <i>user</i>

B. System Relation



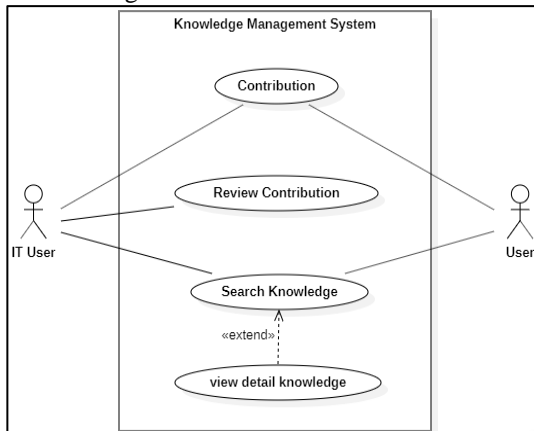
Gambar 3. Entity Relationship Diagram (ERD)

Pada gambar 3 merupakan relasi yang terdapat pada sistem KMS. Seperti yang sudah dijelaskan pada gambar 2, data user diambil dari sistem lain sehingga pada KMS tidak

terdapat proses untuk mengelola user dan hanya digunakan untuk pencarian pengetahuan / informasi.

C. Usecase Diagram

Pada gambar 4 akan ditunjukkan *usecase diagram* yang telah dibuat sebagai analisis kebutuhan sistem.



Gambar 4. Usecase diagram

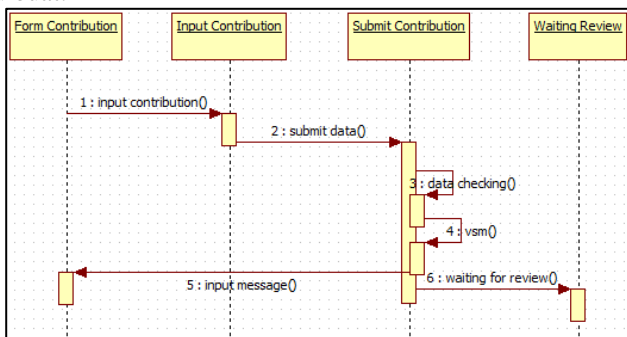
Gambar 4 merupakan rancangan *usecase diagram* untuk KMS. Untuk level *User* biasa, dapat melakukan *contribution*, *search knowledge*, dan *view detail knowledge*. Untuk level *User IT*, dapat melakukan *contribution*, *review contribution*, *search knowledge*, dan *view detail knowledge*.

Penjelasan singkat mengenai *usecase diagram*:

- *Contribution* digunakan oleh semua pengguna yang akan membagikan informasi atau pengetahuannya.
- *Review Contribution* hanya dapat digunakan oleh *IT User* sebagai *approval knowledge* yang telah dibagikan oleh semua *user* sebagai *filter* agar KMS berisikan informasi yang *valid* dan dapat berguna untuk semua orang.
- *Search Knowledge* digunakan oleh semua *user* untuk mencari *knowledge* yang dibutuhkan.
- *View Detail Knowledge* digunakan untuk melihat *detail* informasi dari setiap *knowledge*.

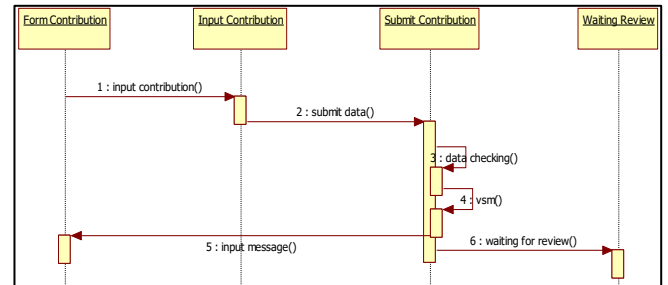
D. Sequence Diagram

Pada bagian ini akan dibahas mengenai sebagian *sequence diagram* berdasarkan *usecase diagram* yang telah dibuat.



Gambar 5. Sequence diagram proses "Contribution"

Pada gambar 5 merupakan rancangan *sequence diagram* untuk proses "Contribution". Proses berawal dari *form contribution*, setelah *user* memasukkan *contribution*, sistem akan melakukan pengecekan *input* apakah sudah sesuai atau belum. Jika *input* sudah sesuai, maka sistem akan menjalankan proses *VSM* dan *user* akan menunggu *review* yang akan dilakukan oleh *IT*. Jika *input* salah, maka akan muncul pesan *error*.

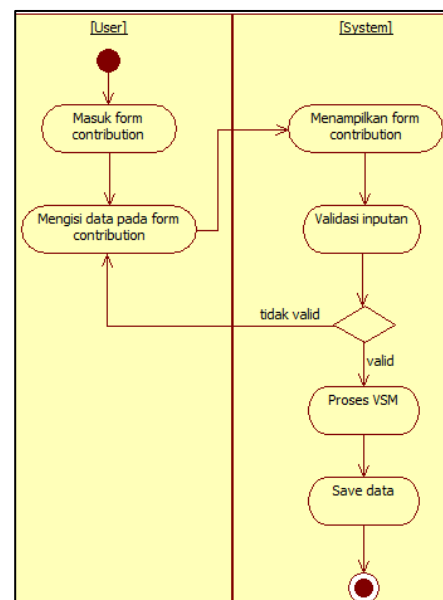


Gambar 6. Sequence diagram proses "Search Knowledge"

Gambar 6 merupakan rancangan *sequence diagram* untuk proses "Search Knowledge". Proses berawal dari halaman awal menu, *user IT* atau *user* biasa melakukan pencarian yang diinginkan. Sistem akan melakukan proses *case folding*, *tokenizing*, *filtering*, dan *TF-IDF*. Setelah semua proses sudah dilakukan, sistem akan menampilkan hasil yang dicari *user*.

E. Activity Diagram

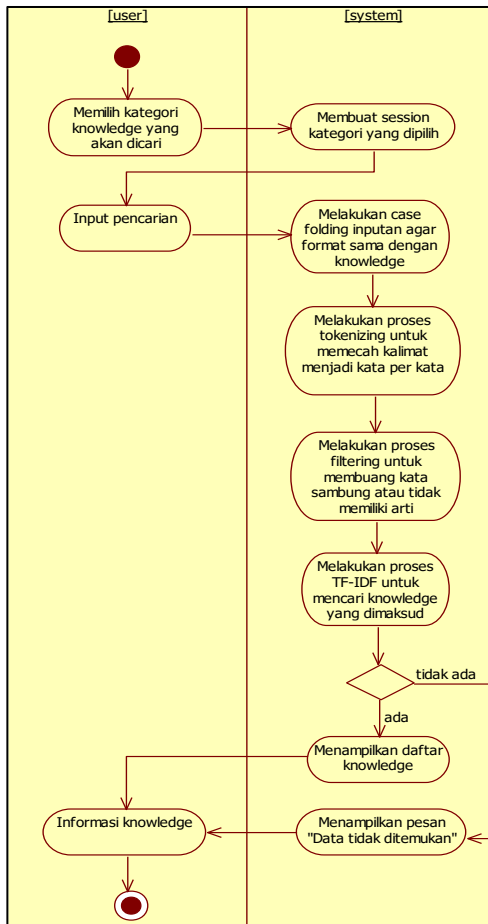
Pada bagian ini akan dibahas mengenai sebagian *activity diagram* berdasarkan *sequence diagram* yang telah dibuat.



Gambar 7. Activity diagram proses "Contribute"

Gambar 7 merupakan rancangan *activity diagram* untuk proses *contribution*. Proses berawal dari masuk kedalam *form contribution*, lalu sistem akan menampilkan *form*

contribution. Setelah itu user akan mengisi data pada form contribution. Setelah user “Submit”, sistem akan melakukan pengecekan input, jika tidak valid, maka akan kembali pada form contribution, jika valid input akan disimpan pada database.



Gambar 8. Activity diagram proses “Search Knowledge”

Gambar 8 merupakan rancangan activity diagram untuk proses “Search Knowledge”. User akan memilih knowledge apa yang akan dicari dan sistem akan membuat “session” untuk menyimpan pilihan user. Setelah user input pencarian, sistem akan melakukan case folding, tokenizing, filtering, dan TF-IDF untuk mencari informasi yang mendekati kata kunci yang dicari. Jika informasi ditemukan, maka hasil akan ditampilkan berupa list, jika tidak ada hasil yang mendekati, maka akan menampilkan pesan “Data tidak ditemukan”.

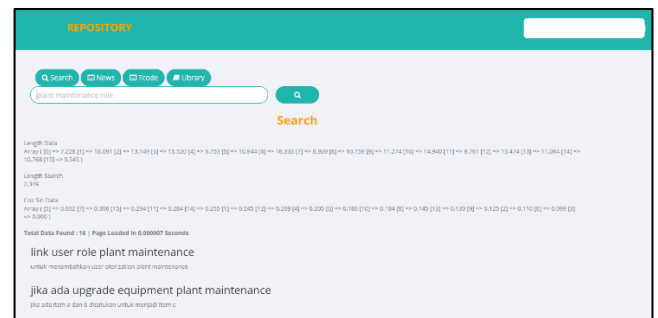
F. Implementasi Sistem

Setelah melewati beberapa proses analisis dan pembuatan sistem, serta dilakukan beberapa pengembangan untuk memenuhi kebutuhan user dan kebutuhan sistem, berikut dipaparkan hasil dari implementasi sistem.



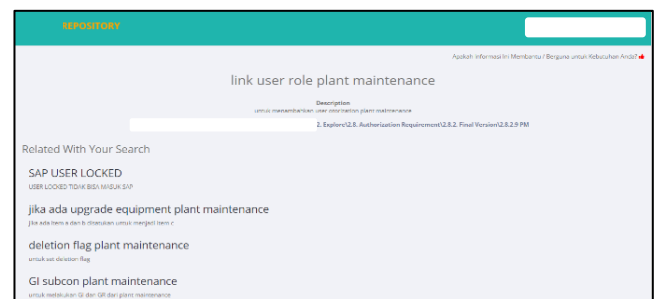
Gambar 9. Tampilan Halaman Awal KMS

Gambar 9 merupakan tampilan dari halaman awal sistem. Terdapat beberapa menu diantaranya menu search, news, TCode, dan library. Berdasarkan beberapa menu tersebut, yang menjadi fokus utama yaitu menu search, dan TCode. Menu search digunakan untuk pencarian pengetahuan, dan menu TCode digunakan untuk pencarian TCode SAP Standard yang diambil dari pihak luar. Hasil pencarian akan ditampilkan setelah user mengisikan pengetahuan yang akan dicari pada field search, lalu menekan tombol search.



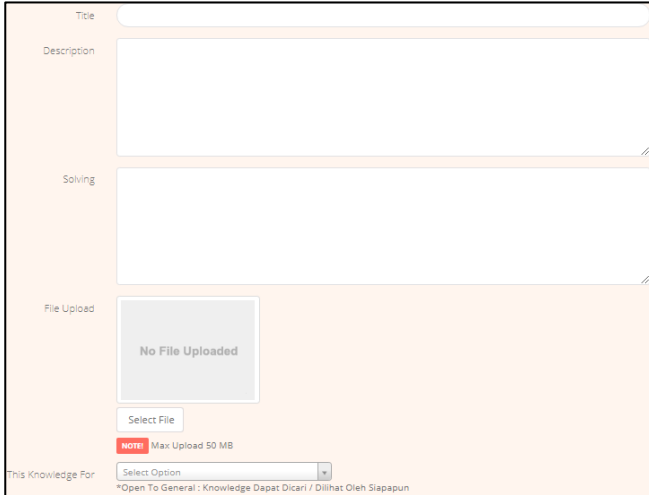
Gambar 10. Tampilan Hasil Pencarian KMS

Gambar 10 merupakan tampilan dari hasil pencarian pada menu search. Hasil yang dikeluarkan berupa daftar yang dapat menampilkan detail informasi jika menekan salah satu dari daftar yang ada. Pada bagian tengah gambar 10, dimunculkan sementara hasil perhitungan Cosine Similarity untuk membuktikan skor yang didapat dalam perhitungan. Pada array index merupakan nilai dari “idKnowledge” dan array value merupakan skor hasil perhitungan cosine similarity. Sebagai contoh untuk pencarian “plant maintenance role”, hasil yang paling mendekati ada pada index 5 dengan nilai Cosine 0,652.



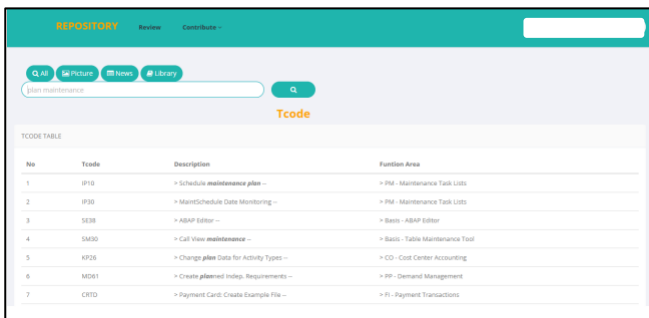
Gambar 11. Tampilan Detail Hasil Pencarian

Gambar 11 merupakan tampilan dari detail hasil pencarian pada menu *search*. Detail hasil pencarian berisi informasi mengenai pesan *error*, judul pengetahuan, dsb; alasan terjadinya *error*, deskripsi pengetahuan, dsb; dan cara mengatasi *error*, isi atau penjabaran pengetahuan, dsb; daftar pencarian yang berhubungan; dan respon pencarian.



Gambar 12. Tampilan *Contribution*

Gambar 12 merupakan tampilan dari menu *Contribution*. Terdapat *field* untuk menambah pengetahuan berupa “*Title*” sebagai pesan *error*, judul pengetahuan, dsb; “*Description*” sebagai alasan terjadinya *error*, deskripsi pengetahuan, dsb; “*Solving*” sebagai cara mengatasi *error*, isi atau penjabaran pengetahuan, dsb; “*Gambar*” sebagai capture pesan *error*, cara mengatasi *error*, dsb; “*This Knowledge For*” sebagai pilihan untuk pencarian pengetahuan yang berupa “*Open To General*” jika pengetahuan dapat ditemukan oleh semua *user*, dan “*Only To Department*” jika pengetahuan hanya dapat ditemukan oleh *user* dengan departemen yang sama.



Gambar 13. Tampilan Hasil Pencarian *TCode SAP*

Gambar 13 merupakan tampilan dari hasil pencarian *TCode Standard SAP* berupa daftar *TCode* berdasarkan *keyword* yang dimasukkan yang diambil dari pihak luar.

V. ANALISIS DATA

Dalam analisis data, akan dijabarkan mengenai analisis yang dilakukan setelah tahap implementasi *Knowledge Management System (KMS)* yang digunakan oleh kurang lebih 1000 *user* yang terdiri dari *user* biasa (*Non IT*) dan *user admin (IT)* yang telah digunakan sejak bulan Juni 2020.

1) *Analisis Data Time Series Dengan ARIMA*: tabel XI merupakan data tiket yang diambil sebelum implementasi sistem.

TABEL XI
TABEL DATA TIKET DARI SISTEM

Bulan	Tahun	Jumlah	Rata - rata per 20 Hari Kerja
7	2019	153	8
8	2019	265	13
9	2019	279	14
10	2019	337	17
11	2019	441	22
12	2019	352	18
1	2020	322	16
2	2020	490	25

Tabel XI merupakan tabel jumlah tiket yang masuk dari Juli 2019 hingga Februari 2020 untuk semua kategori. Pada bulan Oktober 2019 hingga Desember 2019 adalah fase *blackout* untuk implementasi sistem baru (*SAP*), sedangkan fase *Go Live* dan penyesuaian *SAP* terjadi pada bulan Januari 2020 hingga Februari 2020. Puncak tiket terbanyak terdapat pada bulan Februari 2020. Dengan data diatas, dapat diketahui bahwa dalam 1 hari, rata – rata tiket yang masuk berkisar antara 8 - 25 tiket dengan jumlah *helpdesk* hanya 1 orang, dan *programmer* yang membantu *helpdesk* ada 3 orang.

2) *Pembuatan Model ARIMA*: Pada proses ini, akan dilakukan pembentukan model *ARIMA* berdasarkan data yang ada. Pembentukan model dilakukan dengan data tiket yang masuk pada departemen IT.

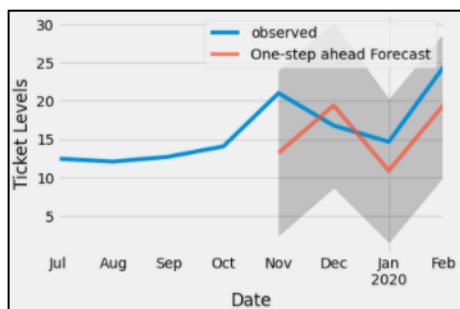
```

ARIMA(0, 1, 1)x(1, 0, 1, 2)12 - AIC:24.736936907328385
ARIMA(0, 1, 1)x(1, 1, 0, 2)12 - AIC:20.94226849245214
ARIMA(1, 0, 0)x(0, 0, 0, 2)12 - AIC:45.7133561124699
ARIMA(1, 0, 0)x(0, 0, 1, 2)12 - AIC:217.41151356632787
ARIMA(1, 0, 0)x(0, 1, 0, 2)12 - AIC:35.966548298471935
ARIMA(1, 0, 0)x(1, 0, 0, 2)12 - AIC:36.832438735524775
ARIMA(1, 0, 0)x(1, 0, 1, 2)12 - AIC:34.63115973739886
ARIMA(1, 0, 0)x(1, 1, 0, 2)12 - AIC:24.75223409211693
ARIMA(1, 0, 1)x(0, 0, 0, 2)12 - AIC:39.2166190932508
ARIMA(1, 0, 1)x(0, 0, 1, 2)12 - AIC:203.50227733779366
ARIMA(1, 0, 1)x(0, 1, 0, 2)12 - AIC:31.95529885561463
ARIMA(1, 0, 1)x(1, 0, 0, 2)12 - AIC:37.39340221991121
ARIMA(1, 0, 1)x(1, 0, 1, 2)12 - AIC:32.815674972161844
ARIMA(1, 0, 1)x(1, 1, 0, 2)12 - AIC:26.023510222427
ARIMA(1, 1, 0)x(0, 0, 0, 2)12 - AIC:40.61506648020065
ARIMA(1, 1, 0)x(0, 0, 1, 2)12 - AIC:214.30059661182614
ARIMA(1, 1, 0)x(0, 1, 0, 2)12 - AIC:32.442142727185235
ARIMA(1, 1, 0)x(1, 0, 0, 2)12 - AIC:30.927598971867553
ARIMA(1, 1, 0)x(1, 0, 1, 2)12 - AIC:32.38489662829727
ARIMA(1, 1, 0)x(1, 1, 0, 2)12 - AIC:16.251385953489706
ARIMA(1, 1, 1)x(0, 0, 0, 2)12 - AIC:36.93661895180699
ARIMA(1, 1, 1)x(0, 0, 1, 2)12 - AIC:153.63175904088206
ARIMA(1, 1, 1)x(0, 1, 0, 2)12 - AIC:27.509048931076176
ARIMA(1, 1, 1)x(1, 0, 0, 2)12 - AIC:32.57583066409603
ARIMA(1, 1, 1)x(1, 0, 1, 2)12 - AIC:27.132230987576756
ARIMA(1, 1, 1)x(1, 1, 0, 2)12 - AIC:18.271507355060228
    
```

Gambar 14. Pembuatan Model *ARIMA*

Gambar 14 merupakan sebagian dari pembuatan model *ARIMA*. Dapat dilihat bahwa model terbaik ada pada *ARIMA* (1, 1, 0) x (1, 1, 0, 2) dengan nilai *AIC* 16.25.

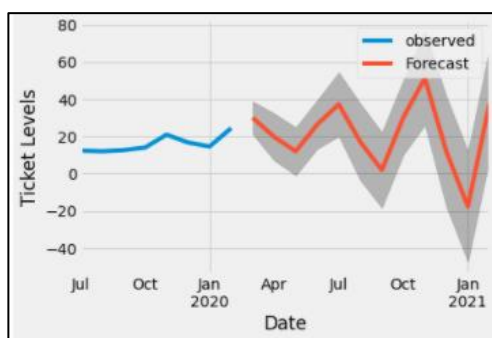
3) *Exploratory Data Training*: Pada proses ini, dilakukan percobaan peramalan untuk bulan November 2019 hingga Februari 2020 menggunakan *ARIMA* dengan data tiket yang ada.



Gambar 15. *Exploratory Data Training*

Gambar 15 merupakan model peramalan data *training* menggunakan *ARIMA* untuk peramalan dari bulan November 2019 hingga Februari 2020. Model yang dibentuk hampir serupa dengan data aktual. Garis berwarna biru merupakan jumlah tiket yang ada, sedangkan garis berwarna jingga merupakan hasil peramalan yang dihasilkan dari jumlah tiket yang ada dengan nilai *Mean Squared Error (MSE)* yang didapat 26.82. Dengan percobaan tersebut dapat dilihat hasil dari peramalan tidak jauh berbeda dengan data aktual.

4) *ARIMA Prediction Analytic*: Setelah dilakukan percobaan peramalan data tiket, pada proses ini akan dibentuk peramalan data tiket yang akan terjadi pada bulan Maret 2020 hingga Februari 2021.

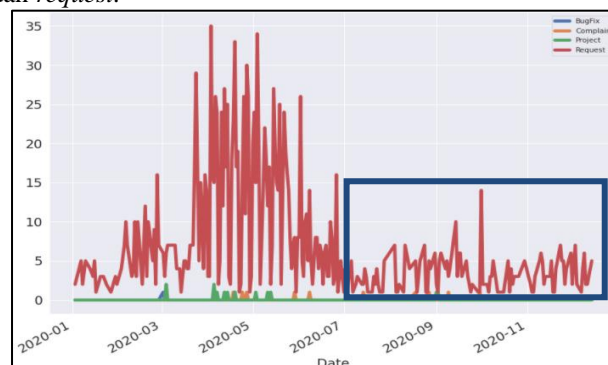


Gambar 16. *ARIMA Prediction Analytic*

Gambar 16 merupakan *time series analytic* menggunakan *ARIMA*. Dari data yang telah dikumpulkan sebelumnya, dilakukan peramalan terhadap *trend* tiket untuk 12 bulan kedepan dimulai dari bulan Maret 2020. Dapat dilihat *trend* terhadap tiket cenderung meningkat. Dengan hasil Perhitungan pada tabel 11, untuk rata – rata tiket per harinya diantara 8 – 25 tiket, dan hasil dari peramalan pada gambar

8 untuk 12 bulan kedepan, maka dapat diasumsikan untuk *trend* tiket 12 bulan kedepan bisa 2 atau 3 kali lipat dari jumlah tiket sebelumnya dengan nilai *Mean Squared Error (MSE)* adalah 40.72.

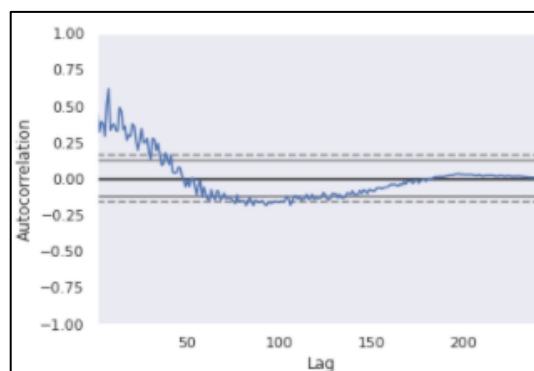
5) *Ticket Data Model*: Pada proses ini, dibuat model berdasarkan data tiket yang ada, yang akan dibagi berdasarkan kategori tiket berupa *bug fix*, *complain*, *project*, dan *request*.



Gambar 17. *Tiket data model*

Dari analisis sebelumnya mengenai peramalan tiket yang akan terjadi 12 bulan kemudian, dan data tiket yang diambil khususnya untuk kategori *SAP*, dapat dilihat pada gambar 17 bahwa jumlah terbanyak ada pada tiket dengan jenis *Request* berada pada bulan April 2020 hingga Juni 2020 kurang lebih sebanyak 25 hingga 35. Dengan melihat *trend* yang akan terjadi dan aktual tiket yang masuk, maka dibuatlah sistem *KMS*. Dari data tersebut sangat terlihat perbedaan antara sebelum penerapan *KMS* dan sesudah *KMS* (ditandai dengan kotak biru) dimana beban *helpdesk* terjadi penurunan. Berdasarkan data yang didapat sebelum dan setelah implementasi *KMS*, dapat dilihat bahwa rata – rata tiket yang masuk berjumlah 14 hingga 25 per hari, dan setelah implementasi *KMS* selama 6 bulan memiliki rata – rata 5 per hari, atau dapat dikatakan bahwa beban *helpdesk*

6) *Autocorrelation Request Ticket*: Pada proses ini, akan dibuat model *autocorrelation* untuk melihat apakah terdapat *varian* atau tidak.



Gambar 18. *Autocorrelation Request Tiket*

Gambar 18 merupakan model *autocorrelation* untuk tiket berdasarkan perhitungan korelasi antara jenis tiket yang masuk dengan *Category SAP* khususnya jenis *request*. Dapat dilihat, *trend* untuk setiap *lag*, cenderung sudah sesuai sehingga tidak terdapat data tidak stasioner dan *difference* pada model

VI. PENGUJIAN

Dalam pengujian, dilakukan survei serta analisis data terkait untuk mengetahui apakah penelitian yang telah dilakukan sesuai dengan tujuan dan hipotesis yang ada.

1) *Survei*: Sebagai evaluasi mengenai solusi yang ditawarkan, dan kegunaan *KMS* yang dapat membantu pekerjaan *helpdesk*, dilakukan survei kepuasan untuk *KMS* menggunakan *platform Google Form* yang dikirimkan kepada 61 responden dari berbagai departemen.

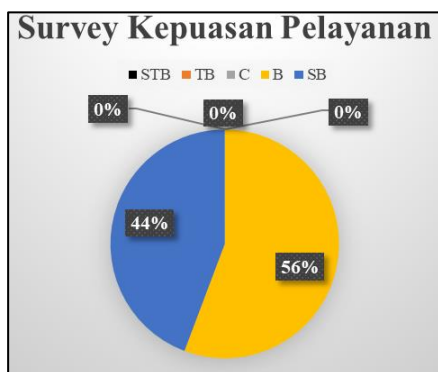
Berikut merupakan keterangan dalam pengisian survei:

- STB (Bernilai 1) = Sangat Tidak Baik
- TB (Bernilai 2) = Tidak Baik
- C (Bernilai 3) = Cukup / > 3 Menit
- B (Bernilai 4) = Baik / 1 – 3 Menit
- SB (Bernilai 5) = Sangat Baik / < 1 Menit

TABEL XII
SURVEI UNTUK SEMUA USER

No	Pertanyaan	STB	TB	C	B	SB
1	Dari Pelayanan yang Telah Anda Pakai, Bagaimana Pelayanan Kami?					
2	Bagaimana Kecepatan Respon menggunakan KMS?					
3	Kecepatan Penyelesaian Solusi / Request (<i>Problem Solving</i>)?					
4	Solusi yang Kami Tawarkan dapat Membantu Pekerjaan Anda?					

Pada tabel XII, terdapat 4 pertanyaan yang akan diajukan dalam survei untuk semua *user*.



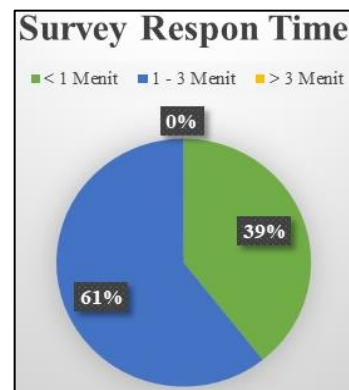
Gambar 19. Hasil Survei Terhadap Kepuasan Pelayanan (Semua User)

Pada gambar 19, dapat dilihat dari total 61 responden untuk pengguna *KMS* dengan *role user*, dan 6 pengguna *KMS* dengan *role IT user*, berikut hasil survei terhadap *KMS* yang telah digunakan. Pada gambar 19, dalam kepuasan pelayanan yang ditawarkan melalui *KMS*, didapatkan 56% responden memilih “Baik”, dan 44% responden memilih “Sangat Baik”.

TABEL XIII
SURVEI UNTUK IT USER

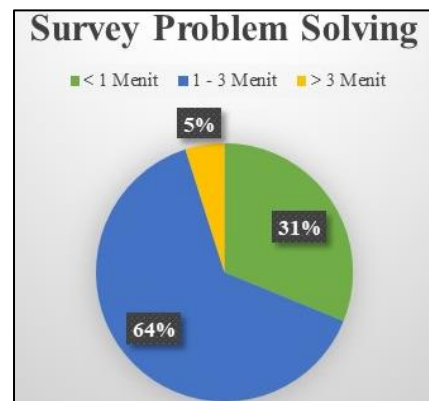
No	Pertanyaan	STB	TB	C	B	SB
1	Dari Pelayanan yang Telah Anda Pakai, Beban Helpdesk Berkurang ±40%?					

Pada tabel XIII, terdapat 1 pertanyaan yang akan diajukan pada survei untuk *IT user*.



Gambar 20. Hasil Survei Terhadap Response Time (Semua User)

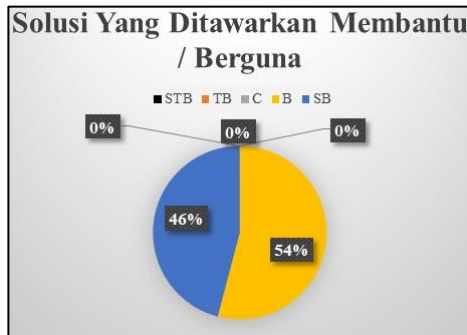
Pada gambar 20 ditampilkan hasil *response time* dalam pencarian data melalui *KMS*, dalam rentang waktu kurang dari 3 menit sebanyak 100% dimana 61% merupakan rentang waktu 1 – 3 menit, dan 39% rentang waktu kurang dari 1 menit.



Gambar 21. Hasil Survei Terhadap Problem Solving (Semua User)

Pada gambar 21, ditampilkan hasil survei dalam pencarian data melalui *KMS*, untuk membantu *user* dalam hal *problem solving*, untuk rentang waktu 1 – 3 menit

sebanyak 64%, kurang dari 1 menit sebanyak 31%, dan lebih dari 3 menit sebanyak 5% yang dikarenakan data pada KMS ini masih tidak terlalu banyak.



Gambar 22. Hasil Survei Terhadap Solusi Dapat Berguna / Membantu (Semua User)

Pada gambar 22, ditampilkan hasil survei terhadap solusi yang ditawarkan melalui KMS, 54% responden menyatakan bahwa solusi dapat berguna atau membantu pekerjaan, dan 46% menyatakan solusi sangat berguna atau membantu pekerjaan.



Gambar 23. Survei Beban Helpdesk (IT User)

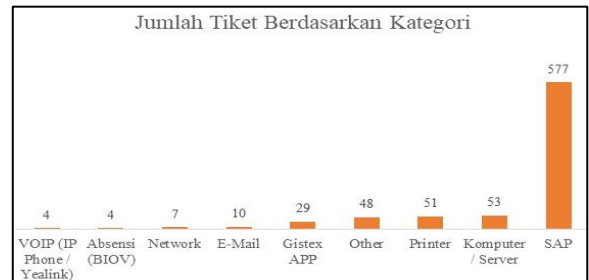
Pada gambar 23, ditampilkan hasil survei dalam membantu menurunkan beban helpdesk untuk user IT, dari 6 responden terdapat 100% yang menyatakan bahwa beban helpdesk turun sebanyak 40%.

Rangkuman survei yang telah didapatkan melalui 61 responden, ditunjukkan melalui tabel XIV. Nilai rata-rata yang diperoleh pada tabel XIV, yaitu 4.52 yang menyatakan antara Baik dan Sangat Baik.

TABEL XIV
RANGKUMAN SURVEI UNTUK IT USER

No	Pertanyaan	Rata - rata
1	Dari Pelayanan yang Telah Anda Pakai, Bagaimana Pelayanan Kami?	4.4
2	Bagaimana Kecepatan Respon menggunakan KMS?	4.4
3	Kecepatan Penyelesaian Solusi / Request (Problem Solving)?	4.3
4	Solusi yang Kami Tawarkan dapat Membantu Pekerjaan Anda?	4.5
5	Beban helpdesk berkurang 40%?	5
Rata - Rata Total		4.52

2) Analisis data setelah implementasi: Setelah implementasi sistem, dilakukan analisis data untuk mengetahui kategori apa yang paling banyak digunakan, kata kunci apa yang paling banyak dicari, serta berapa banyak pengguna aktif KMS.



Gambar 24. Jumlah Tiket Berdasarkan Kategori

Pada gambar 24 ditunjukkan data tiket yang dikelompokkan berdasarkan kategori tiket yang diambil dari bulan Juli 2020 hingga bulan Desember 2020. Dapat dilihat bahwa kategori tiket terbanyak adalah SAP dengan jumlah 577 tiket.



Gambar 25. Jumlah Pencarian Berdasarkan Keyword

Pada gambar 25 ditunjukkan data berdasarkan jumlah keyword yang dianggap populer untuk kategori tiket SAP. Dapat dilihat bahwa keyword "Chop" merupakan yang terbanyak dengan total 95 kali muncul, dan "Cek" merupakan yang terendah dengan jumlah 38. Dalam pengelompokan keyword untuk pemecahan masalah, keyword "Cek" dan "Error" bertotal 95 kali muncul.



Gambar 26. Jumlah Pencarian Berdasarkan Departemen

Pada gambar 26 ditunjukkan data berdasarkan jumlah pengguna tiket yang dikelompokkan berdasarkan departemen. Dapat dilihat bahwa departemen terbanyak adalah "Accounting dan Finance", dan departemen terendah adalah "Project".

VII. SIMPULAN

Dalam penelitian ini, telah dibuat sistem berupa KMS yang diharapkan dapat membantu dalam pencarian pengetahuan / *problem solving* untuk user. Dengan menggunakan metode *information retrieval* mulai dari *tokenizing*, *filtering*, *VSM*, *TF-IDF*, dan juga *Cosine Similarity*, KMS dapat digunakan untuk membantu user dalam penentuan kandidat solusi berdasarkan pencarian pengetahuan, dan *problem solving*, serta membantu dalam menurunkan beban *helpdesk* sebanyak 4 hingga 5%.

Dalam pencarian pengetahuan / *problem solving*, rata – rata respon yang didapat kurang dari atau sama dengan 3 menit, dan dalam mendapatkan pengetahuan / *problem solving*, rata – rata user dapat menemukannya kurang dari atau sama dengan 3 menit dengan kategori terbanyak adalah "SAP", dan pengguna terbanyak adalah departemen "Finance dan Accounting".

DAFTAR PUSTAKA

- [1] W. Dai, "Improvement and Implementation of Feature Weighting Algorithm TF-IDF in Text Classification," *Advances in Intelligent Systems Research*, vol. 147, pp. 583-587, 2018.
- [2] (2016) Education Zone website. [Online]. Tersedia: <https://www.tabraniza.com/2016/11/sekilas-tentang-desain-penelitian.html>
- [3] Wiyanto, W. Priatna and J. S. Hidayat, "Implementasi Term Frequency -Inverse Document Frequency (Tf-Idf) Dan Vector Space Model (Vsm) Untuk Pencarian Berita Bahasa Indonesia," *Pelita Teknologi: Jurnal Ilmiah Informatika, Arsitektur dan Lingkungan*, vol. 14, no. 2, pp. 119-133, 2019.
- [4] (2018) School of Computer Science Binus website. [Online]. Tersedia: <https://socs.binus.ac.id/2018/11/29/vector-space-model-dalam-pengolahan-teks/>
- [5] H. Smuts, A. v. d. Merwe, M. Loock and P. Kotzé, "A Framework and Methodology for Knowledge Management System Implementation," *ACM International Conference Proceeding Series*, 2009, p. 70-79.
- [6] L. Sitorus, *Algoritma Dan Pemrograman*, 1st ed., Yogyakarta, Indonesia: Andi 2015, pp. 14-16
- [7] W. Sahayu. (2013) <http://staff.uny.ac.id/>. [Online]. Tersedia: <http://staff.uny.ac.id/sites/default/files/pendidikan/Dra.%20Wening%20Sahayu,%20M.Pd./Makalah%20Hipotesis.pdf>. [Accessed 31 March 2020]
- [8] E. Sadewo. (2013) ResearchGate. [Online]. Tersedia: https://www.researchgate.net/publication/311558057_Perbandingan_Beberapa_Metode_Time_Series_Pada_Peramalan_Jumlah_Kunjungan_Wisatawan_Mancanegara_Studi_Kasus_Di_Kabupaten_Karimun_Provinsi_Kepulauan_Riau
- [9] O. Karnalim, "Extended Vector Space Model with Semantic Relatedness on Java Archive Search Engine," *Jurnal Teknik Informatika dan Sistem Informasi (JuTiSi)*, vol. 1, no. 2, p. 111, 2015.
- [10] A. Ryansyah and S. Andayani, "Implementasi Algoritma TF-IDF Pada Pengukuran Kesamaan Dokumen," *Jurnal Sistem & Teknologi Informasi Komunikasi*, vol. 1, no. 1, pp. 1-10.
- [11] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25-29, 2018.
- [12] M. Nurjannah, Hamdani and I. F. Astuti, "Penerapan Algoritma Term Frequency-Inverse Document Frequency (Tf-Idf) Untuk Text Mining," *Jurnal Informatika Mulawarman*, vol. 8, no. 3, pp. 110-113, 2013.
- [13] A. A. Maarif. (2015) eprints homepage on UDiNus Repository. [Online]. Tersedia: <http://eprints.dinus.ac.id/15283/>
- [14] Kusri, *Strategi Perancangan dan Pengelolaan Basis Data*, Yogyakarta, Indonesia: Andi 2007, pp. 10, 49-50.
- [15] S. V. Kant and V. K. Singh, "Vector Space Model: An Information Retrieval," *International Journal of Advanced Engineering Research and Studies*, vol. IV, pp. 141-143, 2015.
- [16] A. Josi, L. A. Abdillah and Suryayusra, "Penerapan teknik web scraping pada mesin pencari artikel ilmiah," *Jurnal Sistem Informasi (SISFO)*, vol. 5, no. 2, pp. 159-164, 2014.
- [17] A. Irawan and N. K. Setiyorini, "Rancang Bangun Aplikasi helpdesk Dengan Menggunakan Pendekatan Knowledge Management System Pada Seksi Teknisi pt. Indah kiat pulp & paper tbk," *Jurnal ProTekInfo*, vol. 4, pp. 6-11, 2017.
- [18] B. Herwijayanti, D. E. Ratnawati and L. Muflikhah, "Klasifikasi Berita Online dengan menggunakan Pembobotan TF-IDF dan Cosine Similarity," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 1, pp. 306-3012, 2018.
- [19] M. O. Igbinovia and I. J. Ikenwe, "Knowledge management: processes and systems," *Information Impact: Journal of Information and Knowledge Management*, vol. 8, pp. 26, Feb. 2018.